

UxPrint SDK 集成指南与 API 参考

1. 集成注意事项

最低部署目标

架构支持

权限要求

集成方式

依赖库

初始化

2. API 参考

2.1 UXPCCommonPrinter

打印机查询

查询打印机型号

查询打印机序列号

查询固件版本号

查询电池剩余电量

查询当前纸张介质类型

查询打印宽度数值

查询Wi-Fi工作模式

打印机配置

配置纸张介质类型

开启或关闭任务重打功能

调节打印深浅浓度

设置打印区域宽度

设置打印机界面语言

设置字符编码表

设置国际字符集

其他

重启打印机

2.2 ESC/POS 打印机 — UXPEscPrinter

2.3 TSPL 标签打印机 — UXPTsplPrinter

2.4 CPCL 标签打印机 — UXPCpclPrinter

2.5 ZPL 标签打印机 — UXPZplPrinter

3. 接口说明 — 边缘功能 (Supplementary Features)

3.1 PDF 预览 — UXPPDFRepr

加载 PDF 文件

获取指定页码的渲染图像

附录一：公共设施 (Utilities)

A.1 基类：UXPBasePrinter

A.2 打印选项类

UXPPrintTaskResult

UXPTextPrintOptions

UXPBarcodePrintOptions

UXPQRCodePrintOptions

UXPImageProcessingOptions

UXPImagePrintOptions

A.3 错误处理：UXPError

A.4 CPCL 附加类型

PTCPCLCountryCode (国家/地区代码)

UXPCPCLBarcodeStyle (CPCL 条码类型)

附录二：枚举 & 常量 (Enums & Constants)

UXPPrinterPaperType

UXPPrinterLanguage

UXPAlignment

UXPPrinterBarcodeType

UXPPrinterBarcodeTextPosition

UXPPrinterCompressionType

UXPPrinterDitheringAlgorithm

UXPPrinterQRErrorCorrection

UXPEscPrinterStatus

UXPTsplPrinterStatus

UXPCpclPrinterStatus

UXPZplPrinterStatus

UXPWiFiEncryption

UXPBaseOtaStatus

UXPBitmapCompressMode

UXPBitmapMode

UXPCodePage

UXPCharacterSet

UXPErrorCode

UXPEscBarcodeType

UXPEscQRCodeCorrectionLevel

UXPEscPageModeDirection

UXPTsplQRCodeEccLevel

Typedef Blocks（类型定义回调）

- UXPEscPrinterStatusResultBlock
- UXPTsplPrinterStatusResultBlock
- UXPCpclPrinterStatusResultBlock
- UXPZplPrinterStatusResultBlock
- UXPPrintTaskResultBlock
- UXPCodePageResultBlock
- UXPCharacterSetResultBlock
- UXPEscMissPrintEventHandler

UxPrint SDK 集成指南与 API 参考

版本：1.0.0 | 最低部署目标：iOS 9.0

1. 集成注意事项

最低部署目标

UxPrint SDK 的最低部署目标为 **iOS 9.0**，Example 示例工程要求 **iOS 11.0**。

架构支持

SDK 支持以下 CPU 架构：

架构	说明
arm64	真机设备（iPhone / iPad）
x86_64	Simulator（Intel Mac）

权限要求

在 Info.plist 中添加以下权限描述：

Key	类型	说明
NSBluetoothAlwaysUsageDescription	String	蓝牙打印机发现与连接，例如 "使用蓝牙连接打印机进行打印"

注意：如仅需 TCP/IP (Socket) 方式连接打印机，则无需添加蓝牙权限。

集成方式

UxPrint SDK 不通过 CocoaPods 或 Swift Package Manager 分发。请通过 Xcode 工作区直接集成：

- 1. 克隆或下载 SDK 仓库。
- 2. 打开 project/UxPrint.xcworkspace（若未使用 CocoaPods 则打开 project/UxPrint.xcodeproj）。
- 3. 将 PrinterSDK framework target 添加至主 App 的 **Target Dependencies**。
- 4. 确保以下三个 xcframework 已被链接至主 App：

依赖库

SDK 附带以下 vendored xcframework，位于仓库根目录的 sdk/ 目录下：

Framework	版本	说明
PrinterSDK.xcframework	3.4.7	UxPrint SDK 主库
RYCommunication.xcframework	—	通信层，提供 BLE、TCP/IP (Socket)、MFi (ExternalAccessory)、UDP 等连接能力
CocoaAsyncSocket.xcframework	—	第三方 TCP Socket 库，为 RYCommunication 提供底层 Socket 支持

Example 工程额外通过 CocoaPods 依赖 SVProgressHUD (v2.2.5)，仅用于示例 UI 交互，非 SDK 必需依赖。

初始化

SDK 无需全局初始化。使用方式为直接创建 UXPCommonPrinter 实例，并通过父类 RYSharedAccessory 提供的构造方法指定连接方式，然后调用连接方法即可。

构造方法概览（继承自 RYSharedAccessory）：

构造方法	连接方式
initWith:port:	TCP/IP（Socket）连接，传入 IP 地址和端口号
initWithBleAccessory:	BLE 连接，传入 RYBleAccessory 对象（需自行通过 RYCentralManager 扫描发现）
initWithAccessory:forProtocol:	MFi（ExternalAccessory）连接，传入系统 EAAccessory 对象和协议字符串
initWith:	通用包装，传入已有 RYBaseAccessory 子类实例

连接方法（遵循 RYAccessory 协议）：

```
- (void)connect:(void (^)(_Nullable)(void))successBlock
               fail:(void (^)(_Nullable)(NSError *))failBlock;
```

断开连接：

```
- (void)disconnect;
```

示例：BLE 连接流程

```
// 1. 通过 RYCentralManager 扫描并获取 RYBleAccessory 实例
RYCentralManager *manager = [[RYCentralManager alloc] init];
[manager scanWithDuration:5.0 complete:^(NSArray<RYBleAccessory *>
*accessories) {
    // 2. 选择设备并创建 UXPCommonPrinter 实例
    RYBleAccessory *device = accessories.firstObject;
```

```
UXPCommonPrinter *printer = [[UXPCommonPrinter alloc]
initWithBleAccessory:device];

// 3. 连接
[printer connect:^(
    NSLog(@"连接成功");
    // 4. 连接成功后即可调用查询/配置/打印等方法
} fail:^(NSError *error) {
    NSLog(@"连接失败：%@", error);
}];
};
```

