

# PrinterSDK API Documentation (Windows / C++)

Version: 1.6.3-windows-cpp

Namespace: `hspos`

Entry: `PrinterSdk`

Constants: `PrinterConstants`

Connection: `std::shared_ptr<IPrinterConnection>`

Header: `#include "hspos/HsposPrinterSdk.hpp"`

Companion docs: `PrinterSDK_User_Manual.md`, `PrinterSDK_Windows_CPP_API对照表.md`

Last updated: 2026-07-18

Aligned with Android `PrinterSDK_API_Documentation.pdf` (1.6.3) and the Windows C# handbook style.

---

## Table of Contents

---

1. SDK Introduction
2. FAQ
3. Enums & Macros ( `PrinterConstants` )
4. Interface Specification
5. Not Yet Implemented / Not Ported

- Lifecycle
- `GetInstance`
- `Release`
- Port / Connection
- `ConnectNetworkPrinter`
- `ConnectSerialPrinter`
- `ConnectUsbPrinter`
- `ConnectParallelPrinter`
- `ConnectBluetoothPrinter`
- `DisconnectPrinter`
- `ListSerialPorts`
- `GetUsbDevices`
- `GetParallelDevices`
- `IsBluetoothAvailable`
- `GetBluetoothPairedDevices`
- `DiscoverBluetoothDevices`
- `StopBluetoothDiscovery`
- `PairBluetoothDevice`
- Printer Status
- `QueryRtStatus`
- `QueryTransmitStatus`
- `GetPrinterStatus`
- `IsPrinterOnline`
- `HasPaper`
- `IsCoverOpen`
- POS Printing
- `InitPrinter`
- `ResetPrinter`

- PrintText
- PrintTextInUtf8
- PrintTextInGbk
- PrintTextLineLeftRight
- SetAlignment
- SetBold
- SetUnderline
- SetReverse
- SetTextScale
- SetLineSpacing
- SetChineseMode
- SetSingleByteMode
- SetDoubleByteEncoding
- SetChineseDoubleSize
- SetPrintAreaWidth
- SetPrintAreaLeftMargin
- SetBarcodeHeight
- SetBarcodeWidth
- SetBarcodeHriPosition
- FeedLine
- FeedPaper
- PrintAndFeedLines
- CutPaper
- FeedAndCutPaper
- PrintBarcode
- PrintQrCode
- PrintQrCodeStandard
- PrintPdf417
- PrintDoubleQrCode
- PrintDoubleQrCodeManualExample
- EstimateQrSymbolWidthDots
- OpenCashDrawer
- Beep
- SelfTest
- SendCommand
- PrintRasterImageFromPixels
- PrintRasterImageFromFile
- PrintRasterImageFromData
- Page Mode
- SelectPageMode
- SelectPageModeEx
- ExitPageMode
- PagePrintPage
- CancelPrintDataInPageMode
- PageClearPage
- SetMovementUnits
- SetPrintAreaInPageMode
- SetPageModePrintDirection
- DrawTextInPageMode
- DrawTextUtf8InPageMode
- DrawTextGbkInPageMode
- DrawTextCenteredInPageMode
- DrawTextRightInPageMode
- DrawTextLineLeftRightInPageMode
- DrawBarcodeInPageMode

- DrawBarcodeInPageModeWithCenteredHri
- DrawQrCodeInPageMode
- DrawLineInPageMode
- DrawRectangleInPageMode
- EstimateCode128WidthDots
- EstimateTextWidthDots
- Label Printing
- EnableLabelMode
- DisableLabelMode
- LabelBackPaper
- LabelPageBegin
- LabelPagePrint
- PrintLabelText
- PrintLabelTextUtf8
- PrintLabelTextGbk
- PrintLabelBarcode
- PrintLabelQrCode
- PrintLabelPdf417Code
- PrintLabelLine
- PrintLabelRectangle
- SetLabelBarcodeTextPosition
- DrawLabelImageFromFile
- DrawLabelImageFromData
- DrawLabelImageFromPixels
- DrawLabelImageAdvanced
- DrawLabelBox

## SDK Introduction

---

The Windows C++ HS-POS PrinterSDK aligns with Android 1.6.3 / Windows C# APIs for ESC/POS receipts and JPL labels. Use `PrinterSdk::GetInstance()` for connect, print, and status.

### Typical flow

1. `GetInstance()`
2. `ConnectXxxPrinter(...)` → `shared_ptr<IPrinterConnection>`
3. `InitPrinter(conn)`
4. Text / barcode / page mode / label / raster APIs
5. `FeedAndCutPaper(conn)` or label `LabelPagePrint`
6. `DisconnectPrinter(conn)`

### Return value semantics

- Most print APIs return `bool` (write success)
- Connect APIs return `shared_ptr` (empty on failure)
- Status APIs return `int` / `bool` interpreted via `PrinterConstants`

## FAQ

---

### Garbled Chinese?

Call `SetChineseMode`, then `SetDoubleByteEncoding(..., DoubleByteEncodingGbk)`, and print with `PrintTextInGbk`.

## Cut position wrong?

The cutter sits ~10–20mm above the print head. Prefer `FeedAndCutPaper` , or feed then `CutPaper` .  
Note: C++ `FeedPaper(n)` uses **dots** (ESC J), not lines; use `PrintAndFeedLines` for line feeds.

## USB prints but status is wrong?

Open USB read/write capable; reconnect after upgrading. Prefer `GetPrinterStatus` / `HasPaper` / `IsCoverOpen` .

## Android permission APIs?

Not applicable on Windows. Install drivers and connect directly.

## Enums & Macros ( `PrinterConstants` )

Defined in `include/hspos/PrinterConstants.hpp` (use `PrinterConstants::Xxx` ).

| Constant | Definition |
|----------|------------|
|          |            |

```
| AlignmentLeft | int = 0 |
| AlignmentCenter | int = 1 |
| AlignmentRight | int = 2 |
| CutFull | int = 0 |
| CutPartial | int = 1 |
| PaperWidth58Mm | int = 384 |
| PaperWidth80Mm | int = 576 |
| BarcodeHriNone | int = 0 |
| BarcodeHriBelow | int = 2 |
| BarcodeTypeUpca | int = 0x41 |
| BarcodeTypeUpce | int = 0x42 |
| BarcodeTypeEan13 | int = 0x43 |
| BarcodeTypeEan8 | int = 0x44 |
| BarcodeTypeCode39 | int = 0x45 |
| BarcodeTypeItf | int = 0x46 |
| BarcodeTypeCodabar | int = 0x47 |
| BarcodeTypeCode93 | int = 0x48 |
| BarcodeTypeCode128 | int = 0x49 |
| QrEccLevelL | int = 1 |
| QrEccLevelM | int = 2 |
| QrEccLevelQ | int = 3 |
| QrEccLevelH | int = 4 |
| StatusOffline | int = 0x00 |
| StatusOnline | int = 0x01 |
| StatusPaperOut | int = 0x02 |
| StatusCoverOpen | int = 0x04 |
| StatusTypePrinter | int = 1 |
| StatusTypeOffline | int = 2 |
| StatusTypeError | int = 3 |
| StatusTypePaperSensor | int = 4 |
```

```
| CashDrawerPin2 | int = 0 |
| CashDrawerPin5 | int = 1 |
| BeepDefaultCount | int = 1 |
| BeepDefaultDuration | int = 100 |
| DoubleByteEncodingGbk | int = 0 |
| DoubleByteEncodingUtf8 | int = 1 |
| DoubleByteEncodingBig5 | int = 3 |
| DoubleByteEncodingShiftJis | int = 4 |
| DoubleByteEncodingEucKr | int = 5 |
| PageDirectionLeftToRight | int = 0 |
| PageDirectionBottomToTop | int = 1 |
| PageDirectionRightToLeft | int = 2 |
| PageDirectionTopToBottom | int = 3 |
| LabelBarcodeTypeCode128 | int = 8 |
| LabelBarcodeHeightDefault | int = 80 |
| LabelBarcodeUnitWidth2 | int = 2 |
| RasterCompressionGsV0 | int = 0 |
| RasterCompressionLineRle | int = 1 |
| RasterCompressionJbig | int = 2 |
| RasterCompressionAuto | int = 3 |
| NvBitmapModeNormal | int = 0 |
```

## Interface Specification

---

## Lifecycle

---

### GetInstance

Get the PrinterSdk singleton. No Android Context required.

#### Syntax

```
static PrinterSdk& GetInstance();
```

```
// Get the PrinterSdk singleton. No Android Context required.
//
// Return value
// Returns `PrinterSdk`.
```

### Release

Release SDK resources.

#### Syntax

```
void Release();
```

```
// Release SDK resources.  
//  
// Return value  
// No return value.
```

## Port / Connection

---

### ConnectNetworkPrinter

Connect a network printer synchronously (TCP RAW, default port 9100).

#### Syntax

```
std::shared_ptr<IPrinterConnection> ConnectNetworkPrinter(const std::string& host, int port
```

```
// Connect a network printer synchronously (TCP RAW, default port 9100).  
//  
// host  
// See method description / usage examples.  
// port  
// See method description / usage examples.  
// Return value  
// Connection handle on success; empty shared_ptr on failure.
```

### ConnectSerialPrinter

Connect a serial (COM) printer.

#### Syntax

```
std::shared_ptr<IPrinterConnection> ConnectSerialPrinter(const SerialPortSettings& settings
```

```
// Connect a serial (COM) printer.  
//  
// settings  
// See method description / usage examples.  
// Return value  
// Connection handle on success; empty shared_ptr on failure.
```

### ConnectSerialPrinter

Connect a serial (COM) printer.

