



AsReader Wired BOX-Type

SDK リファレンスガイド V1.3

修正履歴

No.	バージョン	修正内容	日付
1	1.0	新規作成	2020/08/19
2	1.1	1. 属性 isShowPrintNSLog を追加 2. メソッド setRecharging を修正 3. メソッド getRechargedStatus を追加 4. メソッド tX を追加 5. メソッド getFirmwareVersion を追加 6. readTimeForMultiAntennaReceivedAnt1 のデリゲートメソッド名を修正 7. didSetReadTimeForMultiAntenna のデリゲートメソッド名を修正 8. デリゲートメソッド setRecharging を追加 9. デリゲートメソッド rfidModuleVersionReceived を追加 10. デリゲートメソッド receivedRechargedStatus を追加 11. 列挙 AsReaderWiredBOXTypeSDKBOXNetworkState を修正 12. 列挙 AsReaderWiredBOXTypeSDKResultSetRecharging を追加	2022/04/08
3	1.2	1. プロパティ connectMethod を追加 2. 列挙型 AsReaderWiredBOXTypeConnectMethod を追加	2022/5/16
4	1.3	メソッド setFHLBTPParamReadTime にある rfLevel の引数型を「uint16_t」から「int」に修正しました	2022/5/23

目次

1 SDK の使用	5
1.1. SDK を追加.....	5
1.1.1. TARGET -> Build phases -> Link Binary With Libraries	5
1.1.2. 「Add Other...」、「Add Files...」を選択する	6
1.1.3. AsReaderWiredBOXTypeSDK.framework を追加する.....	7
1.2. ヘッダーファイルをインポートする.....	7
1.3. SDK を使用する.....	7
1.3.1. AsReaderWiredBOXType オブジェクトを生成・初期化する(シングルトンモード)	7
1.3.2. デリゲートを設定する.....	8
1.3.3. 検索を開始する.....	8
1.3.4. asReaderWiredBOXType オブジェクトを接続する.....	8
1.3.5. RFID データの受信を開始する.....	8
1.3.6. RFID データの読取を開始する.....	8
1.3.7. RFID データの読取を停止する.....	9
1.3.8. asReaderWiredBOXType オブジェクトの接続を切断する.....	9
2 AsReaderWiredBOXType クラス	10
2.1. 属性.....	10
2.2. メソッド.....	10
2.2.1. sharedInstance	10
2.2.2. getAsWiredBOXPrinter	11
2.2.3. startSearchDevice	11
2.2.4. connectServer.....	11
2.2.5. disconnectServer	12
2.2.6. startReceiveHIDData	12
2.2.7. stopReceiveHIDData	13
2.2.8. startReceiveRFIDData.....	13
2.2.9. stopReceiveRFIDData	14
2.2.10. startInventoryWithTagsMaxTagCount.....	14
2.2.11. stopInventory	16
2.2.12. updateRegistry.....	17
2.2.13. setRecharging.....	17
2.2.14. getRechargedStatus	18
2.2.15. getSession	18
2.2.16. setSession	19
2.2.17. getOutputPowerLevelForMultiAntenna	19
2.2.18. setOutputPowerLevel	20
2.2.19. getOutputPowerLevel	21
2.2.20. setOutputPowerLevel	21
2.2.21. getFHLBTParam.....	22

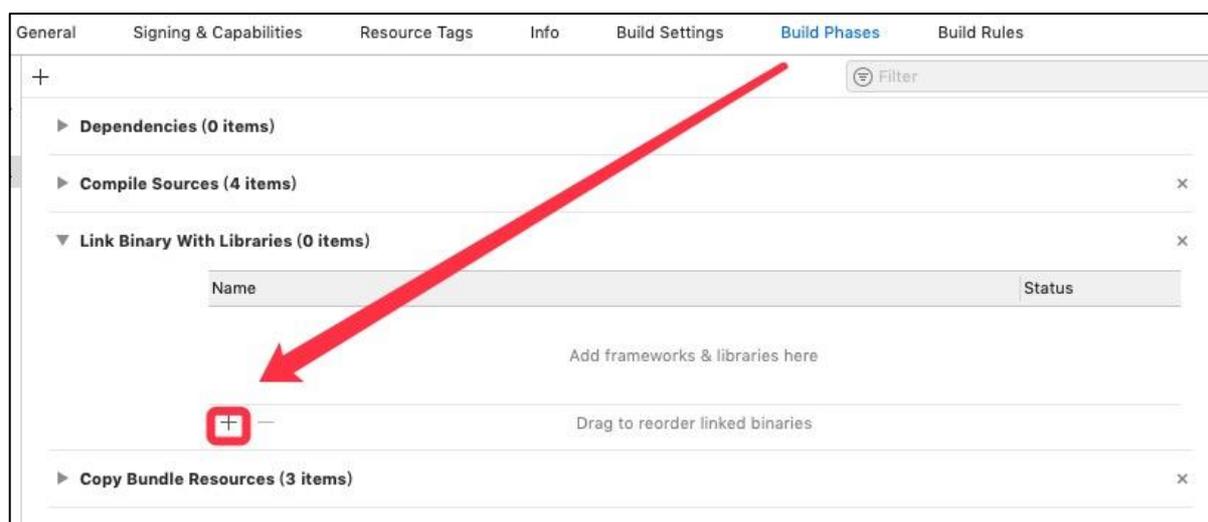
2.2.22. setFHLBTParam	22
2.2.23. getReadTimeForMultiAntenna	24
2.2.24. setReadTimeForMultiAntennaReceived	25
2.2.25. setDHCPOnMac	26
2.2.26. setDHCPOffMac	26
2.2.27. tX.....	27
2.2.28. getFirmwareVersion.....	28
2.3. デリゲート	29
2.3.1. searchingIP	29
2.3.2. networkStateChanged	29
2.3.3. searchedUPDSearchingDeviceInfo	30
2.3.4. inventoryStartedWithStatus	31
2.3.5. inventoryStoppedWithStatus	31
2.3.6. receivedPCEPCData	32
2.3.7. updatedRegistryStatus	33
2.3.8. HIDDataRecived	33
2.3.9. receivedSession	34
2.3.10. didSetSessionStatus	34
2.3.11. receivedOuputPowerAntenna1	35
2.3.12. didSetOutputPowerForMultiAntenna.....	36
2.3.13. receivedOutputPowerLevel	37
2.3.14. didSetOutputPowerLevelStatus	37
2.3.15. receivedFHLBTData	38
2.3.16. didSetFHLBTStatus.....	39
2.3.17. readTimeForMultiAntennaReceivedAnt1	39
2.3.18. didSetReadTimeForMultiAntenna	40
2.3.19. setRecharging.....	41
2.3.20. rfidModuleVersionReceived	41
2.3.21. receivedRechargedStatus	42
3 AsWiredBOXPrinter クラス.....	43
3.1. 属性.....	43
3.2. メソッド	43
3.2.1. initWithAsReaderWiredBOXType.....	43
3.2.2. getPrinterStatus	43
3.2.3. addPrintText.....	44
3.2.4. addPrintQRCode	44
3.2.5. addPrintImage.....	45
3.2.6. addCutCommand.....	45
3.2.7. openDrawer	46
3.2.8. cleanPrintData	46
3.2.9. doPrint.....	46

