

BIXOLON®

API 레퍼런스 가이드

Xamarin SDK

Ver. 1.01

<http://www.bixolon.co.kr>

목차

1. 매뉴얼안내	5
2. 지원 장치	5
3. Xamarin SDK 설치	6
3-1 지원 플랫폼.....	6
3-1-1 iOS.....	6
3-1-2 Android.....	6
3-1-3 Windows	6
3-2 지원 통신.....	7
3-2-1 블루투스.....	7
3-2-2 와이파이 / 이더넷.....	7
3-3 SDK 설치 절차.....	8
3-3-1 Nuget 서버에서 SDK 설치	8
3-3-2 SDK 파일로 설치하기	9
4. IMPosDeviceFactory 레퍼런스.....	12
4-1 개요.....	12
4-2 메소드	12
4-2-1 createDevice	12
5. MPosControllerDevices 레퍼런스.....	13
5-1 개요.....	13
5-2 프로퍼티.....	13
5-2-1 dataEvent.....	13
5-2-2 statusUpdateEvent.....	13
5-2-3 DeviceID	14
5-2-4 IsOpen	14
5-2-5 CommandMode	14
5-2-6 InterfaceType	14
5-3 메소드	15
5-3-1 selectInterface.....	15
5-3-2 selectCommandMode	15
5-3-3 openService	16
5-3-4 closeService.....	16
5-3-5 directIO	16
5-3-6 setTransaction	17
6. MPosControllerPrinter 레퍼런스.....	18
6-1 개요.....	18
6-2 메소드	18
6-2-1 printText.....	18
6-2-2 printText.....	19
6-2-3 setCharacterSet.....	20
6-2-4 setInternationalCharacterSet	21
6-2-5 setPagemode.....	22
6-2-6 setPagemodePrintArea	22

6-2-7 setPagemodeDirection.....	23
6-2-8 setPagemodePosition	23
6-2-9 getFirmwareVersion	23
6-2-10 printBitmap.....	24
6-2-11 printBitmapFile	25
6-2-12 print1DBarcode	26
6-2-13 printQRCode	27
6-2-14 printPDF417	28
6-2-15 checkPrinterStatus	29
6-2-16 checkBattStatus	30
6-2-17 asbEnable	30
6-2-18 printLine	31
6-2-19 printBox.....	31
6-2-20 printPdfFile.....	32
6-2-21 cutPaper.....	33
7. MPosLookup 레퍼런스.....	34
7-1 개요.....	34
7-2 메소드	34
7-2-1 refreshDevicesList	34
7-2-2 getDevicesList.....	35
8. MPosControllerLabelPrinter 레퍼런스.....	36
8-1 개요.....	36
8-2 메소드	36
8-2-1 setTextEncoding	36
8-2-2 setCharacterSet.....	37
8-2-3 checkPrinterStatus	39
8-2-4 printBuffer	39
8-2-5 printRawData	40
8-2-6 drawTextDeviceFont	41
8-2-7 drawTextVectorFont.....	43
8-2-8 drawBarcode1D	44
8-2-9 drawBarcodeMaxiCode.....	45
8-2-10 drawBarcodePDF417	46
8-2-11 drawBarcodeQRCode	48
8-2-12 drawBarcodeDataMatrix.....	49
8-2-13 drawBarcodeAztec	50
8-2-14 drawBarcodeCode49	51
8-2-15 drawBarcodeCodaBlock.....	52
8-2-16 drawBarcodeMicroPDF	53
8-2-17 drawBarcodeIMB	54
8-2-18 drawBarcodeMSI.....	55
8-2-19 drawBarcodePlessey	56
8-2-20 drawBarcodeTLC39	57
8-2-21 drawBarcodeRSS.....	58
8-2-22 drawBlock	59
8-2-23 drawCircle	60
8-2-24 drawImage	61
8-2-25 drawImageFile	61
8-2-26 setPrintingType	62

8-2-27 setMargin	62
8-2-30 setWidth	64
8-2-32 setSpeed.....	64
8-2-33 setDensity	65
8-2-34 setOrientation.....	65
8-2-35 setOffset.....	65
8-2-36 setCuttingPosition	66
8-2-37 setAutoCutter	66
8-2-38 getModelName.....	67
8-2-39 getFirmwareVersion	67
9. 부록	68
9-1 에러 코드 표.....	68

1. 매뉴얼안내

이 SDK 매뉴얼은 Xamarin.Forms PCL(Portable Class Library) 응용 프로그램 개발에 필요한 내용을 제공합니다. Xamarin API 라이브러리 및 데모 응용프로그램은 Visual Studio 2015 기반으로 빌드 되었으므로, 이 매뉴얼은 모든 내용은 Visual Studio 2015 기반으로 설명하고 있습니다.

2. 지원 장치

아래 표는 지원되는 프린터 모델과 사양에 관하여 간략하게 요약한 것 입니다. Xamarin SDK에서 지원하는 아래 장치(프린터)에서 사용 가능합니다.

모델	해상도	지원 통신	최대 인쇄 너비(dot)
SPP-R200II	203 dpi	블루투스, 와이파이	384 dots
SPP-R200III	203 dpi		384 dots
SPP-R210	203 dpi		384 dots
SPP-R300	203 dpi		576 dots
SPP-R310	203 dpi		576 dots
SPP-R400	203 dpi		832 dots
SPP-R410	203 dpi		832 dots
SRP-350IIIOBE	180 dpi		블루투스, 이더넷
SRP-350III	180 dpi	이더넷	512 dots
SRP-352III	203 dpi		576 dots
SRP-330II	180 dpi		512 dots
SRP-332II	203 dpi		576 dots
SRP-350plusIII	180 dpi	블루투스, 와이파이, 이더넷	512 dots
SRP-352plusIII	203 dpi		576 dots
SRP-380	180 dpi		512 dots
SRP-382	203 dpi		576 dots
SRP-383	300 dpi		864 dots
SRP-Q300	180 dpi		512 dots
SRP-Q302	203 dpi		576 dots
SRP-F310II	180 dpi		512 dots
SRP-F312II	203 dpi		576 dots
SRP-F313II	203 dpi		640 dots
SRP-S300	203 dpi	576 dots	
SRP-QE300	180 dpi	이더넷	512 dots
SRP-QE302	203 dpi		576 dots
SRP-E300	180 dpi		512 dots
SRP-E302	203 dpi		576 dots

3. Xamarin SDK 설치

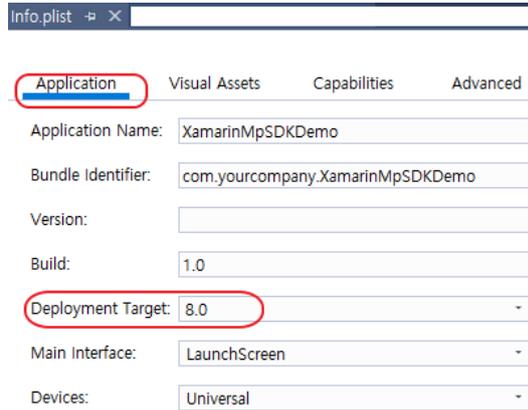
3-1 지원 플랫폼

3-1-1 iOS

iOS 8.0 또는 상위 버전

iOS용 Xamarin 프로젝트에서 개발 타겟 버전을 8.0으로 설정 하십시오.

참고

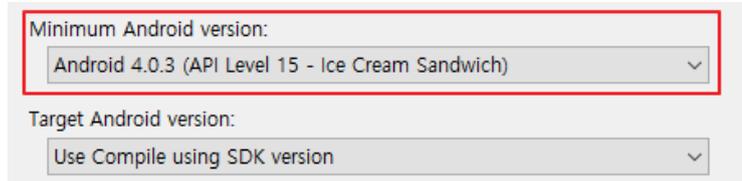


3-1-2 Android

Android 4.0.3(API Level 15: ice cream sandwich) 또는 상위 버전

Android용 Xamarin 프로젝트에서, 최소 Android 버전을 API 레벨 15로 설정하십시오.

참고



위의 그림과 같이 타겟 안드로이드 버전은 “User Compile using SDK version” 으로 설정해야 합니다. 그러면 자동으로 대상 프레임워크 설정과 일치하게 됩니다. 최소 Android 버전은 타겟 Android 버전 보다 높아야 합니다.

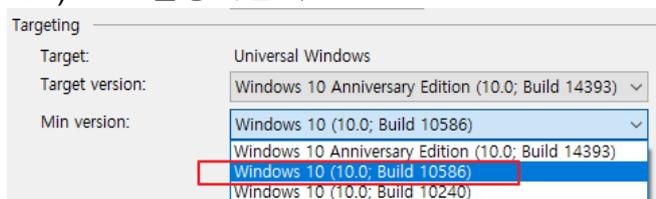
*** Minimum Android Version < Target Android Version < Target Framework.**

3-1-3 Windows

Windows 10 버전 빌드 10586 또는 상위 버전

Universal Windows용 Xamarin 프로젝트에서 최소 버전을 Windows 10 (10.0 Build 10586)으로 설정하십시오.

참고



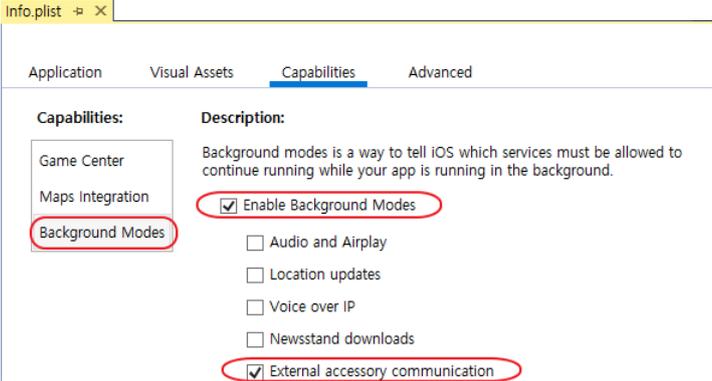
3-2 지원 통신

3-2-1 블루투스

블루투스 통신을 위해서는 블루투스 장치(프린터)와 HOST는 먼저 페어링이 되어야 합니다. 이 매뉴얼에 있는 API를 사용하기 전에, 필요한 권한(permission)를 추가해야 하며, 필요한 권한 설정은 아래 표를 참고하십시오.

3-2-2 와이파이 / 이더넷

HOST 및 장치(프린터)는 동일 유/무선 네트워크에서만 TCP/IP 통신이 가능합니다. 이 매뉴얼에 있는 API를 사용하기 전에, 필요한 권한(permission)를 추가해야 합니다. 필요한 권한 설정은 아래 표를 참고하십시오.

플랫폼	필요한 권한
iOS	<p>1. 블루투스 사용 권한을 획득하기 위하여, XML 편집기 또는 텍스트 편집기를 이용하여 "Info.plist" 파일에 아래 내용을 추가 하십시오.</p> <pre data-bbox="331 869 1442 943"><key>UISupportedExternalAccessoryProtocols</key> <array><string>com.bixolon.protocol</string></array></pre> <p>2. 아래 그림 처럼 "External Accessory communication" 에 체크하십시오.</p> 
Android	<p>"AndroidManifest.xml" 파일에 아래의 권한(permission)를 추가하십시오.</p> <ul data-bbox="331 1503 1310 1574" style="list-style-type: none"> - BLUETOOTH, BLUETOOTH_ADMIN, BLUETOOTH_PRIVILEGED - ACCESS_WIFI_STATE, CHANGE_WIFI_STATE <pre data-bbox="331 1581 1457 1850"><manifest xmlns:android="http://schemas.android.com/apk/res/android" android:installLocation="auto"> <uses-sdk android:minSdkVersion="15" /> <uses-permission android:name="android.permission.BLUETOOTH" /> <uses-permission android:name="android.permission.BLUETOOTH_ADMIN" /> <uses-permission android:name="android.permission.BLUETOOTH_PRIVILEGED" /> <uses-permission android:name="android.permission.ACCESS_WIFI_STATE" /> <uses-permission android:name="android.permission.CHANGE_WIFI_STATE" /> </manifest></pre>
Windows	<p>"Package.appxmanifest" 파일에 아래의 권한(capabilities)를 추가하십시오.</p> <ul data-bbox="331 1910 1238 1942" style="list-style-type: none"> - Bluetooth, Internet (Client), Private Networks (Client & Server)

참고

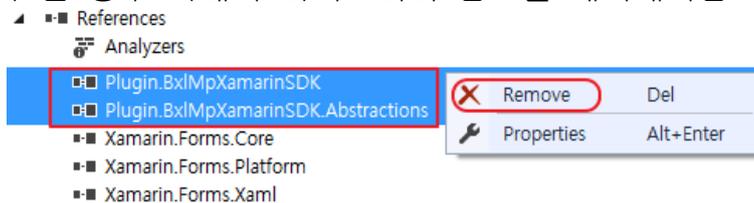
블루투스 및 와이파이 통신을 할 경우, 각 플랫폼에서 권한과 관련된 설정 정보를 프로젝트에 추가해야 합니다. 제공되는 데모 응용 프로그램을 참고하면 필요한 권한 설정을 변경하는 것을 도와줄 것입니다.

3-3 SDK 설치 절차

- Xamarin SDK는 MS Visual Studio에 NuGet 패키지로 설치할 수 있습니다. NuGet을 사용하여 패키지를 설치하면 라이브러리 파일을 솔루션에 복사하고 프로젝트를 자동으로 업데이트합니다 (참조 추가, 구성 파일 변경 등). 패키지를 제거하면 NuGet은 변경 사항을 이전 상태로 되돌릴 수 있습니다. NuGet에 대한 자세한 정보가 필요하면 <https://www.nuget.org/>를 방문하십시오.

참고

Bixolon Xamarin SDK 용 NuGet Package를 설치하기 전에 각 플랫폼에 수동으로 추가 한 경우 아래의 라이브러리 참조를 제거해야 합니다.



