HSPOS iOS PrinterSDK API Documentation

Item	Description
Product	HSPOS iOS PrinterSDK
Platform	iOS 15+
Entry class	`PrinterSDK`
Constants	`PrinterConstants`
Connection protocol	`PrinterConnection`
Companion doc	*PrinterSDK iOS User Manual*
Chinese edition	`PrinterSDK_iOS_函数说明.pdf`

---

1. Conventions

- Most print methods take `PrinterConnection?` first; invalid connections return `false` or `nil`.
- For `Bool` returns: `true` means the command was written to the connection (not a guarantee of successful paper output).
- Prefer calling only the `PrinterSDK` facade from app code.
- Lengths are generally in **dots** (~8 dots/mm at 203 DPI).

2. Helper Types

2.1 `PrinterConnection`

Connection abstraction. Common methods: `connect()` / `disconnect()` / `isConnected()` / `write(_:)` / `read(timeout:)` / `clearInput()` / `flushWrites()`.

2.2 `BluetoothDeviceListener`

```
public protocol BluetoothDeviceListener: AnyObject {
    func onDeviceFound(_ peripheral: CBPeripheral, rssi: Int)
    func onDiscoveryFinished()
}
```

---

3. `PrinterSDK` API

1. Lifecycle

`shared`

Global singleton entry.

```
public static let shared: PrinterSDK
```

`release`

Release SDK resources (disconnect, etc.).

```
public func release()
```

`getCommunicationManager`

Returns the underlying communication manager (advanced).

```
public func getCommunicationManager() -> CommunicationManager
```

Returns: `CommunicationManager`

2. Connection — BLE

`getBluetoothDevices`

Returns currently discovered BLE peripherals.

```
public func getBluetoothDevices() -> [CBPeripheral]
```

Returns: `[CBPeripheral]`

`startBluetoothDiscovery`

Starts BLE scanning.

```
public func startBluetoothDiscovery() -> Bool
```

Returns: `Bool`

`stopBluetoothDiscovery`

Stops BLE scanning.

```
public func stopBluetoothDiscovery()
```

`refreshBluetoothDevices`

Refreshes discovery (stop then start).

```
public func refreshBluetoothDevices() -> Bool
```

Returns: `Bool`

`connectBluetoothPrinter`

Connects a BLE printer; returns the connection on success.

```
public func connectBluetoothPrinter(_ peripheral: CBPeripheral) -> PrinterConnection?
```

Parameter	Type	Description
`peripheral`	`CBPeripheral`	BLE peripheral

Returns: `PrinterConnection?`

`addBluetoothListener`

Registers a BLE discovery listener.

```
public func addBluetoothListener(_ listener: BluetoothDeviceListener)
```

Parameter	Type	Description
`listener`	`BluetoothDeviceListener`	BLE listener

`removeBluetoothListener`

Removes a BLE discovery listener.

```
public func removeBluetoothListener(_ listener: BluetoothDeviceListener)
```

Parameter	Type	Description
`listener`	`BluetoothDeviceListener`	BLE listener

3. Connection — Network TCP

`connectNetworkPrinter`

Connects a network printer synchronously (do not call on the main thread).

```
public func connectNetworkPrinter(ip: String, port: Int) -> PrinterConnection?
```

Parameter	Type	Description
`ip`	`String`	Printer IP
`port`	`Int`	Port, typically 9100

Returns: `PrinterConnection?`

`connectNetworkPrinterAsync`

Connects a network printer asynchronously (main-thread safe).

```
public func connectNetworkPrinterAsync( ip: String, port: Int, onSuccess: @escaping
(PrinterConnection) -> Void, onFailed: @escaping (String) -> Void )
```

Parameter	Type	Description
`ip`	`String`	Printer IP
`port`	`Int`	Port, typically 9100
`onSuccess`	`@escaping (PrinterConnection) -> Void`	—
`onFailed`	`@escaping (String) -> Void`	—

Returns: `Void, onFailed: @escaping (String) -> Void )`

`disconnectPrinter`

Disconnects the printer.

```
public func disconnectPrinter(_ connection: PrinterConnection?)
```

Parameter	Type	Description
`connection`	`PrinterConnection?`	Printer connection; fails if invalid

4. Status & Control

`queryRTStatus`

Queries real-time status.

```
public func queryRTStatus(_ connection: PrinterConnection?, type: Int, timeout: Int = 3000) -> Data?
```

Parameter	Type	Description
`connection`	`PrinterConnection?`	Printer connection; fails if invalid
`type`	`Int`	—
`timeout`	`Int`	Timeout (ms); default `3000`

Returns: `Data?`

`queryTransmitStatus`

Queries transmit status.

```
public func queryTransmitStatus(_ connection: PrinterConnection?, type: Int, timeout: Int = 3000) -> Data?
```

Parameter	Type	Description
`connection`	`PrinterConnection?`	Printer connection; fails if invalid
`type`	`Int`	—
`timeout`	`Int`	Timeout (ms); default `3000`

Returns: `Data?`

`clearPrinterBuffer`

Clears the printer buffer.

```
public func clearPrinterBuffer(_ connection: PrinterConnection?) -> Bool
```

Parameter	Type	Description
`connection`	`PrinterConnection?`	Printer connection; fails if invalid

Returns: `Bool`

`cancelPrint`

Cancels the current print job.

```
public func cancelPrint(_ connection: PrinterConnection?) -> Bool
```

Parameter	Type	Description
`connection`	`PrinterConnection?`	Printer connection; fails if invalid

Returns: `Bool`

`getPrinterStatus`

Returns a composed status bitmask.

```
public func getPrinterStatus(_ connection: PrinterConnection?) -> Int
```

Parameter	Type	Description
`connection`	`PrinterConnection?`	Printer connection; fails if invalid

Returns: `Int`

`isPrinterOnline`

Whether the printer is online.

```
public func isPrinterOnline(_ connection: PrinterConnection?) -> Bool
```

Parameter	Type	Description
`connection`	`PrinterConnection?`	Printer connection; fails if invalid

Returns: `Bool`

`hasPaper`

Whether paper is present.

```
public func hasPaper(_ connection: PrinterConnection?) -> Bool
```

Parameter	Type	Description
`connection`	`PrinterConnection?`	Printer connection; fails if invalid

Returns: `Bool`

`isCoverOpen`

Whether the cover is open.

```
public func isCoverOpen(_ connection: PrinterConnection?) -> Bool
```

Parameter	Type	Description
`connection`	`PrinterConnection?`	Printer connection; fails if invalid

Returns: `Bool`

`clearBuffer`

Sends clear-buffer command.

```
public func clearBuffer(_ connection: PrinterConnection?) -> Bool
```

Parameter	Type	Description
`connection`	`PrinterConnection?`	Printer connection; fails if invalid

Returns: `Bool`

`cancelPrintCmd`

Sends cancel-print command.

```
public func cancelPrintCmd(_ connection: PrinterConnection?) -> Bool
```

Parameter	Type	Description
`connection`	`PrinterConnection?`	Printer connection; fails if invalid

Returns: `Bool`

5. POS — Init & Text

`initPrinter`

Initializes the printer (ESC @).

```
public func initPrinter(_ connection: PrinterConnection?) -> Bool
```

Parameter	Type	Description
`connection`	`PrinterConnection?`	Printer connection; fails if invalid

Returns: `Bool`

`resetPrinter`

Resets the printer.

```
public func resetPrinter(_ connection: PrinterConnection?) -> Bool
```

Parameter	Type	Description
`connection`	`PrinterConnection?`	Printer connection; fails if invalid

Returns: `Bool`

`printText`

Prints text with the default encoding.

```
public func printText(_ connection: PrinterConnection?, text: String) -> Bool
```

Parameter	Type	Description
`connection`	`PrinterConnection?`	Printer connection; fails if invalid
`text`	`String`	Text

Returns: `Bool`

`printTextLineLeftRight`

Prints one left—right aligned line.

```
public func printTextLineLeftRight( _ connection: PrinterConnection?, left: String, right:
String, areaWidthDots: Int = PrinterConstants.PAPER_WIDTH_80MM, charWidthDots: Int = 12 ) ->
Bool
```

Parameter	Type	Description
`connection`	`PrinterConnection?`	Printer connection; fails if invalid
`left`	`String`	Coordinate (dots)
`right`	`String`	Coordinate (dots)
`areaWidthDots`	`Int`	Area width (dots); default `PrinterConstants.PAPER_WIDTH_80MM`
`charWidthDots`	`Int`	default `12`

Returns: `Bool`

```
public func printTextLineLeftRight(_ connection: PrinterConnection?, left: String, right: String) -> Bool
```

Parameter	Type	Description
`connection`	`PrinterConnection?`	Printer connection; fails if invalid
`left`	`String`	Coordinate (dots)
`right`	`String`	Coordinate (dots)

Returns: `Bool`

`printTextLineLeftRightGBK`

Left—right line in GBK.

```
public func printTextLineLeftRightGBK( _ connection: PrinterConnection?, left: String, right: String, lineCharWidth: Int = 48 ) -> Bool
```

Parameter	Type	Description
`connection`	`PrinterConnection?`	Printer connection; fails if invalid
`left`	`String`	Coordinate (dots)
`right`	`String`	Coordinate (dots)
`lineCharWidth`	`Int`	default `48`