示例：Socket (TCP/IP) 连接流程

```
UXPCommonPrinter *printer = [[UXPCommonPrinter alloc]
initWith:@"192.168.1.100" port:9100];
[printer connect:^(
    NSLog(@"连接成功");
} fail:^(NSError *error) {
    NSLog(@"连接失败：%@", error);
}];
```

2. API 参考

2.1 UXPCommonPrinter

项目	内容
类名	UXPCommonPrinter
继承链	UXPCommonPrinter : RYSharedAccessory : RYBaseAccessory : NSObject
头文件	<UxPrint/UXPCommonPrinter.h>
说明	管理打印机连接、信息查询、参数配置、固件升级和数据收发，所有操作的基础入口。

打印机查询

查询打印机型号

获取打印机固件中存储的设备型号标识，用于区分硬件规格。

```
- (void)getPrinterModel:(RYStringResultBlock)completeBlock;
```

参数说明：

参数	类型	说明
completeBlock	RYStringResultBlock	结果回调，返回打印机型号字符串。 ^(NSString * _Nullable result, NSError * _Nullable error)

返回值： 无

查询打印机序列号

获取打印机的唯一序列号，用于设备识别和资产管理。

```
- (void)getSerialNumber:(RYStringResultBlock)completeBlock;
```

参数说明：

参数	类型	说明
completeBlock	RYStringResultBlock	结果回调，返回序列号字符串。 ^(NSString * _Nullable result, NSError * _Nullable error)

查询固件版本号

获取打印机当前运行的固件版本字符串，用于确认是否需要升级。

```
- (void)getPrinterVersion:(RYStringResultBlock)completeBlock;
```

参数说明：

参数	类型	说明
completeBlock	RYStringResultBlock	结果回调，返回固件版本字符串。 ^(NSString * _Nullable result, NSError * _Nullable error)

查询电池剩余电量

返回当前电池电量百分比（0~100），仅便携式打印机有效。

```
- (void)getBatteryLevel:(RYIntegerResultBlock)completeBlock;
```

参数说明：

参数	类型	说明
completeBlock	RYIntegerResultBlock	结果回调，返回电量百分比 0~100。 ^(NSInteger result, NSError * _Nullable error)

返回值： 无

查询当前纸张介质类型

获取打印机当前配置的纸张类型，如连续纸、标签纸或黑标纸。

```
- (void)getPaperType:(RYIntegerResultBlock)completeBlock;
```

参数说明：

参数	类型	说明
completeBlock	RYIntegerResultBlock	结果回调，返回 UXPPrinterPaperType 枚举原始值。 ^(NSInteger result, NSError * _Nullable error)

返回值： 无

查询打印宽度数值

返回打印机当前可打印区域的宽度值。

```
- (void)getPrintWidth:(RYIntegerResultBlock)completeBlock;
```

参数说明：

参数	类型	说明
completeBlock	RYIntegerResultBlock	结果回调，返回打印宽度值。 ^(NSInteger result, NSError * _Nullable error)

查询Wi-Fi工作模式

返回打印机当前的 Wi-Fi 模式：1 表示 AP 模式，2 表示 STA 模式。

```
- (void)getPrinterWifiMode:(RYIntegerResultBlock)completeBlock;
```

参数说明：

参数	类型	说明
completeBlock	RYIntegerResultBlock	结果回调，返回工作模式： 1 =AP 模式， 2 =STA 模式。 ^(NSInteger result, NSError * _Nullable error)

返回值： 无

打印机配置

配置纸张介质类型

设置打印机当前使用的纸张类型，确保走纸和定位方式正确匹配。

```
- (void)setPaperType:(UXPPrinterPaperType)paperType  
    complete:(RYEmptyResultBlock _Nullable)completeBlock;
```

参数说明：

参数	类型	说明
paperType	UXPPrinterPaperType	纸张类型枚举： UXPPrinterPaperTypeContinuous （连续纸）、 UXPPrinterPaperTypeLabel （标签纸）、 UXPPrinterPaperTypeBlackMarkFront2Inch （2寸前黑标）、 UXPPrinterPaperTypeBlackMarkFront3Inch （3寸前黑标）、 UXPPrinterPaperTypeBlackMarkBack2Inch （2寸后黑标）、 UXPPrinterPaperTypeBlackMarkBack3Inch （3寸后黑标）
completeBlock	RYEmptyResultBlock _Nullable	结果回调。 ^(NSError * _Nullable error)

返回值： 无

开启或关闭任务重打功能

开启后打印机在打印中断时可自动重打未完成的任务。

```
- (void)setReprintSwitch:(BOOL)enabled  
      complete:(RYEmptyResultBlock _Nullable)completeBlock;
```

参数说明：

参数	类型	说明
enabled	BOOL	是否开启重打功能
completeBlock	RYEmptyResultBlock _Nullable	结果回调。 ^(NSError * _Nullable error)

调节打印深浅浓度

调整打印头加热强度，数值 0~15，越大打印越深。

```
- (void)setPrintDensity:(NSInteger)density  
        complete:(RYEmptyResultBlock _Nullable)completeBlock;
```

参数说明：

参数	类型	说明
density	NSInteger	打印浓度，范围 0~15，数值越大浓度越深
completeBlock	RYEmptyResultBlock _Nullable	结果回调。 ^(NSError * _Nullable error)

返回值： 无

设置打印区域宽度

限制打印机可打印的水平宽度范围。

```
- (void)setPrintWidth:(NSInteger)width  
        complete:(RYEmptyResultBlock _Nullable)completeBlock;
```

参数说明：

参数	类型	说明
width	NSInteger	打印宽度值
completeBlock	RYEmptyResultBlock _Nullable	结果回调。 ^(NSError * _Nullable error)

返回值： 无

设置打印机界面语言

切换打印机的界面和提示信息显示语言。

```
- (void)setPrinterLanguage:(UXPPrinterLanguage)language
    complete:(RYEmptyResultBlock
    _Nullable)completeBlock;
```

参数说明：

参数	类型	说明
language	UXPPrinterLanguage	打印机语言： UXPPrinterLanguageEnglish （英文）、 UXPPrinterLanguageSimplifiedChinese （简体中文）、 UXPPrinterLanguageTraditionalChinese （繁体中文）
completeBlock	RYEmptyResultBlock _Nullable	结果回调。 ^(NSError * _Nullable error)