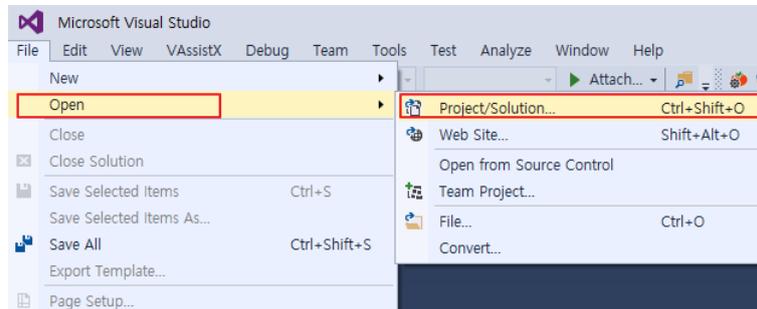
3-3-1 Nuget 서버에서 SDK 설치

이 설치 방법은 아직 지원되지 않습니다.

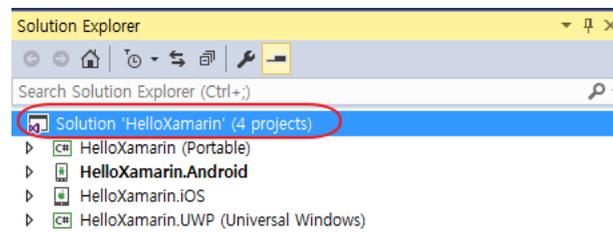
Xamarin SDK를 설치하려면이 아래 "3-3-2" 테이블을 참조하십시오.

3-3-2 SDK 파일로 설치하기

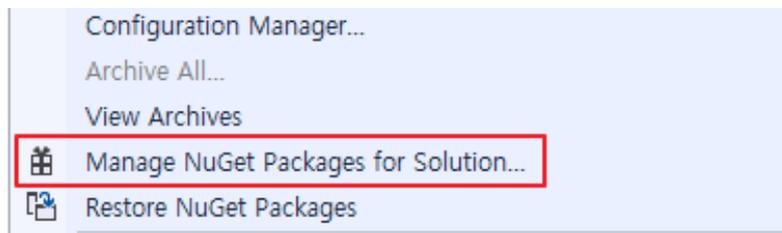
1) Visual Studio를 실행하고, 자신의 Xamarin.Forms 솔루션을 엽니다.



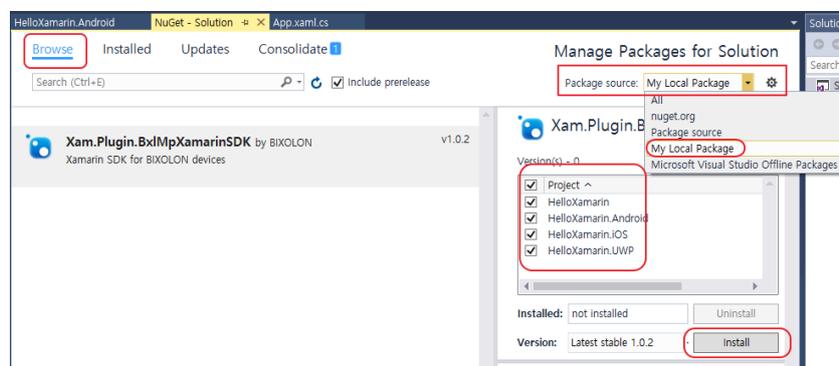
2) 솔루션 탐색기(Solution Explorer)에서 최상위 솔루션을 마우스 오른쪽 단추로 클릭하십시오.



3) " Manage NuGet Packages for Solution..."를 클릭하십시오.



4) 오른쪽 상단의 빨간색 "Setting" 버튼을 클릭하십시오.

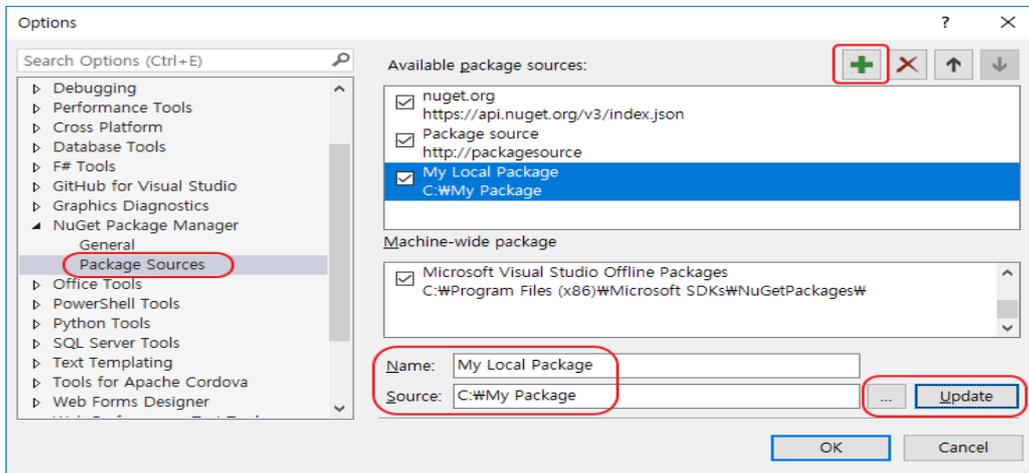


5) " Package Sources "를 클릭 한 다음 "+" 버튼을 클릭하십시오.

6) Xam.Plugin.BxlMpXamarinSDK.x.x.x.nupkg"파일이있는 디렉토리로 이동 한 다음 원하는 이름을 입력합니다.

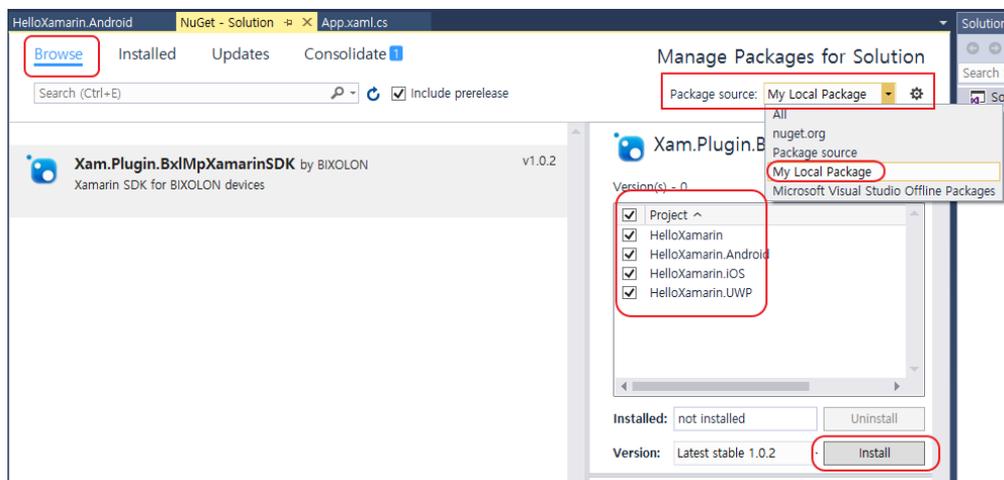
참고 "x.x.x" 는 SDK 버전 번호 입니다.

7) "Update" 버튼을 클릭 하십시오.

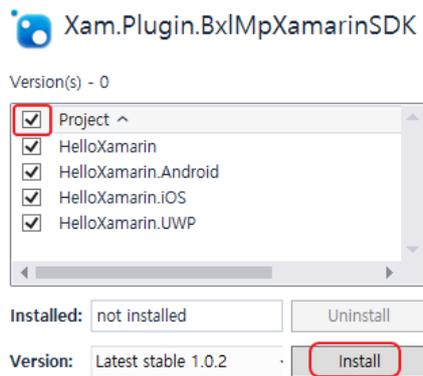


8) "OK" 버튼을 클릭 하십시오.

9) "Package source" 목록에서 6 단계에서 입력 한 이름을 선택하면, "Xam.Plugin.BxlMpXamarinSDK"패키지를 찾을 수 있습니다.



10) 전체 패키지를 설치하기 위해 "**Project**" 체크 박스를 클릭하십시오.



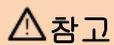
11) "**Install**" 버튼을 클릭하여 각 플랫폼 프로젝트에 대한 라이브러리 참조 설치를 시작하십시오.

4. IMPosDeviceFactory 레퍼런스

4-1 개요

- "IMPosDeviceFactory" 인터페이스는 "MPosControllerDevices" 인터페이스를 구현하는 인스턴스를 생성하는데 사용할 수 있습니다. 장치(프린터) 제어를 위한 모든 인스턴스는 "createDevice" 메소드를 사용하여 생성됩니다.

4-2 메소드



이 인터페이스에 정의된 메서드를 호출하려면 "MPosDeviceFactory" 클래스의 "Current" 프로퍼티를 사용하십시오.

예) MPosDeviceFactory.Current.createDevice (10)

4-2-1 createDevice

특정 장치와의 통신을 위한 인스턴스를 생성합니다.

[구문]

```
MPosControllerDevices createDevice(MPosDeviceType deviceType);
```

[파라미터]

MPosDevice deviceType: 생성될 객체 유형을 나타내는 값.

MPosDeviceType 열거형	값	설명
MPOS_DEVICE_PRINTER	10	POS 프린터

[반환 값]

"MPosControllerDevices" 인터페이스를 구현하는 장치 객체 또는 null

5. MPosControllerDevices 레퍼런스

5-1 개요

- "MPosControllerDevices" 인터페이스는 장치(프린터)와 관련하여 공통 메소드를 제공하며 이 인터페이스를 구현하는 모든 클래스에서 프로퍼티 및 메서드를 사용할 수 있습니다.

5-2 프로퍼티

5-2-1 dataEvent

장치에서 입력 데이터가 발생하면, 이 이벤트를 통하여 응용 프로그램으로 바이트 배열(byte array) 데이터가 전송됩니다.

5-2-2 statusUpdateEvent

프린터 장치의 상태가 변경되면 이 이벤트를 통해 장치의 상태를 나타내는 데이터가 응용 프로그램에 전송됩니다. 전송 되는 값은 아래 테이블과 같이 미리 정의 된 정수 값입니다.

MPosPrinterStatus 열거형	값	설명
MPOS_STATUS_NORMAL	0	인쇄 가능 상태
MPOS_STATUS_PRINTER_COVEROPEN	1	프린터 커버 열림
MPOS_STATUS_PRINTER_PAPEREMPTY	2	용지 없음
MPOS_STATUS_PRINTER_PAPER_NEAREND	4	용지 부족
MPOS_STATUS_PRINTER_ERROR	8	오프라인 (HOST와 연결 끊어짐)
MPOS_STATUS_BATT_HIGH	16	배터리 잔량 수준 높음
MPOS_STATUS_BATT_MID	32	배터리 잔량 수준 중간
MPOS_STATUS_BATT_LOW	64	배터리 잔량 수준 낮음
MPOS_STATUS_PRINTER_OFFLINE	128	오프라인(프린터 전원 OFF)



참고 배터리 잔량과 관련된 값은 SRP-Q300 및 SRP-Q302 모델에서 감지 할 수 있습니다.

5-2-3 DeviceID

장치 식별자 (device ID) 프로퍼티

5-2-4 IsOpen

장치가 현재 사용(Open) 가능한 상태에 대한 프로퍼티

5-2-5 CommandMode

장치와 통신 방법에 대한 프로퍼티 (“B-gate”를 통한 통신 또는 직접 통신)

5-2-6 InterfaceType

통신 인터페이스에 대한 프로퍼티 (블루투스, 와이파이, 이더넷)

5-3 메소드

5-3-1 selectInterface

연결할 장치(프린터)의 인터페이스 유형과 주소를 설정합니다.

참고 이 메소드는 "openService"메소드를 호출하기 전에 사용해야 합니다.

[구문]

```
int selectInterface(int interfaceType, string address);
```

[파라미터]

• int interface type: 연결할 장치의 통신 유형

MPosInterfaceType 열거형	값	설명
MPOS_INTERFACE_WIFI	1	와이파이(Wi-Fi) 통신
MPOS_INTERFACE_ETHERNET	2	이더넷(Ethernet) 통신
MPOS_INTERFACE_BLUETOOTH	4	블루투스(Bluetooth) 통신

• string address:

TCP(와이파이 또는 이더넷) 통신 경우, "IP 주소[:포트번호]" 형식으로 사용해야 하며, 포트 번호는 기본값으로 9100이며 생략 가능합니다. 블루투스 통신 경우, 플랫폼에 따른 아래 표에 기술된 값으로 입력되어야 합니다.

플랫폼	주소 값
iOS	블루투스 S/N (Serial Number)
Android	블루투스 MAC 주소
Windows	블루투스 MAC 주소

[반환 값]

성공 시 반환 값은 0 이며 그 이외 경우 0 아닌 값 반환

5-3-2 selectCommandMode

직접 장치와 통신할지 또는 "B-gate"를 통하여 통신할지 선택합니다.

[구문]

```
int selectCommandMode(int commandMode);
```

[파라미터]

int commandMode: 통신 방식 선택

MPosCommandMode 열거형	값	설명
MPOS_COMMAND_MODE_DEFAULT	0	B-gate를 통하여 장치와 통신
MPOS_COMMAND_MODE_BYPASS	1	장치와 직접 통신

[반환 값]

성공 시 반환 값은 0 이며 그 이외 경우 0 아닌 값 반환

5-3-3 openService

장치와 통신 연결을 설정합니다.

[구문]

```
Task<int> openService(uint timeout = 3 );
```

[파라미터]

uint timeout: 장치 연결에 사용할 최대 시간

[반환 값]

성공 시 반환 값은 0 이며 그 이외 경우 0 아닌 값 반환

5-3-4 closeService

장치와 통신 연결을 종료합니다.

[구문]

```
Task<int> closeService (int closeTimeout = 0);
```

[파라미터]

int closeTimeout: 장치 연결 종료에 사용할 최대 시간

[반환 값]

성공 시 반환 값은 0 이며 그 이외 경우 0 아닌 값 반환

5-3-5 directIO

장치에 사용자 정의 데이터를 전송합니다.

[구문]

```
Task<int> directIO(byte[] data);
```

[파라미터]

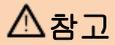
byte[] data: 장치에 전송할 데이터

[반환 값]

성공 시 반환 값은 0 이며 그 이외 경우 0 아닌 값 반환

5-3-6 setTransaction

트랜잭션(Transaction) 모드로 진입 또는 종료하여 버퍼에 있는 데이터를 장치로 전송합니다.