3.2.10. txRawData	47
3.3. デリゲート	47
3.3.1. AsReaderWiredBOXTypePrinterStatus	47
3.3.2. AsReaderWiredBOXTypePrinterConnected	48
3.3.3. retryPrintOnError	48
3.3.4. whenPrintSucceeded	48
3.3.5. rxRawData	49
4 AsReaderWiredBOXTypeSDKDefine クラス.....	50
4.1. 列挙.....	50
4.1.1. AsReaderWiredBOXTypeSDKPrinterStatus	50
4.1.2. AsReaderWiredBOXTypeSDKBOXNetworkState	50
4.1.3. AsReaderWiredBOXTypeSDKBOXSelectAntenna	51
4.1.4. AsReaderWiredBOXTypeSDKCutType	51
4.1.5. AsReaderWiredBOXTypeSDKPrinterAlignment	51
4.1.6. AsReaderWiredBOXTypeSDKMNPrintFont	52
4.1.7. AsReaderWiredBOXTypeSDKQrModel	52
4.1.8. AsReaderWiredBOXTypeSDKQrDotSize	52
4.1.9. AsReaderWiredBOXTypeSDKBOXStatusCode	53
4.1.10. AsReaderWiredBOXTypeSDKCommonStatus	53
4.1.11. AsReaderWiredBOXTypeSDKResultSetRecharging	53
4.1.12. AsReaderWiredBOXTypeConnectMethod.....	54

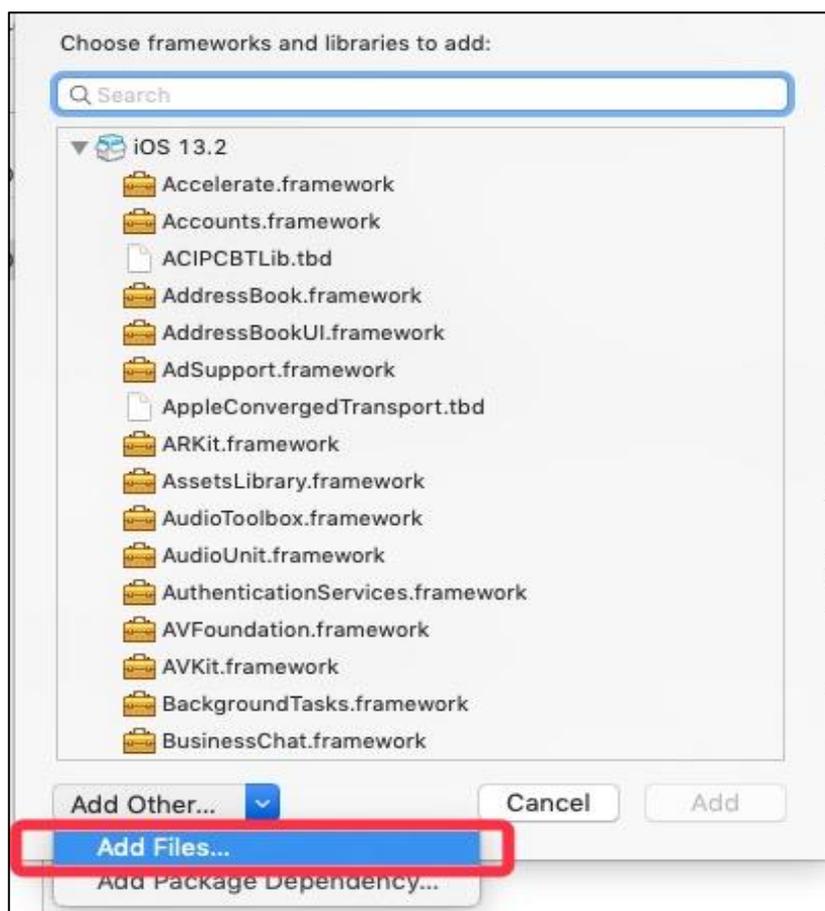
1 SDK の使用

1.1. SDK を追加

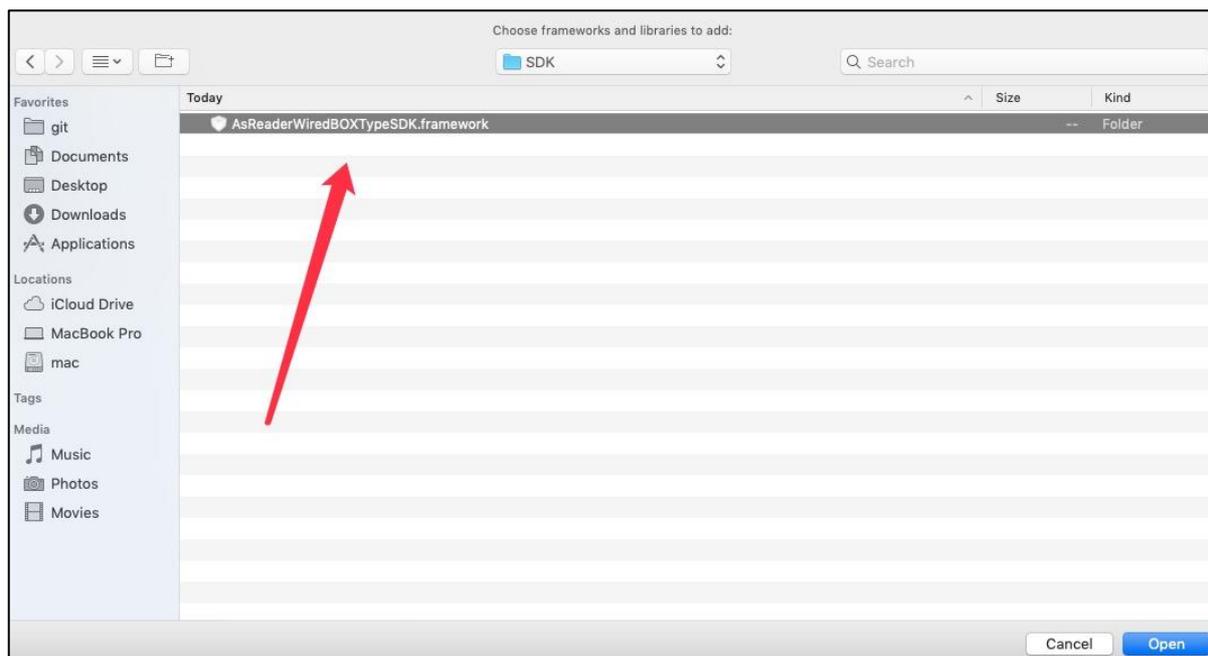
1.1.1. TARGET -> Build phases -> Link Binary With Libraries



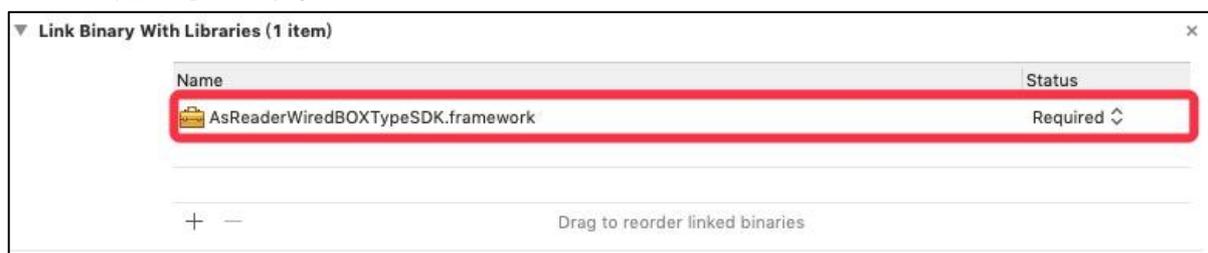
1.1.2. 「Add Other...」、「Add Files...」を選択する



1.1.3. AsReaderWiredBOXTypeSDK.framework を追加する



SDK が追加されます。



1.2. ヘッダーファイルをインポートする

Objective-C プロジェクトは、SDK を使用するクラスにヘッダーファイルをインポートする必要があります。以下参考:

```
#import <AsReaderWiredBOXTypeSDK/AsReaderWiredBOXTypeSDK.h>
```

1.3. SDK を使用する

1.3.1. AsReaderWiredBOXType オブジェクトを生成・初期化する(シングルトンモード)

```
AsReaderWiredBOXType*asReaderWiredBOXType = [AsReaderWiredBOXType  
sharedInstance];
```

