PrinterSDK API 说明书

> 版本：1.6.3
> 包名：`com.hspos.printerSDK`
> 入口类：`PrinterSDK`
> 常量类：`PrinterConstants`
> 配套文档：PrinterSDK_使用说明书.md、PrinterSDK_函数方法说明.md
> English edition: PrinterSDK_API_Documentation.md

目录

1. SDK 简介
2. 常见问题
3. 枚举与宏（`PrinterConstants`）
4. 接口说明

- 生命周期与权限
- getInstance
- release
- onRequestPermissionsResult
- getCommunicationManager
- 端口 / 连接功能
- getUsbDevices
- connectUsbPrinter
- getBluetoothPairedDevices
- connectBluetoothPrinter
- connectNetworkPrinter
- connectNetworkPrinterAsync
- getSerialDevices
- connectSerialPrinter
- disconnectPrinter
- getBluetoothDevices
- getBluetoothPrinters
- startBluetoothDiscovery
- stopBluetoothDiscovery
- refreshBluetoothDevices
- getBluetoothAvailableDevices
- hasUsbPermission
- requestUsbPermission
- addBluetoothListener

- removeBluetoothListener
- 打印机状态功能
- clearPrinterBuffer
- cancelPrint
- queryRTStatus
- queryTransmitStatus
- getPrinterStatus
- isPrinterOnline
- hasPaper
- isCoverOpen
- 票据打印功能
- initPrinter
- resetPrinter
- printText
- printTextLineLeftRight
- printTextLineLeftRight
- printTextInUTF8
- printTextInGBK
- printTextInBIG5
- printTextInShiftJIS
- printTextInEUCKR
- setAlignment
- setBold
- setUnderline
- setReverse
- setLineSpacing
- setCharSpacing
- printBarcode
- printQRCode
- printQRCodeStandard
- printDoubleQRCode
- printPDF417
- printRasterImageFromFile
- printRasterImageFromFile
- printRasterImageFromFile
- printRasterImageFromData
- printRasterImageFromData
- printRasterImageFromData
- printRasterImageFromPixels
- printRasterImageFromPixels
- printRasterImageFromPixels
- cutPaper
- feedAndCutPaper
- feedLine
- feedPaper
- feedPaper

- reverseFeed
- horizontalTab
- setTabPositions
- openCashDrawer
- beep
- beep
- setPrintMode
- setSingleByteMode
- setCharacterSet
- setCharacterCodepage
- setChineseMode
- setChineseDoubleSize
- setChineseDoubleSize
- setDoubleByteEncoding
- setTextScale
- setAsciiTextFontType
- setTextUpsideDown
- setTextRotate
- setTextLineHeight
- setAsciiTextCharRightSpacing
- setKanjiTextCharSpacing
- setPrintPosition
- setPrintAreaLeftMargin
- setPrintAreaWidth
- setHorizontalAbsolutePrintPosition
- setHorizontalRelativePrintPosition
- setVerticalAbsolutePrintPosition
- setVerticalRelativePrintPosition
- setBarcodeHRIPosition
- setBarcodeHeight
- setBarcodeWidth
- setBarcodeReadableTextFontType
- setBarcodeReadableTextPosition
- printHorizontalLine
- fixedHorizontalLine
- printAndFeedDots
- sendSelfTest
- sendCommand
- clearBuffer
- cancelPrintCmd
- 页模式功能
- selectPageMode
- exitPageMode
- selectPageModeEx
- selectPageModeEx
- selectPageModeEx

- pagePrintPage
- pageClearPage
- setMovementUnits
- pageSetPageArea
- pageSetPageDrawDirection
- setPrintAreaInPageMode
- setPrintAreaDirection
- drawLineInPageMode
- drawLineInPageMode
- estimateCode128WidthDots
- drawRectangleInPageMode
- drawRectangleBorderInPageMode
- drawFilledRectangleInPageMode
- drawTextInPageMode
- estimateTextWidthDots
- drawTextCenteredInPageMode
- drawTextCenteredInPageMode
- drawTextRightInPageMode
- drawTextRightInPageMode
- drawTextLineLeftRightInPageMode
- drawTextLineLeftRightInPageMode
- printNvBitmap
- drawNvBitmapInPageMode
- drawNvBitmapCenteredInPageMode
- drawTextUTF8InPageMode
- drawTextGBKInPageMode
- drawTextBIG5InPageMode
- drawTextShiftJISInPageMode
- drawTextEUCKRInPageMode
- drawBarcodeInPageMode
- drawBarcodeInPageMode
- drawBarcodeInPageModeWithCenteredHri
- drawQRCodeInPageMode
- drawImageFromFileInPageMode
- drawImageFromDataInPageMode
- drawImageFromPixelsInPageMode
- 标签打印功能
- enableLabelMode
- disableLabelMode
- labelPageBegin
- backPaperToPrintPosition
- labelPagePrint
- labelPagePrint
- printLabelText
- printLabelText
- printLabelTextUTF8

- printLabelTextUTF8
- printLabelTextGBK
- printLabelTextGBK
- printLabelBarcode
- printLabelBarcode
- setLabelBarcodeTextPosition
- printLabelQRCode
- printLabelQRCode
- printLabelPDF417Code
- printLabelPDF417Code
- drawLabelImageFromFile
- drawLabelImageFromData
- drawLabelImageFromPixels
- drawLabelImageAdvanced
- drawLabelLine
- drawLabelLine
- drawLabelRect
- drawLabelRect
- drawLabelBox
- 其它功能

SDK 简介

1. 开发前请先运行测试 App，确认能正常出纸后再接入业务工程。
2. PrinterSDK 支持 ESC/POS 小票、页模式坐标排版、JPL 标签打印。
3. 对外统一入口：`PrinterSDK.getInstance(context)`。
4. 主要模块：
   - **端口 / 连接**：USB、蓝牙、网络（TCP）、串口
   - **票据 POS**：文本、条码、二维码、图片、切纸、钱箱、蜂鸣
   - **页模式**：按 X/Y 坐标绘制
   - **标签**：JPL 标签布局与打印
5. 典型流程：**连接 → 初始化/复位 → 打印 API → 走纸/切纸 → 断开。**
6. 多数打印 API 返回 `boolean`：`true` 表示指令已写入端口，不代表打印机机械动作已完成。

```
import com.hspos.printerSDK.PrinterSDK;
import com.hspos.printerSDK.PrinterConstants;
import com.hspos.printerSDK.communication.PrinterConnection;

PrinterSDK sdk = PrinterSDK.getInstance(context);
PrinterConnection conn = sdk.connectUsbPrinter(device);
sdk.initPrinter(conn);
sdk.printTextInGBK(conn, "你好\n");
sdk.feedAndCutPaper(conn);
sdk.disconnectPrinter(conn);
```

常见问题

1. **小票纸和标签纸分别用哪套 API?**
   连续纸 / 小票 → POS / 页模式。间隙标签纸 → 先 `enableLabelMode`，再用标签 API。

2. **为什么提示成功但打印机不动?**
   `true` 只表示写入成功。请检查纸张类型、是否进入标签模式、电源，以及是否调用了 `pagePrintPage` / `labelPagePrint`。

3. **条码 TYPE 怎么选?**
   ESC/POS 小票用 `BARCODE_TYPE_*`（推荐 `BARCODE_TYPE_CODE128`）。
   JPL 标签用 `LABEL_BARCODE_TYPE_*`（编号体系不同）。

4. **页模式内容重叠**
   部分固件忽略 `GS $` / `ESC $` 高字节。坐标尽量落在 0–255；文字 Y 多为基线，条码 Y 为顶部。

5. **自检无输出**
   当前 SDK 使用 `GS ( A`（`1D 28 41 02 00 00 02`）。旧版 `DC2 T` 仅作兼容保留。

寻找蓝牙打印机（常用流程）

相关 API 位于下文 **「端口 / 连接功能」**，名称如下：

方法	作用
`getBluetoothPairedDevices()`	获取**已配对**设备（最快，不必扫描）
`startBluetoothDiscovery()`	**开始搜索**附近蓝牙设备
`stopBluetoothDiscovery()`	停止搜索
`refreshBluetoothDevices()`	刷新列表（已配对 + 扫描未配对）
`getBluetoothAvailableDevices()`	获取扫描结果（含未配对）
`getBluetoothPrinters()` / `getBluetoothDevices()`	获取蓝牙打印机列表（内部别名）
`addBluetoothListener(listener)`	注册回调，搜索到设备时 `onDeviceFound`
`removeBluetoothListener(listener)`	移除回调
`connectBluetoothPrinter(device)`	连接选中的设备

```
// 1) 先看已配对
List<BluetoothDevice> paired = sdk.getBluetoothPairedDevices();

// 2) 搜索附近新设备
sdk.addBluetoothListener(device -> {
    // 在 UI 线程更新列表
    Log.d("BT", "found: " + device.getName() + " " + device.getAddress());
});
sdk.startBluetoothDiscovery();
// ... 等待数秒后
List<BluetoothDevice> all = sdk.getBluetoothAvailableDevices();
sdk.stopBluetoothDiscovery();

// 3) 连接
PrinterConnection conn = sdk.connectBluetoothPrinter(selectedDevice);
```

> 也可用 `refreshBluetoothDevices()` 一次刷新已配对与扫描结果，再调用 `getBluetoothAvailableDevices()`。

枚举与宏（`PrinterConstants`）

对齐与切纸

语法

```
public static final int ALIGNMENT_LEFT = 0;
public static final int ALIGNMENT_CENTER = 1;
public static final int ALIGNMENT_RIGHT = 2;
public static final int CUT_FULL = 0;
public static final int CUT_PARTIAL = 1;
// 58mm 纸有效打印宽度（点，8 dot/mm）
public static final int PAPER_WIDTH_58MM = 384;
// 80mm 纸有效打印宽度（点，8 dot/mm）
public static final int PAPER_WIDTH_80MM = 576;
```

NV 位图（打印机内存 Logo，FS p）

语法

```
// NV 位图正常尺寸
public static final int NV_BITMAP_MODE_NORMAL = 0;
// NV 位图倍宽
public static final int NV_BITMAP_MODE_DOUBLE_WIDTH = 1;
// NV 位图倍高
public static final int NV_BITMAP_MODE_DOUBLE_HEIGHT = 2;
// NV 位图倍宽倍高
public static final int NV_BITMAP_MODE_DOUBLE_BOTH = 3;
// 不打印条码可读文字（HRI）
public static final int BARCODE_HRI_NONE = 0;
// 在条码下方打印可读文字（常用）
public static final int BARCODE_HRI_BELOW = 2;
```

条码类型（ESC/POS GS k m 值，用于 printBarcode / drawBarcodeInPageMode）

语法

```
// UPC-A，北美 12 位商品码，数据需 11~12 位数字
public static final int BARCODE_TYPE_UPC_A = 0x41;
// UPC-E，北美 8 位压缩商品码
public static final int BARCODE_TYPE_UPC_E = 0x42;
// EAN-13，国际 13 位商品码，数据需 12~13 位数字
public static final int BARCODE_TYPE_EAN13 = 0x43;
// EAN-8，国际 8 位商品码，数据需 7~8 位数字
public static final int BARCODE_TYPE_EAN8 = 0x44;
// Code 39，工业标识，支持数字、大写字母及 -. $/+%
public static final int BARCODE_TYPE_CODE39 = 0x45;
// ITF（交叉二五码），数据必须为偶数位数字
public static final int BARCODE_TYPE_ITF = 0x46;
// Codabar，以 A~D 开头和结尾
public static final int BARCODE_TYPE_CODABAR = 0x47;
// Code 93，高密度编码
public static final int BARCODE_TYPE_CODE93 = 0x48;
// Code 128，通用首选。测试条码功能时推荐使用此类型，支持字母和数字。 <p>示例：{
public static final int BARCODE_TYPE_CODE128 = 0x49;
public static final int BARCODE_CODE128 = BARCODE_TYPE_CODE128;
public static final int BARCODE_CODE39 = BARCODE_TYPE_CODE39;
public static final int BARCODE_EAN13 = BARCODE_TYPE_EAN13;
```

二维码

语法

```
public static final int QRCODE_SIZE_SMALL = 1;
public static final int QRCODE_SIZE_NORMAL = 2;
public static final int QRCODE_SIZE_LARGE = 3;
public static final int QR_ECC_LEVEL_L = 1;
public static final int QR_ECC_LEVEL_M = 2;
public static final int QR_ECC_LEVEL_Q = 3;
public static final int QR_ECC_LEVEL_H = 4;
```