프린터, 커스터머 디스플레이와 같은 출력 장치를 사용할 경우 트랜잭션(Transaction) 모드의 사용을 권장합니다.

[구문]

```
Task<int> setTransaction(int mode);
```

[파라미터]

int mode: 'Transaction' 모드 진입 또는 종료 설정

MPosTransactionMode 열거형	값	설명
MPOS_PRINTER_TRANSACTION_OUT	0	트랜잭션(Transaction) 모드 종료 및 데이터 전송
MPOS_PRINTER_TRANSACTION_IN	1	트랜잭션(Transaction) 모드 진입

[반환 값]

성공 시 반환 값은 0 이며 그 이외 경우 0 아닌 값 반환

6. MPosControllerPrinter 레퍼런스

6-1 개요

- MPosControllerPrinter 인터페이스는 POS 프린터(Thermal Printer)제어를 위한 메소드를 제공합니다.

6-2 메소드

6-2-1 printText

문자의 다양한 글꼴 속성을 지정하여 문자를 인쇄합니다.

[구문]

```
Task<int> printText(string data, MPosFontAttribute attribute);
```

[파라미터]

- string data: 인쇄 될 문자열 데이터
- MPosFontAttribute: 글꼴 속성을 위한 클래스 객체

```
class MPosFontAttribute
{
    public MPosAlignment Alignment { get; set; }
    public bool Bold { get; set; }
    public int CodePage { get; set; }
    public MPosFontType FontType { get; set; }
    public MPosFontSizeHeight Height { get; set; }
    public bool Reverse { get; set; }
    public MPosFontUnderline Underline { get; set; }
    public MPosFontSizeWidth Width { get; set; }
}
```

프로퍼티

- Alignment: 정렬 (좌측, 중간, 우측)
- Bold: 굵게 인쇄
- CodePage: 문자 인코딩에 사용되는 코드 페이지.
- FontType: 글꼴 타입 (Font A, Font B, Font C)
- Height: 글꼴 높이
- Width: 글꼴 너비
- Reverse: 글꼴의 반전 인쇄
- Underline: 글꼴의 밑줄 속성

[반환 값]

성공 시 반환 값은 0 이며 그 이외 경우 0 아닌 값 반환

6-2-2 printText

문자의 다양한 글꼴 속성을 지정하여 문자를 인쇄합니다.

[구문]

```
Task<int> printText(byte[] data, MPosFontAttribute attribute);
```

[파라미터]

- `byte[] data`: 인쇄 문자열 데이터 배열
- `MPosFontAttribute`: 글꼴 속성을 위한 클래스 객체

```
class MPosFontAttribute  
{  
    public MPosAlignment Alignment { get; set; }  
    public bool Bold { get; set; }  
    public int CodePage { get; set; }  
    public MPosFontType FontType { get; set; }  
    public MPosFontSizeHeight Height { get; set; }  
    public bool Reverse { get; set; }  
    public MPosFontUnderline Underline { get; set; }  
    public MPosFontSizeWidth Width { get; set; }  
}
```

프로퍼티

- `Alignment`: 문자 정렬 (좌측, 중간, 우측)
- `Bold`: 글꼴 굵게 인쇄
- `CodePage`: 문자 인코딩에 사용되는 코드 페이지.
- `FontType`: 글꼴 타입 (Font A, Font B, Font C)
- `Height`: 글꼴 높이
- `Width`: 글꼴 너비
- `Reverse`: 글꼴의 반전 인쇄
- `Underline`: 글꼴의 밑줄 속성

[반환 값]

성공 시 반환 값은 0 이며 그 이외 경우 0 아닌 값 반환

6-2-3 setCharacterSet

문자 인코딩에 사용할 코드 페이지를 설정합니다.

[구문]

```
Task<int> setCharacterSet(int characterSet);
```

[파라미터]

int characterSet: 문자열 인코딩에 사용할 코드 페이지

MPosCodePage 열거형	값	설명
MPOS_CODEPAGE_CP437	437	영어 (PC437)
MPOS_CODEPAGE_CP850	850	서유럽 (PC850)
MPOS_CODEPAGE_CP860	860	포르투갈어 (PC860)
MPOS_CODEPAGE_CP863	863	캐나다-프랑스어 (PC863)
MPOS_CODEPAGE_CP865	865	노르웨이어 (PC865)
MPOS_CODEPAGE_WPC1252	1252	라틴어1 (WPC1252)
MPOS_CODEPAGE_CP866	866	키릴어 (PC866)
MPOS_CODEPAGE_CP852	852	라틴어2 (PC852)
MPOS_CODEPAGE_CP858	858	서유럽 (PC858)
MPOS_CODEPAGE_CP862	862	히브리어 (PC862)
MPOS_CODEPAGE_THAI11	34	태국어 (THAI 11)
MPOS_CODEPAGE_THAI14	31	태국어 (THAI 14)
MPOS_CODEPAGE_THAI16	39	태국어 (THAI 16)
MPOS_CODEPAGE_THAI18	35	태국어 (THAI 18)
MPOS_CODEPAGE_THAI42	23	태국어 (THAI 42)
MPOS_CODEPAGE_WPC1253	1253	그리스어 (WPC1253)
MPOS_CODEPAGE_WPC1254	1254	터키어 (WPC1254)
MPOS_CODEPAGE_WPC1257	1257	발틱어 (WPC1257)
MPOS_CODEPAGE_WPC1251	1251	키릴어 (WPC1251)
MPOS_CODEPAGE_FARSI	27	페르시아어
MPOS_CODEPAGE_FARSI_II	28	페르시아어
MPOS_CODEPAGE_CP737	737	그리스어 (PC737)
MPOS_CODEPAGE_CP775	775	발틱어 (PC775)
MPOS_CODEPAGE_WPC1255	1255	히브리어 (WPC1255)
MPOS_CODEPAGE_CP855	855	키릴어 (PC855)
MPOS_CODEPAGE_CP857	857	터키어 (PC857)
MPOS_CODEPAGE_CP928	928	그리스어 (PC928)
MPOS_CODEPAGE_WPC1258	1258	베트남어 (PC1258)
MPOS_CODEPAGE_WPC1250	1250	체코어 (PC1250)
MPOS_CODEPAGE_LATIN9	28605	라틴어 9 (Latin 9)
MPOS_CODEPAGE_TCVN	49	베트남어 (TCVN3)
MPOS_CODEPAGE_TCVN_CAPITAL	50	베트남어 (TCVN3 Capital)

[반환 값]

성공 시 반환 값은 0 이며 그 이외 경우 0 아닌 값 반환

6-2-4 setInternationalCharacterSet

국제 문자 인수를 위해 사용되는 문자 집합(Character Set)을 설정합니다.

[구문]

```
Task<int> setInternationalCharacterSet(int internationalCharSet);
```

[파라미터]

int internationalCharSet: 국제 문자용 문자 집합(Character Set)

MPosICS 열거형	값	설명
MPOS_ICS_USA	0	미국(USA) 코드
MPOS_ICS_FRANCE	1	프랑스(FRANCE) 코드
MPOS_ICS_GERMANY	2	독일(GERMANY) 코드
MPOS_ICS_UK	3	영국(UK) 코드
MPOS_ICS_DENMARK_I	4	덴마크1(DENMARK1) 코드
MPOS_ICS_SWEDEN	5	스웨덴(SWEDEN) 코드
MPOS_ICS_ITALY	6	이탈리아(ITALY) 코드
MPOS_ICS_SPAIN_I	7	스페인(SPAIN) 코드
MPOS_ICS_JAPAN	8	일본(JAPAN) 코드
MPOS_ICS_NORWAY	9	노르웨이(NORWAY) 코드
MPOS_ICS_DENMARK_II	10	덴마크2(DENMARK 2) 코드
MPOS_ICS_SPAIN_II	11	스페인2(SPAIN 2) 코드
MPOS_ICS_LATIN_AMERICA	12	라틴 아메리카(LATIN AMERICA) 코드
MPOS_ICS_KOREA	13	한국(KOREA) 코드
MPOS_ICS_SLOVENIA_CROATIA	14	슬로베니아(SLOVENIA) 코드
MPOS_ICS_CHINA	15	중국(CHINA) 코드

[반환 값]

성공 시 반환 값은 0 이며 그 이외 경우 0 아닌 값 반환

6-2-5 setPagemode

페이지 모드 진입 또는 종료하여 버퍼에 있는 데이터를 장치로 전송합니다.

참고 페이지 모드는 트랜잭션(Transaction) 모드와 같이 사용할 수 없습니다.

[구문]

```
Task<int> setPagemode (int mode);
```

[파라미터]

int mode: 페이지 모드 진입 또는 종료

MPosPageMode 열거형	값	설명
MPOS_PRINTER_PAGEMODE_IN	1	페이지 모드 진입
MPOS_PRINTER_PAGEMODE_OUT	0	페이지 모드 종료 및 인쇄

[반환 값]

성공 시 반환 값은 0 이며 그 이외 경우 0 아닌 값 반환

6-2-6 setPagemodePrintArea

페이지 모드의 인쇄 영역을 설정합니다.

[구문]

```
int setPagemodePrintArea(int x, int y, int width, int height);
```

[파라미터]

- int x: 인쇄 영역의 시작 X 좌표
- int y: 인쇄 영역의 시작 Y 좌표
- int width: 인쇄 너비
- int height: 인쇄 높이

[반환 값]

성공 시 반환 값은 0 이며 그 이외 경우 0 아닌 값 반환

6-2-7 setPagemodeDirection

페이지 모드의 인쇄 방향을 설정합니다.

[구문]

```
int setPagemodeDirection(int direction);
```

[파라미터]

int direction: 인쇄 방향

MPosPageMode 열거형	값	설명
MPOS_PRINTER_PD_0	0	회전 없음
MPOS_PRINTER_PD_LEFT90	1	90도 회전 (시계방향)
MPOS_PRINTER_PD_180	2	180도 회전
MPOS_PRINTER_PD_RIGHT90	3	90도 회전 (반시계방향)

[반환 값]

성공 시 반환 값은 0 이며 그 이외 경우 0 아닌 값 반환

6-2-8 setPagemodePosition

페이지 모드의 인쇄 좌표를 설정합니다.

[구문]

```
int setPagemodePosition(int x, int y);
```

[파라미터]

- int x: 인쇄될 객체(문자, 바코드, 이미지)의 X좌표
- int y: 인쇄될 객체(문자, 바코드, 이미지)의 Y좌표

[반환 값]

성공 시 반환 값은 0 이며 그 이외 경우 0 아닌 값 반환

6-2-9 getFirmwareVersion

현재 프린터 펌웨어 버전 정보를 가져옵니다.

[구문]

```
Task<int> getFirmwareVersion(Action<string> versionDelegate)
```

[파라미터]

Action<string> versionDelegate: 펌웨어 버전 정보를 받을 델리게이트(delegate)

[반환 값]

성공 시 반환 값은 0 이며 그 이외 경우 0 아닌 값 반환

6-2-10 printBitmap

플랫폼 종속적인 이미지 객체를 사용하여 이미지를 인쇄합니다.

[구문]

```
Task<int> printBitmap(object platformImage, int width, int alignment, int brightness,
                    bool isDithering, bool isCompress);
```

[파라미터]

- object platformImage: 플랫폼 종속 이미지 객체

△참고	iOS	UIKit.UIImage
	Android	Android.Graphics.Bitmap
	Windows	Windows.UI.Xaml.Media.Imaging.WriteableBitmap Windows.UI.Xaml.Media.Imaging.BitmapImage (deprecated)

- int width: 인쇄될 이미지 너비 또는 원본 크기로 인쇄

MPosImageWidth 열거형	값	설명
MPOS_IMAGE_WIDTH_ASIS	-2	원본 크기로 인쇄

- int alignment: 정렬

MPosAlignment 열거형	값	설명
MPOS_ALIGNMENT_LEFT	0	좌측 정렬
MPOS_ALIGNMENT_CENTER	1	중앙 정렬
MPOS_ALIGNMENT_RIGHT	2	우측 정렬

- int brightness: 밝기 레벨 (범위: $0 \leq \text{brightness} \leq 100$)
- bool isDithering: 디더링 알고리즘 적용 여부
- bool isCompress: 데이터 압축 전송 여부

[반환 값]

성공 시 반환 값은 0 이며 그 이외 경우 0 아닌 값 반환

6-2-11 printBitmapFile

지정된 로컬 경로에 있는 비트맵 파일을 인쇄합니다.

[구문]

```
Task<int> printBitmapFile(string path, int width, int alignment, int brightness,
                        bool isDithering, bool isCompress);
```

[파라미터]

- string path: 이미지 파일 경로

△참고 Android 경우, SD카드에 있는 파일만 인쇄할 수 있습니다.

- int width: 인쇄될 이미지 너비 또는 원본 크기로 인쇄

MPosImageWidth 열거형	값	설명
MPOS_IMAGE_WIDTH_ASIS	-2	원본 크기로 인쇄

- int alignment: 정렬

MPosAlignment 열거형	값	설명
MPOS_ALIGNMENT_LEFT	0	좌측 정렬
MPOS_ALIGNMENT_CENTER	1	중앙 정렬
MPOS_ALIGNMENT_RIGHT	2	우측 정렬

- int brightness: 이미지 밝기 (범위: $0 \leq \text{brightness} \leq 100$)
- bool isDithering: 디더링 알고리즘 적용 여부
- bool isCompress: 데이터 압축 기능 적용 여부

[반환 값]

성공 시 반환 값은 0 이며 그 이외 경우 0 아닌 값 반환

6-2-12 print1DBarcode

1차원 바코드를 인쇄합니다.