Returns: `Bool`

`printTextInUTF8`

Prints text as UTF-8.

```
public func printTextInUTF8(_ connection: PrinterConnection?, text: String) -> Bool
```

Parameter	Type	Description
`connection`	`PrinterConnection?`	Printer connection; fails if invalid
`text`	`String`	Text

Returns: `Bool`

`printTextInGBK`

Prints text as GBK (recommended for Simplified Chinese).

```
public func printTextInGBK(_ connection: PrinterConnection?, text: String) -> Bool
```

Parameter	Type	Description
`connection`	`PrinterConnection?`	Printer connection; fails if invalid

`text`	`String`	Text

Returns: `Bool`

`printTextInBIG5`

Prints text as BIG5.

```
public func printTextInBIG5(_ connection: PrinterConnection?, text: String) -> Bool
```

Parameter	Type	Description
`connection`	`PrinterConnection?`	Printer connection; fails if invalid
`text`	`String`	Text

Returns: `Bool`

`printTextInShiftJIS`

Prints text as Shift-JIS.

```
public func printTextInShiftJIS(_ connection: PrinterConnection?, text: String) -> Bool
```

Parameter	Type	Description
`connection`	`PrinterConnection?`	Printer connection; fails if invalid
`text`	`String`	Text

Returns: `Bool`

`printTextInEUCKR`

Prints text as EUC-KR.

```
public func printTextInEUCKR(_ connection: PrinterConnection?, text: String) -> Bool
```

Parameter	Type	Description
`connection`	`PrinterConnection?`	Printer connection; fails if invalid
`text`	`String`	Text

Returns: `Bool`

`printTabLine`

Prints a multi-column tab line.

```
public func printTabLine(_ connection: PrinterConnection?, columns: [String], encodingName:
String? = nil) -> Bool
```

Parameter	Type	Description
`connection`	`PrinterConnection?`	Printer connection; fails if invalid
`columns`	`[String]`	Column texts
`encodingName`	`String?`	default `nil`

Returns: `Bool`

6. POS — Formatting

`setAlignment`

Sets alignment.

```
public func setAlignment(_ connection: PrinterConnection?, align: Int) -> Bool
```

Parameter	Type	Description
`connection`	`PrinterConnection?`	Printer connection; fails if invalid
`align`	`Int`	ALIGNMENT_*

Returns: `Bool`

`setBold`

Sets bold.

```
public func setBold(_ connection: PrinterConnection?, bold: Bool) -> Bool
```

Parameter	Type	Description
`connection`	`PrinterConnection?`	Printer connection; fails if invalid
`bold`	`Bool`	—

Returns: `Bool`

`setUnderline`

Sets underline.

```
public func setUnderline(_ connection: PrinterConnection?, underline: Bool) -> Bool
```

Parameter	Type	Description
`connection`	`PrinterConnection?`	Printer connection; fails if invalid
`underline`	`Bool`	—

Returns: `Bool`

`setReverse`

Sets reverse (white-on-black).

```
public func setReverse(_ connection: PrinterConnection?, reverse: Bool) -> Bool
```

Parameter	Type	Description
`connection`	`PrinterConnection?`	Printer connection; fails if invalid
`reverse`	`Bool`	—

Returns: `Bool`

`setLineSpacing`

Sets line spacing.

```
public func setLineSpacing(_ connection: PrinterConnection?, spacing: Int) -> Bool
```

Parameter	Type	Description
`connection`	`PrinterConnection?`	Printer connection; fails if invalid
`spacing`	`Int`	—

Returns: `Bool`

`setCharSpacing`

Sets character spacing.

```
public func setCharSpacing(_ connection: PrinterConnection?, spacing: Int) -> Bool
```

Parameter	Type	Description
`connection`	`PrinterConnection?`	Printer connection; fails if invalid
`spacing`	`Int`	—

Returns: `Bool`

`setPrintMode`

Sets print mode (ESC !).

```
public func setPrintMode(_ connection: PrinterConnection?, mode: Int) -> Bool
```

Parameter	Type	Description
`connection`	`PrinterConnection?`	Printer connection; fails if invalid
`mode`	`Int`	—

Returns: `Bool`

`setChineseMode`

Enables Chinese mode.

```
public func setChineseMode(_ connection: PrinterConnection?) -> Bool
```

Parameter	Type	Description
`connection`	`PrinterConnection?`	Printer connection; fails if invalid

Returns: `Bool`

`setDoubleByteEncoding`

Sets double-byte encoding.

```
public func setDoubleByteEncoding(_ connection: PrinterConnection?, encoding: Int) -> Bool
```

Parameter	Type	Description
`connection`	`PrinterConnection?`	Printer connection; fails if invalid
`encoding`	`Int`	DOUBLE_BYTE_ENCODING_*

Returns: `Bool`

`setTextScale`

Sets character scale (GS !).

```
public func setTextScale(_ connection: PrinterConnection?, widthScale: Int, heightScale: Int) -> Bool
```

Parameter	Type	Description
`connection`	`PrinterConnection?`	Printer connection; fails if invalid
`widthScale`	`Int`	—
`heightScale`	`Int`	—

Returns: `Bool`

`setChineseDoubleSize`

CJK double size (FS W).

```
public func setChineseDoubleSize(_ connection: PrinterConnection?, enable: Bool) -> Bool
```

Parameter	Type	Description
`connection`	`PrinterConnection?`	Printer connection; fails if invalid

`enable`	`Bool`	—

Returns: `Bool`

```
public func setChineseDoubleSize(_ connection: PrinterConnection?, n: Int) -> Bool
```

Parameter	Type	Description
`connection`	`PrinterConnection?`	Printer connection; fails if invalid
`n`	`Int`	—

Returns: `Bool`

`setCharacterSet`

Sets international character set.

```
public func setCharacterSet(_ connection: PrinterConnection?, charset: Int) -> Bool
```

Parameter	Type	Description
`connection`	`PrinterConnection?`	Printer connection; fails if invalid
`charset`	`Int`	—

Returns: `Bool`

`setCharacterCodepage`

Sets code page.

```
public func setCharacterCodepage(_ connection: PrinterConnection?, codepage: Int) -> Bool
```

Parameter	Type	Description
`connection`	`PrinterConnection?`	Printer connection; fails if invalid
`codepage`	`Int`	—

Returns: `Bool`

`setSingleByteMode`

Switches to single-byte mode.

```
public func setSingleByteMode(_ connection: PrinterConnection?) -> Bool
```

Parameter	Type	Description
`connection`	`PrinterConnection?`	Printer connection; fails if invalid

Returns: `Bool`

`setAsciiTextFontType`

Sets ASCII font type.

```
public func setAsciiTextFontType(_ connection: PrinterConnection?, fontType: Int) -> Bool
```

Parameter	Type	Description
`connection`	`PrinterConnection?`	Printer connection; fails if invalid
`fontType`	`Int`	—

Returns: `Bool`

`setTextUpsideDown`

Sets upside-down printing.

```
public func setTextUpsideDown(_ connection: PrinterConnection?, upsideDown: Bool) -> Bool
```

Parameter	Type	Description
`connection`	`PrinterConnection?`	Printer connection; fails if invalid
`upsideDown`	`Bool`	—

Returns: `Bool`

`setTextRotate`

Sets rotated printing.

```
public func setTextRotate(_ connection: PrinterConnection?, rotate: Bool) -> Bool
```

Parameter	Type	Description
`connection`	`PrinterConnection?`	Printer connection; fails if invalid
`rotate`	`Bool`	Rotation

Returns: `Bool`

`setTextLineHeight`

Sets line height.

```
public func setTextLineHeight(_ connection: PrinterConnection?, lineHeight: Int) -> Bool
```

Parameter	Type	Description
`connection`	`PrinterConnection?`	Printer connection; fails if invalid
`lineHeight`	`Int`	—

Returns: `Bool`

`setAsciiTextCharRightSpacing`

Sets ASCII right spacing.

```
public func setAsciiTextCharRightSpacing(_ connection: PrinterConnection?, spacing: Int) -> Bool
```

Parameter	Type	Description
`connection`	`PrinterConnection?`	Printer connection; fails if invalid
`spacing`	`Int`	—

Returns: `Bool`

`setKanjiTextCharSpacing`

Sets CJK left/right spacing.

```
public func setKanjiTextCharSpacing(_ connection: PrinterConnection?, left: Int, right: Int) -> Bool
```

Parameter	Type	Description
`connection`	`PrinterConnection?`	Printer connection; fails if invalid
`left`	`Int`	Coordinate (dots)
`right`	`Int`	Coordinate (dots)

Returns: `Bool`

`setPrintAreaWidth`

Sets print-area width in dots.

```
public func setPrintAreaWidth(_ connection: PrinterConnection?, width: Int) -> Bool
```

Parameter	Type	Description
`connection`	`PrinterConnection?`	Printer connection; fails if invalid
`width`	`Int`	Size (dots)

Returns: `Bool`

`setPrintAreaLeftMargin`

Sets left margin.

```
public func setPrintAreaLeftMargin(_ connection: PrinterConnection?, leftMargin: Int) -> Bool
```

Parameter	Type	Description

`connection`	`PrinterConnection?`	Printer connection; fails if invalid
`leftMargin`	`Int`	—

Returns: `Bool`

`setPrintPosition`

Sets absolute print position.

```
public func setPrintPosition(_ connection: PrinterConnection?, position: Int) -> Bool
```