返回值： 无

设置字符编码表

配置打印机用于文本打印的代码页，不同语言区域需选择对应代码页。

```
- (void)setCodePage:(UXPCodePage)codePage
    complete:(RYEmptyResultBlock _Nullable)completeBlock;
```

参数说明：

参数	类型	说明
codePage	UXPCodePage	代码页枚举值，如 UXPCodePagePC437 、 UXPCodePagePC850 、 UXPCodePageISO88591 等
completeBlock	RYEmptyResultBlock _Nullable	结果回调。 ^(NSError * _Nullable error)

返回值： 无

设置国际字符集

选择打印机使用的国际字符集，适配不同国家和地区的字符显示。

```
- (void)setCharacterSet:(UXPCharacterSet)characterSet
    complete:(RYEmptyResultBlock _Nullable)completeBlock;
```

参数说明：

参数	类型	说明
characterSet	UXPCharacterSet	字符集枚举值，如 UXPCharacterSetUSA 、 UXPCharacterSetChina 、 UXPCharacterSetJapan 等
completeBlock	RYEmptyResultBlock _Nullable	结果回调。 ^(NSError * _Nullable error)

返回值： 无

其他

重启打印机

向打印机发送重启指令，重启后所有设置保持当前配置。

```
- (void)reboot:(RYEmptyResultBlock _Nullable)completeBlock;
```

参数说明：

参数	类型	说明
completeBlock	RYEmptyResultBlock _Nullable	结果回调。 ^(NSError * _Nullable error)

2.2 ESC/POS 打印机 — UXPEscPrinter

指令详细说明请参阅 [ESC/POS 指令集](#)。

2.3 TSPL 标签打印机 — UXPTsplPrinter

指令详细说明请参阅 [TSPL 指令集](#)。

2.4 CPCL 标签打印机 — UXPCpclPrinter

指令详细说明请参阅 [CPCL 指令集](#)。

2.5 ZPL 标签打印机 — UXPZplPrinter

指令详细说明请参阅 [ZPL 指令集](#)。

3. 接口说明 — 边缘功能 (Supplementary Features)

3.1 PDF 预览 — UXPPDFRepr

用途： 将 PDF 文档逐页渲染为 UIImage 图像，用于打印预览或图片化处理。

类层次： UXPPDFRepr : NSObject

头文件： <UxPrint/UXPPDFRepr.h>

属性：

属性	类型	说明
pageCount	NSInteger (readonly)	PDF 总页数

加载 PDF 文件

通过文件路径初始化渲染器，可选择是否自动旋转页面方向。

```
- (instancetype)initWithPath:(NSURL *)path autoRotate:
(BBOOL)autoRotate;
```

参数说明：

参数	类型	说明
path	NSURL *	PDF 文件路径
autoRotate	BOOL	是否自动旋转页面方向

返回值： `instancetype` ，初始化的 `UXPPDFRepr` 实例。

获取指定页码的渲染图像

将 PDF 的指定页面按目标尺寸渲染为 `UIImage`，用于预览。

```
- (UIImage * _Nullable)imageAtIndex:(NSInteger)index size:
(CGSize)size;
```

参数说明：

参数	类型	说明
index	NSInteger	页码（从 0 开始）
size	CGSize	目标渲染尺寸

返回值： `UIImage *` ，指定页的渲染图像；失败时返回 `nil` 。

附录一：公共设施 (Utilities)

A.1 基类：UXPBasePrinter

用途： 所有协议打印机类的基类，提供向打印机发送原始数据的能力。

类层次： `UXPBasePrinter` : `RYSharedAccessory`

头文件： <UxPrint/UXPBasePrinter.h>

方法：

方法	说明
<code>- (void)uxp_sendData:(NSData *)data complete:(RYEmptyResultBlock _Nullable)completeBlock</code>	向打印机发送数据并通过 block 回调结果。 参数 <code>data</code> 为待发送的数据， <code>completeBlock</code> 为发送完成回调。

A.2 打印选项类

UXPPrintTaskResult

用途： 封装打印任务的结果， 包含任务标识和状态值。

头文件： <UxPrint/UXPType.h>

属性：

属性	类型	说明
<code>taskId</code>	<code>NSInteger</code>	任务标识
<code>status</code>	<code>NSInteger</code>	任务状态值

方法：

方法	说明
<code>- (instancetype)initWithTaskId:(NSInteger)taskId status:(NSInteger)status</code>	使用任务标识和状态 值初始化
<code>+ (instancetype)resultWithTaskId:(NSInteger)taskId status:(NSInteger)status</code>	创建打印任务结果对 象

UXPTextPrintOptions

用途： 通用文本打印选项。

头文件： <UxPrint/UXPType.h>

属性：

属性	类型	默认值	说明
fontSize	NSInteger	24	抽象字号
bold	BOOL	NO	是否加粗
underline	BOOL	NO	是否下划线
alignment	UXPAlignment	Left	文本对齐方式

方法：

方法	说明
+ (instancetype)defaultOptions	返回默认文本选项

UXPBarcodePrintOptions

用途： 通用条码打印选项。

头文件： <UxPrint/UXPType.h>

属性：

属性	类型	默认值	说明
height	NSInteger	50	条码高度
width	NSInteger	2	条码线宽
textPosition	UXPPrinterBarcodeTextPosition	Below	可读文本位置
rotation	NSInteger	0	旋转角度或协议旋转值

方法：

方法	说明
<code>+(instancetype)defaultOptions</code>	返回默认条码选项

UXPQRCodePrintOptions

用途： 通用二维码打印选项。

头文件： `<UxPrint/UXPType.h>`

属性：

属性	类型	默认值	说明
size	NSInteger	6	二维码尺寸
errorCorrection	UXPPrinterQRErrorCorrection	M	纠错等级
rotation	NSInteger	0	旋转角度或协议旋转值

方法：

方法	说明
<code>+(instancetype)defaultOptions</code>	返回默认二维码选项

UXPImageProcessingOptions

用途： 图片预处理选项，用于将 UIImage 转换为单色打印数据前的自定义处理。

头文件： `<UxPrint/UXPType.h>`

属性：

属性	类型	说明
rotation	NSInteger	旋转角度，仅支持 0、90、180、270
paperWidth	NSInteger	目标纸宽（毫米），为 0 时忽略
maxWidth	NSInteger	最大像素宽度，为 0 时表示不限制
brightness	NSInteger	亮度调整，范围建议 -255~255
contrast	CGFloat	对比度调整，1.0 表示不调整
threshold	NSInteger	二值化阈值，传 -1 时自动计算
ditheringAlgorithm	UXPPrinterDitheringAlgorithm	抖动算法
dpi	NSInteger	DPI，用于纸宽换算