[구문]

```
Task<int> print1DBarcode(string data, int symbology, int height, int barWidth, int alignment, int textPosition);
```

[파라미터]

- string data: 바코드 데이터
- int symbology: 바코드 유형

MPos1dCodeType 열거형	값	설명
MPOS_BARCODE_UPCA	101	UPC A 바코드
MPOS_BARCODE_UPCE	102	UPC E 바코드
MPOS_BARCODE_EAN8 MPOS_BARCODE_JAN8	103	EAN 8/JAN 8바코드
MPOS_BARCODE_EAN13 MPOS_BARCODE_JAN13	104	EAN 13/JAN 13바코드
MPOS_BARCODE_ITF	106	ITF 바코드
MPOS_BARCODE_CODABAR	107	CODABAR 바코드
MPOS_BARCODE_CODE39	108	CODE39 바코드
MPOS_BARCODE_CODE93	109	CODE93 바코드
MPOS_BARCODE_CODE128	110	CODE128 바코드

참고 Code128 서브셋 A, B 또는 C 경우 바코드 데이터 시작에 "{A", "{B", "C"와 같은 데이터를 입력하여 인쇄할 수 있습니다. Code128 서브셋 C 경우 짝수개의 숫자로 구성된 데이터만 허용됩니다.

- int height: 바코드 높이 (범위: 1 ≤ height ≤ 255)
- int barWidth: 바코드 너비 (범위: 2 ≤ barWidth ≤ 6)
- int alignment: 정렬

MPosAlignment 열거형	값	설명
MPOS_ALIGNMENT_LEFT	0	좌측 정렬
MPOS_ALIGNMENT_CENTER	1	중앙 정렬
MPOS_ALIGNMENT_RIGHT	2	우측 정렬

- int textPosition: HRI (Human Readable Interface) 인쇄 위치

MPosLabelHRI 열거형	값	설명
MPOS_BARCODE_TEXTNONE	0	인쇄 안함
MPOS_BARCODE_TEXTABOVE	1	바코드 상단
MPOS_BARCODE_TEXTBELOW	2	바코드 하단

[반환 값]

성공 시 반환 값은 0 이며 그 이외 경우 0 아닌 값 반환

6-2-13 printQRCode

QR 코드를 인쇄합니다.

[구문]

```
int printQRCode (string data, int model, int alignment, int moduleSize, int eccLevel);
```

[파라미터]

- string data: QR 코드 데이터
- int model: QR 코드 모델

MPos2dCodeType 열거형	값	설명
MPOS_BARCODE_QRCODE_MODEL1	204	모델 1
MPOS_BARCODE_QRCODE_MODEL2	205	모델 2

- int alignment: 정렬

MPosAlignment 열거형	값	설명
MPOS_ALIGNMENT_LEFT	0	좌측 정렬
MPOS_ALIGNMENT_CENTER	1	중앙 정렬
MPOS_ALIGNMENT_RIGHT	2	우측 정렬

- int moduleSize: 모듈 크기 (범위: $1 \leq \text{moduleSize} \leq 8$)

- int eccLevel: 에러 보정 레벨

MPosQRCodeECCLevel 열거형	값	설명
MPOS_QRCODE_ECC_LEVEL_L	48	레벨 L
MPOS_QRCODE_ECC_LEVEL_M	49	레벨 M
MPOS_QRCODE_ECC_LEVEL_Q	50	레벨 Q
MPOS_QRCODE_ECC_LEVEL_H	51	레벨 H

[반환 값]

성공 시 반환 값은 0 이며 그 이외 경우 0 아닌 값 반환

6-2-14 printPDF417

PDF417 코드를 인쇄합니다.

[구문]

Task<int> **printPDF417**(string data, int symbol, int alignment, int columnNumber, int rowNumber, int moduleWidth, int moduleHeight, int eccLevel);

[파라미터]

- string data: PDF417 코드 데이터
- int symbol: PDF417 코드 유형

MPos2dCodeType 열거형	값	설명
MPOS_BARCODE_PDF417	201	PDF417 표준
MPOS_BARCODE_PDF417_SIMPLIFIED	202	PDF417 간소화

- int alignment: 정렬

MPosAlignment 열거형	값	설명
MPOS_ALIGNMENT_LEFT	0	좌측 정렬
MPOS_ALIGNMENT_CENTER	1	중앙 정렬
MPOS_ALIGNMENT_RIGHT	2	우측 정렬

- int columnNumber: 행(Columns) 개수 (범위: $0 \leq \text{columnNumber} \leq 30$)
- int rowNumber: 열(Row) 개수 (범위: $3 \leq \text{rowNumber} \leq 90$)
- int moduleWidth: 모듈 너비 (범위: $1 \leq \text{moduleWidth} \leq 4$)
- int moduleHeight: 모듈 높이 (범위: $2 \leq \text{moduleHeight} \leq 8$)
- int eccLevel: 에러 보정 레벨

MPosPDF417ECCLevel 열거형	값	설명
MPOS_PDF417_ECC_LEVEL_0	48	에러 보정 레벨 0
MPOS_PDF417_ECC_LEVEL_1	49	에러 보정 레벨 1
MPOS_PDF417_ECC_LEVEL_2	50	에러 보정 레벨 2
MPOS_PDF417_ECC_LEVEL_3	51	에러 보정 레벨 3
MPOS_PDF417_ECC_LEVEL_4	52	에러 보정 레벨 4
MPOS_PDF417_ECC_LEVEL_5	53	에러 보정 레벨 5
MPOS_PDF417_ECC_LEVEL_6	54	에러 보정 레벨 6
MPOS_PDF417_ECC_LEVEL_7	55	에러 보정 레벨 7
MPOS_PDF417_ECC_LEVEL_8	56	에러 보정 레벨 8

[반환 값]

성공 시 반환 값은 0 이며 그 이외 경우 0 아닌 값 반환

6-2-15 checkPrinterStatus

프린터의 현재 상태를 가져옵니다.

[구문]

```
Task<uint> checkPrinterStatus(void);
```

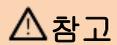
[반환 값]

성공 시 반환 값은 0 이며 그 이외 경우 0 아닌 값 반환

MPosPrinterStatus 열거형	값	설명
MPOS_STATUS_NORMAL	0	인쇄 가능 상태
MPOS_STATUS_PRINTER_COVEROPEN	1	프린터 커버 열림
MPOS_STATUS_PRINTER_PAPEREMPTY	2	용지 없음
MPOS_STATUS_PRINTER_PAPER_NEAREND	4	용지 부족
MPOS_STATUS_PRINTER_ERROR	8	오프라인 (HOST와 연결 끊어짐)
MPOS_STATUS_BATT_HIGH	16	배터리 잔량 수준 높음
MPOS_STATUS_BATT_MID	32	배터리 잔량 수준 중간
MPOS_STATUS_BATT_LOW	64	배터리 잔량 수준 낮음
MPOS_STATUS_PRINTER_OFFLINE	128	오프라인(프린터 전원 OFF)

6-2-16 checkBattStatus

프린터의 현재 배터리 상태를 가져옵니다.



이 메소드는 “SRP-Q300”, “SRP-Q302” 또는 모바일 프린터에서만 사용가능합니다.

[구문]

```
Task<uint> checkBattStatus(void);
```

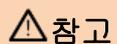
[반환 값]

성공 시 반환 값은 0 이며 그 이외 경우 0 아닌 값 반환

MPosPrinterStatus 열거형	값	설명
MPOS_STATUS_NORMAL	0	배터리 잔량: 완충
MPOS_STATUS_BATT_HIGH	16	배터리 잔량: 높음
MPOS_STATUS_BATT_MID	32	배터리 잔량: 중간
MPOS_STATUS_BATT_LOW	64	배터리 잔량: 부족

6-2-17 asbEnable

ASB(Automatic Status Back) 기능을 활성화 또는 비 활성화 합니다.



- 활성화 시, 프린터 상태가 변경될 때 마다 'statusUpdateEvent'에 의해서 등록된 메소드가 호출됩니다. 프린터 상태 값과 관련해서는 5-2-2를 참고하십시오.
- 이 메소드는 Windows, Android 플랫폼에서만 지원됩니다.

[구문]

```
Task<uint> asbEnable(bool enable);
```

[파라미터]

bool enable: 프린터 상태 통지 활성화 또는 비 활성화

[반환 값]

성공 시 반환 값은 0 이며 그 이외 경우 0 아닌 값 반환

6-2-18 printLine

직선을 인쇄합니다.

⚠참고

- 이 메소드는 페이지 모드에서 회전 없이(좌측에서 우측 방향) 인쇄 했을 때만 사용 가능하며, 직선이 아닌 사선은 인쇄할 수 없습니다.
- 이 메소드는 모바일 프린터에서만 사용가능 합니다.

[구문]

```
Task<int> printLine(int x1, int y1, int x2, int y2, int thickness);
```

[파라미터]

- int x1: 직선의 시작 X 좌표 (dot 단위)
- int y1: 직선의 시작 Y 좌표 (dot 단위)
- int x2: 직선의 종료 X 좌표 (dot 단위)
- int y2: 직선의 종료 Y 좌표 (dot 단위)
- int thickness: 직선의 굵기(범위: $0 \leq \text{thickness} \leq 16$)

[반환 값]

성공 시 반환 값은 0 이며 그 이외 경우 0 아닌 값 반환

6-2-19 printBox

사각형을 인쇄합니다.

⚠참고

- 이 메소드는 페이지 모드에서 회전 없이(좌측에서 우측 방향) 인쇄 했을 때만 사용 가능합니다.
- 이 메소드는 모바일 프린터에서만 사용가능 합니다.

[구문]

```
Task<int> printBox(int left, int top, int right, int bottom, int thickness);
```

[파라미터]

- int left: 사각형의 좌측 상단 X 좌표 (dot 단위)
- int top: 사각형의 좌측 상단 Y 좌표 (dot 단위)
- int right: 사각형의 우측 하단 X 좌표 (dot 단위)
- int y2: 사각형의 우측 하단 Y 좌표 (dot 단위)
- int thickness: 사각형의 굵기(범위: $0 \leq \text{thickness} \leq 16$)

[반환 값]

성공 시 반환 값은 0 이며 그 이외 경우 0 아닌 값 반환

6-2-20 printPdfFile

지정된 로컬 경로에 있는 PDF 파일을 인쇄합니다.

[구문]

```
Task<int> printPdfFile(object context, string filePath, int startPage, int endPage, int width,
    int alignment, int brightness, bool isDithering, bool isCompress);
```

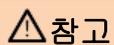
[파라미터]

- object context: Android context 객체



이 파라미터는 Android에서만 사용됩니다. iOS 또는 Windows에서는 null 값으로 설정 하십시오.

- string filePath: PDF 파일 경로



Android 경우, SD카드에 있는 파일만 인쇄할 수 있습니다.

- int startPage: 인쇄 시작 페이지 번호



첫 시작 페이지는 0부터 시작합니다.

- int endPage: 인쇄 마지막 페이지 번호



모든 페이지를 인쇄하려면 -1 값을 사용 하십시오.

- int width: 인쇄될 이미지 너비 또는 원본 크기로 인쇄

MPosImageWidth 열거형	값	설명
MPOS_IMAGE_WIDTH_ASIS	-2	원본 크기로 인쇄

- int alignment: 정렬

MPosAlignment 열거형	값	설명
MPOS_ALIGNMENT_LEFT	0	좌측 정렬
MPOS_ALIGNMENT_CENTER	1	중앙 정렬
MPOS_ALIGNMENT_RIGHT	2	우측 정렬

- int brightness: 이미지 밝기 (범위: $0 \leq \text{brightness} \leq 100$)
- bool isDithering: 디더링 알고리즘 적용 여부
- bool isCompress: 데이터 압축 기능 적용 여부

[반환 값]

성공 시 반환 값은 0 이며 그 이외 경우 0 아닌 값 반환

6-2-21 cutPaper

용지를 절단합니다.

△Note 오토 커터가 장착된 프린터에서만 이 메소드를 호출하십시오.

[구문]

```
Task<int> cutPaper(bool feed, bool fullCut);
```

[파라미터]

- bool feed: 절단 위치까지 용지 피딩 또는 피딩 없음.
- bool fullCut: 완전 절단(Full Cut) 또는 부분 절단(Partial Cut)

[반환 값]

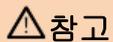
성공 시 반환 값은 0 이며 그 이외 경우 0 아닌 값 반환

7. MPosLookup 레퍼런스

7-1 개요

- “MPosLookup” 인터페이스는 장치(프린터) 연결에 필요한 정보를 검색하고 가져오는데 유용한 메소드를 제공합니다.

7-2 메소드



이 인터페이스에 정의된 메서드를 호출하려면 "MPosLookupUtil" 클래스의 "Current" 프로퍼티를 사용하십시오.

예) MPosLookupUtil.Current.refreshDevicesList()

예) MPosLookupUtil.Current.getDeviceList()

7-2-1 refreshDevicesList

연결 가능한 장치 검색을 시작합니다.

[구문]

Task<int> refreshDevicesList (int interfaceType)

[파라미터]

int interface type: 검색할 장치의 통신 유형

MPosInterfaceType 열거형	값	설명
MPOS_INTERFACE_WIFI	1	와이파이(Wi-Fi)
MPOS_INTERFACE_ETHERNET	2	이더넷(Ethernet)
MPOS_INTERFACE_BLUETOOTH	4	블루투스(Bluetooth)

[반환 값]

성공 시 반환 값은 0 이며 그 이외 경우 0 아닌 값 반환

7-2-2 getDevicesList

“refreshDeviceList” 메소드에 의해서 검색된 장치 목록을 가져옵니다.

[파라미터]

int interface type: 장치 목록을 가져오기 위한 통신 유형

MPosInterfaceType 열거형	값	설명
MPOS_INTERFACE_WIFI	1	와이파이(Wi-Fi)
MPOS_INTERFACE_ETHERNET	2	이더넷(Ethernet)
MPOS_INTERFACE_BLUETOOTH	4	블루투스(Bluetooth)

[구문]

ObservableCollection<MposConnectionInformation> getDevicesList (int interfaceType)

[반환 값]

ObservableCollection<MposConnectionInformation>: 장치 연결 정보에 대한 데이터 목록

```
class MposConnectionInformation
{
    public MPosInterfaceType IntefaceType { get; set; }
    public string Name { get; set; }
    public string Address
    public string MacAddress { get; set; }
    public string IpAddress { get; set; }
    public string PortNumber { get; set; }
}
```

프로퍼티

- InterfaceType: 통신 유형 (블루투스, 와이파이, 이더넷 등)
- Name: 장치 이름
- Address: 장치 주소 (MAC 주소, S/N, IP 주소 등)
- MACAddress: 블루투스 장치용 S/N 또는 MAC 주소
- IPAddress: 네트워크 통신을 위한 IP 주소
- PortNumber: 네트워크 통신을 위한 포트 번호

8. MPosControllerLabelPrinter 레퍼런스

8-1 개요

• MPosControllerLabelPrinter 인터페이스는 Label 프린터 제어를 위한 메소드를 제공합니다.