页模式方向

语法

```
public static final int PAGE_DIRECTION_LEFT_TO_RIGHT = 0;
public static final int PAGE_DIRECTION_BOTTOM_TO_TOP = 1;
public static final int PAGE_DIRECTION_RIGHT_TO_LEFT = 2;
public static final int PAGE_DIRECTION_TOP_TO_BOTTOM = 3;
```

打印机状态位（getPrinterStatus 组合返回值，非原始 DLE EOT 字节）

语法

```
// 查询失败或无响应
public static final int STATUS_OFFLINE = 0x00;
// 在线（由 DLE EOT n=1 的 bit3=0 合成）
public static final int STATUS_ONLINE = 0x01;
// 缺纸（由 DLE EOT n=4 纸尽位合成）
public static final int STATUS_PAPER_OUT = 0x02;
// 开盖（由 DLE EOT n=2 的 bit2 合成）
public static final int STATUS_COVER_OPEN = 0x04;
// GS r / DLE EOT 查询类型：打印机状态
public static final int STATUS_TYPE_PRINTER = 1;
// GS r / DLE EOT 查询类型：脱机状态
public static final int STATUS_TYPE_OFFLINE = 2;
// GS r / DLE EOT 查询类型：错误状态
public static final int STATUS_TYPE_ERROR = 3;
// GS r / DLE EOT 查询类型：纸传感器状态
public static final int STATUS_TYPE_PAPER_SENSOR = 4;
```

钱箱

语法

```
public static final int CASH_DRAWER_PIN2 = 0;
public static final int CASH_DRAWER_PIN5 = 1;
```

蜂鸣默认值

语法

```
public static final int BEEP_DEFAULT_COUNT = 1;
// 默认蜂鸣时长参数（ESC B 的 t 值，与 {
public static final int BEEP_DEFAULT_DURATION = 100;
```

光栅图压缩模式（printRasterImageFrom*）

语法

```
// GS v 0 标准光栅，不压缩
public static final int RASTER_COMPRESSION_GS_V0 = 0;
// US ( P 逐行 RLE（压缩算法1）；需打印机固件支持，多数机型不兼容，勿用于 AUTO
public static final int RASTER_COMPRESSION_LINE_RLE = 1;
// US ( J JBIG（压缩算法2）
public static final int RASTER_COMPRESSION_JBIG = 2;
// 自动选择传输量最小的方案
public static final int RASTER_COMPRESSION_AUTO = 3;
```

标签模式

语法

```
public static final int LABEL_FONT_HEIGHT_24 = 24;
// 标签模式 Code 128 条码类型（JPL 编号 8，与 ESC/POS 的 BARCODE_TYPE_* 不同）
public static final int LABEL_BARCODE_TYPE_CODE128 = 8;
public static final int LABEL_BARCODE_HEIGHT_DEFAULT = 80;
public static final int LABEL_BARCODE_UNIT_WIDTH_2 = 2;
```

双字节编码

语法

```
public static final int DOUBLE_BYTE_ENCODING_GBK = 0;
public static final int DOUBLE_BYTE_ENCODING_UTF8 = 1;
public static final int DOUBLE_BYTE_ENCODING_BIG5 = 3;
```

ASCII 字体

语法

```
public static final int ASCII_FONT_TYPE_A = 0;
public static final int ASCII_FONT_TYPE_B = 1;
```

行间距

语法

```
public static final int LINE_SPACING_DEFAULT = 0;
public static final int LINE_SPACING_1_8 = 1;
public static final int LINE_SPACING_1_6 = 2;
```

打印模式（ESC ! n）

语法

```
public static final int PRINT_MODE_NORMAL = 0;
public static final int PRINT_MODE_DOUBLE_WIDTH = 1;
public static final int PRINT_MODE_DOUBLE_HEIGHT = 2;
public static final int PRINT_MODE_DOUBLE = 3;
public static final int PRINT_MODE_BOLD = 8;
public static final int PRINT_MODE_REVERSE = 16;
public static final int PRINT_MODE_UNDERLINE = 128;
```

汉字倍宽倍高（FS W n）

语法

```
public static final int CHINESE_DOUBLE_SIZE_NORMAL = 0;
public static final int CHINESE_DOUBLE_SIZE_ON = 1;
```

ESC/POS 指令字节

语法

```
public static final byte[] CMD_INIT = new byte[] { 0x1B, 0x40 };
public static final byte[] CMD_CUT = new byte[] { 0x1D, 0x56, 0x00 };
public static final byte[] CMD_FEED_LINE = new byte[] { 0x0A };
public static final byte[] CMD_FEED_PAPER = new byte[] { 0x1B, 0x4A, 0x00 };
public static final byte[] CMD_CLEAR_BUFFER = new byte[] { 0x18 };
public static final byte[] CMD_CANCEL_PRINT = new byte[] { 0x18 };
public static final byte[] CMD_RESET_PRINTER = new byte[] { 0x1B, 0x40 };
public static final byte[] CMD_SELECT_PAGE_MODE = new byte[] { 0x1B, 0x4C };
public static final byte[] CMD_EXIT_PAGE_MODE = new byte[] { 0x1B, 0x53 };
// 自检：GS ( A pL pH n m — 打印十六进制转储（Epson 兼容，多数机型有效） 旧版 DC2 T（12 54）在部分机型上无动作，已弃用。
public static final byte[] CMD_SELF_TEST = new byte[] {
        0x1D, 0x28, 0x41, 0x02, 0x00, 0x00, 0x02
    };
// 兼容旧机型的 DC2 T 自检（部分国产机）
public static final byte[] CMD_SELF_TEST_DC2_T = new byte[] { 0x12, 0x54 };
```

JPL 标签指令

语法

```
public static final byte[] JPL_ENABLE_LABEL_MODE = new byte[] { 0x1C, 0x4C };
public static final byte[] JPL_DISABLE_LABEL_MODE = new byte[] { 0x1C, 0x4C, 0x00 };
public static final byte[] JPL_BACK_PAPER_TO_PRINT_POSITION = new byte[] { 0x1C, 0x42 };
// 标签页打印 1 份：1A 4F 01 n（与 pageBegin/draw* 同属 1A 指令族） 注意：旧值 FS P（1C 50 01）与绘制指令体系不一致，会导致“发送成功但不出纸”。
public static final byte[] JPL_PAGE_PRINT_SINGLE = new byte[] { 0x1A, 0x4F, 0x01, 0x01 };
// 兼容旧固件的 FS P 打印（少数机型）
public static final byte[] JPL_PAGE_PRINT_SINGLE_FS = new byte[] { 0x1C, 0x50, 0x01 };
public static final byte JPL_DRAW_TEXT_BASIC = 0x00;
public static final byte JPL_DRAW_TEXT_ADVANCED = 0x01;
```

接口说明

生命周期与权限

getInstance

getInstance

语法

```
public PrinterSDK getInstance(Context context);
```

```
// getInstance
//
//
// context
// Android Context（Application 或 Activity）。
//
// 返回值
// PrinterSDK 单例。
```

release

释放资源

语法

```
public void release();
```

```
// 释放资源
//
//
// 返回值
// 无返回值。
```

onRequestPermissionsResult

处理权限申请结果（由Activity调用） 必须在Activity的onRequestPermissionsResult中调用此方法

语法

```
public void onRequestPermissionsResult(int requestCode, String[] permissions, int[]
grantResults);
```

```
// 处理权限申请结果（由Activity调用） 必须在Activity的onRequestPermissionsResult中调用此方法
//
//
// requestCode
// 权限请求码。
//
// permissions
// 权限名数组。
//
// grantResults
// 授权结果数组。
//
// 返回值
// 无返回值。
```

getCommunicationManager

获取通信管理器（用于高级功能）

语法

```
public CommunicationManager getCommunicationManager();
```

```
// 获取通信管理器（用于高级功能）
//
//
// 返回值
// CommunicationManager 实例。
```

端口 / 连接功能

getUsbDevices

获取USB设备列表

语法

```
public List<UsbDevice> getUsbDevices();
```

```
// 获取USB设备列表
//
//
// 返回值
// 设备列表；失败时返回空列表。
```

connectUsbPrinter

连接USB打印机（自动处理权限申请）

语法

```
public PrinterConnection connectUsbPrinter(UsbDevice device);
```

```
// 连接USB打印机（自动处理权限申请）
//
//
// device
// 目标设备对象。
//
// 返回值
// 成功返回连接句柄，失败返回 null。
```

getBluetoothPairedDevices

获取已配对的蓝牙设备（自动处理权限申请）

语法

```
public List<BluetoothDevice> getBluetoothPairedDevices();
```

```
// 获取已配对的蓝牙设备（自动处理权限申请）
//
//
// 返回值
// 设备列表；失败时返回空列表。
```

connectBluetoothPrinter

连接蓝牙打印机（自动处理权限申请）

语法

```
public PrinterConnection connectBluetoothPrinter(BluetoothDevice device);
```

```
// 连接蓝牙打印机（自动处理权限申请）
//
//
// device
// 目标设备对象。
//
// 返回值
// 成功返回连接句柄，失败返回 null。
```

connectNetworkPrinter

连接网络打印机（同步方法，不推荐在主线程中使用）

语法

```
public PrinterConnection connectNetworkPrinter(String ip, int port);
```

```
// 连接网络打印机（同步方法，不推荐在主线程中使用）
//
//
// ip
// 打印机 IP 地址。
//
// port
// TCP 端口（通常 9100）。
//
// 返回值
// 成功返回连接句柄，失败返回 null。
```

connectNetworkPrinterAsync

异步连接网络打印机（推荐使用，自动在后台线程执行）

语法

```
public void connectNetworkPrinterAsync(String ip, int port,
CommunicationManager.NetworkConnectionCallback callback);
```

```
// 异步连接网络打印机（推荐使用，自动在后台线程执行）
//
//
// ip
// 打印机 IP 地址。
//
// port
// TCP 端口（通常 9100）。
//
// callback
// 异步连接回调。
//
// 返回值
// 无返回值。
```

getSerialDevices

获取串口设备列表

语法

```
public List<String> getSerialDevices();
```

```
// 获取串口设备列表
//
//
// 返回值
// 设备列表；失败时返回空列表。
```

connectSerialPrinter

连接串口打印机

语法

```
public PrinterConnection connectSerialPrinter(String devicePath, int baudRate);
```

```
// 连接串口打印机
//
//
// devicePath
// 串口设备路径，如 /dev/ttyS1。
//
// baudRate
// 串口波特率，如 9600。
//
// 返回值
// 成功返回连接句柄，失败返回 null。
```

disconnectPrinter

断开打印机连接

语法

```
public void disconnectPrinter(PrinterConnection connection);
```

```
// 断开打印机连接
//
//
// connection
// 由 connectXxx() 返回的打印机连接句柄。
//
// 返回值
// 无返回值。
```

getBluetoothDevices

获取蓝牙打印机列表（自动处理权限申请）

语法

```
public List<BluetoothDevice> getBluetoothDevices();
```

```
// 获取蓝牙打印机列表（自动处理权限申请）
//
//
// 返回值
// 设备列表；失败时返回空列表。
```

getBluetoothPrinters

获取蓝牙打印机列表（别名函数，与内部函数名保持一致）

语法

```
public List<BluetoothDevice> getBluetoothPrinters();
```

```
// 获取蓝牙打印机列表（别名函数，与内部函数名保持一致）
//
//
// 返回值
// 设备列表；失败时返回空列表。
```

startBluetoothDiscovery

开始搜索附近的蓝牙设备（寻找蓝牙打印机）。需配合 `addBluetoothListener` 或之后调用 `getBluetoothAvailableDevices()` 取结果。

语法

```
public boolean startBluetoothDiscovery();
```

```
// 开始搜索附近的蓝牙设备（寻找蓝牙打印机）。需配合 addBluetoothListener 或之后调用 getBluetoothAvailableDevices() 取结果。
//
//
// 返回值
// 成功启动扫描返回 true，否则返回 false（权限不足、蓝牙未开等）。
```

stopBluetoothDiscovery

停止蓝牙设备搜索

语法

```
public void stopBluetoothDiscovery();
```

```
// 停止蓝牙设备搜索
//
//
// 返回值
// 无返回值。
```

refreshBluetoothDevices

刷新蓝牙设备列表（已配对 + 扫描未配对）。适合"一键寻找打印机"。

语法

```
public boolean refreshBluetoothDevices();
```

```
// 刷新蓝牙设备列表（已配对 + 扫描未配对）。适合"一键寻找打印机"。
//
//
// 返回值
// 刷新流程启动成功返回 true，否则返回 false。
```

getBluetoothAvailableDevices

获取扫描到的蓝牙设备（含已配对与未配对），通常在 `startBluetoothDiscovery` / `refreshBluetoothDevices` 之后调用。