方法：

方法	说明
<code>+(instancetype)defaultOptions</code>	返回默认图片预处理选项

UXPImagePrintOptions

用途： 图片打印选项，用于直接打印位图时的压缩与二值化配置。

头文件： `<UxPrint/UXPType.h>`

属性：

属性	类型	默认值	说明
compression	UXPBitmapCompressMode	None	图片压缩方式
threshold	NSInteger	-	二值化阈值
ditheringAlgorithm	UXPBitmapMode	-	二值算法

方法：

方法	说明
<code>+ (instancetype)defaultOptions</code>	返回默认图片打印选项

A.3 错误处理：UXPError

头文件：`<UxPrint/UXPError.h>`

符号常量：

符号	类型	说明
<code>UXPErrorDomain</code>	<code>NSErrorDomain</code>	SDK 错误域
<code>UXPInnerErrorUserInfoKey</code>	<code>NSErrorUserInfoKey</code>	内部错误对象键名

函数：

函数	说明
<code>NSError *UXPErrorFromCode(UXPErrCode code)</code>	根据错误码生成 SDK 错误对象
<code>NSError *UXPErrorFromCodeWithInnerError(UXPErrCode code, NSError * _Nullable error)</code>	根据错误码和底层错误生成 SDK 错误对象
<code>NSError *UXPErrorFromCodeWithOtaStatus(UXPBaseOtaStatus status)</code>	根据 OTA 状态生成 SDK 错误对象

A.4 CPCL 附加类型

头文件：`<UxPrint/UXPCpclType.h>`

PTCPCLCountryCode（国家/地区代码）

用途： CPCL 打印机国家/地区代码，使用 NS_STRING_ENUM 定义。

```
typedef NSString * PTCPCLCountryCode NS_STRING_ENUM;
```

常量名	字符串值	说明
PTCPCLCountryCodeUSA	@ "USA"	美国
PTCPCLCountryCodeUK	@ "UK"	英国
PTCPCLCountryCodeGermany	@ "Germany"	德国
PTCPCLCountryCodeFrance	@ "France"	法国
PTCPCLCountryCodeSweden	@ "Sweden"	瑞典
PTCPCLCountryCodeNorway	@ "Norway"	挪威
PTCPCLCountryCodeItaly	@ "Italy"	意大利
PTCPCLCountryCodeSpain	@ "Spain"	西班牙
PTCPCLCountryCodeDenmark	@ "Denmark"	丹麦
PTCPCLCountryCodeBelgium	@ "Belgium"	比利时
PTCPCLCountryCodePortugal	@ "Portugal"	葡萄牙
PTCPCLCountryCodeJapan	@ "Japan"	日本
PTCPCLCountryCodeKorea	@ "Korea"	韩国
PTCPCLCountryCodeChina	@ "China"	中国
PTCPCLCountryCodeTaiwan	@ "Taiwan"	台湾
PTCPCLCountryCodeHongKong	@ "HongKong"	香港

UXPCPCLBarcodeStyle （CPCL 条码类型）

用途： CPCL 协议使用的条码类型枚举。

```
typedef NS_ENUM(NSUInteger, UXPCPCLBarcodeStyle) {
    UXPCPCLBarcodeStyleUPCA = 0,
    UXPCPCLBarcodeStyleUPCE = 1,
    UXPCPCLBarcodeStyleEAN13 = 2,
    UXPCPCLBarcodeStyleEAN8 = 3,
    UXPCPCLBarcodeStyleCode39 = 4,
    UXPCPCLBarcodeStyleCode93 = 5,
    UXPCPCLBarcodeStyleCode128 = 6,
    UXPCPCLBarcodeStyleCodeBAR = 7
};
```

值名	值	说明
UXPCPCLBarcodeStyleUPCA	0	UPC-A
UXPCPCLBarcodeStyleUPCE	1	UPC-E
UXPCPCLBarcodeStyleEAN13	2	EAN-13
UXPCPCLBarcodeStyleEAN8	3	EAN-8
UXPCPCLBarcodeStyleCode39	4	Code 39
UXPCPCLBarcodeStyleCode93	5	Code 93
UXPCPCLBarcodeStyleCode128	6	Code 128
UXPCPCLBarcodeStyleCodeBAR	7	CodeBAR

附录二：枚举 & 常量 (Enums & Constants)

UXPPrinterPaperType

用途： 打印介质类型。

```
typedef NS_ENUM(Byte, UXPPrinterPaperType) {
    UXPPrinterPaperTypeContinuous = 0x00,
    UXPPrinterPaperTypeLabel = 0x02,
    UXPPrinterPaperTypeBlackMarkFront2Inch = 0x07,
    UXPPrinterPaperTypeBlackMarkFront3Inch = 0x08,
    UXPPrinterPaperTypeBlackMarkBack2Inch = 0x0D,
    UXPPrinterPaperTypeBlackMarkBack3Inch = 0x0E,
};
```

值名	值	说明
UXPPrinterPaperTypeContinuous	0x00	连续纸
UXPPrinterPaperTypeLabel	0x02	标签纸
UXPPrinterPaperTypeBlackMarkFront2Inch	0x07	2寸前黑标
UXPPrinterPaperTypeBlackMarkFront3Inch	0x08	3寸前黑标
UXPPrinterPaperTypeBlackMarkBack2Inch	0x0D	2寸后黑标
UXPPrinterPaperTypeBlackMarkBack3Inch	0x0E	3寸后黑标

UXPPrinterLanguage

用途： 打印机语言类型。

```
typedef NS_ENUM(Byte, UXPPrinterLanguage) {
    UXPPrinterLanguageEnglish = 0,
    UXPPrinterLanguageSimplifiedChinese = 1,
    UXPPrinterLanguageTraditionalChinese = 2,
};
```

值名	值	说明
UXPPrinterLanguageEnglish	0	英文
UXPPrinterLanguageSimplifiedChinese	1	简体中文
UXPPrinterLanguageTraditionalChinese	2	繁体中文

UXPAlignment

用途： 通用对齐方式。

```
typedef NS_ENUM(NSInteger, UXPAlignment) {
    UXPAlignmentLeft = 0,
    UXPAlignmentCenter = 1,
    UXPAlignmentRight = 2,
};
```

