

**Nemonic**

# **Printer SDK commands**

(iOS)

V 1.0.9

MANGOSLAB Co.,Ltd.

## History

Ver	내용	비고
1.0.0	Draft	
1.0.1	Fixed bug about getPrinterStatus on printing	
1.0.2	Fixed bug about image data processing	
1.0.3	Slightly improved printing speed for Nemonic AI printer.	
1.0.4	Add print quality printing options.	
1.0.5	Improve disconnected on BLE	
1.0.6	Fixed bug about print result for Nemonic AI printer.	
1.0.7	Fixed bug about print complete result.	
1.0.8	Add manual print timeout. Change auto print timeout for Nemonic AI.	
1.0.9	Improved Bluetooth classic connectivity (MIP-001, MIP-001L, MIP-101)	

## **1. Summary**

This document describes the printer control SDK for Nemonic printers.

## **Scan printer(NPrinterScanController)**

### Basic

1. startScan() -> Int

Start scan.

return: Start scan result (NResult).

2. stopScan()

Stop scan.

3. isInScan() -> Bool

Whether scanning.

return: Whether scanning.

### Callback(NPrinterScanControllerDelegate)

1. deviceFound(printer: NPrinter)

Called when a printer is found.

printer: Printer information found.

## Control printer(NPrinterController)

### Basic

1. connect(printer: NPrinter) -> Int

Connect the printer selecting Bluetooth mode automatically depending on the printer type.

printer: Printer to connect to.

return: Connect result (NResult).

2. disconnect()

Disconnect printer.

3. getConnectState() -> Int

Get connection state.

return Connection state.

4. cancel()

Cancel image transfer or template settings.

5. setPrintTimeout(\_ enableAuto: Bool, \_ manualTime: Int)

Set print timeout.

enableAuto: Whether to enable automatic timeout.

manualTime: Manual timeout time.(Applies only when enableAuto is false.)

6. print(\_ printInfo: NPrintInfo) -> Int

Print images.

If the width is not 576 pixels, it will be resized.

And after resizing, there are height restrictions for each printer model as shown below.

- Nemonic(MIP-001): 2240
- Nemonic Label(MIP-001L): 2240
- Nemonic mini(MIP-101): 2240
- Nemonic AI(MIP-201): 2000

printInfo: Information for printing.

return: Print result (NResult & NPrinterStatus).

7. setTemplate(\_ image: UIImage, withPrint: Bool, enableDither: Bool) -> Int

Set a template for the printer.

If the width is not 576 pixels, it will be resized.

And after resizing, there are height restrictions for each printer model as shown below.

- Nemonic(MIP-001): 900
- Nemonic Label(MIP-001L): 900

- Nemonic mini(MIP-101): 900
- Nemonic AI(MIP-201): 2000

image: Image to set as template.

withPrint: Whether to print after setting a template.

enableDither: Whether to process image dither.

return: Template setting result (NResult & NPrinterStatus).

#### 8. clearTemplate() -> Int

Initializing the template of the printer.

return: Template initialization result (NResult).

#### 9. getPrinterStatus() -> Int

Get printer status.

return: Printer status (NPrinterStatus or NResult(<0)).

#### 10. getCartridgeType() -> Int

Get cartridge type.

return Cartridge type (NCartridgeType or NResult(<0)).

#### 11. getPrinterName() -> NResultString

Get printer Bluetooth name.

return: Printer Bluetooth name(value: printer Bluetooth name,  
result: NResult).

12. getBatteryLevel() -> Int

Get battery level.

return: Battery level (level or NResult(<0)).

13. getBatteryStatus() -> Int

Get battery status.

return: Battery status (status or Nresult(<0)).

14. getDefaultConnectDelay() -> UInt32

Get default delay after connecting.(Only Nemonic AI printer)

return: Default delay after connecting (milly second).

15. getConnectDelay() -> UInt32

Get delay after connecting.(Only Nemonic AI printer)

return: Delay after connecting (milly second).



#### 16. setConnectDelay(\_ delay: UInt32)

Set delay after connecting.(Only Nemonic AI printer)

As the time decreases, the faster the connection speed.

As the time increases, the stability of the connection increases.

delay: Delay after connecting (milly second).

#### Callback(NPrinterControllerDelegate)

##### 1. disconnected()

Called when the printer is disconnected due to external factors.

##### 2. printProgress(index: Int, total: Int, result: Int)

Called to notify the progress of each print image when printing multiple images.

index: Current print completion index (start from 0).

total: Total number of printing images.

result: Print result of the currently printed index.

##### 3. printComplete(result: Int)

Called when all printing is complete.

result: Print result.

## Object

### NPrinter

#### 1. isEmpty() -> Bool

Whether the printer information is empty.

return: Whether the printer information is empty.