#### Syntax

```
std::shared_ptr<IPrinterConnection> ConnectSerialPrinter(const std::string& portName, int b
```

```
// Connect a serial (COM) printer.  
//  
// portName  
// See method description / usage examples.  
// baudRate  
// See method description / usage examples.  
// Return value  
// Connection handle on success; empty shared_ptr on failure.
```

## ConnectUsbPrinter

Connect a USB printer (USB00x port or SetupAPI device path).

### Syntax

```
std::shared_ptr<IPrinterConnection> ConnectUsbPrinter(const std::string& portNameOrPath);
```

```
// Connect a USB printer (USB00x port or SetupAPI device path).  
//  
// portNameOrPath  
// See method description / usage examples.  
// Return value  
// Connection handle on success; empty shared_ptr on failure.
```

## ConnectUsbPrinter

Connect a USB printer (USB00x port or SetupAPI device path).

### Syntax

```
std::shared_ptr<IPrinterConnection> ConnectUsbPrinter(const UsbPrinterInfo& device);
```

```
// Connect a USB printer (USB00x port or SetupAPI device path).  
//  
// device  
// See method description / usage examples.  
// Return value  
// Connection handle on success; empty shared_ptr on failure.
```

## ConnectParallelPrinter

Connect a parallel (LPT) printer (Windows extra).

## Syntax

```
std::shared_ptr<IPrinterConnection> ConnectParallelPrinter(const std::string& portName);
```

```
// Connect a parallel (LPT) printer (Windows extra).  
//  
// portName  
// See method description / usage examples.  
// Return value  
// Connection handle on success; empty shared_ptr on failure.
```

## ConnectBluetoothPrinter

Connect a Bluetooth SPP printer.

### Syntax

```
std::shared_ptr<IPrinterConnection> ConnectBluetoothPrinter(const std::string& address);
```

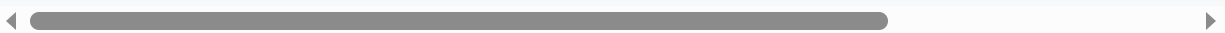
```
// Connect a Bluetooth SPP printer.  
//  
// address  
// See method description / usage examples.  
// Return value  
// Connection handle on success; empty shared_ptr on failure.
```

## ConnectBluetoothPrinter

Connect a Bluetooth SPP printer.

### Syntax

```
std::shared_ptr<IPrinterConnection> ConnectBluetoothPrinter(const BluetoothPrinterInfo& dev:
```



```
// Connect a Bluetooth SPP printer.  
//  
// device  
// See method description / usage examples.  
// pairIfNeeded  
// See method description / usage examples.  
// Return value  
// Connection handle on success; empty shared_ptr on failure.
```

## DisconnectPrinter

Disconnect and release the connection.

## Syntax

```
bool DisconnectPrinter(const std::shared_ptr<IPrinterConnection>& connection);
```

```
// Disconnect and release the connection.  
//  
// connection  
// See method description / usage examples.  
// Return value  
// true if the command was written successfully; otherwise false.
```

## ListSerialPorts

List serial port names (COM1...).

### Syntax

```
static std::vector<std::string> ListSerialPorts();
```

```
// List serial port names (COM1...).  
//  
// Return value  
// Returns `static std::vector<std::string>`.
```

## GetUsbDevices

Enumerate USB printer devices.

### Syntax

```
static std::vector<UsbPrinterInfo> GetUsbDevices();
```

```
// Enumerate USB printer devices.  
//  
// Return value  
// Returns `static std::vector<UsbPrinterInfo>`.
```

## GetParallelDevices

Enumerate parallel ports (LPT1...).

### Syntax

```
static std::vector<std::string> GetParallelDevices();
```

```
// Enumerate parallel ports (LPT1...).\n//\n// Return value\n// Returns `static std::vector<std::string>`.
```

## IsBluetoothAvailable

Check whether a Bluetooth adapter is available.

### Syntax

```
static bool IsBluetoothAvailable();
```

```
// Check whether a Bluetooth adapter is available.\n//\n// Return value\n// Returns `static bool`.
```

## GetBluetoothPairedDevices

List paired Bluetooth devices.

### Syntax

```
static std::vector<BluetoothPrinterInfo> GetBluetoothPairedDevices();
```

```
// List paired Bluetooth devices.\n//\n// Return value\n// Returns `static std::vector<BluetoothPrinterInfo>`.
```

## DiscoverBluetoothDevices

Discover nearby Bluetooth devices (blocking).

### Syntax

```
static std::vector<BluetoothPrinterInfo> DiscoverBluetoothDevices(bool excludePaired = true
```



```
// Discover nearby Bluetooth devices (blocking).\n//\n// excludePaired\n// See method description / usage examples.\n// Return value\n// Returns `static std::vector<BluetoothPrinterInfo>`.
```

## StopBluetoothDiscovery

Stop Bluetooth discovery.

### Syntax

```
static void StopBluetoothDiscovery();
```

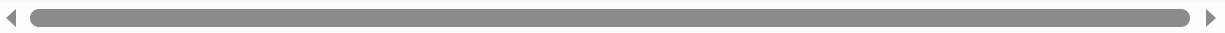
```
// Stop Bluetooth discovery.  
//  
// Return value  
// Returns `static void`.
```

## PairBluetoothDevice

Pair a Bluetooth device.

### Syntax

```
static bool PairBluetoothDevice(const std::string& address, const std::string& pin = "");
```



```
// Pair a Bluetooth device.  
//  
// address  
// See method description / usage examples.  
// pin  
// See method description / usage examples.  
// Return value  
// Returns `static bool`.
```

## Printer Status

---

### QueryRtStatus

Real-time status via DLE EOT.

### Syntax

```
int QueryRtStatus(const std::shared_ptr<IPrinterConnection>& connection, int type);
```

```
// Real-time status via DLE EOT.  
//  
// connection  
// See method description / usage examples.  
// type  
// See method description / usage examples.  
// Return value  
// Returns `int`.
```

## QueryTransmitStatus

Transmit status via GS r.

### Syntax

```
int QueryTransmitStatus(const std::shared_ptr<IPrinterConnection>& connection, int type);
```

```
// Transmit status via GS r.  
//  
// connection  
// See method description / usage examples.  
// type  
// See method description / usage examples.  
// Return value  
// Returns `int`.
```

## GetPrinterStatus

Combined status flags (online/paper/cover).

### Syntax

```
int GetPrinterStatus(const std::shared_ptr<IPrinterConnection>& connection);
```

```
// Combined status flags (online/paper/cover).  
//  
// connection  
// See method description / usage examples.  
// Return value  
// Returns `int`.
```

## IsPrinterOnline

Whether the printer is online.

### Syntax

```
bool IsPrinterOnline(const std::shared_ptr<IPrinterConnection>& connection);
```

```
// Whether the printer is online.  
//  
// connection  
// See method description / usage examples.  
// Return value  
// true if the command was written successfully; otherwise false.
```

## HasPaper

Whether paper is present.

### Syntax

```
bool HasPaper(const std::shared_ptr<IPrinterConnection>& connection);
```

```
// Whether paper is present.  
//  
// connection  
// See method description / usage examples.  
// Return value  
// true if the command was written successfully; otherwise false.
```

## IsCoverOpen

Whether the cover is open.

### Syntax

```
bool IsCoverOpen(const std::shared_ptr<IPrinterConnection>& connection);
```

```
// Whether the cover is open.  
//  
// connection  
// See method description / usage examples.  
// Return value  
// true if the command was written successfully; otherwise false.
```

## POS Printing

---

### InitPrinter

Initialize printer (ESC @).

### Syntax

```
bool InitPrinter(const std::shared_ptr<IPrinterConnection>& connection);
```

```
// Initialize printer (ESC @).  
//  
// connection  
// See method description / usage examples.  
// Return value  
// true if the command was written successfully; otherwise false.
```

## ResetPrinter

Reset printer.

### Syntax

```
bool ResetPrinter(const std::shared_ptr<IPrinterConnection>& connection);
```

```
// Reset printer.  
//  
// connection  
// See method description / usage examples.  
// Return value  
// true if the command was written successfully; otherwise false.
```

## PrintText

Print text (default encoding path).

### Syntax

```
bool PrintText(const std::shared_ptr<IPrinterConnection>& connection, const std::string& text);
```

```
// Print text (default encoding path).  
//  
// connection  
// See method description / usage examples.  
// text  
// See method description / usage examples.  
// Return value  
// true if the command was written successfully; otherwise false.
```

## PrintTextInUtf8

Print text encoded as UTF-8.

### Syntax

```
bool PrintTextInUtf8(const std::shared_ptr<IPrinterConnection>& connection, const std::string& text)
```

```
// Print text encoded as UTF-8.
//
// connection
// See method description / usage examples.
// text
// See method description / usage examples.
// Return value
// true if the command was written successfully; otherwise false.
```

## PrintTextInGbk

Print text encoded as GBK (recommended for Chinese receipts).

### Syntax

```
bool PrintTextInGbk(const std::shared_ptr<IPrinterConnection>& connection, const std::string& text)
```

```
// Print text encoded as GBK (recommended for Chinese receipts).
//
// connection
// See method description / usage examples.
// text
// See method description / usage examples.
// Return value
// true if the command was written successfully; otherwise false.
```

## PrintTextLineLeftRight

Print one line with left/right columns.

### Syntax

```
bool PrintTextLineLeftRight(const std::shared_ptr<IPrinterConnection>& connection, const std::string& text, int left, int right)
```

```

// Print one line with left/right columns.
//
// connection
// See method description / usage examples.
// left
// See method description / usage examples.
// right
// See method description / usage examples.
// areaWidthDots
// See method description / usage examples.
// charWidthDots
// See method description / usage examples.
// Return value
// true if the command was written successfully; otherwise false.