8-2 메소드

8-2-1 setTextEncoding

[구문]

```
int setTextEncoding(int textEncoding);
```

문자 인코딩에 사용할 코드 페이지를 설정합니다.

[파라미터]

• int textEncoding: 문자열을 인코딩 할 때 사용하는 codepage 값

MPosLabelCodePage 열거형	값	설명
MPOS_CODEPAGE_CP437	437	영어 (PC437)
MPOS_CODEPAGE_CP850	850	서유럽 (PC850)
MPOS_CODEPAGE_CP852	852	라틴어2 (PC852)
MPOS_CODEPAGE_CP860	860	포르투갈어 (PC860)
MPOS_CODEPAGE_CP863	863	캐나다-프랑스어 (PC863)
MPOS_CODEPAGE_CP865	865	노르웨이어 (PC865)
MPOS_CODEPAGE_WPC1252	1252	라틴어1 (WPC1252)
MPOS_CODEPAGE_CP857	857	터키어 (PC857)
MPOS_CODEPAGE_CP737	737	그리스어 (PC737)
MPOS_CODEPAGE_WPC1250	1250	체코어 (PC1250)
MPOS_CODEPAGE_WPC1253	1253	그리스어 (WPC1253)
MPOS_CODEPAGE_WPC1254	1254	터키어 (WPC1254)
MPOS_CODEPAGE_CP855	855	키릴어 (PC855)
MPOS_CODEPAGE_CP862	862	히브리어 (PC862)
MPOS_CODEPAGE_CP866	866	키릴어 (PC866)
MPOS_CODEPAGE_WPC1251	1251	키릴어 (WPC1251)
MPOS_CODEPAGE_WPC1255	1255	히브리어 (WPC1255)
MPOS_CODEPAGE_CP928	928	그리스어 (PC928)
MPOS_CODEPAGE_CP775	775	발틱어 (PC775)
MPOS_CODEPAGE_WPC1257	1257	발틱어 (WPC1257)
MPOS_CODEPAGE_CP858	858	서유럽 (PC858)
MPOS_CODEPAGE_CHINA_GB2312	936	중국어 (GB2312)
MPOS_CODEPAGE_CHINA_BIG5	950	중국어 (BIG5)
MPOS_CODEPAGE_KOREA_KS5601	949	한국어 (KS5601)
MPOS_CODEPAGE_JAPAN_SHIFTJIS	932	일본어 (SHIFT JIS)

[반환 값]

성공 시 반환 값은 0 이며 그 이외 경우 0 아닌 값 반환

8-2-2 setCharacterSet

문자 인코딩에 사용할 코드 페이지 및 국제 문자 집합을 설정합니다.

[구문]

Task<int> setCharacterSet(int characterSet, int internationalCharSet);

[파라미터]

• int characterSet: 문자열 인코딩에 사용할 코드 페이지

MPosLabelCodePage 열거형	값	설명
MPOS_CODEPAGE_CP437	437	영어 (PC437)
MPOS_CODEPAGE_CP850	850	서유럽 (PC850)
MPOS_CODEPAGE_CP852	852	라틴어2 (PC852)
MPOS_CODEPAGE_CP860	860	포르투갈어 (PC860)
MPOS_CODEPAGE_CP863	863	캐나다-프랑스어 (PC863)
MPOS_CODEPAGE_CP865	865	노르웨이어 (PC865)
MPOS_CODEPAGE_WPC1252	1252	라틴어1 (WPC1252)
MPOS_CODEPAGE_CP857	857	터키어 (PC857)
MPOS_CODEPAGE_CP737	737	그리스어 (PC737)
MPOS_CODEPAGE_WPC1250	1250	체코어 (PC1250)
MPOS_CODEPAGE_WPC1253	1253	그리스어 (WPC1253)
MPOS_CODEPAGE_WPC1254	1254	터키어 (WPC1254)
MPOS_CODEPAGE_CP855	855	키릴어 (PC855)
MPOS_CODEPAGE_CP862	862	히브리어 (PC862)
MPOS_CODEPAGE_CP866	866	키릴어 (PC866)
MPOS_CODEPAGE_WPC1251	1251	키릴어 (WPC1251)
MPOS_CODEPAGE_WPC1255	1255	히브리어 (WPC1255)
MPOS_CODEPAGE_CP928	928	그리스어 (PC928)
MPOS_CODEPAGE_CP775	775	발틱어 (PC775)
MPOS_CODEPAGE_WPC1257	1257	발틱어 (WPC1257)
MPOS_CODEPAGE_CP858	858	서유럽 (PC858)
MPOS_CODEPAGE_CHINA_GB2312	936	중국어 (GB2312)
MPOS_CODEPAGE_CHINA_BIG5	950	중국어 (BIG5)
MPOS_CODEPAGE_KOREA_KS5601	949	한국어 (KS5601)
MPOS_CODEPAGE_JAPAN_SHIFTJIS	932	일본어 (SHIFT JIS)

• int internationalCharSet: 국제 문자용 문자 집합(Character Set)

MPosICS 열거형	값	설명
MPOS_ICS_USA	0	미국(USA) 코드
MPOS_ICS_FRANCE	1	프랑스(FRANCE) 코드
MPOS_ICS_GERMANY	2	독일(GERMANY) 코드
MPOS_ICS_UK	3	영국(UK) 코드
MPOS_ICS_DENMARK_I	4	덴마크1(DENMARK1) 코드
MPOS_ICS_SWEDEN	5	스웨덴(SWEDEN) 코드
MPOS_ICS_ITALY	6	이탈리아(ITALY) 코드
MPOS_ICS_SPAIN_I	7	스페인(SPAIN) 코드

MPOS_ICS_JAPAN	8	일본(JAPAN) 코드
MPOS_ICS_NORWAY	9	노르웨이(NORWAY) 코드
MPOS_ICS_DENMARK_II	10	덴마크2(DENMARK 2) 코드
MPOS_ICS_SPAIN_II	11	스페인2(SPAIN 2) 코드
MPOS_ICS_LATIN_AMERICA	12	라틴 아메리카(LATIN AMERICA) 코드
MPOS_ICS_KOREA	13	한국(KOREA) 코드
MPOS_ICS_SLOVENIA_CROATIA	14	슬로베니아(SLOVENIA) 코드
MPOS_ICS_CHINA	15	중국(CHINA) 코드

[반환 값]

성공 시 반환 값은 0 이며 그 이외 경우 0 아닌 값 반환

8-2-3 checkPrinterStatus

프린터의 현재 상태를 가져옵니다.

[구문]

```
Task<uint> checkPrinterStatus(void);;
```

[반환 값]

MPosLabelPrinterStatus 열거형	값	설명
MPOS_LABEL_STATUS_IDLE	0	인쇄 가능한 상태
MPOS_LABEL_STATUS_PRINTER_PAPEREMPTY	1	종이 없음
MPOS_LABEL_STATUS_PRINTER_COVEROPEN	2	커버 열림
MPOS_LABEL_STATUS_PRINTER_CUTTER_JAMMED	4	커터 에러
MPOS_LABEL_STATUS_PRINTER_TPH_OVERHEAT	8	TPH 과열
MPOS_LABEL_STATUS_PRINTER_GAP_ERROR	16	GAP 센서 에러
MPOS_LABEL_STATUS_PRINTER_RIBBON_END	32	리본 없음
MPOS_LABEL_STATUS_PRINTER_ON_BUILDING_LABEL_TO_BE_PRINTED	64	라벨 버퍼 구성 중
MPOS_LABEL_STATUS_PRINTER_ON_PRINTING_LABEL	128	인쇄 중
MPOS_LABEL_STATUS_PRINTER_ISSUED_LABEL_ISPAUSED	256	필러에 용지 붙음

8-2-4 printBuffer

프린터에 저장된 콘텐츠 인쇄를 시작합니다.

[구문]

```
Task<int> printBuffer(int numberOfCopies);
```

[파라미터]

- int numberOfCopies: 복사본 매수 (범위 : 1 ~ 65535)

[반환 값]

성공 시 반환 값은 0 이며 그 이외 경우 0 아닌 값 반환

8-2-5 printRawData

사용자 정의 데이터를 프린터로 전송합니다.

[구문]

```
Task<int> printRawData(byte[] data);
```

[파라미터]

- byte[] data: 전송할 데이터

[반환 값]

성공 시 반환 값은 0 이며 그 이외 경우 0 아닌 값 반환

8-2-6 drawTextDeviceFont

비트맵 폰트를 이용하여 문자열 데이터를 프린터 버퍼에 저장합니다.

[구문]

```
Task<int> drawTextDeviceFont(string data, int xPos, int yPos, char fontSelection,
                             int fontWidth, int fontHeight, int rightSpace, int rotation,
                             bool reverse, bool bold, bool rightToLeft,
                             int alignment);
```

[파라미터]

- string data : 인쇄할 문자열 데이터
- int xPos : 문자열의 x축 좌표
- int yPos : 문자열의 y축 좌표
- char fontSelection : 폰트 선택

MPosLabelBitmapFont 열거형	값	설명
MPOS_LABEL_DEVICE_FONT_6PT	'0'	9 X 15 (dots)
MPOS_LABEL_DEVICE_FONT_8PT	'1'	12 X 20 (dots)
MPOS_LABEL_DEVICE_FONT_10PT	'2'	16 X 25 (dots)
MPOS_LABEL_DEVICE_FONT_12PT	'3'	19 X 30 (dots)
MPOS_LABEL_DEVICE_FONT_15PT	'4'	24 X 38 (dots)
MPOS_LABEL_DEVICE_FONT_20PT	'5'	32 X 40 (dots)
MPOS_LABEL_DEVICE_FONT_30PT	'6'	48 X 76 (dots)
MPOS_LABEL_DEVICE_FONT_14PT	'7'	22 X 34 (dots)
MPOS_LABEL_DEVICE_FONT_18PT	'8'	28 X 44 (dots)
MPOS_LABEL_DEVICE_FONT_24PT	'9'	37 X 58 (dots)
MPOS_LABEL_DEVICE_FONT_KOREAN1	'a'	16 X 16 (dots) (영.숫자 9 X 15)
MPOS_LABEL_DEVICE_FONT_KOREAN2	'b'	24 X 24 (dots) (영.숫자12 X 24)
MPOS_LABEL_DEVICE_FONT_KOREAN3	'c'	20 X 20 (dots) (영.숫자12 X 20)
MPOS_LABEL_DEVICE_FONT_KOREAN4	'd'	26 X 26 (dots) (영.숫자16 X 30)
MPOS_LABEL_DEVICE_FONT_KOREAN5	'e'	20 X 26 (dots) (영.숫자16 X 30)
MPOS_LABEL_DEVICE_FONT_KOREAN6	'f'	38 X 38 (dots) (영.숫자22 X 34)
MPOS_LABEL_DEVICE_FONT_GB2312	'm'	24 X 24 (dots) (영.숫자12 X 24)
MPOS_LABEL_DEVICE_FONT_BIG5	'n'	24 X 24 (dots) (영.숫자12 X 24)
MPOS_LABEL_DEVICE_FONT_SHIFT_JIS	'j'	24 X 24 (dots) (영.숫자12 X 24)

- int fontWidth : 문자의 가로 확대 비율 (범위: 1 ~ 4)
- int fontHeight : 문자의 세로 확대 비율 (범위: 1 ~ 4)
- int rightSpace : 문자의 오른쪽 자간 (기본 값 : 0)
- int rotation : 문자열의 출력 방향

MPosLabelRotation 열거형	값	설명
MPOS_LABEL_ROTATION_DEGREES_0	0	0도 회전
MPOS_LABEL_ROTATION_DEGREES_90	1	90도 회전 (시계 방향)
MPOS_LABEL_ROTATION_DEGREES_180	2	180도 회전
MPOS_LABEL_ROTATION_DEGREES_270	3	270도 회전 (시계 방향)

- bool reverse : 역상 인쇄 여부
- bool bold : 굵게 인쇄 여부
- bool rightToLeft : 문자열 인쇄 방향 (좌측에서 우측, 우측에서 좌측)

- int alignment : 문자열 정렬

MPosLabelAlignment 열거형	값	설명
MPOS_LABEL_ALIGNMENT_LEFT	0	왼쪽 정렬
MPOS_LABEL_ALIGNMENT_RIGHT	1	오른쪽 정렬

[반환 값]

성공 시 반환 값은 0 이며 그 이외 경우 0 아닌 값 반환

8-2-7 drawTextVectorFont

벡터 폰트를 이용하여 문자열 데이터를 프린터 버퍼에 저장합니다.