语法

```
public List<BluetoothDevice> getBluetoothAvailableDevices();
```

```
// 获取扫描到的蓝牙设备（含已配对与未配对），通常在 startBluetoothDiscovery / refreshBluetoothDevices 之后调用。
//
//
// 返回值
// 设备列表；失败时返回空列表。
```

hasUsbPermission

检查USB设备权限

语法

```
public boolean hasUsbPermission(UsbDevice device);
```

```
// 检查USB设备权限
//
//
// device
// 目标设备对象。
//
// 返回值
// 指令写入成功返回 true，否则返回 false。
```

requestUsbPermission

请求USB设备权限

语法

```
public boolean requestUsbPermission(UsbDevice device);
```

```
// 请求USB设备权限
//
//
// device
// 目标设备对象。
//
// 返回值
// 指令写入成功返回 true，否则返回 false。
```

addBluetoothListener

添加蓝牙设备监听器

语法

```
public void
addBluetoothListener(com.hspos.printerSDK.communication.bluetooth.BluetoothPrinterManager.Bluet
oothDeviceListener listener);
```

```
// 添加蓝牙设备监听器
//
//
// listener
// 蓝牙发现监听器。
//
// 返回值
// 无返回值。
```

removeBluetoothListener

移除蓝牙设备监听器

语法

```
public void
removeBluetoothListener(com.hspos.printerSDK.communication.bluetooth.BluetoothPrinterManager.Bl
uetoothDeviceListener listener);
```

```
// 移除蓝牙设备监听器
//
//
// listener
// 蓝牙发现监听器。
//
// 返回值
// 无返回值。
```

打印机状态功能

clearPrinterBuffer

清除打印机缓冲区

语法

```
public boolean clearPrinterBuffer(PrinterConnection connection);
```

```
// 清除打印机缓冲区
//
//
// connection
// 由 connectXxx() 返回的打印机连接句柄。
//
// 返回值
// 指令写入成功返回 true，否则返回 false。
```

cancelPrint

取消打印

语法

```
public boolean cancelPrint(PrinterConnection connection);
```

```
// 取消打印
//
//
// connection
// 由 connectXxx() 返回的打印机连接句柄。
//
// 返回值
// 指令写入成功返回 true，否则返回 false。
```

queryRTStatus

查询实时状态

语法

```
public byte[] queryRTStatus(PrinterConnection connection, int type, int timeout);
```

```
// 查询实时状态
//
//
// connection
// 由 connectXxx() 返回的打印机连接句柄。
//
// type
// 类型常量（条码/切纸/状态等）。
//
// timeout
// 读取超时（毫秒）。
//
// 返回值
// 成功返回原始状态字节；超时或无响应返回 null。
```

queryTransmitStatus

传送状态（GS r）

语法

```
public byte[] queryTransmitStatus(PrinterConnection connection, int type, int timeout);
```

```
// 传送状态（GS r）
//
//
// connection
// 由 connectXxx() 返回的打印机连接句柄。
//
// type
// 类型常量（条码/切纸/状态等）。
//
// timeout
// 读取超时（毫秒）。
//
// 返回值
// 成功返回原始状态字节；超时或无响应返回 null。
```

getPrinterStatus

按 ESC/POS（DLE EOT）查询并合成状态标志。 返回值为 {@link PrinterConstants#STATUS_ONLINE} / {@link PrinterConstants#STATUS_PAPER_OUT} / {@link PrinterConstants#STATUS_COVER_OPEN} 的按位组合； 查询失败返回 {@link PrinterConstants#STATUS_OFFLINE}。 位解析依据 Epson ESC/POS：

语法

```
public int getPrinterStatus(PrinterConnection connection);
```

```
// 按 ESC/POS（DLE EOT）查询并合成状态标志。 返回值为 {@link PrinterConstants#STATUS_ONLINE} / {@link PrinterConstants#STATUS_PAPER_OUT} / {@link PrinterConstants#STATUS_COVER_OPEN} 的按位组合； 查询失败返回 {@link PrinterConstants#STATUS_OFFLINE}。 位解析依据 Epson ESC/POS：
//
//
// connection
// 由 connectXxx() 返回的打印机连接句柄。
//
// 返回值
// 整型结果；状态类 API 失败时可能返回 STATUS_OFFLINE。
```

isPrinterOnline

检查打印机是否在线

语法

```
public boolean isPrinterOnline(PrinterConnection connection);
```

```
// 检查打印机是否在线
//
//
// connection
// 由 connectXxx() 返回的打印机连接句柄。
//
// 返回值
// 指令写入成功返回 true，否则返回 false。
```

hasPaper

检查是否有纸（未置缺纸位则视为有纸）

语法

```
public boolean hasPaper(PrinterConnection connection);
```

```
// 检查是否有纸（未置缺纸位则视为有纸）
//
//
// connection
// 由 connectXxx() 返回的打印机连接句柄。
//
// 返回值
// 指令写入成功返回 true，否则返回 false。
```

isCoverOpen

检查是否开盖

语法

```
public boolean isCoverOpen(PrinterConnection connection);
```

```
// 检查是否开盖
//
//
// connection
// 由 connectXxx() 返回的打印机连接句柄。
//
// 返回值
// 指令写入成功返回 true，否则返回 false。
```

票据打印功能

initPrinter

初始化打印机

语法

```
public boolean initPrinter(PrinterConnection connection);
```

```
// 初始化打印机
//
//
// connection
// 由 connectXxx() 返回的打印机连接句柄。
//
// 返回值
// 指令写入成功返回 true，否则返回 false。
```

resetPrinter

重置打印机

语法

```
public boolean resetPrinter(PrinterConnection connection);
```

```
// 重置打印机
//
//
// connection
// 由 connectXxx() 返回的打印机连接句柄。
//
// 返回值
// 指令写入成功返回 true，否则返回 false。
```

printText

打印文本

语法

```
public boolean printText(PrinterConnection connection, String text);
```

```
// 打印文本
//
//
// connection
// 由 connectXxx() 返回的打印机连接句柄。
//
// text
// 要打印的文本。
//
// 返回值
// 指令写入成功返回 true，否则返回 false。
```

printTextLineLeftRight

标准模式：一行左右分列（左对齐 + 右对齐）

语法

```
public boolean printTextLineLeftRight(PrinterConnection connection, String left, String right,
int areaWidthDots, int charWidthDots);
```

```
// 标准模式：一行左右分列（左对齐 + 右对齐）
//
//
// connection
// 由 connectXxx() 返回的打印机连接句柄。
//
// left
// 左侧文本。
//
// right
// 右侧文本。
//
// areaWidthDots
// 详见方法说明。
//
// charWidthDots
// 字符宽度估算（Font A 约 12 点）。
//
// 返回值
// 指令写入成功返回 true，否则返回 false。
```

printTextLineLeftRight

80mm 纸默认宽度（576 点）左右分列

语法

```
public boolean printTextLineLeftRight(PrinterConnection connection, String left, String right);
```

```
// 80mm 纸默认宽度（576 点）左右分列
//
//
// connection
// 由 connectXxx() 返回的打印机连接句柄。
//
// left
// 左侧文本。
//
// right
// 右侧文本。
//
// 返回值
// 指令写入成功返回 true，否则返回 false。
```

printTextInUTF8

UTF8编码打印文本

语法

```
public boolean printTextInUTF8(PrinterConnection connection, String text);
```

```
// UTF8编码打印文本
//
//
// connection
// 由 connectXxx() 返回的打印机连接句柄。
//
// text
// 要打印的文本。
//
// 返回值
// 指令写入成功返回 true，否则返回 false。
```

printTextInGBK

GBK编码打印文本

语法

```
public boolean printTextInGBK(PrinterConnection connection, String text);
```

```
// GBK编码打印文本
//
//
// connection
// 由 connectXxx() 返回的打印机连接句柄。
//
// text
// 要打印的文本。
//
// 返回值
// 指令写入成功返回 true，否则返回 false。
```

printTextInBIG5

BIG5编码打印文本

语法

```
public boolean printTextInBIG5(PrinterConnection connection, String text);
```

```
// BIG5编码打印文本
//
//
// connection
// 由 connectXxx() 返回的打印机连接句柄。
//
// text
// 要打印的文本。
//
// 返回值
// 指令写入成功返回 true，否则返回 false。
```

printTextInShiftJIS

Shift_JIS编码打印文本

语法

```
public boolean printTextInShiftJIS(PrinterConnection connection, String text);
```

```
// Shift_JIS编码打印文本
//
//
// connection
// 由 connectXxx() 返回的打印机连接句柄。
//
// text
// 要打印的文本。
//
// 返回值
// 指令写入成功返回 true，否则返回 false。
```

printTextInEUCKR

EUC-KR编码打印文本

语法

```
public boolean printTextInEUCKR(PrinterConnection connection, String text);
```

```
// EUC-KR编码打印文本
//
//
// connection
// 由 connectXxx() 返回的打印机连接句柄。
//
// text
// 要打印的文本。
//
// 返回值
// 指令写入成功返回 true，否则返回 false。
```

setAlignment

设置对齐方式

语法

```
public boolean setAlignment(PrinterConnection connection, int alignment);
```

```
// 设置对齐方式
//
//
// connection
// 由 connectXxx() 返回的打印机连接句柄。
//
// alignment
// ALIGNMENT_LEFT / CENTER / RIGHT。
//
// 返回值
// 指令写入成功返回 true，否则返回 false。
```

setBold

设置加粗

语法

```
public boolean setBold(PrinterConnection connection, boolean bold);
```

```
// 设置加粗
//
//
// connection
// 由 connectXxx() 返回的打印机连接句柄。
//
// bold
// true 表示加粗。
//
// 返回值
// 指令写入成功返回 true，否则返回 false。
```

setUnderline

设置下划线

语法

```
public boolean setUnderline(PrinterConnection connection, boolean underline);
```

```
// 设置下划线
//
//
// connection
// 由 connectXxx() 返回的打印机连接句柄。
//
// underline
// true 表示下划线。
//
// 返回值
// 指令写入成功返回 true，否则返回 false。
```

setReverse

设置反白打印

语法

```
public boolean setReverse(PrinterConnection connection, boolean reverse);
```

```
// 设置反白打印
//
//
// connection
// 由 connectXxx() 返回的打印机连接句柄。
//
// reverse
// true 表示反白。
//
// 返回值
// 指令写入成功返回 true，否则返回 false。
```

setLineSpacing

设置行间距

语法

```
public boolean setLineSpacing(PrinterConnection connection, int spacing);
```

```
// 设置行间距
//
//
// connection
// 由 connectXxx() 返回的打印机连接句柄。
//
// spacing
// 间距（点）。
//
// 返回值
// 指令写入成功返回 true，否则返回 false。
```

setCharSpacing

设置字符间距

语法

```
public boolean setCharSpacing(PrinterConnection connection, int spacing);
```

```
// 设置字符间距
//
//
// connection
// 由 connectXxx() 返回的打印机连接句柄。
//
// spacing
// 间距（点）。
//
// 返回值
// 指令写入成功返回 true，否则返回 false。
```

printBarcode

打印条形码 测试时推荐 {@link PrinterConstants#BARCODE_TYPE_CODE128} + 任意字母数字数据。不要手写数字，也不要与标签模式的条码类型（0~29）混用。

