

Class

XWIFIManager

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

```
#import <Foundation/Foundation.h>
#import <Cocoa/Cocoa.h>
#import "NetProfile.h"
@class XWIFIManager;
```

XWIFIManagerDelegate

Delegate Methods

xWifiConnectedToHost:port:

Callback for successful connection

```
- (void)xWifiConnectedToHost:(NSString *)host port:(UInt16)port;
```

- **host**
The host address of the connected device.
- **port**
The port number used for the connection.

xWifiDisconnectWithError:host:

Callback for disconnection error

```
- (void)xWifiDisconnectWithError:(NSError *)error host:(NSString *)host;
```

- **error**
The error information related to the disconnection.
- **host**
The host address of the connected device.

xWifiWriteValueWithTag:host:

Callback for successful data transmission

```
- (void)xWifiWriteValueWithTag:(long)tag host:(NSString *)host;
```

- **tag**
The tag value associated with the data transmission.
- **host**
The host address of the connected device.

xWifiReceiveValueForData:host:

Callback for receiving data from the printer

```
- (void)xWifiReceiveValueForData:(NSData *)data host:(NSString *)host;
```

- **data**
The received data.
- **host**
The host address of the connected device.

Properties

Network Communication Class

```
@interface XWIFIManager : NSObject
```

- **hostStr**
The host address string of the printer.

```
@property (nonatomic, copy) NSString *hostStr;
```

- **port**
The port number used for communication.

```
@property (nonatomic, assign) UInt16 port;
```

- **isConnect**
The connection status of the printer.

```
@property (nonatomic, assign) BOOL isConnect;
```

- **delegate**
The delegate to receive WiFi manager events.

```
@property (nonatomic, weak) id<XWIFIManagerDelegate> delegate;
```

- **receiveBlock**

The callback block called when data is received.

```
@property (nonatomic, copy) XWIFIManagerReceiveBlock receiveBlock;
```

- **writeBlock**

The callback block called when data is written.

```
@property (nonatomic, copy) XWIFIManagerWriteBlock writeBlock;
```

- **statusPOSBlock**

The callback block called when reporting POS printer status.

```
@property (nonatomic, copy) XWIFIPOSPrinterStatusBlock statusPOSBlock;
```

- **statusLabelBlock**

The callback block called when reporting label printer status.

```
@property (nonatomic, copy) XWIFILabelPrinterStatusBlock statusLabelBlock;
```

- **snBlock**

The callback block called when reporting the printer's serial number.

```
@property (nonatomic, copy) XWIFIPrinterSNBlock snBlock;
```

- **checkPOSBlock**

The callback block called when checking POS printer status.

```
@property (nonatomic, copy) XWIFIPOSPrinterCheckBlock checkPOSBlock;
```

- **foundPrinterBlock**

The callback block called when a printer is discovered.

```
@property (nonatomic, copy) XWIFIManagerFoundPrinterBlock foundPrinterBlock;
```

- **cashBoxBlock**

The callback block called when reporting cash drawer status.

```
@property (nonatomic, copy) XWIFICashBoxBlock cashBoxBlock;
```

- **connectedPrinter**

The configuration profile of the connected printer.

```
@property (nonatomic, strong) NetProfile *connectedPrinter;
```

Methods

Singleton Object

```
+ (instancetype)sharedInstance;
```

Remove a Delegate Object

```
- (void)removeDelegate:(id<XWIFIManagerDelegate>)delegate;
```

- **delegate**
The delegate object to be removed.

Remove All Delegate Objects

```
- (void)removeAllDelegates;
```

Connect to the Printer Address

```
- (void)connectWithHost:(NSString *)hostStr port:(UInt16)port;
```

- **hostStr**
The host address of the printer.
- **port**
The port number to connect to.

Disconnect from the Printer

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

Write Commands to the Printer

```
- (void)writeCommandWithData:(NSData *)data;
```

- **data**
The command data to be sent.

Write Commands with a Receive Callback

```
- (void)writeCommandWithData:(NSData *)data receiveCallBack:
(XWIFIManagerReceiveBlock)receiveBlock;
```

- **data**
The command data to be sent.
- **receiveBlock**
The callback block when data is received.

Write Commands with a Write Completion Callback

```
- (void)writeCommandWithData:(NSData *)data writeCallBack:
(XWIFIManagerWriteBlock)writeBlock;
```

- **data**
The command data to be sent.
- **writeBlock**
The callback block when the write operation is complete.

Create a UDP Socket

```
- (BOOL)createUdpSocket;
```

- **return**
Whether the socket creation was successful.

Close the UDP Socket

```
- (void)closeUdpSocket;
```

Search for Printers

```
- (void)sendFindCmd:(XWIFIManagerFoundPrinterBlock)foundPrinterBlock;
```

- **foundPrinterBlock**
A callback block that is triggered when a printer is found. It returns a `NetProfile` object containing the following printer network configuration details:
 - `printerName`: The name of the printer. The default is the printer's MAC address.
 - `printerDesc`: A description of the printer, typically used to provide basic information about the printer.

- `modelName`: The printer's model name. Some printers may not provide this, so it might be `nil` or unset.
- `MAC`: The printer's MAC address, used to uniquely identify the device on the network.
- `IP`: The printer's IP address, used for network communication to locate the device.
- `mask`: The printer's subnet mask, used for defining the network's range.
- `gateway`: The printer's gateway address, used for routing network traffic.
- `DHCP`: The printer's DHCP setting, indicating whether dynamic IP assignment is enabled. `1` means enabled, `0` means disabled.

Set IP Configuration

```
- (void)setIPConfigWithIP:(NSString *)ip Mask:(NSString *)mask Gateway:(NSString *)gateway DHCP:(BOOL)dhcp;
```

- **ip**
The IP address of the printer.
- **mask**
The subnet mask.
- **gateway**
The gateway address.
- **dhcp**
Whether to enable DHCP.

Set WiFi Configuration

```
- (void)setWiFiConfigWithIP:(NSString *)ip mask:(NSString *)mask gateway:(NSString *)gateway ssid:(NSString *)ssid password:(NSString *)password encrypt:(NSInteger)encrypt;
```

- **ip**
The IP address of the printer.
- **mask**
The subnet mask.
- **gateway**
The gateway address.
- **ssid**
The WiFi SSID.
- **password**
The WiFi password.
- **encrypt**
The encryption method.

Check Printer Connection Status

```
- (BOOL)printerIsConnect;
```

- **return**

Whether the printer is connected.

Get Printer Serial Number

```
- (void)printerSN:(XWIFIPrinterSNBlock)snBlock;
```

- **snBlock**

The callback block for the serial number.

Check Printer Status (POS Printers)

```
- (void)printerPOSStatus:(XWIFIPosPrinterStatusBlock)statusBlock;
```

- **statusBlock**

The callback block for the POS printer status.

Check Printer Status (Label Printers)

```
- (void)printerLabelStatus:(XWIFILabelPrinterStatusBlock)statusBlock;
```

- **statusBlock**

The callback block for the label printer status.

Query Printer Status

```
- (void)printerPOSCheck:(int)type checkBlock:(XWIFIPosPrinterCheckBlock)checkBlock;
```

- **type**

The query type (1: common status, 2: offline status, 3: error status, 4: paper feed status).

- **checkBlock**

The callback block for the query result.

Check Cash Drawer Status

```
- (void)cashBoxCheck:(XWIFICashBoxBlock)cashBoxBlock;
```

- **cashBoxBlock**

The callback block for the cash drawer status.

Get Copyright Information

```
+ (NSString *)GetCopyright;
```

- **return**

The copyright information string.