```

## SetAlignment

Set alignment (left/center/right).

### Syntax

```
bool SetAlignment(const std::shared_ptr<IPrinterConnection>& connection, int align);
```

```

// Set alignment (left/center/right).
//
// connection
// See method description / usage examples.
// align
// See method description / usage examples.
// Return value
// true if the command was written successfully; otherwise false.

```

## SetBold

Enable/disable bold.

### Syntax

```
bool SetBold(const std::shared_ptr<IPrinterConnection>& connection, bool bold);
```

```

// Enable/disable bold.
//
// connection
// See method description / usage examples.
// bold
// See method description / usage examples.
// Return value
// true if the command was written successfully; otherwise false.

```

## SetUnderline

Enable/disable underline.

### Syntax

```
bool SetUnderline(const std::shared_ptr<IPrinterConnection>& connection, bool underline);
```

```
// Enable/disable underline.  
//  
// connection  
// See method description / usage examples.  
// underline  
// See method description / usage examples.  
// Return value  
// true if the command was written successfully; otherwise false.
```

## SetReverse

Enable/disable reverse video.

### Syntax

```
bool SetReverse(const std::shared_ptr<IPrinterConnection>& connection, bool reverse);
```

```
// Enable/disable reverse video.  
//  
// connection  
// See method description / usage examples.  
// reverse  
// See method description / usage examples.  
// Return value  
// true if the command was written successfully; otherwise false.
```

## SetTextScale

Set character width/height magnification (0–7).

### Syntax

```
bool SetTextScale(const std::shared_ptr<IPrinterConnection>& connection, int widthScale, int
```

```
// Set character width/height magnification (0-7).  
//  
// connection  
// See method description / usage examples.  
// widthScale  
// See method description / usage examples.  
// heightScale  
// See method description / usage examples.  
// Return value  
// true if the command was written successfully; otherwise false.
```

## SetLineSpacing

Set line spacing in dots.

### Syntax

```
bool SetLineSpacing(const std::shared_ptr<IPrinterConnection>& connection, int spacing);
```

```
// Set line spacing in dots.  
//  
// connection  
// See method description / usage examples.  
// spacing  
// See method description / usage examples.  
// Return value  
// true if the command was written successfully; otherwise false.
```

## SetChineseMode

Enter Chinese character mode.

### Syntax

```
bool SetChineseMode(const std::shared_ptr<IPrinterConnection>& connection);
```

```
// Enter Chinese character mode.  
//  
// connection  
// See method description / usage examples.  
// Return value  
// true if the command was written successfully; otherwise false.
```

## SetSingleByteMode

Enter single-byte mode.

### Syntax

```
bool SetSingleByteMode(const std::shared_ptr<IPrinterConnection>& connection);
```

```
// Enter single-byte mode.  
//  
// connection  
// See method description / usage examples.  
// Return value  
// true if the command was written successfully; otherwise false.
```

## SetDoubleByteEncoding

Set double-byte encoding (GBK/UTF-8/BIG5/ShiftJIS/EUCKR).

### Syntax

```
bool SetDoubleByteEncoding(const std::shared_ptr<IPrinterConnection>& connection, int encoding);
```

```
// Set double-byte encoding (GBK/UTF-8/BIG5/ShiftJIS/EUCKR).  
//  
// connection  
// See method description / usage examples.  
// encoding  
// See method description / usage examples.  
// Return value  
// true if the command was written successfully; otherwise false.
```

## SetChineseDoubleSize

Chinese double-width/double-height.

### Syntax

```
bool SetChineseDoubleSize(const std::shared_ptr<IPrinterConnection>& connection, bool enable);
```

```
// Chinese double-width/double-height.  
//  
// connection  
// See method description / usage examples.  
// enable  
// See method description / usage examples.  
// Return value  
// true if the command was written successfully; otherwise false.
```

## SetPrintAreaWidth

Set print area width in dots.

### Syntax

```
bool SetPrintAreaWidth(const std::shared_ptr<IPrinterConnection>& connection, int width);
```

```
// Set print area width in dots.  
//  
// connection  
// See method description / usage examples.  
// width  
// See method description / usage examples.  
// Return value  
// true if the command was written successfully; otherwise false.
```

## SetPrintAreaLeftMargin

Set left margin in dots.

### Syntax

```
bool SetPrintAreaLeftMargin(const std::shared_ptr<IPrinterConnection>& connection, int leftMargin);
```

```
// Set left margin in dots.  
//  
// connection  
// See method description / usage examples.  
// leftMargin  
// See method description / usage examples.  
// Return value  
// true if the command was written successfully; otherwise false.
```

## SetBarcodeHeight

Set barcode height.

### Syntax

```
bool SetBarcodeHeight(const std::shared_ptr<IPrinterConnection>& connection, int height);
```

```
// Set barcode height.
//
// connection
// See method description / usage examples.
// height
// See method description / usage examples.
// Return value
// true if the command was written successfully; otherwise false.
```

## SetBarcodeWidth

Set barcode module width.

### Syntax

```
bool SetBarcodeWidth(const std::shared_ptr<IPrinterConnection>& connection, int width);
```

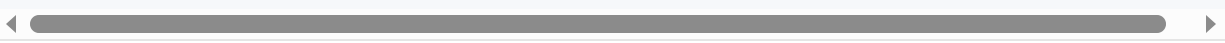
```
// Set barcode module width.
//
// connection
// See method description / usage examples.
// width
// See method description / usage examples.
// Return value
// true if the command was written successfully; otherwise false.
```

## SetBarcodeHriPosition

Set barcode HRI position.

### Syntax

```
bool SetBarcodeHriPosition(const std::shared_ptr<IPrinterConnection>& connection, int pos);
```



```
// Set barcode HRI position.
//
// connection
// See method description / usage examples.
// pos
// See method description / usage examples.
// Return value
// true if the command was written successfully; otherwise false.
```

## FeedLine

Line feed (LF).

## Syntax

```
bool FeedLine(const std::shared_ptr<IPrinterConnection>& connection);
```

```
// Line feed (LF).  
//  
// connection  
// See method description / usage examples.  
// Return value  
// true if the command was written successfully; otherwise false.
```

## FeedPaper

Feed paper by dots (ESC J n). Note: unit is dots, not lines.

### Syntax

```
bool FeedPaper(const std::shared_ptr<IPrinterConnection>& connection, int dots);
```

```
// Feed paper by dots (ESC J n). Note: unit is dots, not lines.  
//  
// connection  
// See method description / usage examples.  
// dots  
// See method description / usage examples.  
// Return value  
// true if the command was written successfully; otherwise false.
```

## PrintAndFeedLines

Print buffer and feed n lines (ESC d n).

### Syntax

```
bool PrintAndFeedLines(const std::shared_ptr<IPrinterConnection>& connection, int lines);
```

```
// Print buffer and feed n lines (ESC d n).  
//  
// connection  
// See method description / usage examples.  
// lines  
// See method description / usage examples.  
// Return value  
// true if the command was written successfully; otherwise false.
```

## CutPaper

Cut paper. CutPartial=ESC m, CutFull=ESC i.

### Syntax

```
bool CutPaper(const std::shared_ptr<IPrinterConnection>& connection, int cutType);
```

```
// Cut paper. CutPartial=ESC m, CutFull=ESC i.  
//  
// connection  
// See method description / usage examples.  
// cutType  
// See method description / usage examples.  
// Return value  
// true if the command was written successfully; otherwise false.
```

## FeedAndCutPaper

Feed past the cutter then partial-cut (recommended receipt finish).

### Syntax

```
bool FeedAndCutPaper(const std::shared_ptr<IPrinterConnection>& connection);
```

```
// Feed past the cutter then partial-cut (recommended receipt finish).  
//  
// connection  
// See method description / usage examples.  
// Return value  
// true if the command was written successfully; otherwise false.
```

## PrintBarcode

Print a 1D barcode.

### Syntax

```
bool PrintBarcode(const std::shared_ptr<IPrinterConnection>& connection, int type, const std::string& data);
```

```

// Print a 1D barcode.
//
// connection
// See method description / usage examples.
// type
// See method description / usage examples.
// data
// See method description / usage examples.
// hriPos
// See method description / usage examples.
// height
// See method description / usage examples.
// width
// See method description / usage examples.
// Return value
// true if the command was written successfully; otherwise false.

```

## PrintQrCode

Print a QR code (simplified).

### Syntax

```
bool PrintQrCode(const std::shared_ptr<IPrinterConnection>& connection, const std::string& c
```

```

// Print a QR code (simplified).
//
// connection
// See method description / usage examples.
// content
// See method description / usage examples.
// size
// See method description / usage examples.
// Return value
// true if the command was written successfully; otherwise false.

```

## PrintQrCodeStandard

Print a QR code (size/align/ECC).

### Syntax

```
bool PrintQrCodeStandard(const std::shared_ptr<IPrinterConnection>& connection, const std::s
```

```

// Print a QR code (size/align/ECC).
//
// connection
// See method description / usage examples.
// content
// See method description / usage examples.
// size
// See method description / usage examples.
// align
// See method description / usage examples.
// eccLevel
// See method description / usage examples.
// Return value
// true if the command was written successfully; otherwise false.

```

## PrintPdf417

Print PDF417 (POS GS ( k).

### Syntax

```
bool PrintPdf417(const std::shared_ptr<IPrinterConnection>& connection, int columns, int rows,
```

```

// Print PDF417 (POS GS ( k).
//
// connection
// See method description / usage examples.
// columns
// See method description / usage examples.
// rows
// See method description / usage examples.
// moduleWidth
// See method description / usage examples.
// rowHeight
// See method description / usage examples.
// errorCorrectionLevel
// See method description / usage examples.
// option
// See method description / usage examples.
// data
// See method description / usage examples.
// Return value
// true if the command was written successfully; otherwise false.