语法

```
public boolean printBarcode(PrinterConnection connection, int type, String data, int hri, int height, int width);
```

```
// 打印条形码 测试时推荐 {@link PrinterConstants#BARCODE_TYPE_CODE128} + 任意字母数字数据。 不要手写数字，也不要与标签模式的条码类型（0~29）混用。
//
//
// connection
// 由 connectXxx() 返回的打印机连接句柄。
//
// type
// 类型常量（条码/切纸/状态等）。
//
// data
// 数据内容（字符串或字节）。
//
// hri
// 条码 HRI 位置（BARCODE_HRI_*）。
//
// height
// 高度（点）。
//
// width
// 宽度（点）/ 模块宽度。
//
// 返回值
// 指令写入成功返回 true，否则返回 false。
```

printQRCode

打印二维码

语法

```
public boolean printQRCode(PrinterConnection connection, String content, int size);
```

```
// 打印二维码
//
//
// connection
// 由 connectXxx() 返回的打印机连接句柄。
//
// content
// 二维码 / 文本内容。
//
// size
// 模块大小。
//
// 返回值
// 指令写入成功返回 true，否则返回 false。
```

printQRCodeStandard

打印二维码（标准版）

语法

```
public boolean printQRCodeStandard(PrinterConnection connection, String content, int size, int
align, int eccLevel);
```

```
// 打印二维码（标准版）
//
//
// connection
// 由 connectXxx() 返回的打印机连接句柄。
//
// content
// 二维码 / 文本内容。
//
// size
// 模块大小。
//
// align
// 对齐方式。
//
// eccLevel
// 二维码纠错等级（QR_ECC_LEVEL_*）。
//
// 返回值
// 指令写入成功返回 true，否则返回 false。
```

printDoubleQRCode

打印双二维码

语法

```
public boolean printDoubleQRCode(PrinterConnection connection, int n, int qr1Pos, String qr1Data, int qr1Ecc, int qr1Ver, int qr2Pos, String qr2Data, int qr2Ecc, int qr2Ver);
```

```
// 打印双二维码
//
//
// connection
// 由 connectXxx() 返回的打印机连接句柄。
//
// n
// 数值参数（编码/点数/份数等，视 API 而定）。
//
// qr1Pos
// 左侧二维码水平位置。
//
// qr1Data
// 左侧二维码数据。
//
// qr1Ecc
// 左侧纠错等级。
//
// qr1Ver
// 左侧版本。
//
// qr2Pos
// 右侧二维码水平位置。
//
// qr2Data
// 右侧二维码数据。
//
// qr2Ecc
// 右侧纠错等级。
//
// qr2Ver
// 右侧版本。
//
// 返回值
// 指令写入成功返回 true，否则返回 false。
```

printPDF417

打印PDF417码

语法

```
public boolean printPDF417(PrinterConnection connection, int columns, int rows, int moduleWidth, int rowHeight, int errorCorrectionLevel, int option, String data);
```

```
// 打印PDF417码
//
//
// connection
// 由 connectXxx() 返回的打印机连接句柄。
//
// columns
// PDF417 列数。
//
// rows
// PDF417 行数。
//
// moduleWidth
// 条码模块宽度 2-6。
//
// rowHeight
// PDF417 行高。
//
// errorCorrectionLevel
// 纠错等级。
//
// option
// PDF417 选项。
//
// data
// 数据内容（字符串或字节）。
//
// 返回值
// 指令写入成功返回 true，否则返回 false。
```

printRasterImageFromFile

打印图片

语法

```
public boolean printRasterImageFromFile(PrinterConnection connection, String filePath);
```

```
// 打印图片
//
//
// connection
// 由 connectXxx() 返回的打印机连接句柄。
//
// filePath
// 图片文件路径。
//
// 返回值
// 指令写入成功返回 true，否则返回 false。
```

printRasterImageFromFile

打印图片（自定义尺寸）

语法

```
public boolean printRasterImageFromFile(PrinterConnection connection, String filePath, int targetWidth, int targetHeight);
```

```
// 打印图片（自定义尺寸）
//
//
// connection
// 由 connectXxx() 返回的打印机连接句柄。
//
// filePath
// 图片文件路径。
//
// targetWidth
// 目标打印宽度（点）。
//
// targetHeight
// 目标打印高度（点）。
//
// 返回值
// 指令写入成功返回 true，否则返回 false。
```

printRasterImageFromFile

打印图片（自定义尺寸与压缩模式）

语法

```
public boolean printRasterImageFromFile(PrinterConnection connection, String filePath, int targetWidth, int targetHeight, int compressionMode);
```

```
// 打印图片（自定义尺寸与压缩模式）
//
//
// connection
// 由 connectXxx() 返回的打印机连接句柄。
//
// filePath
// 图片文件路径。
//
// targetWidth
// 目标打印宽度（点）。
//
// targetHeight
// 目标打印高度（点）。
//
// compressionMode
// RASTER_COMPRESSION_* 常量。
//
// 返回值
// 指令写入成功返回 true，否则返回 false。
```

printRasterImageFromData

打印图片（字节数据）

语法

```
public boolean printRasterImageFromData(PrinterConnection connection, byte[] imageData);
```

```
// 打印图片（字节数据）
//
//
// connection
// 由 connectXxx() 返回的打印机连接句柄。
//
// imageData
// 编码后的图片字节。
//
// 返回值
// 指令写入成功返回 true，否则返回 false。
```

printRasterImageFromData

打印图片（字节数据，自定义尺寸）

语法

```
public boolean printRasterImageFromData(PrinterConnection connection, byte[] imageData, int targetWidth, int targetHeight);
```

```
// 打印图片（字节数据，自定义尺寸）
//
//
// connection
// 由 connectXxx() 返回的打印机连接句柄。
//
// imageData
// 编码后的图片字节。
//
// targetWidth
// 目标打印宽度（点）。
//
// targetHeight
// 目标打印高度（点）。
//
// 返回值
// 指令写入成功返回 true，否则返回 false。
```

printRasterImageFromData

打印图片（字节数据，自定义尺寸与压缩模式）

语法

```
public boolean printRasterImageFromData(PrinterConnection connection, byte[] imageData, int targetWidth, int targetHeight, int compressionMode);
```

```
// 打印图片（字节数据，自定义尺寸与压缩模式）
//
//
// connection
// 由 connectXxx() 返回的打印机连接句柄。
//
// imageData
// 编码后的图片字节。
//
// targetWidth
// 目标打印宽度（点）。
//
// targetHeight
// 目标打印高度（点）。
//
// compressionMode
// RASTER_COMPRESSION_* 常量。
//
// 返回值
// 指令写入成功返回 true，否则返回 false。
```

printRasterImageFromPixels

打印图片（像素数据）

语法

```
public boolean printRasterImageFromPixels(PrinterConnection connection, int[] pixels, int width, int height);
```

```
// 打印图片（像素数据）
//
//
// connection
// 由 connectXxx() 返回的打印机连接句柄。
//
// pixels
// ARGB 像素数组。
//
// width
// 宽度（点）/ 模块宽度。
//
// height
// 高度（点）。
//
// 返回值
// 指令写入成功返回 true，否则返回 false。
```

printRasterImageFromPixels

打印图片（像素数据，自定义尺寸）

语法

```
public boolean printRasterImageFromPixels(PrinterConnection connection, int[] pixels, int width, int height, int targetWidth, int targetHeight);
```

```
// 打印图片（像素数据，自定义尺寸）
//
//
// connection
// 由 connectXxx() 返回的打印机连接句柄。
//
// pixels
// ARGB 像素数组。
//
// width
// 宽度（点）/ 模块宽度。
//
// height
// 高度（点）。
//
// targetWidth
// 目标打印宽度（点）。
//
// targetHeight
// 目标打印高度（点）。
//
// 返回值
// 指令写入成功返回 true，否则返回 false。
```

printRasterImageFromPixels

打印图片（像素数据，自定义尺寸与压缩模式）

语法

```
public boolean printRasterImageFromPixels(PrinterConnection connection, int[] pixels, int width, int height, int targetWidth, int targetHeight, int compressionMode);
```

```
// 打印图片（像素数据，自定义尺寸与压缩模式）
//
//
// connection
// 由 connectXxx() 返回的打印机连接句柄。
//
// pixels
// ARGB 像素数组。
//
// width
// 宽度（点）/ 模块宽度。
//
// height
// 高度（点）。
//
// targetWidth
// 目标打印宽度（点）。
//
// targetHeight
// 目标打印高度（点）。
//
// compressionMode
// RASTER_COMPRESSION_* 常量。
//
// 返回值
// 指令写入成功返回 true，否则返回 false。
```

cutPaper

切纸

语法

```
public boolean cutPaper(PrinterConnection connection, int cutType);
```

```
// 切纸
//
//
// connection
// 由 connectXxx() 返回的打印机连接句柄。
//
// cutType
// CUT_FULL 或 CUT_PARTIAL。
//
// 返回值
// 指令写入成功返回 true，否则返回 false。
```

feedAndCutPaper

进纸并切纸

语法

```
public boolean feedAndCutPaper(PrinterConnection connection);
```

```
// 进纸并切纸
//
//
// connection
// 由 connectXxx() 返回的打印机连接句柄。
//
// 返回值
// 指令写入成功返回 true，否则返回 false。
```

feedLine

换行

语法

```
public boolean feedLine(PrinterConnection connection);
```

```
// 换行
//
//
// connection
// 由 connectXxx() 返回的打印机连接句柄。
//
// 返回值
// 指令写入成功返回 true，否则返回 false。
```

feedPaper

进纸

语法

```
public boolean feedPaper(PrinterConnection connection, int lines);
```

```
// 进纸
//
//
// connection
// 由 connectXxx() 返回的打印机连接句柄。
//
// lines
// 行数 / 走纸量。
//
// 返回值
// 指令写入成功返回 true，否则返回 false。
```

feedPaper

进纸

语法

```
public boolean feedPaper(PrinterConnection connection);
```

```
// 进纸
//
//
// connection
// 由 connectXxx() 返回的打印机连接句柄。
//
// 返回值
// 指令写入成功返回 true，否则返回 false。
```

reverseFeed

反向走纸（ESC e）

语法

```
public boolean reverseFeed(PrinterConnection connection, int lines);
```

```
// 反向走纸（ESC e）
//
//
// connection
// 由 connectXxx() 返回的打印机连接句柄。
//
// lines
// 行数 / 走纸量。
//
// 返回值
// 指令写入成功返回 true，否则返回 false。
```

horizontalTab

水平制表（HT）

语法

```
public boolean horizontalTab(PrinterConnection connection);
```

```
// 水平制表（HT）
//
//
// connection
// 由 connectXxx() 返回的打印机连接句柄。
//
// 返回值
// 指令写入成功返回 true，否则返回 false。
```

setTabPositions

设置水平制表位（ESC D）

语法

```
public boolean setTabPositions(PrinterConnection connection, int[] positions);
```

```
// 设置水平制表位（ESC D）
//
//
// connection
// 由 connectXxx() 返回的打印机连接句柄。
//
// positions
// 水平制表位。
//
// 返回值
// 指令写入成功返回 true，否则返回 false。
```

openCashDrawer

打开钱箱

语法

```
public boolean openCashDrawer(PrinterConnection connection, int drawerIndex);
```

```
// 打开钱箱
//
//
// connection
// 由 connectXxx() 返回的打印机连接句柄。
//
// drawerIndex
// CASH_DRAWER_PIN2 或 PIN5。
//
// 返回值
// 指令写入成功返回 true，否则返回 false。
```

beep

蜂鸣器

语法

```
public boolean beep(PrinterConnection connection, int count, int duration);
```

```
// 蜂鸣器
//
//
// connection
// 由 connectXxx() 返回的打印机连接句柄。
//
// count
// 蜂鸣次数。
//
// duration
// 蜂鸣时长参数。
//
// 返回值
// 指令写入成功返回 true，否则返回 false。
```

beep

蜂鸣器（默认）

语法

```
public boolean beep(PrinterConnection connection);
```

```
// 蜂鸣器（默认）
//
//
// connection
// 由 connectXxx() 返回的打印机连接句柄。
//
// 返回值
// 指令写入成功返回 true，否则返回 false。
```

setPrintMode

设置打印模式

语法

```
public boolean setPrintMode(PrinterConnection connection, int mode);
```

```
// 设置打印模式
//
//
// connection
// 由 connectXxx() 返回的打印机连接句柄。
//
// mode
// 模式标志 / NV 位图模式。
//
// 返回值
// 指令写入成功返回 true，否则返回 false。
```

setSingleByteMode

设置单字节模式

语法

```
public boolean setSingleByteMode(PrinterConnection connection);
```

```
// 设置单字节模式
//
//
// connection
// 由 connectXxx() 返回的打印机连接句柄。
//
// 返回值
// 指令写入成功返回 true，否则返回 false。
```

setCharacterSet

设置国际字符集

语法

```
public boolean setCharacterSet(PrinterConnection connection, int charset);
```

```
// 设置国际字符集
//
//
// connection
// 由 connectXxx() 返回的打印机连接句柄。
//
// charset
// 国际字符集编号。
//
// 返回值
// 指令写入成功返回 true，否则返回 false。
```

setCharacterCodepage

设置字符代码页

语法

```
public boolean setCharacterCodepage(PrinterConnection connection, int codepage);
```

```
// 设置字符代码页
//
//
// connection
// 由 connectXxx() 返回的打印机连接句柄。
//
// codepage
// 代码页编号。
//
// 返回值
// 指令写入成功返回 true，否则返回 false。
```