#### 2. isValid() -> Bool

Whether the printer information is a Nemonic product.

return: Whether the printer information is a Nemonic product.

#### 3. reset()

Reset printer information.

#### 4. getName() -> String

Get printer Bluetooth name.

return: Printer Bluetooth name.

#### 5. setName(\_ name: String)

Set printer Bluetooth name.

name: Printer Bluetooth name.([A-Za-z0-9]{4,20}(MIP201:  
[A-Za-z0-9]{4,13}))

6. checkName(\_ name: String) -> Bool

Check printer Bluetooth name validation.

name: Printer Bluetooth name.

return Printer Bluetooth name valid.

7. static checkName(type: NPrinterType, name: String) -> Bool

Check printer Bluetooth name validation.

type: Printer type.

name: Printer Bluetooth name.

return: Printer Bluetooth name valid.

8. static checkNameWithoutCartridgeType(type: NPrinterType, name: String) -> Bool

Check printer Bluetooth name without cartridge type(using setting printer name).

type: Printer type.

name: Printer Bluetooth name.

return: Printer Bluetooth name valid.

9. getAddress() -> String

Get Mac address.

return: Mac address.

10. setAddress(\_ address: String)-> Bool

Set Mac address.

address: Mac address.

return: Mac address valid.

11. checkAddress(\_ address: String) -> Bool

Check Mac address valid.

address: Mac address.

return Mac address valid.

12. getType() -> NPrinterType

Get printer type.

return: Printer type.

13. isLabel() -> Bool

Label printer or not(It does not refer to a Nemonic Label product(MIP-001L)).

return: Label printer or not.(When the result is false, it does not mean that it is not a Nemonic Label product (MIP-001L)).

14. isMini() -> Bool

Whether the printer is Nemonic mini printer.

return: Whether the printer is Nemonic mini printer.

15. isFixedPaperSize() -> Bool

Whether it is a fixed length cartridge type.

return: Whether it is a fixed length cartridge type.

16. isSupportedBattery() -> Bool

Whether the printer supports batteries.

return: Whether the printer supports batteries.

17. setCartridgeType(\_ type: NCartridgeType)

Set cartridge type.

type: Cartridge type.

18. getCartridgeType() -> NCartridgeType

Get cartridge type (based on printer name).

return Cartridge type.

19. getWaitTimeForPrint(int height, int copies) -> Int

Get print time.

height: Print image height.

copies: Print copies.

return: Print time (msec).

20. isEqual(\_ object: Any?) -> Bool

Whether it is the same printer (based on Mac address).

object: Other NPrinter object for comparison.

## NPrintInfo

1. isEmpty() -> Bool

Whether the images for printing are empty.

return: Whether the images for printing are empty.

2. `setPrinter(_ printer: NPrinter) -> NPrintInfo`

Set the printer for generating print data.

printer: The printer for creating print images (Connected or connectable printer).

return: Object itself.

3. `getPrinter() -> NPrinter`

Get the printer for generating print data.

return: The printer for generating print data.

4. `setPrintQuality(_ quality: NPrintQuality) -> NPrintInfo`

Set the print quality. (Only Nemonic AI firmware 1.03 or later.)

quality: Print quality.

return: Object itself.

5. `getPrintQuality() -> NPrintQuality`

Get the print quality.

return: Print quality.

6. setImage(\_ image: UIImage) -> NPrintInfo

Set the image for printing.

image: The image for printing.

return: Object itself.

7. getImage() -> UIImage?

Get the image for printing.

return: The image for printing.

8. setImages(\_ images: [UIImage]) -> NPrintInfo

Set the images for printing.

images: The images array for printing.

return: Object itself.

9. getImages() -> [UIImage]

Get the images for printing.

images: The images list for printing.



return: Object itself.

10. `setCopies(_ copies: Int) -> NPrintInfo`

Set copies.

`copies`: Copy quantity.

return: Object itself.

11. `getCopies() -> Int`

Get copies.

return: Copy quantity.

12. `setEnabledLastPageCut(_ enable: Bool) -> NPrintInfo`

Set enable last page cut.

`enable`: Enabled or not.

return Object itself.

13. `isLastPageCutEnable() -> Bool`

Whether to cut the last page.

return: Whether to cut the last page.

14. `setEnabledDither(_ enable: Bool) -> NPrintInfo`

Set enable dither processing.

enable: Enabled or not.

return: Object itself.

15. `isEnabledDither() -> Bool`

Whether to enable print image dither processing.

return: Whether to enable print image dither processing.

16. `setEnabledCheckPrinterStatus(_ enable: Bool) -> NPrintInfo`

Set enable checking printer status.

enable: Enable or not.

return: Object itself.