```

## PrintDoubleQrCode

Print two QR codes side by side.

### Syntax

```
bool PrintDoubleQrCode(const std::shared_ptr<IPrinterConnection>& connection, int n, int qr:
```

```
// Print two QR codes side by side.
//
// connection
// See method description / usage examples.
// n
// See method description / usage examples.
// qr1Pos
// See method description / usage examples.
// qr1Data
// See method description / usage examples.
// qr1Ecc
// See method description / usage examples.
// qr1Ver
// See method description / usage examples.
// qr2Pos
// See method description / usage examples.
// qr2Data
// See method description / usage examples.
// qr2Ecc
// See method description / usage examples.
// qr2Ver
// See method description / usage examples.
// Return value
// true if the command was written successfully; otherwise false.
```

## PrintDoubleQrCodeManualExample

Manual double-QR sample command.

### Syntax

```
bool PrintDoubleQrCodeManualExample(const std::shared_ptr<IPrinterConnection>& connection);
```

```
// Manual double-QR sample command.
//
// connection
// See method description / usage examples.
// Return value
// true if the command was written successfully; otherwise false.
```

## EstimateQrSymbolWidthDots

Estimate QR symbol width in dots.

### Syntax

```
int EstimateQrSymbolWidthDots(int version, int moduleSize) const;
```

```
// Estimate QR symbol width in dots.  
//  
// version  
// See method description / usage examples.  
// moduleSize  
// See method description / usage examples.  
// Return value  
// Returns `int`.
```

## OpenCashDrawer

Open cash drawer.

### Syntax

```
bool OpenCashDrawer(const std::shared_ptr<IPrinterConnection>& connection, int drawerIndex)
```



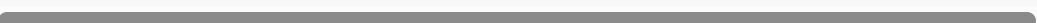
```
// Open cash drawer.  
//  
// connection  
// See method description / usage examples.  
// drawerIndex  
// See method description / usage examples.  
// Return value  
// true if the command was written successfully; otherwise false.
```

## Beep

Beep.

### Syntax

```
bool Beep(const std::shared_ptr<IPrinterConnection>& connection, int count = 1, int duration)
```



```

// Beep.
//
// connection
// See method description / usage examples.
// count
// See method description / usage examples.
// duration
// See method description / usage examples.
// Return value
// true if the command was written successfully; otherwise false.

```

## SelfTest

Print self-test page.

### Syntax

```
bool SelfTest(const std::shared_ptr<IPrinterConnection>& connection);
```

```

// Print self-test page.
//
// connection
// See method description / usage examples.
// Return value
// true if the command was written successfully; otherwise false.

```

## SendCommand

Send raw ESC/POS or JPL bytes.

### Syntax

```
bool SendCommand(const std::shared_ptr<IPrinterConnection>& connection, const ByteBuffer& command);
```

```

// Send raw ESC/POS or JPL bytes.
//
// connection
// See method description / usage examples.
// command
// See method description / usage examples.
// Return value
// true if the command was written successfully; otherwise false.

```

## PrintRasterImageFromPixels

Print raster image from ARGB pixels (GS v 0).

## Syntax

```
bool PrintRasterImageFromPixels(const std::shared_ptr<IPrinterConnection>& connection, const
```

```
// Print raster image from ARGB pixels (GS v 0).  
//  
// connection  
// See method description / usage examples.  
// pixels  
// See method description / usage examples.  
// width  
// See method description / usage examples.  
// height  
// See method description / usage examples.  
// targetWidth  
// See method description / usage examples.  
// targetHeight  
// See method description / usage examples.  
// compressionMode  
// See method description / usage examples.  
// Return value  
// true if the command was written successfully; otherwise false.
```

## PrintRasterImageFromFile

Print raster image from file (WIC decode).

### Syntax

```
bool PrintRasterImageFromFile(const std::shared_ptr<IPrinterConnection>& connection, const
```

```
// Print raster image from file (WIC decode).  
//  
// connection  
// See method description / usage examples.  
// path  
// See method description / usage examples.  
// Return value  
// true if the command was written successfully; otherwise false.
```

## PrintRasterImageFromFile

Print raster image from file (WIC decode).

### Syntax

```
bool PrintRasterImageFromFile(const std::shared_ptr<IPrinterConnection>& connection, const std::string& path, const std::string& fileName, const std::string& targetWidth, const std::string& targetHeight, const std::string& compressionMode, const std::string& return value, const std::string& true if the command was written successfully; otherwise false.
```

```
// Print raster image from file (WIC decode).  
//  
// connection  
// See method description / usage examples.  
// path  
// See method description / usage examples.  
// targetWidth  
// See method description / usage examples.  
// targetHeight  
// See method description / usage examples.  
// compressionMode  
// See method description / usage examples.  
// Return value  
// true if the command was written successfully; otherwise false.
```

## PrintRasterImageFromData

Print raster image from memory bytes.

### Syntax

```
bool PrintRasterImageFromData(const std::shared_ptr<IPrinterConnection>& connection, const std::string& data, const std::string& size, const std::string& return value, const std::string& true if the command was written successfully; otherwise false.
```

```
// Print raster image from memory bytes.  
//  
// connection  
// See method description / usage examples.  
// data  
// See method description / usage examples.  
// size  
// See method description / usage examples.  
// Return value  
// true if the command was written successfully; otherwise false.
```

## PrintRasterImageFromData

Print raster image from memory bytes.

### Syntax

```
bool PrintRasterImageFromData(const std::shared_ptr<IPrinterConnection>& connection, const std::string& data, const std::string& size, const std::string& return value, const std::string& true if the command was written successfully; otherwise false.
```

```

// Print raster image from memory bytes.
//
// connection
// See method description / usage examples.
// data
// See method description / usage examples.
// size
// See method description / usage examples.
// targetWidth
// See method description / usage examples.
// targetHeight
// See method description / usage examples.
// compressionMode
// See method description / usage examples.
// Return value
// true if the command was written successfully; otherwise false.

```

## PrintRasterImageFromData

Print raster image from memory bytes.

### Syntax

```
bool PrintRasterImageFromData(const std::shared_ptr<IPrinterConnection>& connection, const :
```

```

// Print raster image from memory bytes.
//
// connection
// See method description / usage examples.
// data
// See method description / usage examples.
// Return value
// true if the command was written successfully; otherwise false.

```

## PrintRasterImageFromData

Print raster image from memory bytes.

### Syntax

```
bool PrintRasterImageFromData(const std::shared_ptr<IPrinterConnection>& connection, const :
```

```

    // Print raster image from memory bytes.
    //
    // connection
    // See method description / usage examples.
    // data
    // See method description / usage examples.
    // targetWidth
    // See method description / usage examples.
    // targetHeight
    // See method description / usage examples.
    // compressionMode
    // See method description / usage examples.
    // Return value
    // true if the command was written successfully; otherwise false.

```

## Page Mode

---

### SelectPageMode

Enter page mode.

#### Syntax

```
bool SelectPageMode(const std::shared_ptr<IPrinterConnection>& connection);
```

```

    // Enter page mode.
    //
    // connection
    // See method description / usage examples.
    // Return value
    // true if the command was written successfully; otherwise false.

```

### SelectPageModeEx

Enter page mode with area/units.

#### Syntax

```
bool SelectPageModeEx(const std::shared_ptr<IPrinterConnection>& connection);
```

```

    // Enter page mode with area/units.
    //
    // connection
    // See method description / usage examples.
    // Return value
    // true if the command was written successfully; otherwise false.

```

## SelectPageModeEx

Enter page mode with area/units.

### Syntax

```
bool SelectPageModeEx(const std::shared_ptr<IPrinterConnection>& connection, int x, int y, :
```

```
// Enter page mode with area/units.  
//  
// connection  
// See method description / usage examples.  
// x  
// See method description / usage examples.  
// y  
// See method description / usage examples.  
// width  
// See method description / usage examples.  
// height  
// See method description / usage examples.  
// Return value  
// true if the command was written successfully; otherwise false.
```

## SelectPageModeEx

Enter page mode with area/units.

### Syntax

```
bool SelectPageModeEx(const std::shared_ptr<IPrinterConnection>& connection, int hu, int vu
```

```

// Enter page mode with area/units.
//
// connection
// See method description / usage examples.
// hu
// See method description / usage examples.
// vu
// See method description / usage examples.
// x
// See method description / usage examples.
// y
// See method description / usage examples.
// w
// See method description / usage examples.
// h
// See method description / usage examples.
// Return value
// true if the command was written successfully; otherwise false.

```

## ExitPageMode

Exit page mode.

### Syntax

```
bool ExitPageMode(const std::shared_ptr<IPrinterConnection>& connection);
```

```

// Exit page mode.
//
// connection
// See method description / usage examples.
// Return value
// true if the command was written successfully; otherwise false.

```

## PagePrintPage

Print page-mode buffer.

### Syntax

```
bool PagePrintPage(const std::shared_ptr<IPrinterConnection>& connection);
```

```

// Print page-mode buffer.
//
// connection
// See method description / usage examples.
// Return value
// true if the command was written successfully; otherwise false.

```

## CancelPrintDataInPageMode

Cancel page-mode data.

### Syntax

```
bool CancelPrintDataInPageMode(const std::shared_ptr<IPrinterConnection>& connection);
```

```
// Cancel page-mode data.  
//  
// connection  
// See method description / usage examples.  
// Return value  
// true if the command was written successfully; otherwise false.
```

## PageClearPage

Clear page-mode buffer.

### Syntax

```
bool PageClearPage(const std::shared_ptr<IPrinterConnection>& connection);
```

```
// Clear page-mode buffer.  
//  
// connection  
// See method description / usage examples.  
// Return value  
// true if the command was written successfully; otherwise false.
```

## SetMovementUnits

Set movement units.