值名	值	说明
UXPAlignmentLeft	0	左对齐
UXPAlignmentCenter	1	居中对齐
UXPAlignmentRight	2	右对齐

UXPPrinterBarcodeType

用途： 通用条码类型。

```
typedef NSInteger, UXPPrinterBarcodeType) {
    UXPPrinterBarcodeTypeCode128 = 0,
    UXPPrinterBarcodeTypeCode39 = 1,
    UXPPrinterBarcodeTypeEAN13 = 2,
    UXPPrinterBarcodeTypeEAN8 = 3,
    UXPPrinterBarcodeTypeUPCA = 4,
    UXPPrinterBarcodeTypeUPCE = 5,
    UXPPrinterBarcodeTypeITF = 6,
    UXPPrinterBarcodeTypeCodabar = 7,
};
```

值名	值	说明
UXPPrinterBarcodeTypeCode128	0	Code 128
UXPPrinterBarcodeTypeCode39	1	Code 39
UXPPrinterBarcodeTypeEAN13	2	EAN-13
UXPPrinterBarcodeTypeEAN8	3	EAN-8
UXPPrinterBarcodeTypeUPCA	4	UPC-A
UXPPrinterBarcodeTypeUPCE	5	UPC-E
UXPPrinterBarcodeTypeITF	6	ITF
UXPPrinterBarcodeTypeCodabar	7	Codabar

UXPPrinterBarcodeTextPosition

用途： 条码可读文本位置。

```
typedef NS_ENUM(NSInteger, UXPPrinterBarcodeTextPosition) {
    UXPPrinterBarcodeTextPositionNone = 0,
    UXPPrinterBarcodeTextPositionAbove = 1,
    UXPPrinterBarcodeTextPositionBelow = 2,
    UXPPrinterBarcodeTextPositionBoth = 3,
};
```

值名	值	说明
UXPPrinterBarcodeTextPositionNone	0	不打印可读文本
UXPPrinterBarcodeTextPositionAbove	1	打印在条码上方
UXPPrinterBarcodeTextPositionBelow	2	打印在条码下方
UXPPrinterBarcodeTextPositionBoth	3	同时打印在上下方

UXPPrinterCompressionType

用途： 图片压缩方式。

```
typedef NS_ENUM(NSInteger, UXPPrinterCompressionType) {
    UXPPrinterCompressionTypeNone = 0,
    UXPPrinterCompressionTypeLZO = 1,
};
```

值名	值	说明
UXPPrinterCompressionTypeNone	0	不压缩
UXPPrinterCompressionTypeLZO	1	使用 LZO 压缩

UXPPrinterDitheringAlgorithm

用途： 图片二值化抖动算法。

```
typedef NS_ENUM(NSInteger, UXPPrinterDitheringAlgorithm) {
    UXPPrinterDitheringAlgorithmThreshold = 0,
    UXPPrinterDitheringAlgorithmDiffusion = 1,
    UXPPrinterDitheringAlgorithmCluster = 2,
};
```

值名	值	说明
UXPPrinterDitheringAlgorithmThreshold	0	直接阈值法
UXPPrinterDitheringAlgorithmDiffusion	1	扩散抖动
UXPPrinterDitheringAlgorithmCluster	2	聚集抖动

UXPPrinterQRErrorCorrection

用途： 二维码纠错等级。

```
typedef NS_ENUM(NSInteger, UXPPrinterQRErrorCorrection) {
    UXPPrinterQRErrorCorrectionL = 0,
    UXPPrinterQRErrorCorrectionM = 1,
    UXPPrinterQRErrorCorrectionQ = 2,
    UXPPrinterQRErrorCorrectionH = 3,
};
```

值名	值	说明
UXPPrinterQRErrorCorrectionL	0	L 级
UXPPrinterQRErrorCorrectionM	1	M 级
UXPPrinterQRErrorCorrectionQ	2	Q 级
UXPPrinterQRErrorCorrectionH	3	H 级

UXPEscPrinterStatus

用途：ESC/POS 打印机状态（位掩码）。

```
typedef NS_OPTIONS(NSUInteger, UXPEscPrinterStatus) {
    UXPEscPrinterStatusNone = 0,
    UXPEscPrinterStatusCoverOpen = 1 << 0,
    UXPEscPrinterStatusPaperNearEnd = 1 << 1,
    UXPEscPrinterStatusPaperOut = 1 << 2,
    UXPEscPrinterStatusCutterError = 1 << 3,
    UXPEscPrinterStatusVoltageError = 1 << 4,
    UXPEscPrinterStatusOverheating = 1 << 5,
};
```

值名	值	说明
UXPEscPrinterStatusNone	0	无异常
UXPEscPrinterStatusCoverOpen	1 << 0	开盖
UXPEscPrinterStatusPaperNearEnd	1 << 1	纸将尽
UXPEscPrinterStatusPaperOut	1 << 2	缺纸
UXPEscPrinterStatusCutterError	1 << 3	切刀错误
UXPEscPrinterStatusVoltageError	1 << 4	电压异常
UXPEscPrinterStatusOverheating	1 << 5	过热

UXPTSplPrinterStatus

用途：TSPL 打印机状态（位掩码）。

```
typedef NS_OPTIONS(NSUInteger, UXPTsplPrinterStatus) {
    UXPTsplPrinterStatusNone = 0,
    UXPTsplPrinterStatusCoverOpen = 1 << 0,
    UXPTsplPrinterStatusPaperOut = 1 << 1,
    UXPTsplPrinterStatusOverheating = 1 << 2,
    UXPTsplPrinterStatusPrinting = 1 << 3,
};
```

值名	值	说明
UXPTsplPrinterStatusNone	0	无异常
UXPTsplPrinterStatusCoverOpen	1 << 0	开盖
UXPTsplPrinterStatusPaperOut	1 << 1	缺纸
UXPTsplPrinterStatusOverheating	1 << 2	过热
UXPTsplPrinterStatusPrinting	1 << 3	打印中

UXPCpclPrinterStatus

用途： CPCL 打印机状态（位掩码）。

```
typedef NS_OPTIONS(NSUInteger, UXPCpclPrinterStatus) {
    UXPCpclPrinterStatusNone = 0,
    UXPCpclPrinterStatusPaperOut = 1 << 0,
    UXPCpclPrinterStatusCoverOpen = 1 << 1,
};
```

值名	值	说明
UXPCpclPrinterStatusNone	0	无异常
UXPCpclPrinterStatusPaperOut	1 << 0	缺纸
UXPCpclPrinterStatusCoverOpen	1 << 1	开盖