Parameter	Type	Description
`connection`	`PrinterConnection?`	Printer connection; fails if invalid
`position`	`Int`	—

Returns: `Bool`

`setHorizontalAbsolutePrintPosition`

Horizontal absolute position.

```
public func setHorizontalAbsolutePrintPosition(_ connection: PrinterConnection?, position: Int)
-> Bool
```

Parameter	Type	Description
`connection`	`PrinterConnection?`	Printer connection; fails if invalid
`position`	`Int`	—

Returns: `Bool`

`setHorizontalRelativePrintPosition`

Horizontal relative position.

```
public func setHorizontalRelativePrintPosition(_ connection: PrinterConnection?, offset: Int) ->
Bool
```

Parameter	Type	Description
`connection`	`PrinterConnection?`	Printer connection; fails if invalid
`offset`	`Int`	—

Returns: `Bool`

`setVerticalAbsolutePrintPosition`

Vertical absolute position.

```
public func setVerticalAbsolutePrintPosition(_ connection: PrinterConnection?, position: Int) -> Bool
```

Parameter	Type	Description
`connection`	`PrinterConnection?`	Printer connection; fails if invalid
`position`	`Int`	—

Returns: `Bool`

`setVerticalRelativePrintPosition`

Vertical relative position.

```
public func setVerticalRelativePrintPosition(_ connection: PrinterConnection?, offset: Int) -> Bool
```

Parameter	Type	Description
`connection`	`PrinterConnection?`	Printer connection; fails if invalid
`offset`	`Int`	—

Returns: `Bool`

`horizontalTab`

Horizontal tab.

```
public func horizontalTab(_ connection: PrinterConnection?) -> Bool
```

Parameter	Type	Description
`connection`	`PrinterConnection?`	Printer connection; fails if invalid

Returns: `Bool`

`setTabPositions`

Sets tab stops.

```
public func setTabPositions(_ connection: PrinterConnection?, positions: [Int]) -> Bool
```

Parameter	Type	Description
`connection`	`PrinterConnection?`	Printer connection; fails if invalid
`positions`	`[Int]`	Tab stops

Returns: `Bool`

7. POS — Barcode / QR / PDF417

`printBarcode`

Prints a 1D barcode.

```
public func printBarcode( _ connection: PrinterConnection?, type: Int, data: String, hri: Int = PrinterConstants.BARCODE_HRI_BELOW, height: Int = 80, width: Int = 2 ) -> Bool
```

Parameter	Type	Description
`connection`	`PrinterConnection?`	Printer connection; fails if invalid
`type`	`Int`	—
`data`	`String`	Payload
`hri`	`Int`	default `PrinterConstants.BARCODE_HRI_BELOW`
`height`	`Int`	Size (dots); default `80`
`width`	`Int`	Size (dots); default `2`

Returns: `Bool`

`setBarcodeHeight`

Sets barcode height.

```
public func setBarcodeHeight(_ connection: PrinterConnection?, height: Int) -> Bool
```

Parameter	Type	Description
`connection`	`PrinterConnection?`	Printer connection; fails if invalid
`height`	`Int`	Size (dots)

Returns: `Bool`

`setBarcodeWidth`

Sets barcode module width.

```
public func setBarcodeWidth(_ connection: PrinterConnection?, width: Int) -> Bool
```

Parameter	Type	Description
`connection`	`PrinterConnection?`	Printer connection; fails if invalid
`width`	`Int`	Size (dots)

Returns: `Bool`

`setBarcodeHRIPosition`

Sets barcode HRI position.

```
public func setBarcodeHRIPosition(_ connection: PrinterConnection?, pos: Int) -> Bool
```

Parameter	Type	Description
`connection`	`PrinterConnection?`	Printer connection; fails if invalid
`pos`	`Int`	—

Returns: `Bool`

`setBarcodeReadableTextFontType`

Sets HRI font.

```
public func setBarcodeReadableTextFontType(_ connection: PrinterConnection?, fontType: Int) ->
Bool
```

Parameter	Type	Description
`connection`	`PrinterConnection?`	Printer connection; fails if invalid
`fontType`	`Int`	—

Returns: `Bool`

`setBarcodeReadableTextPosition`

Sets HRI position.

```
public func setBarcodeReadableTextPosition(_ connection: PrinterConnection?, position: Int) ->
Bool
```

Parameter	Type	Description
`connection`	`PrinterConnection?`	Printer connection; fails if invalid
`position`	`Int`	—

Returns: `Bool`

`printQRCode`

Prints a QR code (simple API).

```
public func printQRCode(_ connection: PrinterConnection?, content: String, size: Int = 3) ->
Bool
```

Parameter	Type	Description
`connection`	`PrinterConnection?`	Printer connection; fails if invalid
`content`	`String`	Content
`size`	`Int`	default `3`

Returns: `Bool`

`printQRCodeStandard`

Prints a QR code (full parameters).

```
public func printQRCodeStandard( _ connection: PrinterConnection?, content: String, size: Int, align: Int, eccLevel: Int ) -> Bool
```

Parameter	Type	Description
`connection`	`PrinterConnection?`	Printer connection; fails if invalid
`content`	`String`	Content
`size`	`Int`	—
`align`	`Int`	ALIGNMENT_*
`eccLevel`	`Int`	—

Returns: `Bool`

`printDoubleQRCode`

Prints dual QR codes.

```
public func printDoubleQRCode( _ connection: PrinterConnection?, n: Int, qr1Pos: Int, qr1Data: String, qr1Ecc: Int, qr1Ver: Int, qr2Pos: Int, qr2Data: String, qr2Ecc: Int, qr2Ver: Int ) -> Bool
```

Parameter	Type	Description
`connection`	`PrinterConnection?`	Printer connection; fails if invalid
`n`	`Int`	—
`qr1Pos`	`Int`	—
`qr1Data`	`String`	—
`qr1Ecc`	`Int`	—
`qr1Ver`	`Int`	—
`qr2Pos`	`Int`	—
`qr2Data`	`String`	—
`qr2Ecc`	`Int`	—
`qr2Ver`	`Int`	—

Returns: `Bool`

`printPDF417`

Prints PDF417.

```
public func printPDF417( _ connection: PrinterConnection?, columns: Int, rows: Int, moduleWidth: Int, rowHeight: Int, errorCorrectionLevel: Int, option: Int, data: String ) -> Bool
```

Parameter	Type	Description
`connection`	`PrinterConnection?`	Printer connection; fails if invalid
`columns`	`Int`	Column texts
`rows`	`Int`	—
`moduleWidth`	`Int`	—
`rowHeight`	`Int`	—
`errorCorrectionLevel`	`Int`	—
`option`	`Int`	—
`data`	`String`	Payload

Returns: `Bool`

8. POS — Image / Cut / Peripherals

`printRasterImageFromFile`

Prints a raster image from a file.

```
public func printRasterImageFromFile( _ connection: PrinterConnection?, filePath: String, targetWidth: Int = 384, targetHeight: Int = 0, compressionMode: Int = PrinterConstants.RASTER_COMPRESSION_GS_V0 ) -> Bool
```

Parameter	Type	Description
`connection`	`PrinterConnection?`	Printer connection; fails if invalid
`filePath`	`String`	File path
`targetWidth`	`Int`	Target width (dots); for labels ⩽0 uses source width (max 576); default `384`
`targetHeight`	`Int`	Target height; 0 often keeps aspect; default `0`
`compressionMode`	`Int`	Currently falls back to GS v 0; default `PrinterConstants.RASTER_COMPRESSION_GS_V0`

Returns: `Bool`

`printRasterImageFromData`

Prints a raster image from image Data.

```
public func printRasterImageFromData( _ connection: PrinterConnection?, imageData: Data, targetWidth: Int = 384, targetHeight: Int = 0, compressionMode: Int = PrinterConstants.RASTER_COMPRESSION_GS_V0 ) -> Bool
```

Parameter	Type	Description
`connection`	`PrinterConnection?`	Printer connection; fails if invalid
`imageData`	`Data`	Image data
`targetWidth`	`Int`	Target width (dots); for labels ⩽0 uses source width (max 576); default `384`
`targetHeight`	`Int`	Target height; 0 often keeps aspect; default `0`
`compressionMode`	`Int`	Currently falls back to GS v 0; default `PrinterConstants.RASTER_COMPRESSION_GS_V0`

Returns: `Bool`

`printRasterImageFromPixels`

Prints a raster image from pixels.

```
public func printRasterImageFromPixels( _ connection: PrinterConnection?, pixels: [UInt32], width: Int, height: Int, targetWidth: Int = 384, targetHeight: Int = 0, compressionMode: Int = PrinterConstants.RASTER_COMPRESSION_GS_V0 ) -> Bool
```

Parameter	Type	Description
`connection`	`PrinterConnection?`	Printer connection; fails if invalid
`pixels`	`[UInt32]`	ARGB pixels
`width`	`Int`	Size (dots)
`height`	`Int`	Size (dots)
`targetWidth`	`Int`	Target width (dots); for labels ⩽0 uses source width (max 576); default `384`
`targetHeight`	`Int`	Target height; 0 often keeps aspect; default `0`
`compressionMode`	`Int`	Currently falls back to GS v 0; default `PrinterConstants.RASTER_COMPRESSION_GS_V0`

Returns: `Bool`

`printImage`

Prints a UIImage.

```
public func printImage(_ connection: PrinterConnection?, image: UIImage) -> Bool
```