### Syntax

```
bool SetMovementUnits(const std::shared_ptr<IPrinterConnection>& connection, int x, int y);
```



```

    // Set movement units.
    //
    // connection
    // See method description / usage examples.
    // x
    // See method description / usage examples.
    // y
    // See method description / usage examples.
    // Return value
    // true if the command was written successfully; otherwise false.

```

## SetPrintAreaInPageMode

Set page-mode print area.

### Syntax

```
bool SetPrintAreaInPageMode(const std::shared_ptr<IPrinterConnection>& connection, int x, int y, int width, int height)
```

```

    // Set page-mode print area.
    //
    // connection
    // See method description / usage examples.
    // x
    // See method description / usage examples.
    // y
    // See method description / usage examples.
    // width
    // See method description / usage examples.
    // height
    // See method description / usage examples.
    // Return value
    // true if the command was written successfully; otherwise false.

```

## SetPageModePrintDirection

Set page-mode print direction.

### Syntax

```
bool SetPageModePrintDirection(const std::shared_ptr<IPrinterConnection>& connection, int direction)
```

```
// Set page-mode print direction.
//
// connection
// See method description / usage examples.
// direction
// See method description / usage examples.
// Return value
// true if the command was written successfully; otherwise false.
```

## DrawTextInPageMode

Draw text in page mode.

### Syntax

```
bool DrawTextInPageMode(const std::shared_ptr<IPrinterConnection>& connection, int x, int y
```

```
// Draw text in page mode.
//
// connection
// See method description / usage examples.
// x
// See method description / usage examples.
// y
// See method description / usage examples.
// text
// See method description / usage examples.
// Return value
// true if the command was written successfully; otherwise false.
```

## DrawTextUtf8InPageMode

Draw UTF-8 text in page mode.

### Syntax

```
bool DrawTextUtf8InPageMode(const std::shared_ptr<IPrinterConnection>& connection, int x, in
```

```

// Draw UTF-8 text in page mode.
//
// connection
// See method description / usage examples.
// x
// See method description / usage examples.
// y
// See method description / usage examples.
// text
// See method description / usage examples.
// Return value
// true if the command was written successfully; otherwise false.

```

## DrawTextGbkInPageMode

Draw GBK text in page mode.

### Syntax

```
bool DrawTextGbkInPageMode(const std::shared_ptr<IPrinterConnection>& connection, int x, int y, const std::string& text)
```

```

// Draw GBK text in page mode.
//
// connection
// See method description / usage examples.
// x
// See method description / usage examples.
// y
// See method description / usage examples.
// text
// See method description / usage examples.
// Return value
// true if the command was written successfully; otherwise false.

```

## DrawTextCenteredInPageMode

Draw centered text in page mode.

### Syntax

```
bool DrawTextCenteredInPageMode(const std::shared_ptr<IPrinterConnection>& connection, int x, int y, const std::string& text)
```

```

// Draw centered text in page mode.
//
// connection
// See method description / usage examples.
// pageWidth
// See method description / usage examples.
// y
// See method description / usage examples.
// text
// See method description / usage examples.
// charWidthDots
// See method description / usage examples.
// Return value
// true if the command was written successfully; otherwise false.

```

## DrawTextRightInPageMode

Draw right-aligned text in page mode.

### Syntax

```
bool DrawTextRightInPageMode(const std::shared_ptr<IPrinterConnection>& connection, int page,
```

```

// Draw right-aligned text in page mode.
//
// connection
// See method description / usage examples.
// pageWidth
// See method description / usage examples.
// y
// See method description / usage examples.
// text
// See method description / usage examples.
// charWidthDots
// See method description / usage examples.
// Return value
// true if the command was written successfully; otherwise false.

```

## DrawTextLineLeftRightInPageMode

Draw left/right columns in page mode.

### Syntax

```
bool DrawTextLineLeftRightInPageMode(const std::shared_ptr<IPrinterConnection>& connection,
```

```

// Draw left/right columns in page mode.
//
// connection
// See method description / usage examples.
// pageWidth
// See method description / usage examples.
// y
// See method description / usage examples.
// left
// See method description / usage examples.
// right
// See method description / usage examples.
// charWidthDots
// See method description / usage examples.
// Return value
// true if the command was written successfully; otherwise false.

```

## DrawBarcodeInPageMode

Draw barcode in page mode.

### Syntax

```
bool DrawBarcodeInPageMode(const std::shared_ptr<IPrinterConnection>& connection, int x, int y, int type, int data, int hriPos, int height, int width)
```

```

// Draw barcode in page mode.
//
// connection
// See method description / usage examples.
// x
// See method description / usage examples.
// y
// See method description / usage examples.
// type
// See method description / usage examples.
// data
// See method description / usage examples.
// hriPos
// See method description / usage examples.
// height
// See method description / usage examples.
// width
// See method description / usage examples.
// Return value
// true if the command was written successfully; otherwise false.

```

## DrawBarcodeInPageModeWithCenteredHri

Draw barcode with centered HRI.

### Syntax

```
bool DrawBarcodeInPageModeWithCenteredHri(const std::shared_ptr<IPrinterConnection>& connect
```

```
// Draw barcode with centered HRI.
//
// connection
// See method description / usage examples.
// x
// See method description / usage examples.
// y
// See method description / usage examples.
// type
// See method description / usage examples.
// data
// See method description / usage examples.
// height
// See method description / usage examples.
// width
// See method description / usage examples.
// hriGapDots
// See method description / usage examples.
// Return value
// true if the command was written successfully; otherwise false.
```

## DrawQrCodeInPageMode

Draw QR in page mode.

### Syntax

```
bool DrawQrCodeInPageMode(const std::shared_ptr<IPrinterConnection>& connection, int x, int
```

```

// Draw QR in page mode.
//
// connection
// See method description / usage examples.
// x
// See method description / usage examples.
// y
// See method description / usage examples.
// version
// See method description / usage examples.
// eccLevel
// See method description / usage examples.
// data
// See method description / usage examples.
// Return value
// true if the command was written successfully; otherwise false.

```

## DrawLineInPageMode

Draw line in page mode.

### Syntax

```
bool DrawLineInPageMode(const std::shared_ptr<IPrinterConnection>& connection, int x1, int y1, int x2, int y2, int lineType)
```

```

// Draw line in page mode.
//
// connection
// See method description / usage examples.
// x1
// See method description / usage examples.
// y1
// See method description / usage examples.
// x2
// See method description / usage examples.
// y2
// See method description / usage examples.
// lineType
// See method description / usage examples.
// Return value
// true if the command was written successfully; otherwise false.

```

## DrawRectangleInPageMode

Draw rectangle in page mode.

### Syntax

```
bool DrawRectangleInPageMode(const std::shared_ptr<IPrinterConnection>& connection, int mode
```

```
// Draw rectangle in page mode.
//
// connection
// See method description / usage examples.
// mode
// See method description / usage examples.
// x
// See method description / usage examples.
// y
// See method description / usage examples.
// width
// See method description / usage examples.
// height
// See method description / usage examples.
// Return value
// true if the command was written successfully; otherwise false.
```

## EstimateCode128WidthDots

Estimate Code128 width.

### Syntax

```
int EstimateCode128WidthDots(const std::string& data, int moduleWidth) const;
```

```
// Estimate Code128 width.
//
// data
// See method description / usage examples.
// moduleWidth
// See method description / usage examples.
// Return value
// Returns `int`.
```

## EstimateTextWidthDots

Estimate text width.

### Syntax

```
int EstimateTextWidthDots(const std::string& text, int charWidthDots = 12) const;
```

```
// Estimate text width.
//
// text
// See method description / usage examples.
// charWidthDots
// See method description / usage examples.
// Return value
// Returns `int`.
```

## Label Printing

---

### EnableLabelMode

Enter label (JPL) mode.

#### Syntax

```
bool EnableLabelMode(const std::shared_ptr<IPrinterConnection>& connection);
```

```
// Enter label (JPL) mode.
//
// connection
// See method description / usage examples.
// Return value
// true if the command was written successfully; otherwise false.
```

### DisableLabelMode

Exit label mode.

#### Syntax

```
bool DisableLabelMode(const std::shared_ptr<IPrinterConnection>& connection);
```

```
// Exit label mode.
//
// connection
// See method description / usage examples.
// Return value
// true if the command was written successfully; otherwise false.
```

### LabelBackPaper

Back-feed label to print position.

#### Syntax

```
bool LabelBackPaper(const std::shared_ptr<IPrinterConnection>& connection);
```

```
// Back-feed label to print position.  
//  
// connection  
// See method description / usage examples.  
// Return value  
// true if the command was written successfully; otherwise false.
```

## LabelPageBegin

Begin a label page layout.

### Syntax

```
bool LabelPageBegin(const std::shared_ptr<IPrinterConnection>& connection, int x, int y, int
```

```
// Begin a label page layout.  
//  
// connection  
// See method description / usage examples.  
// x  
// See method description / usage examples.  
// y  
// See method description / usage examples.  
// width  
// See method description / usage examples.  
// height  
// See method description / usage examples.  
// rotation  
// See method description / usage examples.  
// Return value  
// true if the command was written successfully; otherwise false.
```

## LabelPagePrint

Print the label page (optional copies).

### Syntax

```
bool LabelPagePrint(const std::shared_ptr<IPrinterConnection>& connection, int printNum = 1
```

```
// Print the label page (optional copies).  
//  
// connection  
// See method description / usage examples.  
// printNum  
// See method description / usage examples.  
// Return value  
// true if the command was written successfully; otherwise false.
```

## PrintLabelText

Label text.