UXPZplPrinterStatus

用途： ZPL 打印机状态（位掩码）。

```
typedef NS_OPTIONS(NSUInteger, UXPZplPrinterStatus) {
    UXPZplPrinterStatusNone = 0,
    UXPZplPrinterStatusPaperOut = 1 << 0,
    UXPZplPrinterStatusCoverOpen = 1 << 1,
    UXPZplPrinterStatusOverheating = 1 << 2,
    UXPZplPrinterStatusRibbonOut = 1 << 3,
    UXPZplPrinterStatusPrinting = 1 << 4,
    UXPZplPrinterStatusError = 1 << 5,
};
```

值名	值	说明
UXPZplPrinterStatusNone	0	无异常
UXPZplPrinterStatusPaperOut	1 << 0	缺纸
UXPZplPrinterStatusCoverOpen	1 << 1	开盖
UXPZplPrinterStatusOverheating	1 << 2	过热
UXPZplPrinterStatusRibbonOut	1 << 3	缺碳带
UXPZplPrinterStatusPrinting	1 << 4	打印中
UXPZplPrinterStatusError	1 << 5	其他错误

UXPWiFiEncryption

用途： Wi-Fi 加密方式。

```
typedef NS_ENUM(Byte, UXWiFiEncryption) {
    UXWiFiEncryptionOpen = 0,
    UXWiFiEncryptionWep = 1,
    UXWiFiEncryptionWpa2_psk = 2,
    UXWiFiEncryptionWpa_wpa2_psk = 3,
    UXWiFiEncryptionWpa_psk = 4,
    UXWiFiEncryptionWpa = 5,
    UXWiFiEncryptionWpa2 = 6,
    UXWiFiEncryptionSae = 7,
    UXWiFiEncryptionWpa3_wpa2_psk_mix = 8,
};
```

值名	值	说明
UXWiFiEncryptionOpen	0	开放网络
UXWiFiEncryptionWep	1	WEP
UXWiFiEncryptionWpa2_psk	2	WPA2-PSK
UXWiFiEncryptionWpa_wpa2_psk	3	WPA/WPA2-PSK 混合模式
UXWiFiEncryptionWpa_psk	4	WPA-PSK
UXWiFiEncryptionWpa	5	WPA
UXWiFiEncryptionWpa2	6	WPA2
UXWiFiEncryptionSae	7	SAE
UXWiFiEncryptionWpa3_wpa2_psk_mix	8	WPA3/WPA2-PSK 混合模式

UXPBaseOtaStatus

用途： OTA 升级状态。

```
typedef NS_ENUM(Byte, UXPBaseOtaStatus) {
    UXPBaseOtaStatusOk = 0,
    UXPBaseOtaStatusLengthError,
    UXPBaseOtaStatusCrcError,
    UXPBaseOtaStatusTimeOut,
    UXPBaseOtaStatusWriteFail,
    UXPBaseOtaStatusFail,
};
```

值名	值	说明
UXPBaseOtaStatusOk	0	成功
UXPBaseOtaStatusLengthError	1	长度错误
UXPBaseOtaStatusCrcError	2	CRC 错误
UXPBaseOtaStatusTimeOut	3	超时
UXPBaseOtaStatusWriteFail	4	写入失败
UXPBaseOtaStatusFail	5	执行失败

UXPBitmapCompressMode

用途： 位图压缩模式。

```
typedef NS_ENUM(NSInteger, UXPBitmapCompressMode) {
    UXPBitmapCompressModeNone = 0,
    UXPBitmapCompressModeLZO = 48,
};
```

值名	值	说明
UXPBitmapCompressModeNone	0	不压缩
UXPBitmapCompressModeLZO	48	LZO 压缩算法

UXPBitmapMode

用途： 位图模式（二值化算法）。

```
typedef NS_ENUM(NSInteger, UXPBitmapMode) {
    UXPBitmapModeBinary = 0,
    UXPBitmapModeDithering = 1,
    UXPBitmapModeCluster = 2,
};
```

值名	值	说明
UXPBitmapModeBinary	0	黑白二值图像
UXPBitmapModeDithering	1	扩散抖动
UXPBitmapModeCluster	2	聚集抖动算法

UXPCodePage

用途： ESC/POS 代码页。

```
typedef NS_ENUM(NSInteger, UXPCodePage) {
    UXPCodePagePC437 = 0,
    UXPCodePageKatakana = 1,
    UXPCodePagePC850 = 2,
    UXPCodePagePC860 = 3,
    UXPCodePagePC863 = 4,
    UXPCodePagePC865 = 5,
    UXPCodePagePC857 = 13,
    UXPCodePagePC737 = 14,
    UXPCodePageISO88597 = 15,
    UXPCodePageWindows1252 = 16,
    UXPCodePagePC866 = 17,
    UXPCodePagePC852 = 18,
    UXPCodePagePC858 = 19,
```

```

UXPCodePageKU42 = 20,
UXPCodePageTIS11 = 21,
UXPCodePageTIS18 = 26,
UXPCodePagePC720 = 32,
UXPCodePageWPC775 = 33,
UXPCodePagePC855 = 34,
UXPCodePagePC862 = 36,
UXPCodePagePC864 = 37,
UXPCodePageISO88592 = 39,
UXPCodePageISO885915 = 40,
UXPCodePageWPC1250 = 45,
UXPCodePageWPC1251 = 46,
UXPCodePageWPC1253 = 47,
UXPCodePageWPC1254 = 48,
UXPCodePageWPC1255 = 49,
UXPCodePageWPC1256 = 50,
UXPCodePageWPC1257 = 51,
UXPCodePageWPC1258 = 52,
UXPCodePageMIK = 54,
UXPCodePageCP755 = 55,
UXPCodePageIRAN = 56,
UXPCodePageIRANII = 57,
UXPCodePageLatvian = 58,
UXPCodePageISO88591 = 59,
UXPCodePageISO88593 = 60,
UXPCodePageISO88594 = 61,
UXPCodePageISO88595 = 62,
UXPCodePageISO88596 = 63,
UXPCodePageISO88598 = 64,
UXPCodePageISO88599 = 65,
UXPCodePagePC856 = 66,
UXPCodePageABIC0IM = 67,
};