Parameter	Type	Description
`connection`	`PrinterConnection?`	Printer connection; fails if invalid
`image`	`UIImage`	UIImage

Returns: `Bool`

`printNvBitmap`

Prints an NV bitmap.

```
public func printNvBitmap(_ connection: PrinterConnection?, nvNumber: Int, mode: Int) -> Bool
```

Parameter	Type	Description
`connection`	`PrinterConnection?`	Printer connection; fails if invalid
`nvNumber`	`Int`	—
`mode`	`Int`	—

Returns: `Bool`

`cutPaper`

Cuts paper.

```
public func cutPaper(_ connection: PrinterConnection?, cutType: Int) -> Bool
```

Parameter	Type	Description
`connection`	`PrinterConnection?`	Printer connection; fails if invalid
`cutType`	`Int`	CUT_FULL / CUT_PARTIAL

Returns: `Bool`

`feedAndCutPaper`

Feeds then cuts.

```
public func feedAndCutPaper(_ connection: PrinterConnection?) -> Bool
```

Parameter	Type	Description
`connection`	`PrinterConnection?`	Printer connection; fails if invalid

Returns: `Bool`

`feedLine`

Feeds one line.

```
public func feedLine(_ connection: PrinterConnection?) -> Bool
```

Parameter	Type	Description
`connection`	`PrinterConnection?`	Printer connection; fails if invalid

Returns: `Bool`

`feedPaper`

Feeds paper.

```
public func feedPaper(_ connection: PrinterConnection?, lines: Int) -> Bool
```

Parameter	Type	Description
`connection`	`PrinterConnection?`	Printer connection; fails if invalid
`lines`	`Int`	—

Returns: `Bool`

```
public func feedPaper(_ connection: PrinterConnection?) -> Bool
```

Parameter	Type	Description
`connection`	`PrinterConnection?`	Printer connection; fails if invalid

Returns: `Bool`

`reverseFeed`

Reverse-feeds paper.

```
public func reverseFeed(_ connection: PrinterConnection?, lines: Int) -> Bool
```

Parameter	Type	Description
`connection`	`PrinterConnection?`	Printer connection; fails if invalid
`lines`	`Int`	—

Returns: `Bool`

`printAndFeedDots`

Prints and feeds by dots.

```
public func printAndFeedDots(_ connection: PrinterConnection?, dots: Int) -> Bool
```

Parameter	Type	Description
`connection`	`PrinterConnection?`	Printer connection; fails if invalid
`dots`	`Int`	—

Returns: `Bool`

`printHorizontalLine`

Prints a horizontal rule.

```
public func printHorizontalLine(_ connection: PrinterConnection?, start: Int, end: Int) -> Bool
```

Parameter	Type	Description
`connection`	`PrinterConnection?`	Printer connection; fails if invalid
`start`	`Int`	—
`end`	`Int`	—

Returns: `Bool`

`fixedHorizontalLine`

Prints a fixed-width horizontal line.

```
public func fixedHorizontalLine(_ connection: PrinterConnection?) -> Bool
```

Parameter	Type	Description
`connection`	`PrinterConnection?`	Printer connection; fails if invalid

Returns: `Bool`

`openCashDrawer`

Opens the cash drawer.

```
public func openCashDrawer(_ connection: PrinterConnection?, drawerIndex: Int) -> Bool
```

Parameter	Type	Description
`connection`	`PrinterConnection?`	Printer connection; fails if invalid
`drawerIndex`	`Int`	—

Returns: `Bool`

`beep`

Beeps.

```
public func beep(_ connection: PrinterConnection?) -> Bool
```

Parameter	Type	Description
`connection`	`PrinterConnection?`	Printer connection; fails if invalid

Returns: `Bool`

```
public func beep(_ connection: PrinterConnection?, count: Int, duration: Int) -> Bool
```

Parameter	Type	Description

`connection`	`PrinterConnection?`	Printer connection; fails if invalid
`count`	`Int`	—
`duration`	`Int`	—

Returns: `Bool`

`sendSelfTest`

Sends a self-test command.

```
public func sendSelfTest(_ connection: PrinterConnection?) -> Bool
```

Parameter	Type	Description
`connection`	`PrinterConnection?`	Printer connection; fails if invalid

Returns: `Bool`

`sendCommand`

Sends raw command bytes.

```
public func sendCommand(_ connection: PrinterConnection?, command: Data) -> Bool
```

Parameter	Type	Description
`connection`	`PrinterConnection?`	Printer connection; fails if invalid
`command`	`Data`	—

Returns: `Bool`

9. Page Mode

`selectPageMode`

Enters page mode.

```
public func selectPageMode(_ connection: PrinterConnection?) -> Bool
```

Parameter	Type	Description
`connection`	`PrinterConnection?`	Printer connection; fails if invalid

Returns: `Bool`

`selectPageModeEx`

Enters page mode with area.

```
public func selectPageModeEx( _ connection: PrinterConnection?, nHorizontalMovementUnit: Int, nVerticalMovementUnit: Int, x: Int, y: Int, width: Int, height: Int ) -> Bool
```

Parameter	Type	Description
`connection`	`PrinterConnection?`	Printer connection; fails if invalid
`nHorizontalMovementUnit`	`Int`	—
`nVerticalMovementUnit`	`Int`	—
`x`	`Int`	Coordinate (dots)
`y`	`Int`	Coordinate (dots)
`width`	`Int`	Size (dots)
`height`	`Int`	Size (dots)

Returns: `Bool`

```
public func selectPageModeEx(_ connection: PrinterConnection?, x: Int, y: Int, width: Int, height: Int) -> Bool
```

Parameter	Type	Description
`connection`	`PrinterConnection?`	Printer connection; fails if invalid
`x`	`Int`	Coordinate (dots)
`y`	`Int`	Coordinate (dots)
`width`	`Int`	Size (dots)
`height`	`Int`	Size (dots)

Returns: `Bool`

```
public func selectPageModeEx(_ connection: PrinterConnection?) -> Bool
```

Parameter	Type	Description
`connection`	`PrinterConnection?`	Printer connection; fails if invalid

Returns: `Bool`

`exitPageMode`

Exits page mode.

```
public func exitPageMode(_ connection: PrinterConnection?) -> Bool
```

Parameter	Type	Description
`connection`	`PrinterConnection?`	Printer connection; fails if invalid

Returns: `Bool`

`pagePrintPage`

Prints the page-mode buffer.

```
public func pagePrintPage(_ connection: PrinterConnection?) -> Bool
```

Parameter	Type	Description
`connection`	`PrinterConnection?`	Printer connection; fails if invalid

Returns: `Bool`

`pageClearPage`

Clears the page-mode buffer.

```
public func pageClearPage(_ connection: PrinterConnection?) -> Bool
```

Parameter	Type	Description
`connection`	`PrinterConnection?`	Printer connection; fails if invalid

Returns: `Bool`

`setMovementUnits`

Sets movement units.

```
public func setMovementUnits(_ connection: PrinterConnection?, x: Int, y: Int) -> Bool
```

Parameter	Type	Description
`connection`	`PrinterConnection?`	Printer connection; fails if invalid
`x`	`Int`	Coordinate (dots)
`y`	`Int`	Coordinate (dots)

Returns: `Bool`

`setPrintAreaInPageMode`

Sets page-mode print area.

```
public func setPrintAreaInPageMode(_ connection: PrinterConnection?, x: Int, y: Int, width: Int,
height: Int) -> Bool
```

Parameter	Type	Description
`connection`	`PrinterConnection?`	Printer connection; fails if invalid
`x`	`Int`	Coordinate (dots)

`y`	`Int`	Coordinate (dots)
`width`	`Int`	Size (dots)
`height`	`Int`	Size (dots)

Returns: `Bool`

`setPrintAreaDirection`

Sets page-mode print direction.

```
public func setPrintAreaDirection(_ connection: PrinterConnection?, direction: Int) -> Bool
```

Parameter	Type	Description
`connection`	`PrinterConnection?`	Printer connection; fails if invalid
`direction`	`Int`	—

Returns: `Bool`

`pageSetPageArea`

Sets page-mode print area (alias).

```
public func pageSetPageArea(_ connection: PrinterConnection?, x: Int, y: Int, width: Int,
height: Int) -> Bool
```

Parameter	Type	Description
`connection`	`PrinterConnection?`	Printer connection; fails if invalid
`x`	`Int`	Coordinate (dots)
`y`	`Int`	Coordinate (dots)
`width`	`Int`	Size (dots)
`height`	`Int`	Size (dots)

Returns: `Bool`

`pageSetPageDrawDirection`

Sets page-mode draw direction (alias).

```
public func pageSetPageDrawDirection(_ connection: PrinterConnection?, direction: Int) -> Bool
```

Parameter	Type	Description
`connection`	`PrinterConnection?`	Printer connection; fails if invalid
`direction`	`Int`	—

Returns: `Bool`

`drawLineInPageMode`

Draws a line in page mode.

```
public func drawLineInPageMode(_ connection: PrinterConnection?, x1: Int, y1: Int, x2: Int, y2: Int) -> Bool
```

Parameter	Type	Description
`connection`	`PrinterConnection?`	Printer connection; fails if invalid
`x1`	`Int`	—
`y1`	`Int`	—
`x2`	`Int`	—
`y2`	`Int`	—