1.3.2. デリゲートを設定する

```
asReaderWiredBOXType.delegate = self;
```

1.3.3. 検索を開始する

```
[asReaderWiredBOXType startSearchDevice];
```

検索した IP を受信するデリゲート「searchingIP」([2.3.1](#)を参照)。

1.3.4. asReaderWiredBOXType オブジェクトを接続する

「searchingIP」([2.3.1](#)を参照)を通じて IP アドレスを取得し、接続します。

```
[asReaderWiredBOXType connectServer:@"192.168.1.10"port:9600];
```

接続状態の変更を通知するデリゲート「networkStateChanged」([2.3.2](#)を参照)。

1.3.5. RFID データの受信を開始する

```
[asReaderWiredBOXType startReceiveRFIDData];
```

1.3.6. RFID データの読取を開始する

```
[asReaderWiredBOXType startInventoryWithTagsMaxTagCount:0  
                                readTime:0  
                                repeatCycle:0  
                                enableAntenna1:YES  
                                enableAntenna2:YES  
                                enableAntenna3:YES  
                                enableAntenna4:YES  
                                enableAntenna5:YES  
                                enableAntenna6:YES  
                                enableAntenna7:YES  
                                enableAntenna8:YES];
```

読取開始の実行結果を受け取るデリゲート「inventoryStartedWithStatus」([2.3.4](#)を参照)。

読み取った RFID データを受け取るデリゲート「receivedPCEPCData」([2.3.6](#)を参照)。

1.3.7. RFID データの読取を停止する

```
[asReaderWiredBOXType stopInventory];
```

読取停止の実行結果を受け取るデリゲート「inventoryStoppedWithStatus」([2.3.5](#)を参照)。

1.3.8. asReaderWiredBOXType オブジェクトの接続を切断する

```
[asReaderWiredBOXType disconnectServer];
```

接続状態の変更を通知するデリゲート「networkStateChanged」([2.3.2](#)を参照)。

2 AsReaderWiredBOXType クラス

AsReaderWiredBOXType オブジェクトは Wired BOX-Type の検索・接続・切断、RFID データの読取、読取停止、読取に関する設定、USB 接続した機器の HID データを受信などの機能があります。

2.1. 属性

属性名	属性	型	説明
host	nonatomic, strong, readonly	NSString	IP アドレス
delegate	nonatomic, assign	AsReaderWiredBOXTypeDelegate	AsReaderWiredBOXTypeDelegate (2.3 を参照)
isConnected	nonatomic, assign, readonly	BOOL	接続状態
isShowPrint NSLog	nonatomic, assign	BOOL	Log 出力要否
connectMethod	nonatomic, assign	AsReaderWiredBOXTypeConnectMethod	AsReaderWiredBOXTypeConnectMethod (4.1.12 を参照)

2.2. メソッド

2.2.1. sharedInstance

+ (AsReaderWiredBOXType *)sharedInstance;				
	引数名	In/Out	型	説明
戻り値	-	Out	AsReaderWiredBOXType	AsReaderWiredBOXType オブジェクト
メソッド説明: シングルトンモードで AsReaderWiredBOXType オブジェクトを生成・初期化を行います。 サンプルコード: <pre>AsReaderWiredBOXType *asReaderWiredBOXType = [AsReaderWiredBOXType sharedInstance];</pre>				

2.2.2. getAsWiredBOXPrinter

- (AsWiredBOXPrinter *)getAsWiredBOXPrinter;				
	引数名	In/Out	型	説明
戻り値	-	Out	AsWiredBOXPrinter	AsWiredBOXPrinter オブジェクト
<p>メソッド説明: AsWiredBOXPrinter オブジェクトを生成します。</p> <p>サンプルコード: (注: asReaderWiredBOXType は AsReaderWiredBOXType クラスのインスタンスオブジェクトです。)</p> <pre>AsWiredBOXPrinter *asPrinter = [asReaderWiredBOXType getAsWiredBOXPrinter];</pre>				

2.2.3. startSearchDevice

- (BOOL)startSearchDevice;				
	引数名	In/Out	型	説明
戻り値	-	Out	BOOL	YES: コマンドの送信に成功 NO: コマンドの送信に失敗
<p>メソッド説明: 同じネットワークセグメント上の Wired BOX-Type デバイスを検索します。 このメソッドを実行した後、「searchingIP」(2.3.1を参照)、 「searchedUPDSearchingDeviceInfo」(2.3.3を参照)デリゲートがコールバックされます。</p> <p>サンプルコード: (注: asReaderWiredBOXType は AsReaderWiredBOXType クラスのインスタンスオブジェクトです。)</p> <pre>BOOL status = [asReaderWiredBOXType startSearchDevice]; if (status) { //コマンドの送信に成功 } else { //コマンドの送信に失敗 }</pre>				

2.2.4. connectServer

- (BOOL)connectServer:(NSString*)ip port:(int)port;				
	引数名	In/Out	型	説明
引数	ip	In	NSString	IP アドレス
引数	port	In	int	ポート(デフォルトポート: 9600)

戻り値	-	Out	BOOL	YES: コマンドの送信に成功 NO: コマンドの送信に失敗
<p>メソッド説明: TCP ポートと IP を指定し、Wired BOX-Type デバイスに接続します。 このメソッドを実行した後、「networkStateChanged」(2.3.2を参照)デリゲートがコールバックされます。</p> <p>サンプルコード:(注: asReaderWiredBOXType は AsReaderWiredBOXType クラスのインスタンスオブジェクトです。)</p> <pre> BOOL status = [asReaderWiredBOXType connectServer:@"192.168.1.10" port:9600]; if (status) { //コマンドの送信に成功 } else { //コマンドの送信に失敗 } </pre>				

2.2.5. disconnectServer

- (BOOL)disconnectServer;				
	引数名	In/Out	型	説明
戻り値	-	Out	BOOL	YES: コマンドの送信に成功 NO: コマンドの送信に失敗
<p>メソッド説明: Wired BOX-Type デバイスから切断します。 このメソッドを実行した後、「networkStateChanged」(2.3.2を参照)デリゲートがコールバックされます。</p> <p>サンプルコード:(注: asReaderWiredBOXType は AsReaderWiredBOXType クラスのインスタンスオブジェクトです。)</p> <pre> BOOL status = [asReaderWiredBOXType disconnectServer]; if (status) { //コマンドの送信に成功 } else { //コマンドの送信に失敗 } </pre>				

2.2.6. startReceiveHIDData

- (BOOL)startReceiveHIDData;				
	引数名	In/Out	型	説明

戻り値	-	Out	BOOL	YES: コマンドの送信に成功 NO: コマンドの送信に失敗
<p>メソッド説明: Wired BOX-Type デバイスに USB 接続した HID 入力デバイスから HID データの受信を開始します。Wired BOX-Type デバイスが HID 入力デバイスに接続されている場合、アプリは HID 入力デバイスからの入力データを受信できます。</p> <p>サンプルコード:(注: asReaderWiredBOXType は AsReaderWiredBOXType クラスのインスタンスオブジェクトです。)</p> <pre> BOOL status = [asReaderWiredBOXType startReceiveHIDData]; if (status) { //コマンドの送信に成功 } else { //コマンドの送信に失敗 } </pre>				

2.2.7. stopReceiveHIDData

- (BOOL)stopReceiveHIDData;				
	引 数 名	In/Out	型	説明
戻り値	-	Out	BOO L	YES: コマンドの送信に成功 NO: コマンドの送信に失敗
<p>メソッド説明: HID データの受信を停止します。アプリは HID 入力デバイスからの入力データを受信できなくなります。</p> <p>サンプルコード:(注: asReaderWiredBOXType は AsReaderWiredBOXType クラスのインスタンスオブジェクトです。)</p> <pre> BOOL status = [asReaderWiredBOXType stopReceiveHIDData]; if (status) { //コマンドの送信に成功 } else { //コマンドの送信に失敗 } </pre>				