### Syntax

```
bool PrintLabelText(const std::shared_ptr<IPrinterConnection>& connection, const std::string
```

```
// Label text.  
//  
// connection  
// See method description / usage examples.  
// text  
// See method description / usage examples.  
// x  
// See method description / usage examples.  
// y  
// See method description / usage examples.  
// Return value  
// true if the command was written successfully; otherwise false.
```

## PrintLabelText

Label text.

### Syntax

```
bool PrintLabelText(const std::shared_ptr<IPrinterConnection>& connection, const std::string
```

```

// Label text.
//
// connection
// See method description / usage examples.
// text
// See method description / usage examples.
// x
// See method description / usage examples.
// y
// See method description / usage examples.
// fontHeight
// See method description / usage examples.
// Return value
// true if the command was written successfully; otherwise false.

```

## PrintLabelText

Label text.

### Syntax

```
bool PrintLabelText(const std::shared_ptr<IPrinterConnection>& connection, const std::string
```

```

// Label text.
//
// connection
// See method description / usage examples.
// text
// See method description / usage examples.
// x
// See method description / usage examples.
// y
// See method description / usage examples.
// fontHeight
// See method description / usage examples.
// fontType
// See method description / usage examples.
// Return value
// true if the command was written successfully; otherwise false.

```

## PrintLabelTextUtf8

Label UTF-8 text.

### Syntax

```
bool PrintLabelTextUtf8(const std::shared_ptr<IPrinterConnection>& connection, const std::s
```

```

// Label UTF-8 text.
//
// connection
// See method description / usage examples.
// text
// See method description / usage examples.
// x
// See method description / usage examples.
// y
// See method description / usage examples.
// Return value
// true if the command was written successfully; otherwise false.

```

## PrintLabelTextGbk

Label GBK text.

### Syntax

```
bool PrintLabelTextGbk(const std::shared_ptr<IPrinterConnection>& connection, const std::str
```

```

// Label GBK text.
//
// connection
// See method description / usage examples.
// text
// See method description / usage examples.
// x
// See method description / usage examples.
// y
// See method description / usage examples.
// Return value
// true if the command was written successfully; otherwise false.

```

## PrintLabelBarcode

Label 1D barcode.

### Syntax

```
bool PrintLabelBarcode(const std::shared_ptr<IPrinterConnection>& connection, int type, con
```

```

// Label 1D barcode.
//
// connection
// See method description / usage examples.
// type
// See method description / usage examples.
// data
// See method description / usage examples.
// x
// See method description / usage examples.
// y
// See method description / usage examples.
// height
// See method description / usage examples.
// unitWidth
// See method description / usage examples.
// Return value
// true if the command was written successfully; otherwise false.

```

## PrintLabelBarcode

Label 1D barcode.

### Syntax

```
bool PrintLabelBarcode(const std::shared_ptr<IPrinterConnection>& connection, int x, int y,
```

```

// Label 1D barcode.
//
// connection
// See method description / usage examples.
// x
// See method description / usage examples.
// y
// See method description / usage examples.
// type
// See method description / usage examples.
// height
// See method description / usage examples.
// unitWidth
// See method description / usage examples.
// rotate
// See method description / usage examples.
// data
// See method description / usage examples.
// Return value
// true if the command was written successfully; otherwise false.

```

## PrintLabelQrCode

Label QR code.

### Syntax

```
bool PrintLabelQrCode(const std::shared_ptr<IPrinterConnection>& connection, const std::str:
```

```
// Label QR code.  
//  
// connection  
// See method description / usage examples.  
// data  
// See method description / usage examples.  
// x  
// See method description / usage examples.  
// y  
// See method description / usage examples.  
// Return value  
// true if the command was written successfully; otherwise false.
```

## PrintLabelQrCode

Label QR code.

### Syntax

```
bool PrintLabelQrCode(const std::shared_ptr<IPrinterConnection>& connection, int x, int y, :
```

```

// Label QR code.
//
// connection
// See method description / usage examples.
// x
// See method description / usage examples.
// y
// See method description / usage examples.
// version
// See method description / usage examples.
// eccLevel
// See method description / usage examples.
// unitWidth
// See method description / usage examples.
// rotate
// See method description / usage examples.
// data
// See method description / usage examples.
// Return value
// true if the command was written successfully; otherwise false.

```

## PrintLabelPdf417Code

Label PDF417.

### Syntax

```
bool PrintLabelPdf417Code(const std::shared_ptr<IPrinterConnection>& connection, const std:
```

```

// Label PDF417.
//
// connection
// See method description / usage examples.
// data
// See method description / usage examples.
// x
// See method description / usage examples.
// y
// See method description / usage examples.
// Return value
// true if the command was written successfully; otherwise false.

```

## PrintLabelPdf417Code

Label PDF417.

### Syntax

```
bool PrintLabelPdf417Code(const std::shared_ptr<IPrinterConnection>& connection, int x, int
```

```
// Label PDF417.  
//  
// connection  
// See method description / usage examples.  
// x  
// See method description / usage examples.  
// y  
// See method description / usage examples.  
// colNum  
// See method description / usage examples.  
// eccLevel  
// See method description / usage examples.  
// lwRatio  
// See method description / usage examples.  
// unitWidth  
// See method description / usage examples.  
// rotate  
// See method description / usage examples.  
// data  
// See method description / usage examples.  
// Return value  
// true if the command was written successfully; otherwise false.
```

## PrintLabelLine

Label line.

### Syntax

```
bool PrintLabelLine(const std::shared_ptr<IPrinterConnection>& connection, int startX, int :
```

```
// Label line.  
//  
// connection  
// See method description / usage examples.  
// startX  
// See method description / usage examples.  
// startY  
// See method description / usage examples.  
// endX  
// See method description / usage examples.  
// endY  
// See method description / usage examples.  
// Return value  
// true if the command was written successfully; otherwise false.
```

## PrintLabelRectangle

Label rectangle.

### Syntax

```
bool PrintLabelRectangle(const std::shared_ptr<IPrinterConnection>& connection, int x, int y,
```

```
// Label rectangle.  
//  
// connection  
// See method description / usage examples.  
// x  
// See method description / usage examples.  
// y  
// See method description / usage examples.  
// width  
// See method description / usage examples.  
// height  
// See method description / usage examples.  
// Return value  
// true if the command was written successfully; otherwise false.
```

## SetLabelBarcodeTextPosition

Label barcode readable text position.

### Syntax

```
bool SetLabelBarcodeTextPosition(const std::shared_ptr<IPrinterConnection>& connection, int
```

```
// Label barcode readable text position.  
//  
// connection  
// See method description / usage examples.  
// position  
// See method description / usage examples.  
// Return value  
// true if the command was written successfully; otherwise false.
```

## DrawLabelImageFromFile

Label image from file.

### Syntax

```
bool DrawLabelImageFromFile(const std::shared_ptr<IPrinterConnection>& connection, int x, int y, const std::string& imagePath)
```

```
// Label image from file.
//
// connection
// See method description / usage examples.
// x
// See method description / usage examples.
// y
// See method description / usage examples.
// imagePath
// See method description / usage examples.
// Return value
// true if the command was written successfully; otherwise false.
```

## DrawLabelImageFromData

Label image from memory.

### Syntax

```
bool DrawLabelImageFromData(const std::shared_ptr<IPrinterConnection>& connection, int x, int y, const std::vector<uint8_t>& imageData)
```

```
// Label image from memory.
//
// connection
// See method description / usage examples.
// x
// See method description / usage examples.
// y
// See method description / usage examples.
// imageData
// See method description / usage examples.
// imageSize
// See method description / usage examples.
// Return value
// true if the command was written successfully; otherwise false.
```

## DrawLabelImageFromData

Label image from memory.

### Syntax

```
bool DrawLabelImageFromData(const std::shared_ptr<IPrinterConnection>& connection, int x, int y, const std::vector<uint8_t>& imageData)
```

```

// Label image from memory.
//
// connection
// See method description / usage examples.
// x
// See method description / usage examples.
// y
// See method description / usage examples.
// imageData
// See method description / usage examples.
// Return value
// true if the command was written successfully; otherwise false.

```

## DrawLabelImageFromPixels

Label image from ARGB pixels.

### Syntax

```
bool DrawLabelImageFromPixels(const std::shared_ptr<IPrinterConnection>& connection, int x,
```

```

// Label image from ARGB pixels.
//
// connection
// See method description / usage examples.
// x
// See method description / usage examples.
// y
// See method description / usage examples.
// pixels
// See method description / usage examples.
// width
// See method description / usage examples.
// height
// See method description / usage examples.
// Return value
// true if the command was written successfully; otherwise false.

```

## DrawLabelImageAdvanced

Advanced label image draw.

### Syntax

```
bool DrawLabelImageAdvanced(const std::shared_ptr<IPrinterConnection>& connection, int x, int y,
```

```

// Advanced label image draw.
//
// connection
// See method description / usage examples.
// x
// See method description / usage examples.
// y
// See method description / usage examples.
// pixels
// See method description / usage examples.
// width
// See method description / usage examples.
// height
// See method description / usage examples.
// showType
// See method description / usage examples.
// Return value
// true if the command was written successfully; otherwise false.

```

## DrawLabelBox

Label filled box.

### Syntax

```
bool DrawLabelBox(const std::shared_ptr<IPrinterConnection>& connection, int left, int top,
```

```

// Label filled box.
//
// connection
// See method description / usage examples.
// left
// See method description / usage examples.
// top
// See method description / usage examples.
// right
// See method description / usage examples.
// bottom
// See method description / usage examples.
// color
// See method description / usage examples.
// Return value
// true if the command was written successfully; otherwise false.