Returns: `Bool`

```
public func drawLineInPageMode(_ connection: PrinterConnection?, x1: Int, y1: Int, x2: Int, y2: Int, lineType: Int) -> Bool
```

Parameter	Type	Description
`connection`	`PrinterConnection?`	Printer connection; fails if invalid
`x1`	`Int`	—
`y1`	`Int`	—
`x2`	`Int`	—
`y2`	`Int`	—
`lineType`	`Int`	—

Returns: `Bool`

`drawRectangleInPageMode`

Draws a rectangle in page mode.

```
public func drawRectangleInPageMode(_ connection: PrinterConnection?, mode: Int, x: Int, y: Int, width: Int, height: Int) -> Bool
```

Parameter	Type	Description
`connection`	`PrinterConnection?`	Printer connection; fails if invalid
`mode`	`Int`	—
`x`	`Int`	Coordinate (dots)
`y`	`Int`	Coordinate (dots)
`width`	`Int`	Size (dots)

`height`	`Int`	Size (dots)

Returns: `Bool`

`drawRectangleBorderInPageMode`

Draws a rectangle border in page mode.

```
public func drawRectangleBorderInPageMode(_ connection: PrinterConnection?, x: Int, y: Int, width: Int, height: Int) -> Bool
```

Parameter	Type	Description
`connection`	`PrinterConnection?`	Printer connection; fails if invalid
`x`	`Int`	Coordinate (dots)
`y`	`Int`	Coordinate (dots)
`width`	`Int`	Size (dots)
`height`	`Int`	Size (dots)

Returns: `Bool`

`drawFilledRectangleInPageMode`

Draws a filled rectangle in page mode.

```
public func drawFilledRectangleInPageMode(_ connection: PrinterConnection?, x: Int, y: Int, width: Int, height: Int) -> Bool
```

Parameter	Type	Description
`connection`	`PrinterConnection?`	Printer connection; fails if invalid
`x`	`Int`	Coordinate (dots)
`y`	`Int`	Coordinate (dots)
`width`	`Int`	Size (dots)
`height`	`Int`	Size (dots)

Returns: `Bool`

`drawTextInPageMode`

Draws text in page mode.

```
public func drawTextInPageMode(_ connection: PrinterConnection?, x: Int, y: Int, text: String) -> Bool
```

Parameter	Type	Description
`connection`	`PrinterConnection?`	Printer connection; fails if invalid

`x`	`Int`	Coordinate (dots)
`y`	`Int`	Coordinate (dots)
`text`	`String`	Text

Returns: `Bool`

`drawTextCenteredInPageMode`

Draws centered text in page mode.

```
public func drawTextCenteredInPageMode(_ connection: PrinterConnection?, pageWidth: Int, y: Int, text: String) -> Bool
```

Parameter	Type	Description
`connection`	`PrinterConnection?`	Printer connection; fails if invalid
`pageWidth`	`Int`	—
`y`	`Int`	Coordinate (dots)
`text`	`String`	Text

Returns: `Bool`

```
public func drawTextCenteredInPageMode(_ connection: PrinterConnection?, pageWidth: Int, y: Int, text: String, charWidthDots: Int) -> Bool
```

Parameter	Type	Description
`connection`	`PrinterConnection?`	Printer connection; fails if invalid
`pageWidth`	`Int`	—
`y`	`Int`	Coordinate (dots)
`text`	`String`	Text
`charWidthDots`	`Int`	—

Returns: `Bool`

`drawTextRightInPageMode`

Draws right-aligned text in page mode.

```
public func drawTextRightInPageMode(_ connection: PrinterConnection?, pageWidth: Int, y: Int, text: String) -> Bool
```

Parameter	Type	Description
`connection`	`PrinterConnection?`	Printer connection; fails if invalid
`pageWidth`	`Int`	—

`y`	`Int`	Coordinate (dots)
`text`	`String`	Text

Returns: `Bool`

```
public func drawTextRightInPageMode(_ connection: PrinterConnection?, pageWidth: Int, y: Int, text: String, charWidthDots: Int) -> Bool
```

Parameter	Type	Description
`connection`	`PrinterConnection?`	Printer connection; fails if invalid
`pageWidth`	`Int`	—
`y`	`Int`	Coordinate (dots)
`text`	`String`	Text
`charWidthDots`	`Int`	—

Returns: `Bool`

`drawTextLineLeftRightInPageMode`

Draws left—right text in page mode.

```
public func drawTextLineLeftRightInPageMode(_ connection: PrinterConnection?, pageWidth: Int, y: Int, left: String, right: String) -> Bool
```

Parameter	Type	Description
`connection`	`PrinterConnection?`	Printer connection; fails if invalid
`pageWidth`	`Int`	—
`y`	`Int`	Coordinate (dots)
`left`	`String`	Coordinate (dots)
`right`	`String`	Coordinate (dots)

Returns: `Bool`

```
public func drawTextLineLeftRightInPageMode( _ connection: PrinterConnection?, pageWidth: Int, y: Int, left: String, right: String, charWidthDots: Int ) -> Bool
```

Parameter	Type	Description
`connection`	`PrinterConnection?`	Printer connection; fails if invalid
`pageWidth`	`Int`	—
`y`	`Int`	Coordinate (dots)
`left`	`String`	Coordinate (dots)

`right`	`String`	Coordinate (dots)
`charWidthDots`	`Int`	—

Returns: `Bool`

`drawNvBitmapInPageMode`

Draws NV bitmap in page mode.

```
public func drawNvBitmapInPageMode(_ connection: PrinterConnection?, x: Int, y: Int, nvNumber: Int, mode: Int) -> Bool
```

Parameter	Type	Description
`connection`	`PrinterConnection?`	Printer connection; fails if invalid
`x`	`Int`	Coordinate (dots)
`y`	`Int`	Coordinate (dots)
`nvNumber`	`Int`	—
`mode`	`Int`	—

Returns: `Bool`

`drawNvBitmapCenteredInPageMode`

Draws centered NV bitmap in page mode.

```
public func drawNvBitmapCenteredInPageMode( _ connection: PrinterConnection?, pageWidth: Int, y: Int, nvNumber: Int, mode: Int, imageWidthDots: Int ) -> Bool
```

Parameter	Type	Description
`connection`	`PrinterConnection?`	Printer connection; fails if invalid
`pageWidth`	`Int`	—
`y`	`Int`	Coordinate (dots)
`nvNumber`	`Int`	—
`mode`	`Int`	—
`imageWidthDots`	`Int`	—

Returns: `Bool`

`drawTextUTF8InPageMode`

UTF-8 text in page mode.

```
public func drawTextUTF8InPageMode(_ connection: PrinterConnection?, x: Int, y: Int, text: String) -> Bool
```

Parameter	Type	Description
`connection`	`PrinterConnection?`	Printer connection; fails if invalid
`x`	`Int`	Coordinate (dots)
`y`	`Int`	Coordinate (dots)
`text`	`String`	Text

Returns: `Bool`

`drawTextGBKInPageMode`

GBK text in page mode.

```
public func drawTextGBKInPageMode(_ connection: PrinterConnection?, x: Int, y: Int, text: String) -> Bool
```

Parameter	Type	Description
`connection`	`PrinterConnection?`	Printer connection; fails if invalid
`x`	`Int`	Coordinate (dots)
`y`	`Int`	Coordinate (dots)
`text`	`String`	Text

Returns: `Bool`

`drawTextBIG5InPageMode`

BIG5 text in page mode.

```
public func drawTextBIG5InPageMode(_ connection: PrinterConnection?, x: Int, y: Int, text: String) -> Bool
```

Parameter	Type	Description
`connection`	`PrinterConnection?`	Printer connection; fails if invalid
`x`	`Int`	Coordinate (dots)
`y`	`Int`	Coordinate (dots)
`text`	`String`	Text

Returns: `Bool`

`drawTextShiftJISInPageMode`

Shift–JIS text in page mode.

```
public func drawTextShiftJISInPageMode(_ connection: PrinterConnection?, x: Int, y: Int, text: String) -> Bool
```

Parameter	Type	Description
`connection`	`PrinterConnection?`	Printer connection; fails if invalid
`x`	`Int`	Coordinate (dots)
`y`	`Int`	Coordinate (dots)
`text`	`String`	Text

Returns: `Bool`

`drawTextEUCKRInPageMode`

EUC-KR text in page mode.

```
public func drawTextEUCKRInPageMode(_ connection: PrinterConnection?, x: Int, y: Int, text: String) -> Bool
```

Parameter	Type	Description
`connection`	`PrinterConnection?`	Printer connection; fails if invalid
`x`	`Int`	Coordinate (dots)
`y`	`Int`	Coordinate (dots)
`text`	`String`	Text

Returns: `Bool`

`drawImageFromFileInPageMode`

Draws image from file in page mode.

```
public func drawImageFromFileInPageMode( _ connection: PrinterConnection?, x: Int, y: Int, filePath: String, width: Int, height: Int ) -> Bool
```

Parameter	Type	Description
`connection`	`PrinterConnection?`	Printer connection; fails if invalid
`x`	`Int`	Coordinate (dots)
`y`	`Int`	Coordinate (dots)
`filePath`	`String`	File path
`width`	`Int`	Size (dots)
`height`	`Int`	Size (dots)

Returns: `Bool`

`drawImageFromDataInPageMode`

Draws image from Data in page mode.

```
public func drawImageFromDataInPageMode( _ connection: PrinterConnection?, x: Int, y: Int,
imageData: Data, width: Int, height: Int ) -> Bool
```