17. `isCheckPrinterStatus() -> NPrintInfo`

Whether to check printer status when printing.

return: Whether to check printer status when printing.

18. `setEnabledCheckCartridgeType(_ enable: Bool) -> NPrintInfo`

Set enable checking cartridge type.

enable: Enabled or not.

return: Object itself.

19. `isCheckCartridgeType() -> Bool`

Whether to check the cartridge type when printing.

return: Whether to check the cartridge type when printing.

20. `setEnabledCheckPower(_ enable: Bool) -> NPrintInfo`

Set enable checking about power.

enable: Enabled or not.

return: Object itself.

21. `isCheckPower() -> Bool`

Whether to check battery when printing.

return: Whether to check battery when printing.

22. `getPrintImage() -> UIImage?`

Get the image for printing.

return: The image for printing.

23. `getPrintImages()` -> `[UIImage]?`

Get the images for printing.

return: The images for printing.

`NResultString`

1. `getResult()` -> `Int`

Get result

return: `Result (NResult)`.

2. `setResult(_ result: Int)`

Set result.

result: `Result (NResult)`.

3. `getValue()` -> `String`

Get value.

return: `Value`.

4. setValue(\_ value: String)

Set value.

value: Value.

## **Define**

### NBatteryStatus

1. noCharging

Normal battery level without charging.

2. lowNoCharging

Low battery level for printing without charging.

3. charging

Normal battery level with charging.

4. lowCharging

Low battery level with charging.

### NCartridgeType

1. none

None.

2. white

White sticky cartridge.

3. yellow

Yellow sticky cartridge.

4. green

Green sticky cartridge.

5. blue

Blue sticky cartridge.

6. pink

Pink sticky cartridge.

7. l1

3x1 label cartridge.

8. l2

3x2 label cartridge.

9. l3

3x3 label cartridge.

10. l4

3x4 label cartridge.

11. m1

3x1 Nemonic mini cartridge.

12. m2

3x2 Nemonic mini cartridge.

13. m3

3x3 Nemonic mini cartridge.

14. m4

3x4 Nemonic mini cartridge.

## NConnectState

1. disconnected

Disconnected state.

2. connecting

Connecting state.

3. connected

Connected state.

4. disconnecting

Disconnecting state.

## NPrinterStatus

1. ok

Ok.

2. outOfPaper

Out of paper.



3. coverOpened

Over opened.

4. overheat

Overheat.

5. paperJam

Paper jam.

## NPrinterType

1. none

None.

2. nemonic

Nemonic (MIP-001).

3. nemonicLabel

Nemonic Label (MIP-001L).

4. nemonicMini

Nemonic mini (MIP-101).

5. nemonicMIP201

Nemonic AI (MIP-201).

## NPrintQuality

### 1. lowFast

Low quality and fast speed.

### 2. middle

Middle quality and middle speed.

### 3. highSlow

High quality and slow speed.

## NResult

### 1. ok

Ok.

### 2. timeout

Timeout.

### 3. canceled

Canceled.

### 4. batteryLow

Battery low.

### 5. batteryNeedCharge

Battery needs charging.

6. paperNotMatched

Paper not matched.

7. bluetoothUnsupported

Bluetooth not supported.

8. bluetoothDisabled

Bluetooth disabled.

9. bluetoothNoPermission

No Bluetooth permission.

10. bluetoothResetting

Bluetooth resetting.(only iOS)

11. canceledOrBluetoothDisabled

Bluetooth connection canceled or disabled.(only iOS)

12. bluetoothUnknown

Bluetooth unknown.(only iOS)

13. locationNoPermission

No location permission.(only Android)

14. locationDisabled

Location disabled.(only Android)

15. scanFailed

Printer scan failed.

16. noSelectedPrinter

No selected printer.

17. notConnected

Not connected.

18. alreadyConnected

Already connected.

19. notFound

Bluetooth device not found.(only iOS)

20. notConnectable

Bluetooth is not connectable.(only iOS)

21. socketError

Socket error.(only Android)

22. connectError

Connect error.

23. connectFailed

Connect failed.

24. sessionError

Session error.

25. connectServiceNotFound

Connect service not found.

26. connectUnsupportedMode

Unsupported connect mode.

27. ioReceiveError

IO receive error.

28. ioSendError

IO send error.

29. sendFailed

Send failed.

30. unknown

Unknown.

31. invalidParameter

Invalid parameter.

32. notMatchedPrinterType

Not matched printer type.

33. noDelegate

No delegate.

34. notMatchedCommandResultFormat

Not matched printer command result format.

35. invalidPrinterName

Invalid printer name.

36. invalidPrinterResult

Invalid printer result.

37. printerResultFailed

Printer result failed.

38. unsupportedDevice

Unsupported device.