2.2.8. startReceiveRFIDData

- (BOOL)startReceiveRFIDData;				
	引数名	In/Out	型	説明

戻り値	-	Out	BOOL	YES:コマンドの送信に成功 NO: コマンドの送信に失敗
<p>メソッド説明: RFID データの送受信を開始します。アプリは WiredBOX-Type デバイス内の RFID モジュールからのデータを受信できます。</p> <p>サンプルコード:(注: asReaderWiredBOXType は AsReaderWiredBOXType クラスのインスタンスオブジェクトです。)</p> <pre> BOOL status = [asReaderWiredBOXType startReceiveRFIDData]; if (status) { //コマンドの送信に成功 } else { //コマンドの送信に失敗 } </pre>				

2.2.9. stopReceiveRFIDData

- (BOOL)stopReceiveRFIDData;				
	引数名	In/Out	型	説明
戻り値	-	Out	BOOL	YES:コマンドの送信に成功 NO: コマンドの送信に失敗
<p>メソッド説明: RFID データの送受信を停止します。アプリは WiredBOX-Type デバイス内の RFID モジュールからのデータを受信できなくなります。</p> <p>サンプルコード:(注: asReaderWiredBOXType は AsReaderWiredBOXType クラスのインスタンスオブジェクトです。)</p> <pre> BOOL status [asReaderWiredBOXType stopReceiveRFIDData]; if (status) { //コマンドの送信に成功。 } else { //コマンドの送信に失敗。 } </pre>				

2.2.10. startInventoryWithTagsMaxTagCount

- (BOOL)startInventoryWithTagsMaxTagCount:(uint8_t)maxTagCount readTime:(uint8_t)readTime repeatCycle:(uint16_t)repeatCycle enableAntenna1:(BOOL)antenna1 enableAntenna2:(BOOL)antenna2				
---	--	--	--	--

enableAntenna3:(BOOL)antenna3 enableAntenna4:(BOOL)antenna4 enableAntenna5:(BOOL)antenna5 enableAntenna6:(BOOL)antenna6 enableAntenna7:(BOOL)antenna7 enableAntenna8:(BOOL)antenna8;				
	引数名	In/Out	型	説明
引数	maxTagCount	In	uint8_t	一度のインベントリで読み取りするタグの最大数
引数	readTime	In	uint8_t	一度のインベントリの最大継続時間
引数	repeatCycle	In	uint16_t	一度のインベントリでのインベントリサイクルの最大数
引数	antenna1	In	BOOL	アンテナ 1 の使用有無 YES:使用する NO: 使用しない
引数	antenna2	In	BOOL	アンテナ 2 の使用有無 YES:使用する NO: 使用しない
引数	antenna3	In	BOOL	アンテナ 3 の使用有無 YES:使用する NO: 使用しない
引数	antenna4	In	BOOL	アンテナ 4 の使用有無 YES:使用する NO: 使用しない
引数	antenna5	In	BOOL	アンテナ 5 の使用有無 YES:使用する NO: 使用しない
引数	antenna6	In	BOOL	アンテナ 6 の使用有無 YES:使用する NO: 使用しない
引数	antenna7	In	BOOL	アンテナ 7 の使用有無 YES:使用する NO: 使用しない
引数	antenna8	In	BOOL	アンテナ 8 の使用有無 YES:使用する NO: 使用しない
戻り値	-	Out	BOOL	YES:コマンドの送信に成功

				NO: コマンドの送信に失敗
<p>メソッド説明: RFID タグをインベントリします。 このメソッドを実行した後、「inventoryStartedWithStatus」(2.3.4を参照)、 「receivedPCEPCData」(2.3.6を参照)デリゲートがコールバックされます。 サンプルコード:(注: asReaderWiredBOXType は AsReaderWiredBOXType クラスのインスタンスオブジェクトです。) <pre> BOOL status = [asReaderWiredBOXType startInventoryWithTagsMaxTagCount:0 readTime:0 repeatCycle:0 enableAntenna1:YES enableAntenna2:YES enableAntenna3:YES enableAntenna4:YES enableAntenna5:YES enableAntenna6:YES enableAntenna7:YES enableAntenna8:YES]; if (status) { //コマンドの送信に成功 } else { //コマンドの送信に失敗 } </pre> </p>				

2.2.11. stopInventory

- (BOOL)stopInventory;				
	引数名	In/Out	型	説明
戻り値	-	Out	BOOL	YES: コマンドの送信に成功 NO: コマンドの送信に失敗
<p>メソッド説明: RFID タグのインベントリを停止します。 このメソッドを実行した後、「inventoryStoppedWithStatus」(2.3.5を参照)デリゲートがコールバックされます。 サンプルコード:(注: asReaderWiredBOXType は AsReaderWiredBOXType クラスのインスタンスオブジェクトです。) <pre> BOOL status = [asReaderWiredBOXType stopInventory]; if (status) { //コマンドの送信に成功 } </pre> </p>				

```

} else {
//コマンドの送信に失敗
}

```

2.2.12. updateRegistry

- (BOOL)updateRegistry;				
	引数名	In/Out	型	説明
戻り値	-	Out	BOOL	YES: コマンドの送信に成功 NO: コマンドの送信に失敗
<p>メソッド説明: RFID 設定を更新した後、恒久的に設定を保存する場合に使用します。 このメソッドを実行した後、「updatedRegistryStatus」(2.3.7 を参照) デリゲートがコールバックされます。</p> <p>サンプルコード:(注: asReaderWiredBOXType は AsReaderWiredBOXType クラスのインスタンスオブジェクトです。) <pre> BOOL status = [asReaderWiredBOXType updateRegistry]; if (status) { //コマンドの送信に成功 } else { //コマンドの送信に失敗 } </pre> </p>				

2.2.13. setRecharging

- (AsReaderWiredBOXTypeSDKResultSetRecharging)setRecharging:(BOOL)status;				
	引数名	In/Out	型	説明
引数	status	In	BOOL	YES: 充電する NO: 充電しない
戻り値	-	Out	AsReaderWiredBOXType SDKResultSetRecharging	AsReaderWiredBOXTypeSDK ResultSetRecharging 列挙 4.1.11 を参照
<p>メソッド説明: Wired BOX-Type デバイスに接続された iOS デバイスを充電するかどうかを設定します。 このメソッドを実行した後、「setRecharging」(2.3.19 を参照) デリゲートがコールバックします。</p> <p>サンプルコード:(注: asReaderWiredBOXType は AsReaderWiredBOXType クラスのインスタンスオブジェクトです。)</p>				

```
AsReaderWiredBOXTypeSDKResultSetRecharging status = [asReaderWiredBOXType
setRecharging:YES];
```

2.2.14. getRechargedStatus

- (BOOL)getRechargedStatus;				
	引数名	In/Out	型	説明
戻り値	-	Out	BOOL	YES: 充電のステータスの取得に成功 NO: 充電のステータスの取得に失敗
<p>メソッド説明: Wired BOX-Type デバイスに接続された iOS デバイスの充電有無のステータスを取得します。このメソッドを実行した後、「receivedRechargedStatus」(2.3.21を参照)デリゲートがコールバックされます。</p> <p>サンプルコード:(注: asReaderWiredBOXType は AsReaderWiredBOXType クラスのインスタンスオブジェクトです。)</p> <pre>BOOL status = [asReaderWiredBOXType getRechargedStatus]; if (status) { //充電のステータスの取得に成功 } else { //充電のステータスの取得に失敗 }</pre>				