Parameter	Type	Description
`connection`	`PrinterConnection?`	Printer connection; fails if invalid
`x`	`Int`	Coordinate (dots)
`y`	`Int`	Coordinate (dots)
`imageData`	`Data`	Image data
`width`	`Int`	Size (dots)
`height`	`Int`	Size (dots)

Returns: `Bool`

`drawImageFromPixelsInPageMode`

Draws image from pixels in page mode.

```
public func drawImageFromPixelsInPageMode( _ connection: PrinterConnection?, x: Int, y: Int,
pixels: [UInt32], imgWidth: Int, imgHeight: Int, width: Int, height: Int ) -> Bool
```

Parameter	Type	Description
`connection`	`PrinterConnection?`	Printer connection; fails if invalid
`x`	`Int`	Coordinate (dots)
`y`	`Int`	Coordinate (dots)
`pixels`	`[UInt32]`	ARGB pixels
`imgWidth`	`Int`	—
`imgHeight`	`Int`	—
`width`	`Int`	Size (dots)
`height`	`Int`	Size (dots)

Returns: `Bool`

`drawBarcodeInPageMode`

Draws barcode in page mode.

```
public func drawBarcodeInPageMode( _ connection: PrinterConnection?, x: Int, y: Int,
barcodeType: Int, data: String, height: Int = 55, width: Int = 4 ) -> Bool
```

Parameter	Type	Description
`connection`	`PrinterConnection?`	Printer connection; fails if invalid
`x`	`Int`	Coordinate (dots)

`y`	`Int`	Coordinate (dots)
`barcodeType`	`Int`	—
`data`	`String`	Payload
`height`	`Int`	Size (dots); default `55`
`width`	`Int`	Size (dots); default `4`

Returns: `Bool`

```
public func drawBarcodeInPageMode( _ connection: PrinterConnection?, x: Int, y: Int,
barcodeType: Int, data: String, hriPos: Int, height: Int, width: Int ) -> Bool
```

Parameter	Type	Description
`connection`	`PrinterConnection?`	Printer connection; fails if invalid
`x`	`Int`	Coordinate (dots)
`y`	`Int`	Coordinate (dots)
`barcodeType`	`Int`	—
`data`	`String`	Payload
`hriPos`	`Int`	—
`height`	`Int`	Size (dots)
`width`	`Int`	Size (dots)

Returns: `Bool`

`drawBarcodeInPageModeWithCenteredHri`

Draws barcode with centered HRI in page mode.

```
public func drawBarcodeInPageModeWithCenteredHri( _ connection: PrinterConnection?, x: Int, y:
Int, barcodeType: Int, data: String, height: Int, width: Int, hriGapDots: Int ) -> Bool
```

Parameter	Type	Description
`connection`	`PrinterConnection?`	Printer connection; fails if invalid
`x`	`Int`	Coordinate (dots)
`y`	`Int`	Coordinate (dots)
`barcodeType`	`Int`	—
`data`	`String`	Payload
`height`	`Int`	Size (dots)
`width`	`Int`	Size (dots)
`hriGapDots`	`Int`	—

Returns: `Bool`

`drawQRCodeInPageMode`

Draws QR code in page mode.

```
public func drawQRCodeInPageMode(_ connection: PrinterConnection?, x: Int, y: Int, version: Int, eccLevel: Int, data: String) -> Bool
```

Parameter	Type	Description
`connection`	`PrinterConnection?`	Printer connection; fails if invalid
`x`	`Int`	Coordinate (dots)
`y`	`Int`	Coordinate (dots)
`version`	`Int`	—
`eccLevel`	`Int`	—
`data`	`String`	Payload

Returns: `Bool`

10. Label Mode (JPL)

`enableLabelMode`

Enables label (JPL) mode.

```
public func enableLabelMode(_ connection: PrinterConnection?) -> Bool
```

Parameter	Type	Description
`connection`	`PrinterConnection?`	Printer connection; fails if invalid

Returns: `Bool`

`disableLabelMode`

Disables label mode.

```
public func disableLabelMode(_ connection: PrinterConnection?) -> Bool
```

Parameter	Type	Description
`connection`	`PrinterConnection?`	Printer connection; fails if invalid

Returns: `Bool`

`labelPageBegin`

Begins a label page.

```
public func labelPageBegin(_ connection: PrinterConnection?, x: Int, y: Int, width: Int, height: Int, rotation: Int) -> Bool
```

Parameter	Type	Description
`connection`	`PrinterConnection?`	Printer connection; fails if invalid
`x`	`Int`	Coordinate (dots)
`y`	`Int`	Coordinate (dots)
`width`	`Int`	Size (dots)
`height`	`Int`	Size (dots)
`rotation`	`Int`	Rotation

Returns: `Bool`

`backPaperToPrintPosition`

Backs paper to print position.

```
public func backPaperToPrintPosition(_ connection: PrinterConnection?) -> Bool
```

Parameter	Type	Description
`connection`	`PrinterConnection?`	Printer connection; fails if invalid

Returns: `Bool`

`labelPagePrint`

Prints the label page.

```
public func labelPagePrint(_ connection: PrinterConnection?) -> Bool
```

Parameter	Type	Description
`connection`	`PrinterConnection?`	Printer connection; fails if invalid

Returns: `Bool`

```
public func labelPagePrint(_ connection: PrinterConnection?, printNum: Int) -> Bool
```

Parameter	Type	Description
`connection`	`PrinterConnection?`	Printer connection; fails if invalid
`printNum`	`Int`	Copy count

Returns: `Bool`

`printLabelText`

Draws label text.

```
public func printLabelText(_ connection: PrinterConnection?, text: String, x: Int, y: Int, fontSize: Int) -> Bool
```

Parameter	Type	Description
`connection`	`PrinterConnection?`	Printer connection; fails if invalid
`text`	`String`	Text
`x`	`Int`	Coordinate (dots)
`y`	`Int`	Coordinate (dots)
`fontSize`	`Int`	—

Returns: `Bool`

```
public func printLabelText(_ connection: PrinterConnection?, text: String, x: Int, y: Int, fontHeight: Int, fontType: Int) -> Bool
```

Parameter	Type	Description
`connection`	`PrinterConnection?`	Printer connection; fails if invalid
`text`	`String`	Text
`x`	`Int`	Coordinate (dots)
`y`	`Int`	Coordinate (dots)
`fontHeight`	`Int`	—
`fontType`	`Int`	—

Returns: `Bool`

`printLabelTextUTF8`

Draws UTF-8 label text.

```
public func printLabelTextUTF8(_ connection: PrinterConnection?, text: String, x: Int, y: Int) -> Bool
```

Parameter	Type	Description
`connection`	`PrinterConnection?`	Printer connection; fails if invalid
`text`	`String`	Text
`x`	`Int`	Coordinate (dots)
`y`	`Int`	Coordinate (dots)

Returns: `Bool`

```
public func printLabelTextUTF8(_ connection: PrinterConnection?, text: String, x: Int, y: Int, fontHeight: Int, fontType: Int) -> Bool
```

Parameter	Type	Description
`connection`	`PrinterConnection?`	Printer connection; fails if invalid
`text`	`String`	Text
`x`	`Int`	Coordinate (dots)
`y`	`Int`	Coordinate (dots)
`fontHeight`	`Int`	—
`fontType`	`Int`	—

Returns: `Bool`

`printLabelTextGBK`

Draws GBK label text.

```
public func printLabelTextGBK(_ connection: PrinterConnection?, text: String, x: Int, y: Int) -> Bool
```

Parameter	Type	Description
`connection`	`PrinterConnection?`	Printer connection; fails if invalid
`text`	`String`	Text
`x`	`Int`	Coordinate (dots)
`y`	`Int`	Coordinate (dots)

Returns: `Bool`

```
public func printLabelTextGBK(_ connection: PrinterConnection?, text: String, x: Int, y: Int, fontHeight: Int, fontType: Int) -> Bool
```

Parameter	Type	Description
`connection`	`PrinterConnection?`	Printer connection; fails if invalid
`text`	`String`	Text
`x`	`Int`	Coordinate (dots)
`y`	`Int`	Coordinate (dots)
`fontHeight`	`Int`	—
`fontType`	`Int`	—

Returns: `Bool`

`printLabelBarcode`

Draws label barcode.

```
public func printLabelBarcode(_ connection: PrinterConnection?, type: Int, data: String, x: Int, y: Int, height: Int, width: Int) -> Bool
```

Parameter	Type	Description
`connection`	`PrinterConnection?`	Printer connection; fails if invalid
`type`	`Int`	—
`data`	`String`	Payload
`x`	`Int`	Coordinate (dots)
`y`	`Int`	Coordinate (dots)
`height`	`Int`	Size (dots)
`width`	`Int`	Size (dots)

Returns: `Bool`

```
public func printLabelBarcode( _ connection: PrinterConnection?, x: Int, y: Int, barcodeType: Int, barcodeHeight: Int, unitWidth: Int, rotate: Int, data: String ) -> Bool
```

Parameter	Type	Description
`connection`	`PrinterConnection?`	Printer connection; fails if invalid
`x`	`Int`	Coordinate (dots)
`y`	`Int`	Coordinate (dots)
`barcodeType`	`Int`	—
`barcodeHeight`	`Int`	—
`unitWidth`	`Int`	—
`rotate`	`Int`	Rotation
`data`	`String`	Payload