[구문]

```
Task<int> drawTextVectorFont (string data, int xPos, int yPos, char fontSelection,
                             int fontWidth, int fontHeight, int rightSpace, int rotation,
                             bool reverse, bool bold, bool italic,
                             bool rightToLeft, int alignment)
```

[파라미터]

- string data : 인쇄할 문자열 데이터
- int xPos : 문자열의 x축 좌표
- int yPos : 문자열의 y축 좌표
- char fontSelection : 글꼴 선택

MPosLabelVectorFont 열거형	값	설명
MPOS_LABEL_VECTOR_FONT_ASCII	'U'	ASCII (1Byte code)
MPOS_LABEL_VECTOR_FONT_KS5601	'K'	KS5601 (2Byte code)
MPOS_LABEL_VECTOR_FONT_BIG5	'B'	BG5 (2Byte code)
MPOS_LABEL_VECTOR_FONT_GB2312	'G'	GB2312 (2Byte code)
MPOS_LABEL_VECTOR_FONT_SHIFT_JIS	'J'	Shift-JIS (2Byte code)
MPOS_LABEL_VECTOR_FONT_OCR_A	'a'	OCR-A (1Byte code)
MPOS_LABEL_VECTOR_FONT_OCR_B	'b'	OCR-B (1Byte code)

- int fontWidth : 문자 너비
- int fontHeight : 문자 높이
- int rightSpace : 문자의 오른쪽 자간
- int rotation : 출력 방향

MPosLabelRotation 열거형	값	설명
MPOS_LABEL_ROTATION_DEGREES_0	0	0도 회전
MPOS_LABEL_ROTATION_DEGREES_90	1	90도 회전 (시계 방향)
MPOS_LABEL_ROTATION_DEGREES_180	2	180도 회전
MPOS_LABEL_ROTATION_DEGREES_270	3	270도 회전 (시계 방향)

- bool reverse : 역상 인쇄 여부
- bool bold : 굵게 인쇄 여부
- bool italic : 기울임 인쇄 여부
- bool rightToLeft : 문자열 인쇄 방향 (좌측에서 우측, 우측에서 좌측)
- int alignment : 문자열 정렬

MPosLabelAlignment 열거형	값	설명
MPOS_LABEL_ALIGNMENT_LEFT	0	왼쪽 정렬
MPOS_LABEL_ALIGNMENT_RIGHT	1	가운데 정렬
MPOS_LABEL_ALIGNMENT_CENTER	2	가운데 정렬

[반환 값]

성공 시 반환 값은 0 이며 그 이외 경우 0 아닌 값 반환

8-2-8 drawBarcode1D

1차원 바코드 데이터를 프린터 버퍼에 저장합니다.

[구문]

```
Task<int> drawBarcode1D(string data, int xPos, int yPos, int barcodeType, int widthNarrow,
    int widthWide, int height, int hri, int quietZoneWidth, int rotation)
```

[파라미터]

- string data : 바코드 데이터
- int xPos : 바코드의 x축 좌표
- int yPos : 바코드의 y축 좌표
- int barcodeType : 바코드의 종류
- int widthNarrow : 좁은 바(Narrow bar) 너비
- int widthWide : 넓은 바(Wide bar) 너비
- int height : 바코드 높이.
- int hri : HRI(Human Readable Interface) 인쇄 위치

상수명	값	설명
MPOS_BARCODE_TEXTNONE	0	인쇄 안함
MPOS_BARCODE_TEXTBELOW	1	바코드 위에 인쇄
MPOS_BARCODE_TEXTABOVE	2	바코드 아래에 인쇄
MPOS_BARCODE_TEXTBELOW_FONTSIZE2	3	바코드 위에 인쇄 (크기 2)
MPOS_BARCODE_TEXTABOVE_FONTSIZE2	4	바코드 아래에 인쇄 (크기 2)
MPOS_BARCODE_TEXTBELOW_FONTSIZE3	5	바코드 위에 인쇄 (크기 3)
MPOS_BARCODE_TEXTABOVE_FONTSIZE3	6	바코드 아래에 인쇄 (크기 3)
MPOS_BARCODE_TEXTBELOW_FONTSIZE4	7	바코드 위에 인쇄 (크기 4)
MPOS_BARCODE_TEXTABOVE_FONTSIZE4	8	바코드 아래에 인쇄 (크기 4)

- int quietZoneWidth : Quiet zone 너비(범위: 0 ~ 20)
- int rotation : 바코드 인쇄 방향

MPosLabelRotation 열거형	값	설명
MPOS_LABEL_ROTATION_DEGREES_0	0	0도 회전
MPOS_LABEL_ROTATION_DEGREES_90	1	90도 회전 (시계 방향)
MPOS_LABEL_ROTATION_DEGREES_180	2	180도 회전
MPOS_LABEL_ROTATION_DEGREES_270	3	270도 회전 (시계 방향)

[반환 값]

성공 시 반환 값은 0 이며 그 이외 경우 0 아닌 값 반환

8-2-9 drawBarcodeMaxiCode

Maxicode 바코드 데이터를 프린터 버퍼에 저장합니다.

[구문]

```
Task<int> drawBarcodeMaxiCode(string data, int xPos, int yPos, int mode)
```

[파라미터]

- string data : 바코드 데이터
- int xPos : 바코드의 x축 좌표
- int yPos : 바코드의 y축 좌표
- int mode : Maxicode 모드

MPosLabelMaxicodeMode 열거형	값	설명
MPOS_LABEL_MAXICODE_MODE_0	0	MaxiCode 모드 0
MPOS_LABEL_MAXICODE_MODE_2	2	MaxiCode 모드 2
MPOS_LABEL_MAXICODE_MODE_3	3	MaxiCode 모드 3
MPOS_LABEL_MAXICODE_MODE_4	4	MaxiCode 모드 4

[반환 값]

성공 시 반환 값은 0 이며 그 이외 경우 0 아닌 값 반환

8-2-10 drawBarcodePDF417

PDF417 바코드 데이터를 프린터 버퍼에 저장합니다.

[구문]

```
Task<int> drawBarcodePDF417(string data, int xPos, int yPos, int maximumRowCount,
    int maximumColumnCount, int errorCorrectionLevel,
    int dataCompressionMethod, bool hri,
    int barcodeOriginPoint, int moduleWidth, int barHeight,
    int rotation)
```

[파라미터]

- string data : 바코드 데이터
- int xPos : 바코드의 x축 좌표
- int yPos : 바코드의 y축 좌표
- int maximumRowCount: 바코드 Row 갯수. (범위: 3 ~ 90)
- int maximumColumnCount: 바코드 Column 갯수. (범위: 1 ~ 30)
- int errorCorrectionLevel: 에러 교정 레벨(바코드 일부분이 훼손된 경우 인식률)

MPosLabelPDF417ECL 열거형	값	설명
MPOS_LABEL_ERROR_CORRECTION_LEVEL0	0	에러 교정 레벨 0
MPOS_LABEL_ERROR_CORRECTION_LEVEL1	1	에러 교정 레벨 1
MPOS_LABEL_ERROR_CORRECTION_LEVEL2	2	에러 교정 레벨 2
MPOS_LABEL_ERROR_CORRECTION_LEVEL3	3	에러 교정 레벨 3
MPOS_LABEL_ERROR_CORRECTION_LEVEL4	4	에러 교정 레벨 4
MPOS_LABEL_ERROR_CORRECTION_LEVEL5	5	에러 교정 레벨 5
MPOS_LABEL_ERROR_CORRECTION_LEVEL6	6	에러 교정 레벨 6
MPOS_LABEL_ERROR_CORRECTION_LEVEL7	7	에러 교정 레벨 7
MPOS_LABEL_ERROR_CORRECTION_LEVEL8	8	에러 교정 레벨 8

- int dataCompressionMethod: 데이터 압축 방법

PosLabelPDF41Compress 열거형	값	설명
MPOS_LABEL_PDF417_DATA_COMPRESSION_TEXT	0	codeword당 2글자
MPOS_LABEL_PDF417_DATA_COMPRESSION_NUMERIC	1	codeword당 2.93글자
MPOS_LABEL_PDF417_DATA_COMPRESSION_BINARY	2	codeword당 1.2글자

- int hri : HRI(Human Readable Interface) 인쇄 위치

MPosLabelHRI 열거형	값	설명
MPOS_BARCODE_TEXTNONE	0	인쇄 안함
MPOS_BARCODE_TEXTBELOW	2	바코드 아래에 인쇄

- int barcodeOriginPoint: 바코드의 시작지점 (기본값:1)

MPosLabelPDF41StartPosition 열거형	값	설명
MPOS_LABEL_BARCODE_ORIGIN_POINT_CENTER	0	좌표 값이 바코드 가운데 기준
MPOS_LABEL_BARCODE_ORIGIN_POINT_UPPER	1	좌표 값이 바코드 좌측 상단 기준

- int moduleWidth : 바코드 모듈의 너비(범위 : 2 ~ 9)

- int barHeight : 바의 높이(범위 : 4 ~ 99)
- int rotation : 바코드 인쇄 방향

MPosLabelRotation 열거형	값	설명
MPOS_LABEL_ROTATION_DEGREES_0	0	0도 회전
MPOS_LABEL_ROTATION_DEGREES_90	1	90도 회전 (시계 방향)
MPOS_LABEL_ROTATION_DEGREES_180	2	180도 회전
MPOS_LABEL_ROTATION_DEGREES_270	3	270도 회전 (시계 방향)

[반환 값]

성공 시 반환 값은 0 이며 그 이외 경우 0 아닌 값 반환

8-2-11 drawBarcodeQRCode

QRCode 바코드 데이터를 프린터 버퍼에 저장합니다.

[구문]

```
Task<int> drawBarcodeQRCode(string data, int xPos, int yPos, int size, int model,
                             int errorCorrectionLevel, int rotation)
```

[파라미터]

- string data : 바코드 데이터
- int xPos : 바코드의 x축 좌표
- int yPos : 바코드의 y축 좌표
- int size : QRCode 크기 (범위: 1 ~ 9)
- int model : QRCode 모델

MPosLabelQRCodeModel 열거형	값	설명
MPOS_LABEL_QRCODE_MODEL_1	1	모델(Model) 1
MPOS_LABEL_QRCODE_MODEL_2	2	모델(Model) 2

- int errorCorrectionLevel: 에러 교정 레벨

MPosLabelQRCodeECL 열거형	값	설명
MPOS_LABEL_QRCODE_ECCLEVEL_L	'L'	에러 교정률을 7%로 설정
MPOS_LABEL_QRCODE_ECCLEVEL_M	'M'	에러 교정률을 15%로 설정
MPOS_LABEL_QRCODE_ECCLEVEL_Q	'Q'	에러 교정률을 25%로 설정
MPOS_LABEL_QRCODE_ECCLEVEL_H	'H'	에러 교정률을 35%로 설정

- int rotation : 바코드 인쇄 방향

MPosLabelRotation 열거형	값	설명
MPOS_LABEL_ROTATION_DEGREES_0	0	0도 회전
MPOS_LABEL_ROTATION_DEGREES_90	1	90도 회전 (시계 방향)
MPOS_LABEL_ROTATION_DEGREES_180	2	180도 회전
MPOS_LABEL_ROTATION_DEGREES_270	3	270도 회전 (시계 방향)

[반환 값]

성공 시 반환 값은 0 이며 그 이외 경우 0 아닌 값 반환

8-2-12 drawBarcodeDataMatrix

DataMatrix 바코드 데이터를 프린터 버퍼에 저장합니다.

[구문]

```
Task<int> drawBarcodeDataMatrix(string data, int xPos, int yPos, int size, bool reverse, int rotation);
```

[파라미터]

- string data : 바코드 데이터
- int xPos : 바코드의 x축 좌표
- int yPos : 바코드의 y축 좌표
- int size : DataMatrix 크기 (범위: 1 ~ 4)
- bool reverse : 바코드 역상 또는 정상
- int rotation : 바코드 인쇄 방향

MPosLabelRotation 열거형	값	설명
MPOS_LABEL_ROTATION_DEGREES_0	0	0도 회전
MPOS_LABEL_ROTATION_DEGREES_90	1	90도 회전 (시계 방향)
MPOS_LABEL_ROTATION_DEGREES_180	2	180도 회전
MPOS_LABEL_ROTATION_DEGREES_270	3	270도 회전 (시계 방향)

[반환 값]

성공 시 반환 값은 0 이며 그 이외 경우 0 아닌 값 반환

8-2-13 drawBarcodeAztec

Aztec 바코드 데이터를 프린터 버퍼에 저장합니다.

[구문]

```
Task<int> drawBarcodeAztec(string data, int xPos, int yPos, int size,
                           bool extendedChannel, int errorCorrectionLevel,
                           bool menuSymbol, int numberOfSymbols, string optionalID,
                           int rotation)
```

[파라미터]

- string data : 바코드 데이터
- int xPos : 바코드의 x축 좌표
- int yPos : 바코드의 y축 좌표
- int size : Aztec 크기 (범위: 1 ~ 10)
- bool extendedChannel : 확장 채널 해석 코드 설정
- int errorCorrectionLevel: 에러 교정 레벨

상수명	설명
0	에러 교정 레벨을 자동으로 설정
1~99	에러 교정률을 직접 입력
101~104	1~4 레이어 컴팩트 심볼
201~232	1~32 레이어 full range 심볼
300	심플 Aztec "Rune"

- bool menuSymbol : 메뉴 심볼을 설정
- int numberOfSymbols : Structured append 심볼 수 (범위 : 1 ~ 26)
- string optionalID : Structured append 선택 ID 필드. (최대 24글자)
- int rotation : 바코드 인쇄 방향

MPosLabelRotation 열거형	값	설명
MPOS_LABEL_ROTATION_DEGREES_0	0	0도 회전
MPOS_LABEL_ROTATION_DEGREES_90	1	90도 회전 (시계 방향)
MPOS_LABEL_ROTATION_DEGREES_180	2	180도 회전
MPOS_LABEL_ROTATION_DEGREES_270	3	270도 회전 (시계 방향)

[반환 값]

성공 시 반환 값은 0 이며 그 이외 경우 0 아닌 값 반환

8-2-14 drawBarcodeCode49

Code49 바코드 데이터를 프린터 버퍼에 저장합니다.

[구문]

```
Task<int> drawBarcodeCode49(string data, int xPos, int yPos, int widthNarrow, int widthWide,
    int height, int hri, int startingMode, int rotation)
```

[파라미터]

- string data : 바코드 데이터
- int xPos : 바코드의 x축 좌표
- int yPos : 바코드의 y축 좌표
- int widthNarrow : Narrow Bar의 너비
- int widthWide : Wide Bar의 너비
- int height : 바코드 높이
- int hri : HRI(Human Readable Interface) 인쇄 위치.