setChineseMode

设置汉字模式

语法

```
public boolean setChineseMode(PrinterConnection connection);
```

```
// 设置汉字模式
//
//
// connection
// 由 connectXxx() 返回的打印机连接句柄。
//
// 返回值
// 指令写入成功返回 true，否则返回 false。
```

setChineseDoubleSize

选择/取消汉字倍高倍宽（FS W n，0x1C 0x57） 专用于汉字，与 setTextScale（GS ! n）指令不同

语法

```
public boolean setChineseDoubleSize(PrinterConnection connection, boolean enable);
```

```
// 选择/取消汉字倍高倍宽（FS W n，0x1C 0x57） 专用于汉字，与 setTextScale（GS ! n）指令不同
//
//
// connection
// 由 connectXxx() 返回的打印机连接句柄。
//
// enable
// 开关。
//
// 返回值
// 指令写入成功返回 true，否则返回 false。
```

setChineseDoubleSize

选择/取消汉字倍高倍宽（FS W n）

语法

```
public boolean setChineseDoubleSize(PrinterConnection connection, int n);
```

```
// 选择/取消汉字倍高倍宽（FS W n）
//
//
// connection
// 由 connectXxx() 返回的打印机连接句柄。
//
// n
// 数值参数（编码/点数/份数等，视 API 而定）。
//
// 返回值
// 指令写入成功返回 true，否则返回 false。
```

setDoubleByteEncoding

设置双字节编码

语法

```
public boolean setDoubleByteEncoding(PrinterConnection connection, int n);
```

```
// 设置双字节编码
//
//
// connection
// 由 connectXxx() 返回的打印机连接句柄。
//
// n
// 数值参数（编码/点数/份数等，视 API 而定）。
//
// 返回值
// 指令写入成功返回 true，否则返回 false。
```

setTextScale

设置字符大小（GS ! n）

语法

```
public boolean setTextScale(PrinterConnection connection, int widthScale, int heightScale);
```

```
// 设置字符大小（GS ! n）
//
//
// connection
// 由 connectXxx() 返回的打印机连接句柄。
//
// widthScale
// 文字倍宽 0–7（0=1 倍）。
//
// heightScale
// 文字倍高 0–7（0=1 倍）。
//
// 返回值
// 指令写入成功返回 true，否则返回 false。
```

setAsciiTextFontType

设置ASCII文本字体类型

语法

```
public boolean setAsciiTextFontType(PrinterConnection connection, int fontType);
```

```
// 设置ASCII文本字体类型
//
//
// connection
// 由 connectXxx() 返回的打印机连接句柄。
//
// fontType
// ASCII / 条码字体类型。
//
// 返回值
// 指令写入成功返回 true，否则返回 false。
```

setTextUpsideDown

设置文本倒置

语法

```
public boolean setTextUpsideDown(PrinterConnection connection, boolean upsideDown);
```

```
// 设置文本倒置
//
//
// connection
// 由 connectXxx() 返回的打印机连接句柄。
//
// upsideDown
// true 表示倒置。
//
// 返回值
// 指令写入成功返回 true，否则返回 false。
```

setTextRotate

设置文本旋转90度

语法

```
public boolean setTextRotate(PrinterConnection connection, boolean rotate);
```

```
// 设置文本旋转90度
//
//
// connection
// 由 connectXxx() 返回的打印机连接句柄。
//
// rotate
// 旋转标志 / 标签旋转 0-3。
//
// 返回值
// 指令写入成功返回 true，否则返回 false。
```

setTextLineHeight

设置文本行高

语法

```
public boolean setTextLineHeight(PrinterConnection connection, int lineHeight);
```

```
// 设置文本行高
//
//
// connection
// 由 connectXxx() 返回的打印机连接句柄。
//
// lineHeight
// 行高（点）。
//
// 返回值
// 指令写入成功返回 true，否则返回 false。
```

setAsciiTextCharRightSpacing

设置ASCII字符右间距

语法

```
public boolean setAsciiTextCharRightSpacing(PrinterConnection connection, int spacing);
```

```
// 设置ASCII字符右间距
//
//
// connection
// 由 connectXxx() 返回的打印机连接句柄。
//
// spacing
// 间距（点）。
//
// 返回值
// 指令写入成功返回 true，否则返回 false。
```

setKanjiTextCharSpacing

设置汉字字符左右间距

语法

```
public boolean setKanjiTextCharSpacing(PrinterConnection connection, int leftSpacing, int rightSpacing);
```

```
// 设置汉字字符左右间距
//
//
// connection
// 由 connectXxx() 返回的打印机连接句柄。
//
// leftSpacing
// 左字间距。
//
// rightSpacing
// 右字间距。
//
// 返回值
// 指令写入成功返回 true，否则返回 false。
```

setPrintPosition

设置打印起始位置

语法

```
public boolean setPrintPosition(PrinterConnection connection, int position);
```

```
// 设置打印起始位置
//
//
// connection
// 由 connectXxx() 返回的打印机连接句柄。
//
// position
// 绝对打印位置 / HRI 位置。
//
// 返回值
// 指令写入成功返回 true，否则返回 false。
```

setPrintAreaLeftMargin

设置打印区域左边空白

语法

```
public boolean setPrintAreaLeftMargin(PrinterConnection connection, int leftMargin);
```

```
// 设置打印区域左边空白
//
//
// connection
// 由 connectXxx() 返回的打印机连接句柄。
//
// leftMargin
// 左边距（点）。
//
// 返回值
// 指令写入成功返回 true，否则返回 false。
```

setPrintAreaWidth

设置打印区域宽度

语法

```
public boolean setPrintAreaWidth(PrinterConnection connection, int width);
```

```
// 设置打印区域宽度
//
//
// connection
// 由 connectXxx() 返回的打印机连接句柄。
//
// width
// 宽度（点）/ 模块宽度。
//
// 返回值
// 指令写入成功返回 true，否则返回 false。
```

setHorizontalAbsolutePrintPosition

设置水平方向绝对打印位置

语法

```
public boolean setHorizontalAbsolutePrintPosition(PrinterConnection connection, int position);
```

```
// 设置水平方向绝对打印位置
//
//
// connection
// 由 connectXxx() 返回的打印机连接句柄。
//
// position
// 绝对打印位置 / HRI 位置。
//
// 返回值
// 指令写入成功返回 true，否则返回 false。
```

setHorizontalRelativePrintPosition

设置水平方向相对打印位置

语法

```
public boolean setHorizontalRelativePrintPosition(PrinterConnection connection, int offset);
```

```
// 设置水平方向相对打印位置
//
//
// connection
// 由 connectXxx() 返回的打印机连接句柄。
//
// offset
// 相对偏移（点）。
//
// 返回值
// 指令写入成功返回 true，否则返回 false。
```

setVerticalAbsolutePrintPosition

设置垂直方向绝对打印位置

语法

```
public boolean setVerticalAbsolutePrintPosition(PrinterConnection connection, int position);
```

```
// 设置垂直方向绝对打印位置
//
//
// connection
// 由 connectXxx() 返回的打印机连接句柄。
//
// position
// 绝对打印位置 / HRI 位置。
//
// 返回值
// 指令写入成功返回 true，否则返回 false。
```

setVerticalRelativePrintPosition

设置垂直方向相对打印位置

语法

```
public boolean setVerticalRelativePrintPosition(PrinterConnection connection, int offset);
```

```
// 设置垂直方向相对打印位置
//
//
// connection
// 由 connectXxx() 返回的打印机连接句柄。
//
// offset
// 相对偏移（点）。
//
// 返回值
// 指令写入成功返回 true，否则返回 false。
```

setBarcodeHRIPosition

设置条码HRI位置

语法

```
public boolean setBarcodeHRIPosition(PrinterConnection connection, int pos);
```

```
// 设置条码HRI位置
//
//
// connection
// 由 connectXxx() 返回的打印机连接句柄。
//
// pos
// HRI 位置。
//
// 返回值
// 指令写入成功返回 true，否则返回 false。
```

setBarcodeHeight

设置条码高度

语法

```
public boolean setBarcodeHeight(PrinterConnection connection, int height);
```

```
// 设置条码高度
//
//
// connection
// 由 connectXxx() 返回的打印机连接句柄。
//
// height
// 高度（点）。
//
// 返回值
// 指令写入成功返回 true，否则返回 false。
```

setBarcodeWidth

设置条码宽度

语法

```
public boolean setBarcodeWidth(PrinterConnection connection, int width);
```

```
// 设置条码宽度
//
//
// connection
// 由 connectXxx() 返回的打印机连接句柄。
//
// width
// 宽度（点）/ 模块宽度。
//
// 返回值
// 指令写入成功返回 true，否则返回 false。
```

setBarcodeReadableTextFontType

设置条码可识别字符的字体类型

语法

```
public boolean setBarcodeReadableTextFontType(PrinterConnection connection, int fontType);
```

```
// 设置条码可识别字符的字体类型
//
//
// connection
// 由 connectXxx() 返回的打印机连接句柄。
//
// fontType
// ASCII / 条码字体类型。
//
// 返回值
// 指令写入成功返回 true，否则返回 false。
```

setBarcodeReadableTextPosition

设置条码可识别字符的位置

语法

```
public boolean setBarcodeReadableTextPosition(PrinterConnection connection, int position);
```

```
// 设置条码可识别字符的位置
//
//
// connection
// 由 connectXxx() 返回的打印机连接句柄。
//
// position
// 绝对打印位置 / HRI 位置。
//
// 返回值
// 指令写入成功返回 true，否则返回 false。
```

printHorizontalLine

打印水平线

语法

```
public boolean printHorizontalLine(PrinterConnection connection, int start, int end);
```

```
// 打印水平线
//
//
// connection
// 由 connectXxx() 返回的打印机连接句柄。
//
// start
// 线起点。
//
// end
// 线终点。
//
// 返回值
// 指令写入成功返回 true，否则返回 false。
```

fixedHorizontalLine

固定水平线

语法

```
public boolean fixedHorizontalLine(PrinterConnection connection);
```

```
// 固定水平线
//
//
// connection
// 由 connectXxx() 返回的打印机连接句柄。
//
// 返回值
// 指令写入成功返回 true，否则返回 false。
```

printAndFeedDots

打印并进点

语法

```
public boolean printAndFeedDots(PrinterConnection connection, int n);
```

```
// 打印并进点
//
//
// connection
// 由 connectXxx() 返回的打印机连接句柄。
//
// n
// 数值参数（编码/点数/份数等，视 API 而定）。
//
// 返回值
// 指令写入成功返回 true，否则返回 false。
```

sendSelfTest

发送自检命令

语法

```
public boolean sendSelfTest(PrinterConnection connection);
```

```
// 发送自检命令
//
//
// connection
// 由 connectXxx() 返回的打印机连接句柄。
//
// 返回值
// 指令写入成功返回 true，否则返回 false。
```

sendCommand

发送自定义命令

语法

```
public boolean sendCommand(PrinterConnection connection, byte[] command);
```

```
// 发送自定义命令
//
//
// connection
// 由 connectXxx() 返回的打印机连接句柄。
//
// command
// 原始 ESC/POS 或 JPL 指令字节。
//
// 返回值
// 指令写入成功返回 true，否则返回 false。
```

clearBuffer

清除缓冲区

语法

```
public boolean clearBuffer(PrinterConnection connection);
```

```
// 清除缓冲区
//
//
// connection
// 由 connectXxx() 返回的打印机连接句柄。
//
// 返回值
// 指令写入成功返回 true，否则返回 false。
```

cancelPrintCmd

取消打印

语法

```
public boolean cancelPrintCmd(PrinterConnection connection);
```

```
// 取消打印
//
//
// connection
// 由 connectXxx() 返回的打印机连接句柄。
//
// 返回值
// 指令写入成功返回 true，否则返回 false。
```