Returns: `Bool`

`setLabelBarcodeTextPosition`

Sets label barcode HRI position.

```
public func setLabelBarcodeTextPosition(_ connection: PrinterConnection?, position: Int) -> Bool
```

Parameter	Type	Description
`connection`	`PrinterConnection?`	Printer connection; fails if invalid

`position`	`Int`	—

Returns: `Bool`

`printLabelQRCode`

Draws label QR code.

```
public func printLabelQRCode(_ connection: PrinterConnection?, data: String, x: Int, y: Int, size: Int) -> Bool
```

Parameter	Type	Description
`connection`	`PrinterConnection?`	Printer connection; fails if invalid
`data`	`String`	Payload
`x`	`Int`	Coordinate (dots)
`y`	`Int`	Coordinate (dots)
`size`	`Int`	—

Returns: `Bool`

```
public func printLabelQRCode( _ connection: PrinterConnection?, x: Int, y: Int, version: Int, eccLevel: Int, unitWidth: Int, rotate: Int, data: String ) -> Bool
```

Parameter	Type	Description
`connection`	`PrinterConnection?`	Printer connection; fails if invalid
`x`	`Int`	Coordinate (dots)
`y`	`Int`	Coordinate (dots)
`version`	`Int`	—
`eccLevel`	`Int`	—
`unitWidth`	`Int`	—
`rotate`	`Int`	Rotation
`data`	`String`	Payload

Returns: `Bool`

`printLabelPDF417Code`

Draws label PDF417.

```
public func printLabelPDF417Code( _ connection: PrinterConnection?, x: Int, y: Int, colNum: Int, eccLevel: Int, lwRatio: Int, unitWidth: Int, rotate: Int, data: String ) -> Bool
```

Parameter	Type	Description

`connection`	`PrinterConnection?`	Printer connection; fails if invalid
`x`	`Int`	Coordinate (dots)
`y`	`Int`	Coordinate (dots)
`colNum`	`Int`	—
`eccLevel`	`Int`	—
`lwRatio`	`Int`	—
`unitWidth`	`Int`	—
`rotate`	`Int`	Rotation
`data`	`String`	Payload

Returns: `Bool`

```
public func printLabelPDF417Code(_ connection: PrinterConnection?, x: Int, y: Int, data: String)
-> Bool
```

Parameter	Type	Description
`connection`	`PrinterConnection?`	Printer connection; fails if invalid
`x`	`Int`	Coordinate (dots)
`y`	`Int`	Coordinate (dots)
`data`	`String`	Payload

Returns: `Bool`

`drawLabelImageFromFile`

Draws label bitmap from file.

```
public func drawLabelImageFromFile( _ connection: PrinterConnection?, x: Int, y: Int, filePath:
String, targetWidth: Int = 0 ) -> Bool
```

Parameter	Type	Description
`connection`	`PrinterConnection?`	Printer connection; fails if invalid
`x`	`Int`	Coordinate (dots)
`y`	`Int`	Coordinate (dots)
`filePath`	`String`	File path
`targetWidth`	`Int`	Target width (dots); for labels ⩽0 uses source width (max 576); default `0`

Returns: `Bool`

`drawLabelImageFromData`

Draws label bitmap from Data.

```
public func drawLabelImageFromData( _ connection: PrinterConnection?, x: Int, y: Int, imageData: Data, targetWidth: Int = 0 ) -> Bool
```

Parameter	Type	Description
`connection`	`PrinterConnection?`	Printer connection; fails if invalid
`x`	`Int`	Coordinate (dots)
`y`	`Int`	Coordinate (dots)
`imageData`	`Data`	Image data
`targetWidth`	`Int`	Target width (dots); for labels ⩽0 uses source width (max 576); default `0`

Returns: `Bool`

`drawLabelImageFromPixels`

Draws label bitmap from pixels.

```
public func drawLabelImageFromPixels( _ connection: PrinterConnection?, x: Int, y: Int, pixels: [UInt32], width: Int, height: Int, targetWidth: Int = 0 ) -> Bool
```

Parameter	Type	Description
`connection`	`PrinterConnection?`	Printer connection; fails if invalid
`x`	`Int`	Coordinate (dots)
`y`	`Int`	Coordinate (dots)
`pixels`	`[UInt32]`	ARGB pixels
`width`	`Int`	Size (dots)
`height`	`Int`	Size (dots)
`targetWidth`	`Int`	Target width (dots); for labels ⩽0 uses source width (max 576); default `0`

Returns: `Bool`

`drawLabelImageAdvanced`

Advanced label bitmap draw.

```
public func drawLabelImageAdvanced( _ connection: PrinterConnection?, x: Int, y: Int, pixels: [UInt32], width: Int, height: Int, showType: Int, targetWidth: Int = 0 ) -> Bool
```

Parameter	Type	Description

`connection`	`PrinterConnection?`	Printer connection; fails if invalid
`x`	`Int`	Coordinate (dots)
`y`	`Int`	Coordinate (dots)
`pixels`	`[UInt32]`	ARGB pixels
`width`	`Int`	Size (dots)
`height`	`Int`	Size (dots)
`showType`	`Int`	—
`targetWidth`	`Int`	Target width (dots); for labels ⩽0 uses source width (max 576); default `0`

Returns: `Bool`

`drawLabelLine`

Draws a label line.

```
public func drawLabelLine(_ connection: PrinterConnection?, startX: Int, startY: Int, endX: Int, endY: Int) -> Bool
```

Parameter	Type	Description
`connection`	`PrinterConnection?`	Printer connection; fails if invalid
`startX`	`Int`	Coordinate (dots)
`startY`	`Int`	Coordinate (dots)
`endX`	`Int`	Coordinate (dots)
`endY`	`Int`	Coordinate (dots)

Returns: `Bool`

```
public func drawLabelLine( _ connection: PrinterConnection?, startX: Int, startY: Int, endX: Int, endY: Int, width: Int, color: Int ) -> Bool
```

Parameter	Type	Description
`connection`	`PrinterConnection?`	Printer connection; fails if invalid
`startX`	`Int`	Coordinate (dots)
`startY`	`Int`	Coordinate (dots)
`endX`	`Int`	Coordinate (dots)
`endY`	`Int`	Coordinate (dots)
`width`	`Int`	Size (dots)
`color`	`Int`	—

Returns: `Bool`

`drawLabelRect`

Draws a label rectangle.

```
public func drawLabelRect(_ connection: PrinterConnection?, left: Int, top: Int, right: Int,
bottom: Int) -> Bool
```

Parameter	Type	Description
`connection`	`PrinterConnection?`	Printer connection; fails if invalid
`left`	`Int`	Coordinate (dots)
`top`	`Int`	Coordinate (dots)
`right`	`Int`	Coordinate (dots)
`bottom`	`Int`	Coordinate (dots)

Returns: `Bool`

```
public func drawLabelRect( _ connection: PrinterConnection?, left: Int, top: Int, right: Int,
bottom: Int, width: Int, color: Int ) -> Bool
```

Parameter	Type	Description
`connection`	`PrinterConnection?`	Printer connection; fails if invalid
`left`	`Int`	Coordinate (dots)
`top`	`Int`	Coordinate (dots)
`right`	`Int`	Coordinate (dots)
`bottom`	`Int`	Coordinate (dots)
`width`	`Int`	Size (dots)
`color`	`Int`	—

Returns: `Bool`

`drawLabelBox`

Draws a filled label box.

```
public func drawLabelBox(_ connection: PrinterConnection?, left: Int, top: Int, right: Int,
bottom: Int, color: Int) -> Bool
```

Parameter	Type	Description
`connection`	`PrinterConnection?`	Printer connection; fails if invalid
`left`	`Int`	Coordinate (dots)
`top`	`Int`	Coordinate (dots)
`right`	`Int`	Coordinate (dots)

`bottom`	`Int`	Coordinate (dots)
`color`	`Int`	—

Returns: `Bool`

11. Utilities

`estimateCode128WidthDots`

Estimates Code128 width in dots.

```
public func estimateCode128WidthDots(_ data: String, moduleWidth: Int) -> Int
```

Parameter	Type	Description
`data`	`String`	Payload
`moduleWidth`	`Int`	—

Returns: `Int`

`estimateTextWidthDots`

Estimates text width in dots.

```
public func estimateTextWidthDots(_ text: String, charWidthDots: Int) -> Int
```

Parameter	Type	Description
`text`	`String`	Text
`charWidthDots`	`Int`	—

Returns: `Int`

---

4. Constants `PrinterConstants`

Constant	Value	Description
`ALIGNMENT_LEFT`	`0`	Left
`ALIGNMENT_CENTER`	`1`	Center
`ALIGNMENT_RIGHT`	`2`	Right
`CUT_FULL`	`0`	Full cut
`CUT_PARTIAL`	`1`	Partial cut
`PAPER_WIDTH_58MM`	`384`	58 mm printable width
`PAPER_WIDTH_80MM`	`576`	80 mm effective (~72 mm)
`PAPER_WIDTH_72MM`	`PAPER_WIDTH_80MM`	Same as PAPER_WIDTH_80MM
`CHAR_COLUMNS_72MM`	`48`	Font A columns at 72 mm