MPosLabelHRI 열거형	값	설명
MPOS_BARCODE_TEXTNONE	0	인쇄 안함
MPOS_BARCODE_TEXTABOVE	1	바코드 위에 인쇄
MPOS_BARCODE_TEXTBELOW	2	바코드 아래에 인쇄

- int startingMode : 바코드의 Starting Mode

MPosLabelCode49Mode 열거형	값	설명
MPOS_LABEL_STARTINGMODE_REGULAR_ALPHANUMERIC	0	Regular Alphanumeric Mode
MPOS_LABEL_STARTINGMODE_MULTIPLE_READ_ALPHANUMERIC	1	Multiple Read Alphanumeric
MPOS_LABEL_STARTINGMODE_REGULAR_NUMERIC	2	Regular Numeric Mode
MPOS_LABEL_STARTINGMODE_GROUP_ALPHANUMERIC	3	Group Alphanumeric Mode
MPOS_LABEL_STARTINGMODE_REGULAR_ALPHANUMERIC_SHIFT1	4	Regular Alphanumeric Shift 1
MPOS_LABEL_STARTINGMODE_REGULAR_ALPHANUMERIC_SHIFT2	5	Regular Alphanumeric Shift 2
MPOS_LABEL_STARTINGMODE_AUTOMATIC	7	Automatic Mode

- int rotation : 바코드 인쇄 방향

MPosLabelRotation 열거형	값	설명
MPOS_LABEL_ROTATION_DEGREES_0	0	0도 회전
MPOS_LABEL_ROTATION_DEGREES_90	1	90도 회전 (시계 방향)
MPOS_LABEL_ROTATION_DEGREES_180	2	180도 회전
MPOS_LABEL_ROTATION_DEGREES_270	3	270도 회전 (시계 방향)

[반환 값]

성공 시 반환 값은 0 이며 그 이외 경우 0 아닌 값 반환

8-2-15 drawBarcodeCodaBlock

CodaBlock 바코드 데이터를 프린터 버퍼에 저장합니다.

[구문]

```
Task<int> drawBarcodeCodaBlock(string data, int xPos, int yPos, int widthNarrow,
                               int widthWide, int height, bool securityLevel,
                               int numberOfCharactersPerRow, char mode,
                               int numberOfRowToEncode)
```

[파라미터]

- string data : 바코드 데이터
- int xPos : 바코드의 x축 좌표
- int yPos : 바코드의 y축 좌표
- int widthNarrow : 좁은 바(Narrow Bar) 너비
- int widthWide : 넓은 바(Wide Bar) 너비
- int height : 바코드 높이
- bool securityLevel : 바코드 보안 레벨
- int numberOfCharactersPerRow: 라인 당 글자 수(범위 : 2 ~ 62)
- char mode : Codablock 모드

MPosLabelCodaBlockMode 열거형	값	설명
MPOS_LABEL_CODABLOCK_MODE_A	'A'	Code 39 캐릭터 셋을 이용하는 코드 생성
MPOS_LABEL_CODABLOCK_MODE_E	'E'	Code 128 캐릭터 셋을 이용하는 코드 생성
MPOS_LABEL_CODABLOCK_MODE_F	'F'	Code 128 캐릭터 셋을 이용하고 func1을 자동으로 추가된 코드 생성

- int numberOfRowToEncode: 각 모드별로 아래의 값 사용 가능 합니다.

MPosLabelCodaBlockMode 열거형	Value	사용 가능한 값
MPOS_LABEL_CODABLOCK_MODE_A	'A'	1부터 18까지 사용 가능
MPOS_LABEL_CODABLOCK_MODE_E'	'E'	2부터 4까지 사용 가능
MPOS_LABEL_CODABLOCK_MODE_F	'F'	2부터 4까지 사용 가능

[반환 값]

성공 시 반환 값은 0 이며 그 이외 경우 0 아닌 값 반환

8-2-16 drawBarcodeMicroPDF

MicroPDF 바코드 데이터를 프린터 버퍼에 저장합니다.

[구문]

```
Task<int> drawBarcodeMicroPDF(string data, int xPos, int yPos, int moduleWidth,  
int moduleHeight, int mode, int rotation)
```

[파라미터]

- string data : 바코드 데이터
- int xPos : 바코드의 x축 좌표
- int yPos : 바코드의 y축 좌표
- int moduleWidth: 모듈 너비 (범위 2 ~ 8)
- int moduleHeight: 바코드 높이 (범위 1 ~ 99)
- int mode: 바코드 모드(범위 0 ~ 33)
- int rotation : 바코드 인쇄 방향

MPosLabelRotation 열거형	값	설명
MPOS_LABEL_ROTATION_DEGREES_0	0	0도 회전
MPOS_LABEL_ROTATION_DEGREES_90	1	90도 회전 (시계 방향)
MPOS_LABEL_ROTATION_DEGREES_180	2	180도 회전
MPOS_LABEL_ROTATION_DEGREES_270	3	270도 회전 (시계 방향)

[반환 값]

성공 시 반환 값은 0 이며 그 이외 경우 0 아닌 값 반환

8-2-17 drawBarcodeIMB

IMB 바코드 데이터를 프린터 버퍼에 저장합니다.

[구문]

```
Task<int> drawBarcodeIMB(string data, int xPos, int yPos, bool hri, int rotation)
```

[파라미터]

- string data : 바코드 데이터
- int xPos : 바코드의 x축 좌표
- int yPos : 바코드의 y축 좌표
- bool hri : HRI(Human Readable Interface) 인쇄 여부
- int rotation : 바코드 인쇄 방향

MPosLabelRotation 열거형	값	설명
MPOS_LABEL_ROTATION_DEGREES_0	0	0도 회전
MPOS_LABEL_ROTATION_DEGREES_90	1	90도 회전 (시계 방향)
MPOS_LABEL_ROTATION_DEGREES_180	2	180도 회전
MPOS_LABEL_ROTATION_DEGREES_270	3	270도 회전 (시계 방향)

[반환 값]

성공 시 반환 값은 0 이며 그 이외 경우 0 아닌 값 반환

8-2-18 drawBarcodeMSI

MSI 바코드 데이터를 프린터 버퍼에 저장합니다.

[구문]

```
Task<int> drawBarcodeMSI(string data, int xPos, int yPos, int narrowWidth, int widthWide,
                        int height, int checkDigitSelection, bool printCheckDigit, int hri,
                        int rotation);
```

[파라미터]

- string data : 바코드 데이터
- int xPos : 바코드의 x축 좌표
- int yPos : 바코드의 y축 좌표
- int narrowWidth: Narrow Bar의 너비
- int widthWide : Wide Bar의 너비
- int height : 바코드의 높이
- int checkDigitSelection: 체크디지트 모드

MPosLabelMSICheckDigit 열거형	값	설명
MPOS_LABEL_CHECKDIGIT_NONE	0	체크디지트 미설정
MPOS_LABEL_CHECKDIGIT_1MOD10	1	1 Mod 10
MPOS_LABEL_CHECKDIGIT_2MOD10	2	2 Mode 10
MPOS_LABEL_CHECKDIGIT_1MOD11_A ND_1MOD_10	3	1 Mod 11 and Mod 10

- bool printCheckDigit: 체크디지트 출력 여부
- int hri : HRI(Human Readable Interface) 인쇄 위치

MPosLabelHRI 열거형	값	설명
MPOS_BARCODE_TEXTNONE	0	인쇄 안함
MPOS_BARCODE_TEXTABOVE	1	바코드 위에 인쇄
MPOS_BARCODE_TEXTBELOW	2	바코드 아래에 인쇄

- int rotation : 바코드 인쇄 방향

MPosLabelRotation 열거형	값	설명
MPOS_LABEL_ROTATION_DEGREES_0	0	0도 회전
MPOS_LABEL_ROTATION_DEGREES_90	1	90도 회전 (시계 방향)
MPOS_LABEL_ROTATION_DEGREES_180	2	180도 회전
MPOS_LABEL_ROTATION_DEGREES_270	3	270도 회전 (시계 방향)

[반환 값]

성공 시 반환 값은 0 이며 그 이외 경우 0 아닌 값 반환

8-2-19 drawBarcodePlessey

Plessey 바코드 데이터를 프린터 버퍼에 저장합니다.

[구문]

```
Task<int> drawBarcodePlessey(string data, int xPos, int yPos, int narrowWidth, int widthWide,
                             int height, bool printCheckDigit, int hri, int rotation)
```

[파라미터]

- string data : 바코드 데이터
- int xPos : 바코드의 x축 좌표
- int yPos : 바코드의 y축 좌표
- int widthNarrow : Narrow Bar의 너비
- int widthWide : Wide Bar의 너비
- int height : 바코드 높이
- bool printCheckDigit : 체크디지트 출력 여부
- int hri : HRI(Human Readable Interface) 인쇄 위치

상수명	값	설명
MPOS_BARCODE_TEXTNONE	0	인쇄 안함
MPOS_BARCODE_TEXTABOVE	1	바코드 위에 인쇄
MPOS_BARCODE_TEXTBELOW	2	바코드 아래에 인쇄

- int rotation : 바코드 인쇄 방향

MPosLabelRotation 열거형	값	설명
MPOS_LABEL_ROTATION_DEGREES_0	0	0도 회전
MPOS_LABEL_ROTATION_DEGREES_90	1	90도 회전 (시계 방향)
MPOS_LABEL_ROTATION_DEGREES_180	2	180도 회전
MPOS_LABEL_ROTATION_DEGREES_270	3	270도 회전 (시계 방향)

[반환 값]

성공 시 반환 값은 0 이며 그 이외 경우 0 아닌 값 반환

8-2-20 drawBarcodeTLC39

TLC39 바코드 데이터를 프린터 버퍼에 저장합니다.

[구문]

```
Task<int> drawBarcodeTLC39(string data, int xPos, int yPos, int narrowWidth, int widthWide,  
int height, int rowHeightOfMicroPDF417,  
int narrowWidthOfMicroPDF417, int rotation)
```

[파라미터]

- string data : 바코드 데이터
- int xPos : 바코드의 x축 좌표
- int yPos : 바코드의 y축 좌표
- int narrowWidth: Narrow Bar의 너비
- int widthWide : Wide Bar의 너비
- int height : 바코드 높이
- int rowHeightOfMicroPDF417 : Micro PDF417의 행의 높이(범위 : 1 ~ 255)
- int narrowWidthOfMicroPDF417 : Micro PDF417의 Narrow bar 너비(범위 : 1 ~ 10)
- int rotation : 바코드 인쇄 방향

MPosLabelRotation 열거형	값	설명
MPOS_LABEL_ROTATION_DEGREES_0	0	0도 회전
MPOS_LABEL_ROTATION_DEGREES_90	1	90도 회전 (시계 방향)
MPOS_LABEL_ROTATION_DEGREES_180	2	180도 회전
MPOS_LABEL_ROTATION_DEGREES_270	3	270도 회전 (시계 방향)

[반환 값]

성공 시 반환 값은 0 이며 그 이외 경우 0 아닌 값 반환

8-2-21 drawBarcodeRSS

RSS 바코드 데이터를 프린터 버퍼에 저장합니다.

[구문]

```
Task<int> drawBarcodeRSS(string data, int xPos, int yPos, int barcodeType,
                        int magnification, .int separatorHeight, int barcodeHeight,
                        int segmentWidth, int rotation)
```

[파라미터]

- string data : 바코드 데이터
- int xPos : 바코드의 x축 좌표
- int yPos : 바코드의 y축 좌표
- int barcodeType : 바코드 타입 설정

MPosLabelRSSType 열거형	값	설명
MPOS_LABEL_RSS_BARCODE_TYPE_RSS14	0	RSS14
MPOS_LABEL_RSS_BARCODE_TYPE_RSS14_TRUNCATED	1	RSS14 truncated
MPOS_LABEL_RSS_BARCODE_TYPE_RSS14_STACKED	2	RSS14 stacked
MPOS_LABEL_RSS_BARCODE_TYPE_RSS14_STACKED_OMNIDIRECTIONAL	3	RSS14 Stacked omnidirectional
MPOS_LABEL_RSS_BARCODE_TYPE_RSS_LIMITED	4	RSS limited
MPOS_LABEL_RSS_BARCODE_TYPE_RSS_EXPANDED	5	RSS Expanded
MPOS_LABEL_RSS_BARCODE_TYPE_UPC_A	6	RSS UPC A
MPOS_LABEL_RSS_BARCODE_TYPE_UPC_E	7	RSS UPC E
MPOS_LABEL_RSS_BARCODE_TYPE_EAN13	8	EAN13
MPOS_LABEL_RSS_BARCODE_TYPE_EAN8	9	EAN 8
MPOS_LABEL_RSS_BARCODE_TYPE_UCC_EAN128_CC_A_B	10	EAN128 CC-A/B
MPOS_LABEL_RSS_BARCODE_TYPE_UCC_EAN128_CC_C	11	EAN128 CC-C

- int magnification : 1부터 10까지 사용 가능
- int separatorHeight : RSS 바코드 구분자의 높이(범위 : 0 ~ 22)
- int barcodeHeight: 바코드 높이 설정
- int segmentWidth : segment width 설정 (범위 : 0 ~ 22 사이의 짝수)
- int rotation : 바코드 인쇄 방향

MPosLabelRotation 열거형	값	설명
MPOS_LABEL_ROTATION_DEGREES_0	0	0도 회전
MPOS_LABEL_ROTATION_DEGREES_90	1	90도 회전 (시계 방향)
MPOS_LABEL_ROTATION_DEGREES_180	2	180도 회전
MPOS_LABEL_ROTATION_DEGREES_270	3	270도 회전 (시계 방향)

[반환 값]

성공 시 반환 값은 0 이며 그 이외 경우 0 아닌 값 반환

8-2-22 drawBlock

프린터 버퍼에 사각형(quadrangle) 또는 선(Line)의 데이터를 저장합니다.

[구문]

```
Task<int> drawBlock(int startPosX, int startPosY, int endPosX, int endPosY, string option,
    int thickness)
```

[파라미터]

- int startPosX : 좌측 상단 x축 좌표
- int startPosY : 좌측 상단 y축 좌표
- int endPosX : 우측 하단 x축 좌표
- int endPosY : 우측 하단 y축 좌표
- string option : 그리기 옵션

MPosLabelBlock 열거형	값	설명
MPOS_LABEL_DRAW_BLOCK_OPTION_LINE_OVERWRITING	'O'	라인이 겹치는 곳은 중복으로 그림
MPOS_LABEL_DRAW_BLOCK_OPTION_LINE_EXCLUSIVE_OR	'E'	라인이 겹치는 곳은 그리지 않음
MPOS_LABEL_DRAW_BLOCK_OPTION_LINE_DELETE	'D'	라인 삭제
MPOS_LABEL_DRAW_BLOCK_OPTION_SLOPE	'S'	사선
MPOS_LABEL_DRAW_BLOCK_OPTION_BOX	'B'	사각형 테두리

- int thickness : 선의 굵기를 선택. 'option' 값이 'S' 또는 'B'인 경우에만 적용됩니다.