页模式功能

selectPageMode

选择页模式

语法

```
public boolean selectPageMode(PrinterConnection connection);
```

```
// 选择页模式
//
//
// connection
// 由 connectXxx() 返回的打印机连接句柄。
//
// 返回值
// 指令写入成功返回 true，否则返回 false。
```

exitPageMode

退出页模式

语法

```
public boolean exitPageMode(PrinterConnection connection);
```

```
// 退出页模式
//
//
// connection
// 由 connectXxx() 返回的打印机连接句柄。
//
// 返回值
// 指令写入成功返回 true，否则返回 false。
```

selectPageModeEx

选择页模式（扩展版本）

语法

```
public boolean selectPageModeEx(PrinterConnection connection, int nHorizontalMovementUnit, int nVerticalMovementUnit, int x, int y, int width, int height);
```

```
// 选择页模式（扩展版本）
//
//
// connection
// 由 connectXxx() 返回的打印机连接句柄。
//
// nHorizontalMovementUnit
// 横向移动单位（GS P x）。
//
// nVerticalMovementUnit
// 纵向移动单位（GS P y）。
//
// x
// X 坐标（点）。
//
// y
// Y 坐标（点）。
//
// width
// 宽度（点）/ 模块宽度。
//
// height
// 高度（点）。
//
// 返回值
// 指令写入成功返回 true，否则返回 false。
```

selectPageModeEx

选择页模式（简化版本）

语法

```
public boolean selectPageModeEx(PrinterConnection connection, int x, int y, int width, int height);
```

```
// 选择页模式（简化版本）
//
//
// connection
// 由 connectXxx() 返回的打印机连接句柄。
//
// x
// X 坐标（点）。
//
// y
// Y 坐标（点）。
//
// width
// 宽度（点）/ 模块宽度。
//
// height
// 高度（点）。
//
// 返回值
// 指令写入成功返回 true，否则返回 false。
```

selectPageModeEx

选择页模式（默认参数）

语法

```
public boolean selectPageModeEx(PrinterConnection connection);
```

```
// 选择页模式（默认参数）
//
//
// connection
// 由 connectXxx() 返回的打印机连接句柄。
//
// 返回值
// 指令写入成功返回 true，否则返回 false。
```

pagePrintPage

页模式打印

语法

```
public boolean pagePrintPage(PrinterConnection connection);
```

```
// 页模式打印
//
//
// connection
// 由 connectXxx() 返回的打印机连接句柄。
//
// 返回值
// 指令写入成功返回 true，否则返回 false。
```

pageClearPage

清除页模式数据

语法

```
public boolean pageClearPage(PrinterConnection connection);
```

```
// 清除页模式数据
//
//
// connection
// 由 connectXxx() 返回的打印机连接句柄。
//
// 返回值
// 指令写入成功返回 true，否则返回 false。
```

setMovementUnits

设置移动单位

语法

```
public boolean setMovementUnits(PrinterConnection connection, int x, int y);
```

```
// 设置移动单位
//
//
// connection
// 由 connectXxx() 返回的打印机连接句柄。
//
// x
// X 坐标（点）。
//
// y
// Y 坐标（点）。
//
// 返回值
// 指令写入成功返回 true，否则返回 false。
```

pageSetPageArea

设置页模式打印区域

语法

```
public boolean pageSetPageArea(PrinterConnection connection, int x, int y, int width, int height);
```

```
// 设置页模式打印区域
//
//
// connection
// 由 connectXxx() 返回的打印机连接句柄。
//
// x
// X 坐标（点）。
//
// y
// Y 坐标（点）。
//
// width
// 宽度（点）/ 模块宽度。
//
// height
// 高度（点）。
//
// 返回值
// 指令写入成功返回 true，否则返回 false。
```

pageSetPageDrawDirection

设置页模式打印方向

语法

```
public boolean pageSetPageDrawDirection(PrinterConnection connection, int direction);
```

```
// 设置页模式打印方向
//
//
// connection
// 由 connectXxx() 返回的打印机连接句柄。
//
// direction
// PAGE_DIRECTION_*。
//
// 返回值
// 指令写入成功返回 true，否则返回 false。
```

setPrintAreaInPageMode

设置页模式打印区域

语法

```
public boolean setPrintAreaInPageMode(PrinterConnection connection, int x, int y, int width,
int height);
```

```
// 设置页模式打印区域
//
//
// connection
// 由 connectXxx() 返回的打印机连接句柄。
//
// x
// X 坐标（点）。
//
// y
// Y 坐标（点）。
//
// width
// 宽度（点）/ 模块宽度。
//
// height
// 高度（点）。
//
// 返回值
// 指令写入成功返回 true，否则返回 false。
```

setPrintAreaDirection

设置页模式打印方向

语法

```
public boolean setPrintAreaDirection(PrinterConnection connection, int direction);
```

```
// 设置页模式打印方向
//
//
// connection
// 由 connectXxx() 返回的打印机连接句柄。
//
// direction
// PAGE_DIRECTION_*。
//
// 返回值
// 指令写入成功返回 true，否则返回 false。
```

drawLineInPageMode

在页模式下画线（默认实线）

语法

```
public boolean drawLineInPageMode(PrinterConnection connection, int x1, int y1, int x2, int y2);
```

```
// 在页模式下画线（默认实线）
//
//
// connection
// 由 connectXxx() 返回的打印机连接句柄。
//
// x1
// 线段起点 X。
//
// y1
// 线段起点 Y。
//
// x2
// 线段终点 X。
//
// y2
// 线段终点 Y。
//
// 返回值
// 指令写入成功返回 true，否则返回 false。
```

drawLineInPageMode

在页模式下画线

语法

```
public boolean drawLineInPageMode(PrinterConnection connection, int x1, int y1, int x2, int y2, int lineType);
```

```
// 在页模式下画线
//
//
// connection
// 由 connectXxx() 返回的打印机连接句柄。
//
// x1
// 线段起点 X。
//
// y1
// 线段起点 Y。
//
// x2
// 线段终点 X。
//
// y2
// 线段终点 Y。
//
// lineType
// 1=实线，2=虚线，3=点线。
//
// 返回值
// 指令写入成功返回 true，否则返回 false。
```

estimateCode128WidthDots

估算 Code128 条码宽度（点），用于页模式左右排版

语法

```
public int estimateCode128WidthDots(String data, int moduleWidth);
```

```
// 估算 Code128 条码宽度（点），用于页模式左右排版
//
//
// data
// 数据内容（字符串或字节）。
//
// moduleWidth
// 条码模块宽度 2-6。
//
// 返回值
// 整型结果；状态类 API 失败时可能返回 STATUS_OFFLINE。
```

drawRectangleInPageMode

在页模式下绘制矩形

语法

```
public boolean drawRectangleInPageMode(PrinterConnection connection, int mode, int x, int y,
int width, int height);
```

```
// 在页模式下绘制矩形
//
//
// connection
// 由 connectXxx() 返回的打印机连接句柄。
//
// mode
// 模式标志 / NV 位图模式。
//
// x
// X 坐标（点）。
//
// y
// Y 坐标（点）。
//
// width
// 宽度（点）/ 模块宽度。
//
// height
// 高度（点）。
//
// 返回值
// 指令写入成功返回 true，否则返回 false。
```

drawRectangleBorderInPageMode

在页模式下绘制矩形边框

语法

```
public boolean drawRectangleBorderInPageMode(PrinterConnection connection, int x, int y, int
width, int height);
```

```
// 在页模式下绘制矩形边框
//
//
// connection
// 由 connectXxx() 返回的打印机连接句柄。
//
// x
// X 坐标（点）。
//
// y
// Y 坐标（点）。
//
// width
// 宽度（点）/ 模块宽度。
//
// height
// 高度（点）。
//
// 返回值
// 指令写入成功返回 true，否则返回 false。
```

drawFilledRectangleInPageMode

在页模式下绘制实心矩形

语法

```
public boolean drawFilledRectangleInPageMode(PrinterConnection connection, int x, int y, int width, int height);
```

```
// 在页模式下绘制实心矩形
//
//
// connection
// 由 connectXxx() 返回的打印机连接句柄。
//
// x
// X 坐标（点）。
//
// y
// Y 坐标（点）。
//
// width
// 宽度（点）/ 模块宽度。
//
// height
// 高度（点）。
//
// 返回值
// 指令写入成功返回 true，否则返回 false。
```

drawTextInPageMode

在页模式下绘制文本

语法

```
public boolean drawTextInPageMode(PrinterConnection connection, int x, int y, String text);
```

```
// 在页模式下绘制文本
//
//
// connection
// 由 connectXxx() 返回的打印机连接句柄。
//
// x
// X 坐标（点）。
//
// y
// Y 坐标（点）。
//
// text
// 要打印的文本。
//
// 返回值
// 指令写入成功返回 true，否则返回 false。
```

estimateTextWidthDots

估算页模式文本宽度（点）

语法

```
public int estimateTextWidthDots(String text, int charWidthDots);
```

```
// 估算页模式文本宽度（点）
//
//
// text
// 要打印的文本。
//
// charWidthDots
// 字符宽度估算（Font A 约 12 点）。
//
// 返回值
// 整型结果；状态类 API 失败时可能返回 STATUS_OFFLINE。
```

drawTextCenteredInPageMode

页模式居中绘制文本（ASCII/ISO-8859-1）

语法

```
public boolean drawTextCenteredInPageMode(PrinterConnection connection, int pageWidth, int y,
String text);
```

```
// 页模式居中绘制文本（ASCII/ISO-8859-1）
//
//
// connection
// 由 connectXxx() 返回的打印机连接句柄。
//
// pageWidth
// 页宽 / 纸宽（点）。
//
// y
// Y 坐标（点）。
//
// text
// 要打印的文本。
//
// 返回值
// 指令写入成功返回 true，否则返回 false。
```

drawTextCenteredInPageMode

drawTextCenteredInPageMode

语法

```
public boolean drawTextCenteredInPageMode(PrinterConnection connection, int pageWidth, int y,
String text, int charWidthDots);
```

```
// drawTextCenteredInPageMode
//
//
// connection
// 由 connectXxx() 返回的打印机连接句柄。
//
// pageWidth
// 页宽 / 纸宽（点）。
//
// y
// Y 坐标（点）。
//
// text
// 要打印的文本。
//
// charWidthDots
// 字符宽度估算（Font A 约 12 点）。
//
// 返回值
// 指令写入成功返回 true，否则返回 false。
```

drawTextRightInPageMode

页模式右对齐绘制文本

语法

```
public boolean drawTextRightInPageMode(PrinterConnection connection, int pageWidth, int y,
String text);
```

```
// 页模式右对齐绘制文本
//
//
// connection
// 由 connectXxx() 返回的打印机连接句柄。
//
// pageWidth
// 页宽 / 纸宽（点）。
//
// y
// Y 坐标（点）。
//
// text
// 要打印的文本。
//
// 返回值
// 指令写入成功返回 true，否则返回 false。
```

drawTextRightInPageMode

drawTextRightInPageMode

语法

```
public boolean drawTextRightInPageMode(PrinterConnection connection, int pageWidth, int y,
String text, int charWidthDots);
```

```
// drawTextRightInPageMode
//
//
// connection
// 由 connectXxx() 返回的打印机连接句柄。
//
// pageWidth
// 页宽 / 纸宽（点）。
//
// y
// Y 坐标（点）。
//
// text
// 要打印的文本。
//
// charWidthDots
// 字符宽度估算（Font A 约 12 点）。
//
// 返回值
// 指令写入成功返回 true，否则返回 false。
```

drawTextLineLeftRightInPageMode

页模式一行左右分列（左 x=0，右贴边）

语法

```
public boolean drawTextLineLeftRightInPageMode(PrinterConnection connection, int pageWidth, int
y, String left, String right);
```

```
// 页模式一行左右分列（左 x=0，右贴边）
//
//
// connection
// 由 connectXxx() 返回的打印机连接句柄。
//
// pageWidth
// 页宽 / 纸宽（点）。
//
// y
// Y 坐标（点）。
//
// left
// 左侧文本。
//
// right
// 右侧文本。
//
// 返回值
// 指令写入成功返回 true，否则返回 false。
```

drawTextLineLeftRightInPageMode

drawTextLineLeftRightInPageMode

语法

```
public boolean drawTextLineLeftRightInPageMode(PrinterConnection connection, int pageWidth, int y, String left, String right, int charWidthDots);
```

```
// drawTextLineLeftRightInPageMode
//
//
// connection
// 由 connectXxx() 返回的打印机连接句柄。
//
// pageWidth
// 页宽 / 纸宽（点）。
//
// y
// Y 坐标（点）。
//
// left
// 左侧文本。
//
// right
// 右侧文本。
//
// charWidthDots
// 字符宽度估算（Font A 约 12 点）。
//
// 返回值
// 指令写入成功返回 true，否则返回 false。
```

printNvBitmap

标准模式：打印 NV 位图（打印机内存 Logo）

语法

```
public boolean printNvBitmap(PrinterConnection connection, int nvNumber, int mode);
```

```
// 标准模式：打印 NV 位图（打印机内存 Logo）
//
//
// connection
// 由 connectXxx() 返回的打印机连接句柄。
//
// nvNumber
// NV 位图槽位号（FS p）。
//
// mode
// 模式标志 / NV 位图模式。
//
// 返回值
// 指令写入成功返回 true，否则返回 false。
```