```

## Notes

1. Do not mix standard / page / label modes casually.

2. Paper widths: 58mm  $\approx$  384 dots, 80mm  $\approx$  576 dots.
3. `FeedPaper` is dots in C++; C# overload may mean lines — be careful when porting.
4. Android-only permission/manager APIs are not provided.
5. Full placeholders for pending APIs are listed below; see also `Android_API覆盖对照.md`.

## Windows / C++ extras

---

- Parallel port APIs
- `ListSerialPorts`
- `IsBluetoothAvailable` / `PairBluetoothDevice`
- `PrintAndFeedLines` (explicit line feed)

## Not Yet Implemented APIs (Android 1.6.3 placeholders)

---

These APIs exist on Android / Windows C# but are **not wrapped in the current C++ SDK**. Sections are kept for 1:1 alignment with the Android PDF handbooks. Signatures are planned C++ forms.

Status:  Not implemented

### Index (pending)

- `ConnectNetworkPrinterAsync` — Port / Connection
- `ClearPrinterBuffer` — Printer Status
- `CancelPrint` — Printer Status
- `ClearBuffer` — POS Printing
- `CancelPrintCmd` — POS Printing
- `PrintTextInBIG5` — POS Printing
- `PrintTextInShiftJis` — POS Printing
- `PrintTextInEucKr` — POS Printing
- `SetCharSpacing` — POS Printing
- `ReverseFeed` — POS Printing
- `HorizontalTab` — POS Printing
- `SetTabPositions` — POS Printing
- `SetPrintMode` — POS Printing
- `SetCharacterSet` — POS Printing
- `SetCharacterCodepage` — POS Printing
- `SetAsciiTextFontType` — POS Printing
- `SetTextUpsideDown` — POS Printing
- `SetTextRotate` — POS Printing
- `SetTextLineHeight` — POS Printing
- `SetAsciiTextCharRightSpacing` — POS Printing
- `SetKanjiTextCharSpacing` — POS Printing
- `SetPrintPosition` — POS Printing
- `SetHorizontalAbsolutePrintPosition` — POS Printing
- `SetHorizontalRelativePrintPosition` — POS Printing
- `SetVerticalAbsolutePrintPosition` — POS Printing
- `SetVerticalRelativePrintPosition` — POS Printing
- `SetBarcodeReadableTextFontType` — POS Printing
- `SetBarcodeReadableTextPosition` — POS Printing
- `PrintHorizontalLine` — POS Printing
- `FixedHorizontalLine` — POS Printing
- `PrintAndFeedDots` — POS Printing

- `DrawRectangleBorderInPageMode` — Page Mode
- `DrawFilledRectangleInPageMode` — Page Mode
- `PrintNvBitmap` — Page Mode
- `DrawNvBitmapInPageMode` — Page Mode
- `DrawNvBitmapCenteredInPageMode` — Page Mode
- `DrawTextBIG5InPageMode` — Page Mode
- `DrawTextShiftJisInPageMode` — Page Mode
- `DrawTextEucKrInPageMode` — Page Mode
- `DrawImageFromFileInPageMode` — Page Mode
- `DrawImageFromDataInPageMode` — Page Mode
- `DrawImageFromPixelsInPageMode` — Page Mode

---

## Port / Connection (pending)

### ConnectNetworkPrinterAsync

**Status:** ❌ **Not implemented** (Android: `connectNetworkPrinterAsync` )

Connect a network printer via TCP asynchronously.

**Planned syntax (not available yet)**

```
void ConnectNetworkPrinterAsync(const std::string& ip, int port, /* callback */);
```

**Workaround:** Use sync `ConnectNetworkPrinter` on a worker thread for UI apps.

## Printer Status (pending)

### ClearPrinterBuffer

**Status:** ❌ **Not implemented** (Android: `clearPrinterBuffer` )

Clear Printer Buffer

**Planned syntax (not available yet)**

```
bool ClearPrinterBuffer(const std::shared_ptr<IPrinterConnection>& connection);
```

**Workaround:** Temporarily use `SendCommand` with CAN (0x18).

### CancelPrint

**Status:** ❌ **Not implemented** (Android: `cancelPrint` )

Cancel print

**Planned syntax (not available yet)**

```
bool CancelPrint(const std::shared_ptr<IPrinterConnection>& connection);
```

**Workaround:** Temporarily use `SendCommand` with cancel-related bytes.

## POS Printing (pending)

### ClearBuffer

**Status:** ❌ **Not implemented** (Android: `clearBuffer` )

Clear printer buffer

**Planned syntax (not available yet)**

```
bool ClearBuffer(const std::shared_ptr<IPrinterConnection>& connection);
```

**Workaround:** Same as `ClearPrinterBuffer`; use `SendCommand(0x18)` .

### CancelPrintCmd

**Status:** ❌ **Not implemented** (Android: `cancelPrintCmd` )

Cancel print

**Planned syntax (not available yet)**

```
bool CancelPrintCmd(const std::shared_ptr<IPrinterConnection>& connection);
```

**Workaround:** Same as `CancelPrint`.

### PrintTextInBIG5

**Status:** ❌ **Not implemented** (Android: `printTextInBIG5` )

Print text

**Planned syntax (not available yet)**

```
bool PrintTextInBIG5(const std::shared_ptr<IPrinterConnection>& connection, const std::string& text);
```

**Workaround:** Set `DoubleByteEncodingBig5` , encode bytes yourself, then `SendCommand` .

### PrintTextInShiftJis

**Status:** ❌ **Not implemented** (Android: `printTextInShiftJIS` )

Print text

**Planned syntax (not available yet)**

```
bool PrintTextInShiftJis(const std::shared_ptr<IPrinterConnection>& connection, const std::string& text);
```

**Workaround:** Set `DoubleByteEncodingShiftJis` , then send encoded bytes.

## PrintTextInEucKr

**Status:** ❌ Not implemented (Android: `printTextInEUCKR` )

Print text

**Planned syntax (not available yet)**

```
bool PrintTextInEucKr(const std::shared_ptr<IPrinterConnection>& connection, const std::str:
```

**Workaround:** Set `DoubleByteEncodingEucKr` , then send encoded bytes.

## SetCharSpacing

**Status:** ❌ Not implemented (Android: `setCharSpacing` )

Set Char Spacing

**Planned syntax (not available yet)**

```
bool SetCharSpacing(const std::shared_ptr<IPrinterConnection>& connection, int spacing);
```

**Workaround:** Use `SendCommand` for ESC/POS character spacing.

## ReverseFeed

**Status:** ❌ Not implemented (Android: `reverseFeed` )

Feed paper

**Planned syntax (not available yet)**

```
bool ReverseFeed(const std::shared_ptr<IPrinterConnection>& connection, int lines);
```

**Workaround:** Use `SendCommand` with ESC e n.

## HorizontalTab

**Status:** ❌ Not implemented (Android: `horizontalTab` )

Horizontal tab

**Planned syntax (not available yet)**

```
bool HorizontalTab(const std::shared_ptr<IPrinterConnection>& connection);
```

**Workaround:** Use `SendCommand` with HT (0x09).

## SetTabPositions

**Status:** ❌ Not implemented (Android: `setTabPositions` )

Horizontal tab

**Planned syntax (not available yet)**

```
bool SetTabPositions(const std::shared_ptr<IPrinterConnection>& connection, const std::vector<int>& positions);
```

**Workaround:** Use `SendCommand` with ESC D.

## SetPrintMode

**Status:** ❌ **Not implemented** (Android: `setPrintMode` )

Set print mode

**Planned syntax (not available yet)**

```
bool SetPrintMode(const std::shared_ptr<IPrinterConnection>& connection, int mode);
```

**Workaround:** Use `SendCommand` ESC ! n, or combine `SetBold`/`SetTextScale`.

## SetCharacterSet

**Status:** ❌ **Not implemented** (Android: `setCharacterSet` )

Set character set

**Planned syntax (not available yet)**

```
bool SetCharacterSet(const std::shared_ptr<IPrinterConnection>& connection, int n);
```

**Workaround:** Use `SendCommand` for character set select.

## SetCharacterCodepage

**Status:** ❌ **Not implemented** (Android: `setCharacterCodepage` )

Set character code page

**Planned syntax (not available yet)**

```
bool SetCharacterCodepage(const std::shared_ptr<IPrinterConnection>& connection, int n);
```

**Workaround:** Use `SendCommand` for code page select.

## SetAsciiTextFontType

**Status:** ❌ **Not implemented** (Android: `setAsciiTextFontType` )

Set Ascii Text Font Type

**Planned syntax (not available yet)**

```
bool SetAsciiTextFontType(const std::shared_ptr<IPrinterConnection>& connection, int fontType);
```

**Workaround:** Use `SendCommand` for ASCII font select.

## SetTextUpsideDown

**Status:** ❌ **Not implemented** (Android: `setTextUpsideDown` )

Set upside-down text

**Planned syntax (not available yet)**

```
bool SetTextUpsideDown(const std::shared_ptr<IPrinterConnection>& connection, bool enable);
```

**Workaround:** Use `SendCommand` for upside-down print.

## SetTextRotate

**Status:** ❌ **Not implemented** (Android: `setTextRotate` )

Set text rotate

**Planned syntax (not available yet)**

```
bool SetTextRotate(const std::shared_ptr<IPrinterConnection>& connection, bool enable);
```

**Workaround:** Use `SendCommand` for text rotate.

## SetTextLineHeight

**Status:** ❌ **Not implemented** (Android: `setTextLineHeight` )

Set text line height

**Planned syntax (not available yet)**

```
bool SetTextLineHeight(const std::shared_ptr<IPrinterConnection>& connection, int height);
```