2.2.15. getSession

- (BOOL)getSession;				
	引数名	In/Out	型	説明
戻り値	-	Out	BOOL	YES: コマンドの送信に成功 NO: コマンドの送信に失敗
<p>メソッド説明: RFID 設定の session 値を取得します。このメソッドを実行した後、「receivedSession」(2.3.9を参照)デリゲートがコールバックされます。</p> <p>サンプルコード:(注: asReaderWiredBOXType は AsReaderWiredBOXType クラスのインスタンスオブジェクトです。)</p> <pre>BOOL status = [asReaderWiredBOXType getSession]; if (status) { //コマンドの送信に成功</pre>				

```

} else {
//コマンドの送信に失敗
}

```

2.2.16. setSession

- (BOOL)setSession:(uint8_t)session;

	引数名	In/Out	型	説明
引数	session	In	uint8_t	S0(0x00)、S1(0x01)、S2(0x02)、S3(0x03)
戻り値	-	Out	BOOL	YES: コマンドの送信に成功 NO: コマンドの送信に失敗

メソッド説明:

RFID 設定の session 値を設定します。

このメソッドを実行した後、「didSetSessionStatus」([2.3.10](#)を参照)デリゲートがコールバックされます。

サンプルコード:(注: asReaderWiredBOXType は AsReaderWiredBOXType クラスのインスタンスオブジェクトです。)

```

BOOL status = [asReaderWiredBOXType setSession:0];
if (status) {
//コマンドの送信に成功
} else {
//コマンドの送信に失敗
}

```

2.2.17. getOutputPowerLevelForMultiAntenna

- (BOOL)getOutputPowerLevelForMultiAntenna;

	引数名	In/Out	型	説明
戻り値	-	Out	BOOL	YES: コマンドの送信に成功 NO: コマンドの送信に失敗

メソッド説明:

RFID 設定の各アンテナの出力値を取得します。

このメソッドを実行した後、「receivedOutputPowerAntenna1」([2.3.11](#)を参照)デリゲートがコールバックされます。

サンプルコード:(注: asReaderWiredBOXType は AsReaderWiredBOXType クラスのインスタンスオブジェクトです。)

```

BOOL status = [asReaderWiredBOXType getOutputPowerLevelForMultiAntenna];

```

```

if (status) {
//コマンドの送信に成功
} else {
//コマンドの送信に失敗
}

```

2.2.18. setOutputPowerLevel

```

-(BOOL)setOutputPowerLevel Antenna1:(uint16_t)powerLevel1
                          Antenna2:(uint16_t)powerLevel2
                          Antenna3:(uint16_t)powerLevel3
                          Antenna4:(uint16_t)powerLevel4
                          Antenna5:(uint16_t)powerLevel5
                          Antenna6:(uint16_t)powerLevel6
                          Antenna7:(uint16_t)powerLevel7
                          Antenna8:(uint16_t)powerLevel8;

```

	引数名	In/Out	型	説明
引数	powerLevel1	In	uint16_t	アンテナ 1 の出力
引数	powerLevel2	In	uint16_t	アンテナ2の出力
引数	powerLevel3	In	uint16_t	アンテナ3の出力
引数	powerLevel4	In	uint16_t	アンテナ4の出力
引数	powerLevel5	In	uint16_t	アンテナ5の出力
引数	powerLevel6	In	uint16_t	アンテナ6の出力
引数	powerLevel7	In	uint16_t	アンテナ7の出力
引数	powerLevel8	In	uint16_t	アンテナ8の出力
戻り値	-	Out	BOOL	YES: コマンドの送信に成功 NO: コマンドの送信に失敗

メソッド説明:

RFID 設定のアンテナの出力値を設定します。

このメソッドを実行した後、「didSetOutputPowerForMultiAntenna」([2.3.12](#)を参照)デリゲートがコールバックされます。

サンプルコード:(注: asReaderWiredBOXType は AsReaderWiredBOXType クラスのインスタンスオブジェクトです。)

```

BOOL status = [asReaderWiredBOXType setOutputPowerLevel Antenna1:10
                                           Antenna2:10
                                           Antenna3:10
                                           Antenna4:10

```

```

Antenna5:10
Antenna6:10
Antenna7:10
Antenna8:10];

if (status) {
//コマンドの送信に成功
} else {
//コマンドの送信に失敗
}

```

2.2.19. getOutputPowerLevel

- (BOOL)getOutputPowerLevel;				
	引数名	In/Out	型	説明
戻り値	-	Out	BOOL	YES:コマンドの送信に成功 NO: コマンドの送信に失敗
<p>メソッド説明: RFID 設定の現在の各アンテナの共通出力値、及び設定できる最大および最小の出力値を取得します。 このメソッドを実行した後、「receivedOutputPowerLevel」(2.3.13を参照)デリゲートがコールバックされます。</p> <p>サンプルコード:(注: asReaderWiredBOXType は AsReaderWiredBOXType クラスのインスタンスオブジェクトです。) BOOL status = [asReaderWiredBOXType getOutputPowerLevel]; if (status) { //コマンドの送信に成功 } else { //コマンドの送信に失敗 } </p>				

2.2.20. setOutputPowerLevel

- (BOOL)setOutputPowerLevel:(uint16_t)power;				
	引数名	In/Out	型	説明
引数	power	In	uint16_t	出力 日本版の出力範囲:18 ~ 23dbm 日本版以外の出力範囲:18 ~ 27dbm
戻り値	-	Out	BOOL	YES:コマンドの送信に成功

				NO: コマンドの送信に失敗
<p>メソッド説明: RFID 設定の各アンテナ共通の出力値を設定します。 このメソッドを実行した後、「didSetOutputPowerLevelStatus」(2.3.14を参照)デリゲートがコールバックされます。</p> <p>サンプルコード:(注: asReaderWiredBOXType は AsReaderWiredBOXType クラスのインスタンスオブジェクトです。)</p> <pre> BOOL status = [asReaderWiredBOXType setOutputPowerLevel:20]; if (status) { //コマンドの送信に成功 } else { //コマンドの送信に失敗 } </pre>				

2.2.21. getFHLBTParam

- (BOOL)getFHLBTParam;				
	引数名	In/Out	型	説明
戻り値	-	Out	BOOL	YES: コマンドの送信に成功 NO: コマンドの送信に失敗
<p>メソッド説明: RFID 設定の FH および LBT の設定値を取得します。 このメソッドを実行した後、「receivedFHLBTData」(2.3.15を参照)デリゲートがコールバックされます。</p> <p>サンプルコード:(注: asReaderWiredBOXType は AsReaderWiredBOXType クラスのインスタンスオブジェクトです。)</p> <pre> BOOL status = [asReaderWiredBOXType getFHLBTParam]; if (status) { //コマンドの送信に成功 } else { //コマンドの送信に失敗 } </pre>				

2.2.22. setFHLBTParam

<p>- (BOOL)setFHLBTParam ReadTime:(uint16_t)readTime idleTime:(uint16_t)idleTime carrierSenseTime:(uint16_t)carrierSenseTime rfLevel:(uint16_t)rfLevel</p>
--

frequencyHopping:(uint8_t)frequencyHopping listenBeforeTalk:(uint8_t)listenBeforeTalk continuousWave:(uint8_t)continuousWave;				
	引数名	In/Out	型	説明
引数	readTime	In	uint16_t	読み取り時間(単位:ms)
引数	idleTime	In	uint16_t	アイドル時間(単位:ms)
引数	carrierSenseTime	In	uint16_t	キャリアセンス時間、固定値:50
引数	rfLevel	In	int	キャリアセンス RF レベル、固定値:-740
引数	frequencyHopping	In	uint8_t	FH 有効(0x01 以上);FH 無効(0x00)
引数	listenBeforeTalk	In	uint8_t	LBT 有効(0x01 以上);LBT 無効(0x00)
引数	continuousWave	In	uint8_t	CW、固定値:0x00
戻り値	-	Out	BOOL	YES:コマンドの送信に成功 NO: コマンドの送信に失敗