drawNvBitmapInPageMode

页模式：在指定坐标打印 NV 位图

语法

```
public boolean drawNvBitmapInPageMode(PrinterConnection connection, int x, int y, int nvNumber, int mode);
```

```
// 页模式：在指定坐标打印 NV 位图
//
//
// connection
// 由 connectXxx() 返回的打印机连接句柄。
//
// x
// X 坐标（点）。
//
// y
// Y 坐标（点）。
//
// nvNumber
// NV 位图槽位号（FS p）。
//
// mode
// 模式标志 / NV 位图模式。
//
// 返回值
// 指令写入成功返回 true，否则返回 false。
```

drawNvBitmapCenteredInPageMode

页模式：水平居中打印 NV 位图（需已知位图宽度用于算 x）

语法

```
public boolean drawNvBitmapCenteredInPageMode(PrinterConnection connection, int pageWidth, int y, int nvNumber, int mode, int imageWidthDots);
```

```
// 页模式：水平居中打印 NV 位图（需已知位图宽度用于算 x）
//
//
// connection
// 由 connectXxx() 返回的打印机连接句柄。
//
// pageWidth
// 页宽 / 纸宽（点）。
//
// y
// Y 坐标（点）。
//
// nvNumber
// NV 位图槽位号（FS p）。
//
// mode
// 模式标志 / NV 位图模式。
//
// imageWidthDots
// 详见方法说明。
//
// 返回值
// 指令写入成功返回 true，否则返回 false。
```

drawTextUTF8InPageMode

在页模式下绘制文本（UTF8）

语法

```
public boolean drawTextUTF8InPageMode(PrinterConnection connection, int x, int y, String text);
```

```
// 在页模式下绘制文本（UTF8）
//
//
// connection
// 由 connectXxx() 返回的打印机连接句柄。
//
// x
// X 坐标（点）。
//
// y
// Y 坐标（点）。
//
// text
// 要打印的文本。
//
// 返回值
// 指令写入成功返回 true，否则返回 false。
```

drawTextGBKInPageMode

在页模式下绘制文本（GBK）

语法

```
public boolean drawTextGBKInPageMode(PrinterConnection connection, int x, int y, String text);
```

```
// 在页模式下绘制文本（GBK）
//
//
// connection
// 由 connectXxx() 返回的打印机连接句柄。
//
// x
// X 坐标（点）。
//
// y
// Y 坐标（点）。
//
// text
// 要打印的文本。
//
// 返回值
// 指令写入成功返回 true，否则返回 false。
```

drawTextBIG5InPageMode

在页模式下绘制文本（BIG5）

语法

```
public boolean drawTextBIG5InPageMode(PrinterConnection connection, int x, int y, String text);
```

```
// 在页模式下绘制文本（BIG5）
//
//
// connection
// 由 connectXxx() 返回的打印机连接句柄。
//
// x
// X 坐标（点）。
//
// y
// Y 坐标（点）。
//
// text
// 要打印的文本。
//
// 返回值
// 指令写入成功返回 true，否则返回 false。
```

drawTextShiftJISInPageMode

在页模式下绘制文本（ShiftJIS）

语法

```
public boolean drawTextShiftJISInPageMode(PrinterConnection connection, int x, int y, String text);
```

```
// 在页模式下绘制文本（ShiftJIS）
//
//
// connection
// 由 connectXxx() 返回的打印机连接句柄。
//
// x
// X 坐标（点）。
//
// y
// Y 坐标（点）。
//
// text
// 要打印的文本。
//
// 返回值
// 指令写入成功返回 true，否则返回 false。
```

drawTextEUCKRInPageMode

在页模式下绘制文本（EUCKR）

语法

```
public boolean drawTextEUCKRInPageMode(PrinterConnection connection, int x, int y, String text);
```

```
// 在页模式下绘制文本（EUCKR）
//
//
// connection
// 由 connectXxx() 返回的打印机连接句柄。
//
// x
// X 坐标（点）。
//
// y
// Y 坐标（点）。
//
// text
// 要打印的文本。
//
// 返回值
// 指令写入成功返回 true，否则返回 false。
```

drawBarcodeInPageMode

在页模式下绘制条码

语法

```
public boolean drawBarcodeInPageMode(PrinterConnection connection, int x, int y, int barcodeType, String data);
```

```
// 在页模式下绘制条码
//
//
// connection
// 由 connectXxx() 返回的打印机连接句柄。
//
// x
// X 坐标（点）。
//
// y
// Y 坐标（点）。
//
// barcodeType
// ESC/POS BARCODE_TYPE_* 或 LABEL_BARCODE_TYPE_*。
//
// data
// 数据内容（字符串或字节）。
//
// 返回值
// 指令写入成功返回 true，否则返回 false。
```

drawBarcodeInPageMode

在页模式下绘制条码（可指定 HRI 位置、高度、模块宽度）

语法

```
public boolean drawBarcodeInPageMode(PrinterConnection connection, int x, int y, int barcodeType, String data, int hriPos, int height, int width);
```

```
// 在页模式下绘制条码（可指定 HRI 位置、高度、模块宽度）
//
//
// connection
// 由 connectXxx() 返回的打印机连接句柄。
//
// x
// X 坐标（点）。
//
// y
// Y 坐标（点）。
//
// barcodeType
// ESC/POS BARCODE_TYPE_* 或 LABEL_BARCODE_TYPE_*。
//
// data
// 数据内容（字符串或字节）。
//
// hriPos
// 详见方法说明。
//
// height
// 高度（点）。
//
// width
// 宽度（点）/ 模块宽度。
//
// 返回值
// 指令写入成功返回 true，否则返回 false。
```

drawBarcodeInPageModeWithCenteredHri

页模式条码 + 居中可读数字（单次写入，数字紧贴条码下方）

语法

```
public boolean drawBarcodeInPageModeWithCenteredHri(PrinterConnection connection, int x, int y, int barcodeType, String data, int height, int width, int hriGapDots);
```

```
// 页模式条码 + 居中可读数字（单次写入，数字紧贴条码下方）
//
//
// connection
// 由 connectXxx() 返回的打印机连接句柄。
//
// x
// X 坐标（点）。
//
// y
// Y 坐标（点）。
//
// barcodeType
// ESC/POS BARCODE_TYPE_* 或 LABEL_BARCODE_TYPE_*。
//
// data
// 数据内容（字符串或字节）。
//
// height
// 高度（点）。
//
// width
// 宽度（点）/ 模块宽度。
//
// hriGapDots
// 条码底边与 HRI 数字间距（可为负）。
//
// 返回值
// 指令写入成功返回 true，否则返回 false。
```

drawQRCodeInPageMode

在页模式下绘制二维码

语法

```
public boolean drawQRCodeInPageMode(PrinterConnection connection, int x, int y, int version, int eccLevel, String data);
```

```
// 在页模式下绘制二维码
//
//
// connection
// 由 connectXxx() 返回的打印机连接句柄。
//
// x
// X 坐标（点）。
//
// y
// Y 坐标（点）。
//
// version
// 二维码版本 / 模块提示。
//
// eccLevel
// 二维码纠错等级（QR_ECC_LEVEL_*）。
//
// data
// 数据内容（字符串或字节）。
//
// 返回值
// 指令写入成功返回 true，否则返回 false。
```

drawImageFromFileInPageMode

在页模式下绘制图片（从文件）

语法

```
public boolean drawImageFromFileInPageMode(PrinterConnection connection, int x, int y, String filePath, int width, int height);
```

```
// 在页模式下绘制图片（从文件）
//
//
// connection
// 由 connectXxx() 返回的打印机连接句柄。
//
// x
// X 坐标（点）。
//
// y
// Y 坐标（点）。
//
// filePath
// 图片文件路径。
//
// width
// 宽度（点）/ 模块宽度。
//
// height
// 高度（点）。
//
// 返回值
// 指令写入成功返回 true，否则返回 false。
```

drawImageFromDataInPageMode

在页模式下绘制图片（从数据）

语法

```
public boolean drawImageFromDataInPageMode(PrinterConnection connection, int x, int y, byte[] imageData, int width, int height);
```

```
// 在页模式下绘制图片（从数据）
//
//
// connection
// 由 connectXxx() 返回的打印机连接句柄。
//
// x
// X 坐标（点）。
//
// y
// Y 坐标（点）。
//
// imageData
// 编码后的图片字节。
//
// width
// 宽度（点）/ 模块宽度。
//
// height
// 高度（点）。
//
// 返回值
// 指令写入成功返回 true，否则返回 false。
```

drawImageFromPixelsInPageMode

在页模式下绘制图片（从像素）

语法

```
public boolean drawImageFromPixelsInPageMode(PrinterConnection connection, int x, int y, int[] pixels, int imgWidth, int imgHeight, int width, int height);
```

```
// 在页模式下绘制图片（从像素）
//
//
// connection
// 由 connectXxx() 返回的打印机连接句柄。
//
// x
// X 坐标（点）。
//
// y
// Y 坐标（点）。
//
// pixels
// ARGB 像素数组。
//
// imgWidth
// 源图宽度。
//
// imgHeight
// 源图高度。
//
// width
// 宽度（点）/ 模块宽度。
//
// height
// 高度（点）。
//
// 返回值
// 指令写入成功返回 true，否则返回 false。
```

标签打印功能

enableLabelMode

启用标签模式

语法

```
public boolean enableLabelMode(PrinterConnection connection);
```

```
// 启用标签模式
//
//
// connection
// 由 connectXxx() 返回的打印机连接句柄。
//
// 返回值
// 指令写入成功返回 true，否则返回 false。
```

disableLabelMode

禁用标签模式

语法

```
public boolean disableLabelMode(PrinterConnection connection);
```

```
// 禁用标签模式
//
//
// connection
// 由 connectXxx() 返回的打印机连接句柄。
//
// 返回值
// 指令写入成功返回 true，否则返回 false。
```

labelPageBegin

开始标签页

语法

```
public boolean labelPageBegin(PrinterConnection connection, int x, int y, int width, int height, int rotation);
```

```
// 开始标签页
//
//
// connection
// 由 connectXxx() 返回的打印机连接句柄。
//
// x
// X 坐标（点）。
//
// y
// Y 坐标（点）。
//
// width
// 宽度（点）/ 模块宽度。
//
// height
// 高度（点）。
//
// rotation
// 标签页旋转：0 或 1。
//
// 返回值
// 指令写入成功返回 true，否则返回 false。
```

backPaperToPrintPosition

回纸到打印位置

语法

```
public boolean backPaperToPrintPosition(PrinterConnection connection);
```

```
// 回纸到打印位置
//
//
// connection
// 由 connectXxx() 返回的打印机连接句柄。
//
// 返回值
// 指令写入成功返回 true，否则返回 false。
```

labelPagePrint

标签页打印

语法

```
public boolean labelPagePrint(PrinterConnection connection);
```

```
// 标签页打印
//
//
// connection
// 由 connectXxx() 返回的打印机连接句柄。
//
// 返回值
// 指令写入成功返回 true，否则返回 false。
```

labelPagePrint

标签页打印（指定数量）

语法

```
public boolean labelPagePrint(PrinterConnection connection, int printNum);
```

```
// 标签页打印（指定数量）
//
//
// connection
// 由 connectXxx() 返回的打印机连接句柄。
//
// printNum
// 标签打印份数。
//
// 返回值
// 指令写入成功返回 true，否则返回 false。
```

printLabelText

打印标签文本

语法

```
public boolean printLabelText(PrinterConnection connection, String text, int x, int y, int fontSize);
```

```
// 打印标签文本
//
//
// connection
// 由 connectXxx() 返回的打印机连接句柄。
//
// text
// 要打印的文本。
//
// x
// X 坐标（点）。
//
// y
// Y 坐标（点）。
//
// fontSize
// 标签字号（16/20/24/96）。
//
// 返回值
// 指令写入成功返回 true，否则返回 false。
```

printLabelText

打印标签文本（完整参数）

语法

```
public boolean printLabelText(PrinterConnection connection, String text, int x, int y, int fontHeight, int fontType);
```

```
// 打印标签文本（完整参数）
//
//
// connection
// 由 connectXxx() 返回的打印机连接句柄。
//
// text
// 要打印的文本。
//
// x
// X 坐标（点）。
//
// y
// Y 坐标（点）。
//
// fontHeight
// 标签字号（16/20/24/96）。
//
// fontType
// ASCII / 条码字体类型。
//
// 返回值
// 指令写入成功返回 true，否则返回 false。
```