**Workaround:** Try `SetLineSpacing` , or `SendCommand` .

## SetAsciiTextCharRightSpacing

**Status:** ❌ **Not implemented** (Android: `setAsciiTextCharRightSpacing` )

Set Ascii Text Char Right Spacing

**Planned syntax (not available yet)**

```
bool SetAsciiTextCharRightSpacing(const std::shared_ptr<IPrinterConnection>& connection, int
```

**Workaround:** Use `SendCommand` .

## SetKanjiTextCharSpacing

**Status:** ❌ **Not implemented** (Android: `setKanjiTextCharSpacing` )

Print one line with left and right text

**Planned syntax (not available yet)**

```
bool SetKanjiTextCharSpacing(const std::shared_ptr<IPrinterConnection>& connection, int spa
```

**Workaround:** Use `SendCommand` .

## SetPrintPosition

**Status:** ❌ **Not implemented** (Android: `setPrintPosition` )

Set Print Position

**Planned syntax (not available yet)**

```
bool SetPrintPosition(const std::shared_ptr<IPrinterConnection>& connection, int position);
```

**Workaround:** Use `SendCommand` for absolute print position.

## SetHorizontalAbsolutePrintPosition

**Status:** ❌ **Not implemented** (Android: `setHorizontalAbsolutePrintPosition` )

Set print position

**Planned syntax (not available yet)**

```
bool SetHorizontalAbsolutePrintPosition(const std::shared_ptr<IPrinterConnection>& connectio
```

**Workaround:** Use `SendCommand` (ESC \$).

## SetHorizontalRelativePrintPosition

**Status:** ❌ **Not implemented** (Android: `setHorizontalRelativePrintPosition` )

Set print position

**Planned syntax (not available yet)**

```
bool SetHorizontalRelativePrintPosition(const std::shared_ptr<IPrinterConnection>& connection
```

**Workaround:** Use `SendCommand` (ESC ).

## SetVerticalAbsolutePrintPosition

**Status:** ❌ **Not implemented** (Android: `setVerticalAbsolutePrintPosition` )

Set print position

**Planned syntax (not available yet)**

```
bool SetVerticalAbsolutePrintPosition(const std::shared_ptr<IPrinterConnection>& connection
```

**Workaround:** In page mode use draw APIs; otherwise `SendCommand` .

## SetVerticalRelativePrintPosition

**Status:** ❌ **Not implemented** (Android: `setVerticalRelativePrintPosition` )

Set print position

**Planned syntax (not available yet)**

```
bool SetVerticalRelativePrintPosition(const std::shared_ptr<IPrinterConnection>& connection
```

**Workaround:** In page mode use draw APIs; otherwise `SendCommand` .

## SetBarcodeReadableTextFontType

**Status:** ❌ **Not implemented** (Android: `setBarcodeReadableTextFontType` )

Print / draw barcode

**Planned syntax (not available yet)**

```
bool SetBarcodeReadableTextFontType(const std::shared_ptr<IPrinterConnection>& connection, :
```

**Workaround:** Use `SendCommand` for HRI font.

## SetBarcodeReadableTextPosition

**Status:** ❌ **Not implemented** (Android: `setBarcodeReadableTextPosition` )

Print / draw barcode

**Planned syntax (not available yet)**

```
bool SetBarcodeReadableTextPosition(const std::shared_ptr<IPrinterConnection>& connection, :
```

**Workaround:** Use existing `SetBarcodeHriPosition` .

## PrintHorizontalLine

**Status:** ❌ **Not implemented** (Android: `printHorizontalLine` )

Print horizontal line

**Planned syntax (not available yet)**

```
bool PrintHorizontalLine(const std::shared_ptr<IPrinterConnection>& connection, int start, :
```

**Workaround:** Print '-' text, or `SendCommand` .

## FixedHorizontalLine

**Status:** ❌ **Not implemented** (Android: `fixedHorizontalLine` )

Print horizontal line

**Planned syntax (not available yet)**

```
bool FixedHorizontalLine(const std::shared_ptr<IPrinterConnection>& connection);
```

**Workaround:** Print a text separator line instead.

## PrintAndFeedDots

**Status:** ❌ **Not implemented** (Android: `printAndFeedDots` )

Print And Feed Dots

**Planned syntax (not available yet)**

```
bool PrintAndFeedDots(const std::shared_ptr<IPrinterConnection>& connection, int dots);
```

**Workaround:** Use `FeedPaper(connection, dots)` (ESC J, dots).

## Page Mode (pending)

## DrawRectangleBorderInPageMode

**Status:** ❌ **Not implemented** (Android: `drawRectangleBorderInPageMode` )

Page mode

**Planned syntax (not available yet)**

```
bool DrawRectangleBorderInPageMode(const std::shared_ptr<IPrinterConnection>& connection, in
```

**Workaround:** Try existing `DrawRectangleInPageMode(..., mode, ...)` border mode.

## DrawFilledRectangleInPageMode

**Status:** ❌ **Not implemented** (Android: `drawFilledRectangleInPageMode` )

Page mode

**Planned syntax (not available yet)**

```
bool DrawFilledRectangleInPageMode(const std::shared_ptr<IPrinterConnection>& connection, in
```

**Workaround:** Try existing `DrawRectangleInPageMode(..., mode, ...)` filled mode.

## PrintNvBitmap

**Status:** ❌ **Not implemented** (Android: `printNvBitmap` )

NV bitmap

**Planned syntax (not available yet)**

```
bool PrintNvBitmap(const std::shared_ptr<IPrinterConnection>& connection, int n, int mode);
```

**Workaround:** Not wrapped; use `SendCommand` FS p.

## DrawNvBitmapInPageMode

**Status:** ❌ **Not implemented** (Android: `drawNvBitmapInPageMode` )

Page mode

**Planned syntax (not available yet)**

```
bool DrawNvBitmapInPageMode(const std::shared_ptr<IPrinterConnection>& connection, int x, in
```

**Workaround:** Not wrapped; use `SendCommand` .

## DrawNvBitmapCenteredInPageMode

**Status:** ❌ **Not implemented** (Android: `drawNvBitmapCenteredInPageMode` )

Page mode

**Planned syntax (not available yet)**

```
bool DrawNvBitmapCenteredInPageMode(const std::shared_ptr<IPrinterConnection>& connection, :
```

**Workaround:** Not wrapped; use `SendCommand` .

## DrawTextBIG5InPageMode

**Status:** ❌ **Not implemented** (Android: `drawTextBIG5InPageMode` )

Print text in BIG5

**Planned syntax (not available yet)**

```
bool DrawTextBIG5InPageMode(const std::shared_ptr<IPrinterConnection>& connection, int x, in
```

**Workaround:** Set BIG5 encoding then draw/send bytes.

## DrawTextShiftJisInPageMode

**Status:** ❌ **Not implemented** (Android: `drawTextShiftJISInPageMode` )

Print text in Shift-JIS

**Planned syntax (not available yet)**

```
bool DrawTextShiftJisInPageMode(const std::shared_ptr<IPrinterConnection>& connection, int :
```

**Workaround:** Set Shift-JIS encoding then draw/send bytes.

## DrawTextEucKrInPageMode

**Status:** ❌ **Not implemented** (Android: `drawTextEUCKRInPageMode` )

Print text in EUC-KR

**Planned syntax (not available yet)**

```
bool DrawTextEucKrInPageMode(const std::shared_ptr<IPrinterConnection>& connection, int x, :
```

**Workaround:** Set EUC-KR encoding then draw/send bytes.

## DrawImageFromFileInPageMode

**Status:** ❌ **Not implemented** (Android: `drawImageFromFileInPageMode` )

Page mode

**Planned syntax (not available yet)**

```
bool DrawImageFromFileInPageMode(const std::shared_ptr<IPrinterConnection>& connection, int
```

**Workaround:** Decode then use `PrintRasterImageFromFile` or label `DrawLabelImageFromFile` .

## DrawImageFromDataInPageMode

**Status:** ❌ **Not implemented** (Android: `drawImageFromDataInPageMode` )

Page mode

**Planned syntax (not available yet)**

```
bool DrawImageFromDataInPageMode(const std::shared_ptr<IPrinterConnection>& connection, int
```

**Workaround:** Use `PrintRasterImageFromData` in standard mode as a workaround.

## DrawImageFromPixelsInPageMode

**Status:** ❌ **Not implemented** (Android: `drawImageFromPixelsInPageMode` )

Page mode

**Planned syntax (not available yet)**

```
bool DrawImageFromPixelsInPageMode(const std::shared_ptr<IPrinterConnection>& connection, i
```

**Workaround:** Use `PrintRasterImageFromPixels` in standard mode as a workaround.

## Android-only APIs (not ported)

These depend on Android permissions/platform features and are **intentionally not provided** on Windows C++.

### OnRequestPermissionsResult

**Status:** 🚫 **Not ported** (Android: `onRequestPermissionsResult` )

Forward Android permission results.

**Note:** Not applicable on Windows; not ported.

### GetCommunicationManager

**Status:** 🚫 **Not ported** (Android: `getCommunicationManager` )

Get internal communication manager.

**Note:** Not exposed on Windows C++; not ported.

## HasUsbPermission

**Status:**  **Not ported** (Android: `hasUsbPermission` )

Whether USB permission is granted.

**Note:** No Windows USB runtime permission API; not ported.

## RequestUsbPermission

**Status:**  **Not ported** (Android: `requestUsbPermission` )

Request USB permission.

**Note:** No Windows USB runtime permission API; not ported.

## AddBluetoothListener

**Status:**  **Not ported** (Android: `addBluetoothListener` )

Add Bluetooth device listener.

**Note:** Use `DiscoverBluetoothDevices` / `GetBluetoothPairedDevices` instead.

## RemoveBluetoothListener

**Status:**  **Not ported** (Android: `removeBluetoothListener` )

Remove Bluetooth device listener.

**Note:** No listener model in C++; not ported.