メソッド説明:
RFID 設定の FH および LBT パラメータを設定します。
このメソッドを実行した後、「didSetFHLBTStatus」([2.3.16](#)を参照)デリゲートがコールバックされます。
注:Hopping を優先に設定するなら、パラメーターfrequencyHopping を 0x02 に、パラメーターlistenBeforeTalk を 0x01 に設定する必要があります。
LBT を優先に設定するなら、パラメーターfrequencyHopping を 0x01 に、パラメーターlistenBeforeTalk を 0x02 に設定する必要があります

サンプルコード:(注:asReaderWiredBOXType は AsReaderWiredBOXType クラスのインスタンスオブジェクトです。)

```

BOOL status = [asReaderWiredBOXType setFHLBTParam ReadTime:10
                                     idleTime:10
                                     carrierSenseTime:50
                                     rfLevel:-749
                                     frequencyHopping:1
                                     listenBeforeTalk:2
                                     continuousWave:0];

if (status) {
//コマンドの送信に成功。
} else {
//コマンドの送信に失敗。
}

```

2.2.23. getReadTimeForMultiAntenna

- (BOOL)getReadTimeForMultiAntenna;				
	引数名	In/Out	型	説明
戻り値	-	Out	BOOL	YES: コマンドの送信に成功 NO: コマンドの送信に失敗
<p>メソッド説明: RFID 設定の各アンテナの読み取り時間を取得します。 このメソッドを実行した後、「readTimeForMultiAntennaReceivedAnt1」(2.3.17を参照)デリゲートがコールバックされます。</p> <p>サンプルコード:(注: asReaderWiredBOXType は AsReaderWiredBOXType クラスのインスタンスオブジェクトです。)</p> <pre> BOOL status = [asReaderWiredBOXType getReadTimeForMultiAntenna]; if (status) { //コマンドの送信に成功。 } else { //コマンドの送信に失敗。 } </pre>				

2.2.24. setReadTimeForMultiAntennaReceived

- (BOOL)setReadTimeForMultiAntennaReceived Ant1:(uint16_t)ant1 Ant2:(uint16_t)ant2 Ant3:(uint16_t)ant3 Ant4:(uint16_t)ant4 Ant5:(uint16_t)ant5 Ant6:(uint16_t)ant6 Ant7:(uint16_t)ant7 Ant8:(uint16_t)ant8;				
	引数名	In/Out	型	説明
引数	ant1	In	uint16_t	アンテナ 1 の読み取り時間
引数	ant2	In	uint16_t	アンテナ 2 の読み取り時間
引数	ant3	In	uint16_t	アンテナ 3 の読み取り時間
引数	ant4	In	uint16_t	アンテナ 4 の読み取り時間
引数	ant5	In	uint16_t	アンテナ 5 の読み取り時間
引数	ant6	In	uint16_t	アンテナ 6 の読み取り時間
引数	ant7	In	uint16_t	アンテナ 7 の読み取り時間
引数	ant8	In	uint16_t	アンテナ 8 の読み取り時間
戻り値	-	Out	BOOL	YES: コマンドの送信に成功 NO: コマンドの送信に失敗
<p>メソッド説明: RFID 設定の各アンテナの読み取り時間を設定します。 このメソッドを実行した後、「didSetReadTimeForMultiAntenna」(2.3.18を参照)デリゲートがコールバックされます。</p> <p>サンプルコード:(注: asReaderWiredBOXType は AsReaderWiredBOXType クラスのインスタンスオブジェクトです。)</p> <pre> BOOL status = [asReaderWiredBOXType setReadTimeForMultiAntennaReceivedAnt1:10 Ant2:10 Ant3:10 Ant4:10 Ant5:10 Ant6:10 Ant7:10 Ant8:10]; if (status) { //コマンドの送信に成功。 } else { //コマンドの送信に失敗。 } </pre>				

2.2.25. setDHCPOnMac

-(BOOL)setDHCPOnMac:(NSString*)macAddr;				
	引数名	In/Out	型	説明
引数	macAddr	In	NSString	MAC アドレス
戻り値	-	Out	BOOL	YES: コマンドの送信に成功 NO: コマンドの送信に失敗
メソッド説明: ダイナミックな IP アドレスを設定します。 サンプルコード: (注: asReaderWiredBOXType は AsReaderWiredBOXType クラスのインスタンスオブジェクトです。) <pre> BOOL status = [asReaderWiredBOXType setDHCPOnMac:@"AC:BC:32:02:8D:32"]; if (status) { //コマンドの送信に成功。 } else { //コマンドの送信に失敗。 } </pre>				

2.2.26. setDHCPOffMac

-(BOOL)setDHCPOffMac:(NSString*)macAddr IP:(NSString*)ip Subnet:(NSString*)subnet Gateway:(NSString*)gateway DNSServer:(NSString*)dnsServer;				
	引数名	In/Out	型	説明
引数	macAddr	In	NSString	MAC アドレス
引数	ip	In	NSString	IP アドレス
引数	subnet	In	NSString	サブネットマスク
引数	gateway	In	NSString	ゲートウェイ
引数	dnsServer	In	NSString	ドメインネームシステムサーバー
戻り値	-	Out	BOOL	YES: コマンドの送信に成功 NO: コマンドの送信に失敗
メソッド説明: 静的な IP アドレス、ネットワーク情報を設定します。 サンプルコード: (注: asReaderWiredBOXType は AsReaderWiredBOXType クラスのインスタンスオブジェクトです。) <pre> [asReaderWiredBOXType setDHCPOffMac:@"AC:BC:32:02:8D:32"]; </pre>				

```

BOOL status = [asReaderWiredBOXType setDHCPOffMac:@"AC:BC:32:02:8D:32"
IP:@"192.168.1.10" Subnet:@"255.255.255.255" Gateway:@"255.255.255.255"
DNSServer:@"255.255.255.255"];
if (status) {
//コマンドの送信に成功
} else {
//コマンドの送信に失敗
}

```

2.2.27. tX

- (BOOL)tX:(NSData *)data;				
	引数名	In/Out	型	説明
引数	data	In	NSData	データコマンド
戻り値	-	Out	BOOL	YES: コマンドの送信に成功 NO: コマンドの送信に失敗
メソッド説明: Wired BOX-Type デバイスにデータコマンドを送信します。 サンプルコード: (注: asReaderWiredBOXType は AsReaderWiredBOXType クラスのインスタンスオブジェクトです。) <pre> Byte buf[6] = {0x00,0xB1,0x00,0x00,0x7E,0x30}; NSData *data = [NSData dataWithBytes:buf length:8]; BOOL status = [asReaderWiredBOXType tx:data]; if (status) { //コマンドの送信に成功 } else { //コマンドの送信に失敗 } </pre>				

2.2.28. getFirmwareVersion

- (BOOL)getFirmwareVersion;				
	引数名	In/Out	型	説明
戻り値	-	Out	BOOL	YES:ファームウェアバージョンの取得に成功 NO: ファームウェアバージョンの取得に失敗
<p>メソッド説明: Wired BOX-Type デバイスのファームウェアバージョンを取得します。 このメソッドを実行した後、「rfidModuleVersionReceived」(2.3.20を参照)デリゲートがコールバックされます。</p> <p>サンプルコード:(注: asReaderWiredBOXType は AsReaderWiredBOXType クラスのインスタンスオブジェクトです。)</p> <pre> BOOL status = [asReaderWiredBOXType getFirmwareVersion]; if (status) { //ファームウェアバージョンの取得に成功 } else { //ファームウェアバージョンの取得に失敗 } </pre>				

2.3. デリゲート

2.3.1. searchingIP

-(void)AsReaderWiredBOXType:(AsReaderWiredBOXType*)AsReaderWiredBOXType searchingIP:(NSString*)ip isFinish:(BOOL)isFinish;				
	引数名	In/Out	型	説明
引数	AsReaderWiredBOXType	Out	AsReaderWiredBOXType	AsReaderWiredBOXType オブジェクト
引数	ip	Out	NSString	デバイスの IP アドレス
引数	isFinish	Out	BOOL	検索状態。 YES: 検索が完了 NO: 検索を続行する