printLabelTextUTF8

打印标签文本（UTF8）

语法

```
public boolean printLabelTextUTF8(PrinterConnection connection, String text, int x, int y);
```

```
// 打印标签文本（UTF8）
//
//
// connection
// 由 connectXxx() 返回的打印机连接句柄。
//
// text
// 要打印的文本。
//
// x
// X 坐标（点）。
//
// y
// Y 坐标（点）。
//
// 返回值
// 指令写入成功返回 true，否则返回 false。
```

printLabelTextUTF8

打印标签文本（UTF8，完整参数）

语法

```
public boolean printLabelTextUTF8(PrinterConnection connection, String text, int x, int y, int fontHeight, int fontType);
```

```
// 打印标签文本（UTF8，完整参数）
//
//
// connection
// 由 connectXxx() 返回的打印机连接句柄。
//
// text
// 要打印的文本。
//
// x
// X 坐标（点）。
//
// y
// Y 坐标（点）。
//
// fontHeight
// 标签字号（16/20/24/96）。
//
// fontType
// ASCII / 条码字体类型。
//
// 返回值
// 指令写入成功返回 true，否则返回 false。
```

printLabelTextGBK

打印标签文本（GBK）

语法

```
public boolean printLabelTextGBK(PrinterConnection connection, String text, int x, int y);
```

```
// 打印标签文本（GBK）
//
//
// connection
// 由 connectXxx() 返回的打印机连接句柄。
//
// text
// 要打印的文本。
//
// x
// X 坐标（点）。
//
// y
// Y 坐标（点）。
//
// 返回值
// 指令写入成功返回 true，否则返回 false。
```

printLabelTextGBK

打印标签文本（GBK，完整参数）

语法

```
public boolean printLabelTextGBK(PrinterConnection connection, String text, int x, int y, int fontHeight, int fontType);
```

```
// 打印标签文本（GBK，完整参数）
//
//
// connection
// 由 connectXxx() 返回的打印机连接句柄。
//
// text
// 要打印的文本。
//
// x
// X 坐标（点）。
//
// y
// Y 坐标（点）。
//
// fontHeight
// 标签字号（16/20/24/96）。
//
// fontType
// ASCII / 条码字体类型。
//
// 返回值
// 指令写入成功返回 true，否则返回 false。
```

printLabelBarcode

打印标签条码

语法

```
public boolean printLabelBarcode(PrinterConnection connection, int type, String data, int x, int y, int height, int width);
```

```
// 打印标签条码
//
//
// connection
// 由 connectXxx() 返回的打印机连接句柄。
//
// type
// 类型常量（条码/切纸/状态等）。
//
// data
// 数据内容（字符串或字节）。
//
// x
// X 坐标（点）。
//
// y
// Y 坐标（点）。
//
// height
// 高度（点）。
//
// width
// 宽度（点）/ 模块宽度。
//
// 返回值
// 指令写入成功返回 true，否则返回 false。
```

printLabelBarcode

打印标签条码（完整参数）

语法

```
public boolean printLabelBarcode(PrinterConnection connection, int x, int y, int barcodeType, int barcodeHeight, int unitWidth, int rotate, String data);
```

```
// 打印标签条码（完整参数）
//
//
// connection
// 由 connectXxx() 返回的打印机连接句柄。
//
// x
// X 坐标（点）。
//
// y
// Y 坐标（点）。
//
// barcodeType
// ESC/POS BARCODE_TYPE_* 或 LABEL_BARCODE_TYPE_*。
//
// barcodeHeight
// 条码高度（点）。
//
// unitWidth
// 条码 / 二维码单位宽度。
//
// rotate
// 旋转标志 / 标签旋转 0–3。
//
// data
// 数据内容（字符串或字节）。
//
// 返回值
// 指令写入成功返回 true，否则返回 false。
```

setLabelBarcodeTextPosition

设置标签条码文字位置

语法

```
public boolean setLabelBarcodeTextPosition(PrinterConnection connection, int position);
```

```
// 设置标签条码文字位置
//
//
// connection
// 由 connectXxx() 返回的打印机连接句柄。
//
// position
// 绝对打印位置 / HRI 位置。
//
// 返回值
// 指令写入成功返回 true，否则返回 false。
```

printLabelQRCode

打印标签二维码

语法

```
public boolean printLabelQRCode(PrinterConnection connection, String data, int x, int y, int size);
```

```
// 打印标签二维码
//
//
// connection
// 由 connectXxx() 返回的打印机连接句柄。
//
// data
// 数据内容（字符串或字节）。
//
// x
// X 坐标（点）。
//
// y
// Y 坐标（点）。
//
// size
// 模块大小。
//
// 返回值
// 指令写入成功返回 true，否则返回 false。
```

printLabelQRCode

打印标签二维码（完整参数）

语法

```
public boolean printLabelQRCode(PrinterConnection connection, int x, int y, int version, int eccLevel, int unitWidth, int rotate, String data);
```

```
// 打印标签二维码（完整参数）
//
//
// connection
// 由 connectXxx() 返回的打印机连接句柄。
//
// x
// X 坐标（点）。
//
// y
// Y 坐标（点）。
//
// version
// 二维码版本 / 模块提示。
//
// eccLevel
// 二维码纠错等级（QR_ECC_LEVEL_*）。
//
// unitWidth
// 条码 / 二维码单位宽度。
//
// rotate
// 旋转标志 / 标签旋转 0–3。
//
// data
// 数据内容（字符串或字节）。
//
// 返回值
// 指令写入成功返回 true，否则返回 false。
```

printLabelPDF417Code

打印标签PDF417条码

语法

```
public boolean printLabelPDF417Code(PrinterConnection connection, String data, int x, int y);
```

```
// 打印标签PDF417条码
//
//
// connection
// 由 connectXxx() 返回的打印机连接句柄。
//
// data
// 数据内容（字符串或字节）。
//
// x
// X 坐标（点）。
//
// y
// Y 坐标（点）。
//
// 返回值
// 指令写入成功返回 true，否则返回 false。
```

printLabelPDF417Code

打印标签PDF417条码（完整参数）

语法

```
public boolean printLabelPDF417Code(PrinterConnection connection, int x, int y, int colNum, int eccLevel, int lwRatio, int unitWidth, int rotate, String data);
```

```
// 打印标签PDF417条码（完整参数）
//
//
// connection
// 由 connectXxx() 返回的打印机连接句柄。
//
// x
// X 坐标（点）。
//
// y
// Y 坐标（点）。
//
// colNum
// PDF417 列数。
//
// eccLevel
// 二维码纠错等级（QR_ECC_LEVEL_*）。
//
// lwRatio
// PDF417 长宽比。
//
// unitWidth
// 条码 / 二维码单位宽度。
//
// rotate
// 旋转标志 / 标签旋转 0–3。
//
// data
// 数据内容（字符串或字节）。
//
// 返回值
// 指令写入成功返回 true，否则返回 false。
```

drawLabelImageFromFile

绘制标签图片（从文件）

语法

```
public boolean drawLabelImageFromFile(PrinterConnection connection, int x, int y, String imagePath);
```

```
// 绘制标签图片（从文件）
//
//
// connection
// 由 connectXxx() 返回的打印机连接句柄。
//
// x
// X 坐标（点）。
//
// y
// Y 坐标（点）。
//
// imagePath
// 图片文件路径。
//
// 返回值
// 指令写入成功返回 true，否则返回 false。
```

drawLabelImageFromData

绘制标签图片（从数据）

语法

```
public boolean drawLabelImageFromData(PrinterConnection connection, int x, int y, byte[] imageData);
```

```
// 绘制标签图片（从数据）
//
//
// connection
// 由 connectXxx() 返回的打印机连接句柄。
//
// x
// X 坐标（点）。
//
// y
// Y 坐标（点）。
//
// imageData
// 编码后的图片字节。
//
// 返回值
// 指令写入成功返回 true，否则返回 false。
```

drawLabelImageFromPixels

绘制标签图片（从像素）

语法

```
public boolean drawLabelImageFromPixels(PrinterConnection connection, int x, int y, int[] pixels, int width, int height);
```

```
// 绘制标签图片（从像素）
//
//
// connection
// 由 connectXxx() 返回的打印机连接句柄。
//
// x
// X 坐标（点）。
//
// y
// Y 坐标（点）。
//
// pixels
// ARGB 像素数组。
//
// width
// 宽度（点）/ 模块宽度。
//
// height
// 高度（点）。
//
// 返回值
// 指令写入成功返回 true，否则返回 false。
```

drawLabelImageAdvanced

绘制标签图片（高级）

语法

```
public boolean drawLabelImageAdvanced(PrinterConnection connection, int x, int y, int[] pixels, int width, int height, int showType);
```

```
// 绘制标签图片（高级）
//
//
// connection
// 由 connectXxx() 返回的打印机连接句柄。
//
// x
// X 坐标（点）。
//
// y
// Y 坐标（点）。
//
// pixels
// ARGB 像素数组。
//
// width
// 宽度（点）/ 模块宽度。
//
// height
// 高度（点）。
//
// showType
// 图片显示类型。
//
// 返回值
// 指令写入成功返回 true，否则返回 false。
```

drawLabelLine

绘制标签线段

语法

```
public boolean drawLabelLine(PrinterConnection connection, int startX, int startY, int endX,
int endY);
```

```
// 绘制标签线段
//
//
// connection
// 由 connectXxx() 返回的打印机连接句柄。
//
// startX
// 线段起点 X。
//
// startY
// 线段起点 Y。
//
// endX
// 线段终点 X。
//
// endY
// 线段终点 Y。
//
// 返回值
// 指令写入成功返回 true，否则返回 false。
```

drawLabelLine

绘制标签线段（完整参数）

语法

```
public boolean drawLabelLine(PrinterConnection connection, int startX, int startY, int endX,
int endY, int width, int color);
```

```
// 绘制标签线段（完整参数）
//
//
// connection
// 由 connectXxx() 返回的打印机连接句柄。
//
// startX
// 线段起点 X。
//
// startY
// 线段起点 Y。
//
// endX
// 线段终点 X。
//
// endY
// 线段终点 Y。
//
// width
// 宽度（点）/ 模块宽度。
//
// color
// 绘制颜色（视固件 0/1）。
//
// 返回值
// 指令写入成功返回 true，否则返回 false。
```

drawLabelRect

绘制标签矩形

语法

```
public boolean drawLabelRect(PrinterConnection connection, int left, int top, int right, int bottom);
```

```
// 绘制标签矩形
//
//
// connection
// 由 connectXxx() 返回的打印机连接句柄。
//
// left
// 左侧文本。
//
// top
// 详见方法说明。
//
// right
// 右侧文本。
//
// bottom
// 详见方法说明。
//
// 返回值
// 指令写入成功返回 true，否则返回 false。
```

drawLabelRect

绘制标签矩形（完整参数）

语法

```
public boolean drawLabelRect(PrinterConnection connection, int left, int top, int right, int bottom, int width, int color);
```

```
// 绘制标签矩形（完整参数）
//
//
// connection
// 由 connectXxx() 返回的打印机连接句柄。
//
// left
// 左侧文本。
//
// top
// 详见方法说明。
//
// right
// 右侧文本。
//
// bottom
// 详见方法说明。
//
// width
// 宽度（点）/ 模块宽度。
//
// color
// 绘制颜色（视固件 0/1）。
//
// 返回值
// 指令写入成功返回 true，否则返回 false。
```

drawLabelBox

绘制标签实心框

语法

```
public boolean drawLabelBox(PrinterConnection connection, int left, int top, int right, int bottom, int color);
```

```
// 绘制标签实心框
//
//
// connection
// 由 connectXxx() 返回的打印机连接句柄。
//
// left
// 左侧文本。
//
// top
// 详见方法说明。
//
// right
// 右侧文本。
//
// bottom
// 详见方法说明。
//
// color
// 绘制颜色（视固件 0/1）。
//
// 返回值
// 指令写入成功返回 true，否则返回 false。
```

其它功能

备注

- 页模式（ESC/POS）与标签模式（JPL）请勿混用，切换时注意先进入正确模式。
- 多数固件标签字号仅支持：**16 / 20 / 24 / 96**。
- 58mm 纸宽用 `PAPER_WIDTH_58MM`（384）；80mm 用 `PAPER_WIDTH_80MM`（576）。