`NV_BITMAP_MODE_NORMAL`	`0`	—
`NV_BITMAP_MODE_DOUBLE_WIDTH`	`1`	—
`NV_BITMAP_MODE_DOUBLE_HEIGHT`	`2`	—
`NV_BITMAP_MODE_DOUBLE_BOTH`	`3`	—
`BARCODE_HRI_NONE`	`0`	—
`BARCODE_HRI_BELOW`	`2`	—
`BARCODE_TYPE_UPC_A`	`0x41`	—
`BARCODE_TYPE_UPC_E`	`0x42`	—
`BARCODE_TYPE_EAN13`	`0x43`	—
`BARCODE_TYPE_EAN8`	`0x44`	—
`BARCODE_TYPE_CODE39`	`0x45`	—
`BARCODE_TYPE_ITF`	`0x46`	—
`BARCODE_TYPE_CODABAR`	`0x47`	—
`BARCODE_TYPE_CODE93`	`0x48`	—
`BARCODE_TYPE_CODE128`	`0x49`	—
`BARCODE_CODE128`	`BARCODE_TYPE_CODE128`	—
`BARCODE_CODE39`	`BARCODE_TYPE_CODE39`	—
`BARCODE_EAN13`	`BARCODE_TYPE_EAN13`	—
`QRCODE_SIZE_SMALL`	`1`	—
`QRCODE_SIZE_NORMAL`	`2`	—
`QRCODE_SIZE_LARGE`	`3`	—
`QR_ECC_LEVEL_L`	`1`	—
`QR_ECC_LEVEL_M`	`2`	—
`QR_ECC_LEVEL_Q`	`3`	—
`QR_ECC_LEVEL_H`	`4`	—
`PAGE_DIRECTION_LEFT_TO_RIGHT`	`0`	—
`PAGE_DIRECTION_BOTTOM_TO_TOP`	`1`	—
`PAGE_DIRECTION_RIGHT_TO_LEFT`	`2`	—
`PAGE_DIRECTION_TOP_TO_BOTTOM`	`3`	—
`STATUS_OFFLINE`	`0x00`	Offline
`STATUS_ONLINE`	`0x01`	Online
`STATUS_PAPER_OUT`	`0x02`	Paper out
`STATUS_COVER_OPEN`	`0x04`	Cover open
`STATUS_TYPE_PRINTER`	`1`	Printer status

`STATUS_TYPE_OFFLINE`	`2`	Offline status
`STATUS_TYPE_ERROR`	`3`	Error status
`STATUS_TYPE_PAPER_SENSOR`	`4`	Paper sensor
`CASH_DRAWER_PIN2`	`0`	Cash drawer pin2
`CASH_DRAWER_PIN5`	`1`	Cash drawer pin5
`BEEP_DEFAULT_COUNT`	`1`	—
`BEEP_DEFAULT_DURATION`	`100`	—
`RASTER_COMPRESSION_GS_V0`	`0`	GS v 0 (actual output)
`RASTER_COMPRESSION_LINE_RLE`	`1`	RLE (falls back to GS v 0)
`RASTER_COMPRESSION_JBIG`	`2`	JBIG (falls back to GS v 0)
`RASTER_COMPRESSION_AUTO`	`3`	Auto (falls back to GS v 0)
`LABEL_FONT_HEIGHT_24`	`24`	—
`LABEL_BARCODE_TYPE_CODE128`	`8`	—
`LABEL_BARCODE_HEIGHT_DEFAULT`	`80`	—
`LABEL_BARCODE_UNIT_WIDTH_2`	`2`	—
`DOUBLE_BYTE_ENCODING_GBK`	`0`	GBK
`DOUBLE_BYTE_ENCODING_UTF8`	`1`	UTF-8
`DOUBLE_BYTE_ENCODING_SHIFT_JIS`	`2`	Shift-JIS
`DOUBLE_BYTE_ENCODING_BIG5`	`3`	BIG5
`DOUBLE_BYTE_ENCODING_EUC_KR`	`4`	EUC-KR
`ASCII_FONT_TYPE_A`	`0`	—
`ASCII_FONT_TYPE_B`	`1`	—
`LINE_SPACING_DEFAULT`	`0`	—
`LINE_SPACING_1_8`	`1`	—
`LINE_SPACING_1_6`	`2`	—
`PRINT_MODE_NORMAL`	`0`	—
`PRINT_MODE_DOUBLE_WIDTH`	`1`	—
`PRINT_MODE_DOUBLE_HEIGHT`	`2`	—
`PRINT_MODE_DOUBLE`	`3`	—
`PRINT_MODE_BOLD`	`8`	—
`PRINT_MODE_REVERSE`	`16`	—
`PRINT_MODE_UNDERLINE`	`128`	—
`CHINESE_DOUBLE_SIZE_NORMAL`	`0`	CJK double size off
`CHINESE_DOUBLE_SIZE_ON`	`1`	CJK double size on

5. Quick Index

- `addBluetoothListener`
- `backPaperToPrintPosition`
- `beep`
- `cancelPrint`
- `cancelPrintCmd`
- `clearBuffer`
- `clearPrinterBuffer`
- `connectBluetoothPrinter`
- `connectNetworkPrinter`
- `connectNetworkPrinterAsync`
- `cutPaper`
- `disableLabelMode`
- `disconnectPrinter`
- `drawBarcodeInPageMode`
- `drawBarcodeInPageModeWithCenteredHri`
- `drawFilledRectangleInPageMode`
- `drawImageFromDataInPageMode`
- `drawImageFromFileInPageMode`
- `drawImageFromPixelsInPageMode`
- `drawLabelBox`
- `drawLabelImageAdvanced`
- `drawLabelImageFromData`
- `drawLabelImageFromFile`
- `drawLabelImageFromPixels`
- `drawLabelLine`
- `drawLabelRect`
- `drawLineInPageMode`
- `drawNvBitmapCenteredInPageMode`
- `drawNvBitmapInPageMode`
- `drawQRCodeInPageMode`
- `drawRectangleBorderInPageMode`
- `drawRectangleInPageMode`
- `drawTextBIG5InPageMode`
- `drawTextCenteredInPageMode`
- `drawTextEUCKRInPageMode`
- `drawTextGBKInPageMode`
- `drawTextInPageMode`
- `drawTextLineLeftRightInPageMode`
- `drawTextRightInPageMode`
- `drawTextShiftJISInPageMode`
- `drawTextUTF8InPageMode`
- `enableLabelMode`
- `estimateCode128WidthDots`
- `estimateTextWidthDots`
- `exitPageMode`
- `feedAndCutPaper`
- `feedLine`
- `feedPaper`

- `fixedHorizontalLine`
- `getBluetoothDevices`
- `getCommunicationManager`
- `getPrinterStatus`
- `hasPaper`
- `horizontalTab`
- `initPrinter`
- `isCoverOpen`
- `isPrinterOnline`
- `labelPageBegin`
- `labelPagePrint`
- `openCashDrawer`
- `pageClearPage`
- `pagePrintPage`
- `pageSetPageArea`
- `pageSetPageDrawDirection`
- `printAndFeedDots`
- `printBarcode`
- `printDoubleQRCode`
- `printHorizontalLine`
- `printImage`
- `printLabelBarcode`
- `printLabelPDF417Code`
- `printLabelQRCode`
- `printLabelText`
- `printLabelTextGBK`
- `printLabelTextUTF8`
- `printNvBitmap`
- `printPDF417`
- `printQRCode`
- `printQRCodeStandard`
- `printRasterImageFromData`
- `printRasterImageFromFile`
- `printRasterImageFromPixels`
- `printTabLine`
- `printText`
- `printTextInBIG5`
- `printTextInEUCKR`
- `printTextInGBK`
- `printTextInShiftJIS`
- `printTextInUTF8`
- `printTextLineLeftRight`
- `printTextLineLeftRightGBK`
- `queryRTStatus`
- `queryTransmitStatus`
- `refreshBluetoothDevices`
- `release`
- `removeBluetoothListener`
- `resetPrinter`
- `reverseFeed`

- `selectPageMode`
- `selectPageModeEx`
- `sendCommand`
- `sendSelfTest`
- `setAlignment`
- `setAsciiTextCharRightSpacing`
- `setAsciiTextFontType`
- `setBarcodeHRIPosition`
- `setBarcodeHeight`
- `setBarcodeReadableTextFontType`
- `setBarcodeReadableTextPosition`
- `setBarcodeWidth`
- `setBold`
- `setCharSpacing`
- `setCharacterCodepage`
- `setCharacterSet`
- `setChineseDoubleSize`
- `setChineseMode`
- `setDoubleByteEncoding`
- `setHorizontalAbsolutePrintPosition`
- `setHorizontalRelativePrintPosition`
- `setKanjiTextCharSpacing`
- `setLabelBarcodeTextPosition`
- `setLineSpacing`
- `setMovementUnits`
- `setPrintAreaDirection`
- `setPrintAreaInPageMode`
- `setPrintAreaLeftMargin`
- `setPrintAreaWidth`
- `setPrintMode`
- `setPrintPosition`
- `setReverse`
- `setSingleByteMode`
- `setTabPositions`
- `setTextLineHeight`
- `setTextRotate`
- `setTextScale`
- `setTextUpsideDown`
- `setUnderline`
- `setVerticalAbsolutePrintPosition`
- `setVerticalRelativePrintPosition`
- `shared`
- `startBluetoothDiscovery`
- `stopBluetoothDiscovery`

Overloads are defined in `PrinterSDK.swift`.