メソッド説明:
 WiredBOX-Type デバイスの IP アドレスを受け取ります。
 startSearchDevice ([2.2.3](#)を参照)メソッドを呼び出した後、コールバックされます。

サンプルコード:

```

NSMutableArray *arrayIPs = [[NSMutableArray alloc] init];
- (void)AsReaderWiredBOXType:(AsReaderWiredBOXType *)AsReaderWiredBOXType
searchingIP:(NSString*)ip isFinish:(BOOL)isFinish{
    if(isFinish == YES) {
        //検索が完了
    } else {
        //検索を続行する
        if (![arrayIPs containsObject:ip]) {
            [arrayIPs addObject:ip];
        }
    }
}

```

2.3.2. networkStateChanged

-(void)AsReaderWiredBOXType:(AsReaderWiredBOXType*)AsReaderWiredBOXType networkStateChanged:(AsReaderWiredBOXTypeSDKBOXNetworkState)state error:(NSError *)error;				
	引数名	In/Out	型	説明

引数	AsReaderWiredBOX Type	Out	AsReaderWiredBOX Type	AsReaderWiredBOXType オブジェクト
引数	state	Out	AsReaderWiredBOX TypeSDKBOXNetworkState	AsReaderWiredBOXType SDKBOXNetworkState 列挙(4.1.2を参照)
引数	error	Out	NSError	エラーメッセージ

メソッド説明:
 WiredBOX-Type デバイスの接続状態が変化した場合、コールバックされます。
 connectServer (2.2.4を参照)、disconnectServer (2.2.5を参照)メソッドを呼び出した後、コールバックされます。

サンプルコード:

```

- (void)AsReaderWiredBOXType:(AsReaderWiredBOXType *)AsReaderWiredBOXType
networkStateChanged:(AsReaderWiredBOXTypeSDKBOXNetworkState)state
error:(NSError *)error{
    if (state == AsReaderWiredBOXTypeSDKBOXNetworkStateConnected) {
        //接続に成功
    } else if (state == AsReaderWiredBOXTypeSDKBOXNetworkStateDisconnected) {
        //接続を切断する
    } else if (state == AsReaderWiredBOXTypeSDKBOXNetworkStateRecievedData) {
        //データを受信した
    } else {
        //error エラー発生
    }
}
    
```

2.3.3. searchedUPDSearchingDeviceInfo

-				
(void)AsReaderWiredBOXType:(AsReaderWiredBOXType*)AsReaderWiredBOXType searchedUPDSearchingDeviceInfo:(NSArray *)arrInfos;				
	引数名	In/Out	型	説明
引数	AsReaderWiredBOX Type	Out	AsReaderWiredBOX Type	AsReaderWiredBOXType オブジェクト
引数	arrInfos	Out	NSArray	デバイスの IP アドレスの配列

メソッド説明:
 取得した WiredBOX-Type デバイスの IP アドレスを受け取ります。
 startSearchDevice (2.2.3を参照)メソッドを呼び出した後、コールバックされます。

サンプルコード:

```
- (void)AsReaderWiredBOXType:(AsReaderWiredBOXType *)AsReaderWiredBOXType
searchedUPDSearchingDeviceInfo:(NSArray *)arrInfos {
// arrInfos デバイスの IP アドレスの配列
}
```

2.3.4. inventoryStartedWithStatus

```
-
(void)AsReaderWiredBOXType:(AsReaderWiredBOXType*)AsReaderWiredBOXType
inventoryStartedWithStatus:(AsReaderWiredBOXTypeSDKBOXStatusCode)statusCode;
```

	引数名	In/Out	型	説明
引数	AsReaderWiredBOXType	Out	AsReaderWiredBOXType	AsReaderWiredBOXType オブジェクト
引数	statusCode	Out	AsReaderWiredBOXTypeSDKBOXStatusCode	AsReaderWiredBOXType SDKBOXStatusCode 列挙 (4.1.9 を参照)

メソッド説明:

インベントリ開始のステータスを受け取ります。

startInventoryWithTagsMaxTagCount ([2.2.10](#)を参照) メソッドを呼び出した後、コールバックされます。

サンプルコード:

```
- (void)AsReaderWiredBOXType:(AsReaderWiredBOXType *)AsReaderWiredBOXType
inventoryStartedWithStatus:(AsReaderWiredBOXTypeSDKBOXStatusCode)statusCode {
// statusCode AsReaderWiredBOXTypeSDKBOXStatusCode 列挙(4.1.9を参照)。
}
```

2.3.5. inventoryStoppedWithStatus

```
-
(void)AsReaderWiredBOXType:(AsReaderWiredBOXType*)AsReaderWiredBOXType
inventoryStoppedWithStatus:(AsReaderWiredBOXTypeSDKBOXStatusCode)statusCode;
```

	引数名	In/Out	型	説明
引数	AsReaderWiredBOXType	Out	AsReaderWiredBOXType	AsReaderWiredBOXType オブジェクト

引数	statusCode	Out	AsReaderWiredBOXTypeSDKBOXStatus usCode	AsReaderWiredBOXTypeSDKBOXStatus Code 列挙 (4.1.9 を参照)
<p>メソッド説明: インベントリ停止のステータスを受け取ります。 stopInventory(2.2.11を参照)メソッドを呼び出した後、コールバックされます。</p> <p>サンプルコード:</p> <pre>- (void)AsReaderWiredBOXType:(AsReaderWiredBOXType *)AsReaderWiredBOXType inventoryStoppedWithStatus:(AsReaderWiredBOXTypeSDKBOXStatus)statusCode { // statusCode AsReaderWiredBOXTypeSDKBOXStatus 列挙(4.1.9を参照) }</pre>				

2.3.6. receivedPCEPCData

<pre>-(void)AsReaderWiredBOXType:(AsReaderWiredBOXType *)AsReaderWiredBOXType receivedPCEPCData:(NSData*)pcEpc selectAntenna:(AsReaderWiredBOXTypeSDKBOXSelectAntenna)antenna;</pre>				
	引数名	In/Out	型	説明
引数	AsReaderWiredBOXType	Out	AsReaderWiredBOXType	AsReaderWiredBOXType オブジェクト
引数	pcEpc	Out	NSData	読み取った RFID データ
引数	antenna	Out	AsReaderWiredBOXTypeSDKBOXSelectAntenna	AsReaderWiredBOXTypeSDKBOXSelectAntenna 列挙(4.1.3 を参照)
<p>メソッド説明: 読み取った RFID データを受け取ります。 startInventoryWithTagsMaxTagCount(2.2.10を参照)メソッドを呼び出した後、コールバックされます。</p> <p>サンプルコード:</p> <pre>- (void)AsReaderWiredBOXType:(AsReaderWiredBOXType *)AsReaderWiredBOXType receivedPCEPCData:(NSData *)pcEpc selectAntenna:(AsReaderWiredBOXTypeSDKBOXSelectAntenna)antenna { // pcEpc 読み取った RFID データ。 // antenna AsReaderWiredBOXTypeSDKBOXSelectAntenna 列挙(4.1.3を参照) }</pre>				

2.3.7. updatedRegistryStatus

-(void)AsReaderWiredBOXType:(AsReaderWiredBOXType *)AsReaderWiredBOXType updatedRegistryStatus:(AsReaderWiredBOXTypeSDKBOXStatusCode)statusCode;				
	引数名	In/Out	型	説明
引数	AsReaderWiredBOXType	Out	AsReaderWiredBOXType	AsReaderWiredBOXType オブジェクト
引数	statusCode	Out	AsReaderWiredBOXTypeSDKBOXStatusCode	AsReaderWiredBOXType SDKBOXStatusCode 列挙 (4.1.9 を参照)

