

Class

XUSBManager

The `XUSBManager` class handles USB communication and provides callback blocks for printer-related statuses and operations.

```
import Foundation
import IOKit.usb
```

XUSBManagerDelegate

Delegate Methods

xUSBConnected(_:)

USB Connection Successful

```
func xUSBConnected(_ device: XUSBDeviceInfo)
```

- **device**
The connected USB device information.

xUSBDisconnectWithError(_:message:device:)

USB Disconnection with Error

```
func xUSBDisconnectWithError(_ code: Int, message: String, device: XUSBDeviceInfo)
```

- **code**
The error code indicating the reason for disconnection.
- **message**
The error message describing the disconnection.
- **device**
The disconnected USB device information.

xUSBWriteValueWithDevice(_:isSuccess:message:)

USB Data Write Callback

```
func xUSBWriteValueWithDevice(_ device: XUSBDeviceInfo, isSuccess: Bool, message: String)
```

- **device**
The USB device information.

- **isSuccess**
Indicates whether the data write operation was successful.
- **message**
Additional message related to the write operation.

xUSBReceiveValueForData(_:device:)

USB Data Receive Callback

```
func xUSBReceiveValueForData(_ data: Data, device: XUSBDeviceInfo)
```

- **data**
The received data from the USB device.
- **device**
The USB device information.

xUSBDidSearch(_:device:)

USB Device Search Callback

```
func xUSBDidSearch(_ state: Int, device: XUSBDeviceInfo)
```

- **state**
The search state: `0` for offline, `1` for online.
- **device**
The USB device information.

Properties

USB Communication Class

```
class XUSBManager: NSObject
```

- **searchBlock**
The callback block called when USB device search results are available.

```
var searchBlock: UsbDidSearch?
```

- **delegate**
The delegate that receives USB manager events.

```
weak var delegate: XUSBManagerDelegate?
```

- **statusPOSBLOCK**
The callback block called when reporting POS printer status.

```
var statusPOSBlock: XUSBPOSPrinterStatusBlock?
```

- **statusLabelBlock**

The callback block called when reporting label printer status.

```
var statusLabelBlock: XUSBLabelPrinterStatusBlock?
```

- **snBlock**

The callback block called when reporting printer serial number.

```
var snBlock: XUSBPrinterSNBlock?
```

- **cashBoxBlock**

The callback block called when reporting cash box status.

```
var cashBoxBlock: XUSBCashBoxBlock?
```

Methods

Singleton Instance

```
static let shared = XUSBManager()
```

- Returns the singleton instance of `XUSBManager`.

Remove a Delegate Object

```
func removeDelegate(_ delegate: XUSBManagerDelegate)
```

- **delegate**

The delegate object to be removed.

Remove All Delegate Objects

```
func removeAllDelegates()
```

Connect to a Specified USB Device

```
func connectDevice(_ device: XUSBDeviceInfo)
```

- **device**

The USB device information to connect to.

Connect USB Device by Name

```
func connectDeviceName(_ deviceName: String)
```

- **deviceName**

The name of the USB device to connect to.

Send Data

```
func sendData(_ data: Data)
```

- **data**

The data to be sent to the USB device.

Listen for USB Search Events

```
func listenUsbSearchEvent(_ searchResult: @escaping UsbDidSearch)
```

- **searchResult**

The callback block to handle USB device search results.

Disconnect a Specified USB Device

```
func disconnect() -> IOReturn
```

- Disconnects the currently connected USB device.

Search Devices

```
func searchDevices() -> [XUSBDeviceInfo]
```

- Returns an array of discovered USB devices.

Printer Status (for Receipt Printer)

```
func printerPOSStatus(_ statusBlock: @escaping XUSBPOSPrinterStatusBlock)
```

- **statusBlock**

The callback block to handle POS printer status updates.

Printer Status (for Label Printer)

```
func printerLabelStatus(_ statusBlock: @escaping XUSBLabelPrinterStatusBlock)
```

- **statusBlock**
The callback block to handle label printer status updates.

Printer Serial Number

```
func printerSN(_ snBlock: @escaping XUSBPrinterSNBlock)
```

- **snBlock**
The callback block to handle printer serial number updates.

Cash Box Status

```
func cashBoxCheck(_ cashBoxBlock: @escaping XUSBCashBoxBlock)
```

- **cashBoxBlock**
The callback block to handle cash box status updates.

XUSBDeviceInfo

Properties

- **vid**
The Vendor ID of the USB device.

```
var vid: UInt16
```

- **pid**
The Product ID of the USB device.

```
var pid: UInt16
```

- **locationId**
The Location ID of the USB device.

```
var locationId: String
```

- **deviceName**
The name of the USB device.

```
var deviceName: String
```

- **interface**

The USB interface for the device.

```
var interface: UnsafeMutablePointer<IOUSBInterfaceInterface245>?
```

- **dev**

The USB device interface.

```
var dev: UnsafeMutablePointer<IOUSBDeviceInterface245>?
```

- **pipeIn**

The input pipe for data transfer.

```
var pipeIn: UInt8
```

- **pipeOut**

The output pipe for data transfer.

```
var pipeOut: UInt8
```

USBConnectionError Enum

Error Codes

```
enum USBConnectionError: Int {  
    case masterPortCreationFailed = 1          ///< Failed to create master port.  
    case matchingDictionaryCreationFailed      ///< Failed to create matching  
dictionary.  
    case vendorIDCreationFailed                ///< Failed to create Vendor ID.  
    case productIDCreationFailed               ///< Failed to create Product ID.  
    case devicePluginCreationFailed            ///< Failed to create device plugin.  
    case deviceInterfaceCreationFailed          ///< Failed to create device interface.  
    case deviceOpenFailed                      ///< Failed to open device.  
    case configurationCountFailed              ///< Failed to get configuration count.  
    case configDescriptorFailed                ///< Failed to get configuration  
descriptor.  
    case setConfigurationFailed                ///< Failed to set configuration.  
    case interfaceIteratorCreationFailed        ///< Failed to create interface  
iterator.  
    case interfacePluginCreationFailed          ///< Failed to create interface plugin.  
    case interfaceCreationFailed               ///< Failed to create interface.  
    case interfaceOpenFailed                   ///< Failed to open interface.  
    case endpointCountFailed                   ///< Failed to get endpoint count.  
}
```