```

值名	值	说明

UXPCodePagePC437	0	PC437
UXPCodePageKatakana	1	Katakana
UXPCodePagePC850	2	PC850
UXPCodePagePC860	3	PC860
UXPCodePagePC863	4	PC863
UXPCodePagePC865	5	PC865
UXPCodePagePC857	13	PC857
UXPCodePagePC737	14	PC737
UXPCodePageISO88597	15	ISO8859-7
UXPCodePageWindows1252	16	Windows-1252
UXPCodePagePC866	17	PC866
UXPCodePagePC852	18	PC852
UXPCodePagePC858	19	PC858
UXPCodePageKU42	20	KU42
UXPCodePageTIS11	21	TIS-11
UXPCodePageTIS18	26	TIS-18
UXPCodePagePC720	32	PC720
UXPCodePageWPC775	33	WPC775
UXPCodePagePC855	34	PC855
UXPCodePagePC862	36	PC862
UXPCodePagePC864	37	PC864
UXPCodePageISO88592	39	ISO8859-2
UXPCodePageISO885915	40	ISO8859-15
UXPCodePageWPC1250	45	WPC1250
UXPCodePageWPC1251	46	WPC1251

UXPCodePageWPC1253	47	WPC1253
UXPCodePageWPC1254	48	WPC1254
UXPCodePageWPC1255	49	WPC1255
UXPCodePageWPC1256	50	WPC1256
UXPCodePageWPC1257	51	WPC1257
UXPCodePageWPC1258	52	WPC1258
UXPCodePageMIK	54	MIK
UXPCodePageCP755	55	CP755
UXPCodePageIRAN	56	IRAN
UXPCodePageIRANII	57	IRAN II
UXPCodePageLatvian	58	Latvian
UXPCodePageISO88591	59	ISO8859-1
UXPCodePageISO88593	60	ISO8859-3
UXPCodePageISO88594	61	ISO8859-4
UXPCodePageISO88595	62	ISO8859-5
UXPCodePageISO88596	63	ISO8859-6
UXPCodePageISO88598	64	ISO8859-8
UXPCodePageISO88599	65	ISO8859-9
UXPCodePagePC856	66	PC856
UXPCodePageABICOIM	67	ABICOIM

UXPCharacterSet

用途： ESC/POS 字符集。

```
typedef NS_ENUM(NSInteger, UXPCharacterSet) {
    UXPCharacterSetUSA = 0,
```

```
UXPCharacterSetFrance = 1,  
UXPCharacterSetGermany = 2,  
UXPCharacterSetUK = 3,  
UXPCharacterSetDenmarkI = 4,  
UXPCharacterSetSweden = 5,  
UXPCharacterSetItaly = 6,  
UXPCharacterSetSpainI = 7,  
UXPCharacterSetJapan = 8,  
UXPCharacterSetNorway = 9,  
UXPCharacterSetDenmarkII = 10,  
UXPCharacterSetSpainII = 11,  
UXPCharacterSetLatinAmerica = 12,  
UXPCharacterSetKorea = 13,  
UXPCharacterSetSloveniaCroatia = 14,  
UXPCharacterSetChina = 15,  
};
```

值名	值	说明
UXPCharacterSetUSA	0	USA
UXPCharacterSetFrance	1	法国
UXPCharacterSetGermany	2	德国
UXPCharacterSetUK	3	英国
UXPCharacterSetDenmarkI	4	丹麦 I
UXPCharacterSetSweden	5	瑞典
UXPCharacterSetItaly	6	意大利
UXPCharacterSetSpainI	7	西班牙 I
UXPCharacterSetJapan	8	日本
UXPCharacterSetNorway	9	挪威
UXPCharacterSetDenmarkII	10	丹麦 II
UXPCharacterSetSpainII	11	西班牙 II
UXPCharacterSetLatinAmerica	12	拉丁美洲
UXPCharacterSetKorea	13	韩国
UXPCharacterSetSloveniaCroatia	14	斯洛文尼亚/克罗地亚
UXPCharacterSetChina	15	中国

UXPErrorCode

用途： SDK 统一错误码。

```
typedef NS_ENUM(NSInteger, UXPErrorCode) {
    UXPErrorCodeParams = 1,
    UXPErrorCodeUDPConfigEthDhcp = 1000,
    UXPErrorCodeUDPConfigEthIp = 1001,
    UXPErrorCodeUDPConfigEthMask = 1002,
    UXPErrorCodeUDPConfigEthGateway = 1003,
```

```

UXPErrorCodeUDPCConfigWiFiMode = 1004,
UXPErrorCodeUDPCConfigEncryption = 1005,
UXPErrorCodeUDPCConfigApName = 1006,
UXPErrorCodeUDPCConfigApPwd = 1007,
UXPErrorCodeUDPCConfigStaName = 1008,
UXPErrorCodeUDPCConfigStaPwd = 1009,
UXPErrorCodeUDPCConfigApIp = 1010,
UXPErrorCodeUDPCConfigStaIp = 1011,
UXPErrorCodeUDPCConfigWiFiDhcp = 1012,
UXPErrorCodeUDPCConfigWiFiGateway = 1013,
UXPErrorCodeUDPCConfigWiFiMask = 1014,
UXPErrorCodeResponseResolve = 2001,
UXPErrorCodeOtaGetMaxPackageSize = 2002,
UXPErrorCodeOtaRetryTooMuch = 2003,
UXPErrorCodeOtaStatusLengthError = 3001,
UXPErrorCodeOtaStatusCrcError = 3002,
UXPErrorCodeOtaStatusTimeOut = 3003,
UXPErrorCodeOtaStatusWriteFail = 3004,
UXPErrorCodeOtaStatusFail = 3005,
UXPErrorCodeESCPrintFail = 4000,
UXPErrorCodeESCPrintTimeout = 4001,
};

```

值名	值	说明
UXPErrorCodeParams	1	参数错误
UXPErrorCodeUDPCConfigEthDhcp	1000	以太网 DHCP 配置失败
UXPErrorCodeUDPCConfigEthIp	1001	以太网 IP 配置失败
UXPErrorCodeUDPCConfigEthMask	1002	以太网掩码配置失败
UXPErrorCodeUDPCConfigEthGateway	1003	以太网网关配置失败
UXPErrorCodeUDPCConfigWiFiMode	1004	Wi-Fi 模式配置失败
UXPErrorCodeUDPCConfigEncryption	1005	加密方式配置失败
UXPErrorCodeUDPCConfigApName	1006	AP 名称配置失败