[반환 값]

성공 시 반환 값은 0 이며 그 이외 경우 0 아닌 값 반환

8-2-23 drawCircle

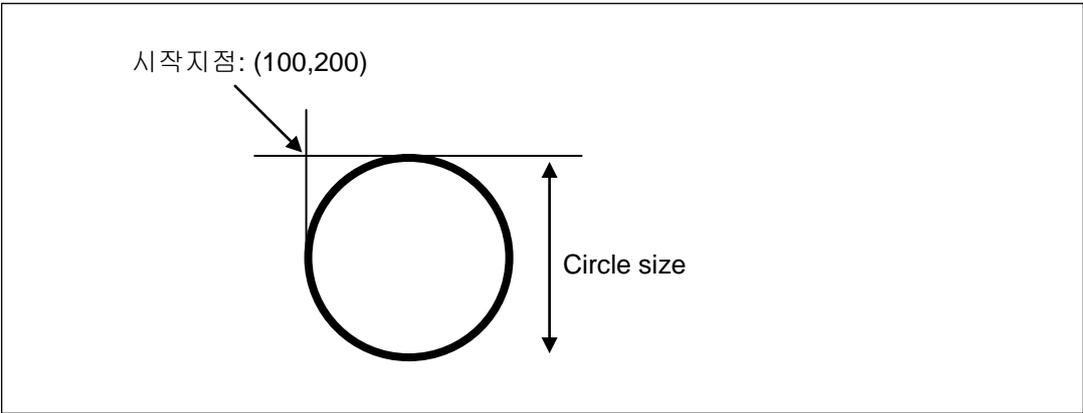
프린터 버퍼에 원(Circle)에 대한 데이터를 저장합니다.

[구문]

```
Task<int> drawCircle(int startPosX, int startPosY, int sizeSelection, int multiplier)
```

[파라미터]

- int startPosX : 원 영역의 시작지점*의 x축 좌표
시작지점(좌측과 상단의 접선의 교점) 설명 추가
- int startPosY : 원 영역의 시작지점*의 y축 좌표
* 시작지점 : 좌측과 상단의 접선의교점



- int sizeSelection: 원의 크기

MPosLabelCircle 열거형	값	W x H 크기 (dots)
MPOS_LABEL_DRAW_CIRCLE_SIZE_40X40	1	40 × 40
MPOS_LABEL_DRAW_CIRCLE_SIZE_56X56	2	56 × 56
MPOS_LABEL_DRAW_CIRCLE_SIZE_72X72	3	72 × 72
MPOS_LABEL_DRAW_CIRCLE_SIZE_88X88	4	88 × 88
MPOS_LABEL_DRAW_CIRCLE_SIZE_104X104	5	104 × 104
MPOS_LABEL_DRAW_CIRCLE_SIZE_168X168	6	168 × 168

- int multiplier : 원을 배율단위로 확대(범위 : 1 ~ 4)

[반환 값]

성공 시 반환 값은 0 이며 그 이외 경우 0 아닌 값 반환

8-2-24 drawImage

프린터 버퍼에 이미지 데이터를 저장합니다

[구문]

Task<int> drawImage(object bitmapSrc, int startPosX, int startPosY, int width, int brightness, bool isDithering, bool isCompress)

[파라미터]

• object bitmapSrc: 플랫폼 종속 이미지 객체

△참고	iOS	UIKit.UIImage
	Android	Android.Graphics.Bitmap
	Windows	Windows.UI.Xaml.Media.Imaging.WriteableBitmap Windows.UI.Xaml.Media.Imaging.BitmapImage (deprecated)

- int startPosX : 인쇄하려는 이미지의 좌측상단의 x축 좌표
- int startPosY : 인쇄하려는 이미지의 좌측상단의 y축 좌표
- int width : 출력물의 이미지 너비
- int brightness: 밝기 레벨 (범위: 0 ≤ brightness ≤ 100)
- bool isDithering: 디더링 알고리즘 적용 여부
- bool isCompress: 데이터 압축 전송 여부

[반환 값]

성공 시 반환 값은 0 이며 그 이외 경우 0 아닌 값 반환

8-2-25 drawImageFile

지정된 경로의 이미지 파일의 데이터를 프린터 버퍼에 저장합니다.

[구문]

Task<int> drawImageFile(string filename, int startPosX, int startPosY, int width, int brightness, bool isDithering, bool isCompress)

[파라미터]

• string path: 이미지 파일 경로

△참고	Android 경우, SD카드에 있는 파일만 인쇄할 수 있습니다.
-----	--------------------------------------

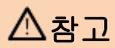
- int startPosX : 인쇄하려는 이미지의 좌측상단의 x축 좌표
- int startPosY : 인쇄하려는 이미지의 좌측상단의 y축 좌표
- int width : 출력물의 이미지 너비
- int brightness: 밝기 레벨 (범위: 0 ≤ brightness ≤ 100)
- bool isDithering: 디더링 알고리즘 적용 여부
- bool isCompress: 데이터 압축 전송 여부

[반환 값]

성공 시 반환 값은 0 이며 그 이외 경우 0 아닌 값 반환

8-2-26 setPrintingType

라벨 프린터의 인쇄 유형을 설정합니다.



라벨 프린터의 경우 Direct Thermal 방식의 프린터는 Transfer Thermal 방식으로 설정할 수 없습니다.

[구문]

```
Task<int> setPrintingType(char printingType)
```

[파라미터]

• char printingType: 인쇄 타입 설정

MPosLabelPrintingType 열거형	값	설명
MPOS_LABEL_PRINTING_TYPE_DIRECT_THERMAL	'd'	용지 직접 감열
MPOS_LABEL_PRINTING_TYPE_THERMAL_TRANSFER	't'	리본 감열

[반환 값]

성공 시 반환 값은 0 이며 그 이외 경우 0 아닌 값 반환

8-2-27 setMargin

인쇄 영역의 여백을 설정합니다.

[구문]

```
Task<int> setMargin(int horizontalMargin, int verticalMargin)
```

[파라미터]

- int horizontalMargin : 좌측 여백
- int verticalMargin : 상단 여백

[반환 값]

성공 시 반환 값은 0 이며 그 이외 경우 0 아닌 값 반환

8-2-28 setLength

라벨 길이, 갭(Gap), 라벨 용지, 오프셋(offset)를 설정합니다.

[구문]

```
Task<int> setLength(int labelLength, int gapLength, char mediaType, int offsetLength)
```

[파라미터]

- int labelLength : 용지 길이
- int gapLength : 용지의 갭 길이
- char mediaType : 용지 타입

MPosLabelMediaType 열거형	값	설명
MPOS_LABEL_MEDIA_TYPE_GAP	'G'	갭 용지
MPOS_LABEL_MEDIA_TYPE_CONTINUOUS	'C'	연속 용지
MPOS_LABEL_MEDIA_TYPE_BLACK_MARK	'B'	블랙마크 용지

- int offsetLength : 오프셋(offset) 길이

[반환 값]

성공 시 반환 값은 0 이며 그 이외 경우 0 아닌 값 반환

8-2-29 setWidth

인쇄 너비를 설정합니다.

[구문]

```
Task<int> setWidth(int labelWidth)
```

[파라미터]

- int labelWidth: 인쇄 너비

[반환 값]

성공 시 반환 값은 0 이며 그 이외 경우 0 아닌 값 반환

8-2-30 setSpeed

인쇄 속도를 설정합니다.

[구문]

```
Task<int> setSpeed(int speed)
```

[파라미터]

- int speed: 인쇄 속도

[반환 값]

성공 시 반환 값은 0 이며 그 이외 경우 0 아닌 값 반환

8-2-31 setDensity

인쇄 농도를 설정합니다.

[구문]

Task<int> setDensity(int density)

[파라미터]

• int density: 인쇄 농도 (범위 : 1 ~ 20)

[반환 값]

성공 시 반환 값은 0 이며 그 이외 경우 0 아닌 값 반환

8-2-32 setOrientation

라벨의 인쇄 방향을 설정합니다.

[구문]

Task<int> setOrientation(char orientation)

[파라미터]

• char orientation: 인쇄 방향

MPosLabelOrientation 열거형	값	설명
MPOS_LABEL_PRINTING_DIRECTION_TOP_TO_BOTTOM	'T'	상단부터 인쇄
MPOS_LABEL_PRINTING_DIRECTION_BOTTOM_TO_TOP	'B'	하단부터 인쇄

[반환 값]

성공 시 반환 값은 0 이며 그 이외 경우 0 아닌 값 반환

8-2-33 setOffset

블랙마크(Black Mar) 또는 갭(Gap)와 절취선 간에 라벨 프린터의 오프셋(offset)을 설정합니다.

[구문]

Task<int> setOffset(int offset)

[파라미터]

• int offset: offset 값 (범위: -100 ~ 100)

[반환 값]

성공 시 반환 값은 0 이며 그 이외 경우 0 아닌 값 반환

8-2-34 setCuttingPosition

라벨 프린터의 커팅 위치 또는 뜯어내기 위치를 설정합니다.

[구문]

```
Task<int> setCuttingPosition(int cuttingPosition)
```

[파라미터]

- int cuttingPosition: 커팅 위치 값(범위: -100 ~ 100)

[반환 값]

성공 시 반환 값은 0 이며 그 이외 경우 0 아닌 값 반환

8-2-35 setAutoCutter

오토커터 사용 여부를 설정합니다.

⚠참고 라벨프린터에 오토 커터가 장착 된 경우만 동작합니다.

[구문]

```
long setAutoCutter(bool enable, int period)
```

[파라미터]

- bool enable: 오토커터 사용 여부
- int period: 커팅주기 설정. 2로 설정할 경우 2장마다 용지를 커팅 합니다.

[반환 값]

성공 시 반환 값은 0 이며 그 이외 경우 0 아닌 값 반환

8-2-36 getModelName

현재 프린터의 모델명을 가져옵니다.

[구문]

Task<int> getModelName (Action<string> modelDelegate)

[파라미터]

Action<string> modelDelegate: 모델명을 받을 델리게이트(delegate)

[반환 값]

성공 시 반환 값은 0 이며 그 이외 경우 0 아닌 값 반환

8-2-37 getFirmwareVersion

현재 프린터 펌웨어 버전 정보를 가져옵니다.

[구문]

Task<int> getFirmwareVersion(Action<string> versionDelegate)

[파라미터]

Action<string> versionDelegate: 펌웨어 버전 정보를 받을 델리게이트(delegate)

[반환 값]

성공 시 반환 값은 0 이며 그 이외 경우 0 아닌 값 반환

9. 부록

9-1 에러 코드 표

- 아래 에러 코드 표는 이 매뉴얼에서 제공되는 API를 호출하였을 때 반환되는 값을 요약한 것 입니다.

MPosResult 열거형	값	설명
MPOS_SUCCESS	0	메소드 실행 성공
MPOS_ALREADY_OPEN	1	이미 오픈(OPEN) 상태
MPOS_FAIL	1000	메소드 실행 실패
MPOS_FAIL_INVALID_INTERFACE	1001	지원되지 않는 통신 유형
MPOS_FAIL_NO_OPEN	1002	오픈되어 있지 않는 장치에 접근
MPOS_FAIL_NOT_SUPPORT	1003	장치에서 지원되는 않는 API 사용
MPOS_FAIL_INVALID_PARAMETER	1004	메소드에 유효하지 않는 인자
MPOS_FAIL_NO_RESPONSE	1005	장치로부터 응답 없음
MPOS_FAIL_NOT_CONNECT	1006	장치 연결 실패
MPOS_FAIL_NO_FILE	1008	지정된 경로에 파일 없음
MPOS_FAIL_NOT_SUPPORT_CODEPAGE	1012	지원되지 않는 코드 페이지
MPOS_FAIL_NOT_SUPPORT_ICS	1013	지원되지 않는 국제 코드 페이지(International code page)
MPOS_PAGEMODE_ALREADY_IN	1014	이미 페이지 모드 진입 상태
MPOS_TRANSACTION_ALREADY_IN	1015	이미 트랜잭션 모드 진입 상태
MPOS_FAIL_NOT_SUPPORT_ESCSEQ	1017	지원되는 않는 이스케이프 시퀀스(escape sequence)
MPOS_FAIL_NO_DEVICE_FOUND	5000	검색된 장치 없음
MPOS_FAIL_NOT_PERMISSION_ALLOWD	5001	필요한 권한 획득 실패
MPOS_FAIL_NOT_SUPPORTED_IMAGE_TYPE	5002	지원되지 않는 파일 확장자
MPOS_FAIL_IMAGE_DECODE	5003	BASE64로 이미지 디코딩 실패

저작권

© BIXOLON Co., Ltd. 모든 권한을 소유합니다.

이 사용설명서와 제품에 사용된 저작물은 저작권법에 의해 보호되어 있습니다.
(주)빅솔론의 사전 서면 동의 없이 사용 설명서 및 제품에 사용된 저작물에 대한 일부 또는 전체를 무단으로 복제, 저장, 전송하는 것을 금합니다.
제공된 정보는 본 제품에만 해당되며 다른 제품에 대해서는 적용되지 않습니다.
또한 본 정보 사용으로 인해 발생하는 직/간접적 손해에 대해 책임지지 않습니다.

- 빅솔론 로고는 (주)빅솔론의 등록상표입니다.
- 모든 다른 상표 또는 제품 이름은 해당하는 회사 또는 조직의 상표입니다.

(주)빅솔론은 제품의 기능과 품질 향상을 위하여 지속적인 개선을 하고 있습니다.
이로 인하여 제품의 사양과 매뉴얼의 내용은 사전 통보 없이 변경될 수 있습니다.

사용시 주의사항

프린터와 같은 전자 제품은 정전기에 의해 쉽게 훼손될 수 있습니다.
정전기로부터 프린터를 보호하기 위해서는 프린터 후면 부에 케이블을 연결하거나 제거하기 전에 반드시 프린터 전원을 끄십시오.
만약 프린터가 정전기로부터 손상을 입었을 경우에는 가까운 구입처에 문의하십시오.