UXPErrorCodeUDPConfigApPwd	1007	AP 密码配置失败
UXPErrorCodeUDPConfigStaName	1008	STA 名称配置失败
UXPErrorCodeUDPConfigStaPwd	1009	STA 密码配置失败
UXPErrorCodeUDPConfigApIp	1010	AP IP 配置失败
UXPErrorCodeUDPConfigStaIp	1011	STA IP 配置失败
UXPErrorCodeUDPConfigWiFiDhcp	1012	Wi-Fi DHCP 配置失败
UXPErrorCodeUDPConfigWiFiGateway	1013	Wi-Fi 网关配置失败
UXPErrorCodeUDPConfigWiFiMask	1014	Wi-Fi 掩码配置失败
UXPErrorCodeResponseResolve	2001	响应解析失败
UXPErrorCodeOtaGetMaxPackageSize	2002	获取 OTA 最大包大小失败
UXPErrorCodeOtaRetryTooMuch	2003	OTA 重发次数过多
UXPErrorCodeOtaStatusLengthError	3001	OTA 长度错误
UXPErrorCodeOtaStatusCrcError	3002	OTA CRC 错误
UXPErrorCodeOtaStatusTimeOut	3003	OTA 超时
UXPErrorCodeOtaStatusWriteFail	3004	OTA 写入失败
UXPErrorCodeOtaStatusFail	3005	OTA 执行失败
UXPErrorCodeESCPrintFail	4000	ESC 打印失败
UXPErrorCodeESCPrintTimeout	4001	ESC 打印超时

UXPEscBarcodeType

用途： ESC/POS 条码类型。

```
typedef NS_ENUM(NSInteger, UXPEscBarcodeType) {
    UXPEscBarcodeTypeUPCA = 65,
    UXPEscBarcodeTypeUPCE = 66,
    UXPEscBarcodeTypeEAN13 = 67,
    UXPEscBarcodeTypeEAN8 = 68,
    UXPEscBarcodeTypeCode39 = 69,
    UXPEscBarcodeTypeITF = 70,
    UXPEscBarcodeTypeCodabar = 71,
    UXPEscBarcodeTypeCode128 = 73,
};
```

值名	值	说明
UXPEscBarcodeTypeUPCA	65	UPC-A
UXPEscBarcodeTypeUPCE	66	UPC-E
UXPEscBarcodeTypeEAN13	67	EAN-13
UXPEscBarcodeTypeEAN8	68	EAN-8
UXPEscBarcodeTypeCode39	69	Code 39
UXPEscBarcodeTypeITF	70	ITF
UXPEscBarcodeTypeCodabar	71	Codabar
UXPEscBarcodeTypeCode128	73	Code 128

UXPEscQRCodeCorrectionLevel

用途： ESC/POS 二维码纠错等级。

```
typedef NS_ENUM(NSInteger, UXPEscQRCodeCorrectionLevel) {
    UXPEscQRCodeCorrectionLevelL = 48,
    UXPEscQRCodeCorrectionLevelM = 49,
    UXPEscQRCodeCorrectionLevelQ = 50,
    UXPEscQRCodeCorrectionLevelH = 51,
};
```

值名	值	说明
UXPEscQRCodeCorrectionLevelL	48	L 级
UXPEscQRCodeCorrectionLevelM	49	M 级
UXPEscQRCodeCorrectionLevelQ	50	Q 级
UXPEscQRCodeCorrectionLevelH	51	H 级

UXPEscPageModeDirection

用途：ESC/POS 页模式方向。

```
typedef NS_ENUM(NSInteger, UXPEscPageModeDirection) {
    UXPEscPageModeDirectionLeftToRight = 0,
    UXPEscPageModeDirectionBottomToTop = 1,
    UXPEscPageModeDirectionRightToLeft = 2,
    UXPEscPageModeDirectionTopToBottom = 3,
};
```

值名	值	说明
UXPEscPageModeDirectionLeftToRight	0	从左到右
UXPEscPageModeDirectionBottomToTop	1	从下到上
UXPEscPageModeDirectionRightToLeft	2	从右到左
UXPEscPageModeDirectionTopToBottom	3	从上到下

UXPTsplQRCodeEccLevel

用途： TSPL 二维码纠错等级（字符串枚举）。

```
typedef NSString * UXPTsplQRCodeEccLevel NS_STRING_ENUM;
```

常量名	字符串值	说明
UXPTsplQRCodeEccLevelL	@ "L"	L 级
UXPTsplQRCodeEccLevelM	@ "M"	M 级
UXPTsplQRCodeEccLevelQ	@ "Q"	Q 级
UXPTsplQRCodeEccLevelH	@ "H"	H 级

Typedef Blocks（类型定义回调）

UXPEscPrinterStatusResultBlock

用途： ESC/POS 打印机状态结果回调。

```
typedef void (^UXPEscPrinterStatusResultBlock)(UXPEscPrinterStatus status, NSError * _Nullable error);
```

UXPTsplPrinterStatusResultBlock

用途： TSPL 打印机状态结果回调。

```
typedef void (^UXPTsplPrinterStatusResultBlock)(UXPTsplPrinterStatus status, NSError * _Nullable error);
```

UXPCpclPrinterStatusResultBlock

用途： CPCL 打印机状态结果回调。

```
typedef void (^UXPCpclPrinterStatusResultBlock)(UXPCpclPrinterStatus
status, NSError * _Nullable error);
```

UXPZplPrinterStatusResultBlock

用途： ZPL 打印机状态结果回调。

```
typedef void (^UXPZplPrinterStatusResultBlock)(UXPZplPrinterStatus
status, NSError * _Nullable error);
```

UXPPrintTaskResultBlock

用途： 打印任务结果回调。

```
typedef void (^UXPPrintTaskResultBlock)(UXPPrintTaskResult * _Nullable
result, NSError * _Nullable error);
```

UXPCodePageResultBlock

用途： 代码页查询结果回调。

```
typedef void (^UXPCodePageResultBlock)(UXPCodePage codePage, NSError *
_Nullable error);
```

UXPCharacterSetResultBlock

用途： 字符集查询结果回调。

```
typedef void (^UXPCharacterSetResultBlock)(UXPCharacterSet
characterSet, NSError * _Nullable error);
```

UXPEscMissPrintEventHandler

用途： ESC/POS 漏打事件回调。回调返回设备侧上报的原始事件数据；若轮询或解析失败，`error` 会携带具体错误。

```
typedef void (^UXPEscMissPrintEventHandler)(NSData * _Nullable  
eventData, NSError * _Nullable error);
```