メソッド説明:
 RFID 設定を恒久的に保存した後にコールバックされます。
 updateRegistry ([2.2.12](#)を参照)メソッドを呼び出した後、コールバックされます。

サンプルコード:

```

- (void)AsReaderWiredBOXType:(AsReaderWiredBOXType *)AsReaderWiredBOXType
updatedRegistryStatus:(AsReaderWiredBOXTypeSDKBOXStatusCode)statusCode {
//statusCode AsReaderWiredBOXTypeSDKBOXStatusCode 列挙(4.1.9を参照)
}
  
```

2.3.8. HIDDataRecived

-(void)AsReaderWiredBOXType:(AsReaderWiredBOXType *)AsReaderWiredBOXType HIDDataRecived:(NSData *)HIDData;				
	引数名	In/Out	型	説明
引数	AsReaderWiredBOXType	Out	AsReaderWiredBOXType	AsReaderWiredBOXType オブジェクト
引数	HIDData	Out	NSData	HID インターフェイスデータ

メソッド説明:
 HID 入力デバイスから読み取られた HID データを受け取ります。

サンプルコード:

```

- (void)AsReaderWiredBOXType:(AsReaderWiredBOXType *)AsReaderWiredBOXType
HIDDataRecived:(NSData *)HIDData {
// HIDData HID インターフェイスデータ
}
  
```

2.3.9. receivedSession

-(void)AsReaderWiredBOXType:(AsReaderWiredBOXType *)AsReaderWiredBOXType receivedSession:(uint8_t)session;				
	引数名	In/Out	型	説明
引数	AsReaderWiredBOXType	Out	AsReaderWiredBOXType	AsReaderWiredBOXType オブジェクト
引数	session	Out	uint8_t	設定した RFID の Session 値
メソッド説明: session 値を受け取ります。 getSession (2.2.15 を参照) メソッドを呼び出した後、コールバックされます。 サンプルコード: <pre> - (void)AsReaderWiredBOXType:(AsReaderWiredBOXType *)AsReaderWiredBOXType receivedSession:(uint8_t)session { // session 設定した RFID の Session 値 } </pre>				

2.3.10. didSetSessionStatus

-(void)AsReaderWiredBOXType:(AsReaderWiredBOXType *)AsReaderWiredBOXType didSetSessionStatus:(AsReaderWiredBOXTypeSDKBOXStatusCode)state;				
	引数名	In/Out	型	説明
引数	AsReaderWiredBOXType	Out	AsReaderWiredBOXType	AsReaderWiredBOXType オブジェクト
引数	state	Out	AsReaderWiredBOXTypeSDKBOXStatusCode	AsReaderWiredBOXType SDKBOXStatusCode 列挙 (4.1.9 を参照)
メソッド説明: session 設定の実行結果を受け取ります。 setSession (2.2.16 を参照) メソッドを呼び出した後、コールバックされます。 サンプルコード: <pre> - (void)AsReaderWiredBOXType:(AsReaderWiredBOXType *)AsReaderWiredBOXType didSetSessionStatus:(AsReaderWiredBOXTypeSDKBOXStatusCode)state { // state AsReaderWiredBOXTypeSDKBOXStatusCode 列挙(4.1.9 を参照) } </pre>				

2.3.11. receivedOuputPowerAntenna1

```
-(void)AsReaderWiredBOXType:(AsReaderWiredBOXType
*)AsReaderWiredBOXType receivedOuputPowerAntenna1:(uint16_t)antenna1
    Antenna2:(uint16_t)antenna2
    Antenna3:(uint16_t)antenna3
    Antenna4:(uint16_t)antenna4
    Antenna5:(uint16_t)antenna5
    Antenna6:(uint16_t)antenna6
    Antenna7:(uint16_t)antenna7
    Antenna8:(uint16_t)antenna8
    Min:(uint16_t)min
    Max:(uint16_t)max;
```

	引数名	In/Out	型	説明
引数	AsReaderWiredBOXType	Out	AsReaderWiredBOXType	AsReaderWiredBOXType オブジェクト
引数	antenna1	Out	uint16_t	アンテナ 1 の出力
引数	antenna2	Out	uint16_t	アンテナ 2 の出力
引数	antenna3	Out	uint16_t	アンテナ 3 の出力
引数	antenna4	Out	uint16_t	アンテナ 4 の出力
引数	antenna5	Out	uint16_t	アンテナ 5 の出力
引数	antenna6	Out	uint16_t	アンテナ 6 の出力
引数	antenna7	Out	uint16_t	アンテナ 7 の出力
引数	antenna8	Out	uint16_t	アンテナ 8 の出力
引数	min	Out	uint16_t	最小出力
引数	max	Out	uint16_t	最大出力

メソッド説明:

各アンテナの出力値を受け取ります。

getOutputPowerLevelForMultiAntenna ([2.2.17](#) を参照) メソッドを呼び出した後、コールバックされます。

サンプルコード:

```
-(void)AsReaderWiredBOXType:(AsReaderWiredBOXType *)AsReaderWiredBOXType
receivedOuputPowerAntenna1:(uint16_t)antenna1 Antenna2:(uint16_t)antenna2
    Antenna3:(uint16_t)antenna3 Antenna4:(uint16_t)antenna4 Antenna5:(uint16_t)antenna5
    Antenna6:(uint16_t)antenna6 Antenna7:(uint16_t)antenna7 Antenna8:(uint16_t)antenna8
    Min:(uint16_t)min Max:(uint16_t)max {
// antenna1 アンテナ 1 の出力
```

```
// antenna2 アンテナ 2 の出力
// antenna3 アンテナ 3 の出力
// antenna4 アンテナ 4 の出力
// antenna5 アンテナ 5 の出力
// antenna6 アンテナ 6 の出力
// antenna7 アンテナ 7 の出力
// antenna8 アンテナ 8 の出力
// min      最小出力
// max      最大出力
}
```

2.3.12. didSetOutputPowerForMultiAntenna

```
-(void)AsReaderWiredBOXType:(AsReaderWiredBOXType *)AsReaderWiredBOXType didSetOutputPowerForMultiAntenna:(AsReaderWiredBOXTypeSDKCommonStatus)statusCode;
```

	引数名	In/Out	型	説明
引数	AsReaderWiredBOXType	Out	AsReaderWiredBOXType	AsReaderWiredBOXType オブジェクト
引数	state	Out	AsReaderWiredBOXTypeSDKCommonStatus	AsReaderWiredBOXType SDKCommonStatus 列挙 (4.1.10 を参照)

メソッド説明:

各アンテナの出力値設定の実行結果を受け取ります。

setOutputPowerLevelAntenna1 ([2.2.18](#)を参照)メソッドを呼び出した後、コールバックされます。

サンプルコード:

```
-(void)AsReaderWiredBOXType:(AsReaderWiredBOXType *)AsReaderWiredBOXType didSetOutputPowerForMultiAntenna:(AsReaderWiredBOXTypeSDKCommonStatus)statusCode {
// statusCode AsReaderWiredBOXTypeSDKCommonStatus 列挙(4.1.10を参照)
}
```

2.3.13. receivedOutputPowerLevel

-(void)AsReaderWiredBOXType:(AsReaderWiredBOXType *)AsReaderWiredBOXType receivedOutputPowerLevel:(uint16_t)powerLevel maxPower:(uint16_t)maxPower minPower:(uint16_t)minPower;				
	引数名	In/Out	型	説明
引数	AsReaderWiredBOXType	Out	AsReaderWiredBOXType	AsReaderWiredBOXType オブジェクト
引数	powerLevel	Out	uint16_t	現在 RFID の出力
引数	maxPower	Out	uint16_t	設定可能な最大出力
引数	minPower	Out	uint16_t	設定可能な最小出力
<p>メソッド説明: アンテナの共通出力値と設定可能な出力値を受け取ります。 getOutputPowerLevel(2.2.19を参照)メソッドを呼び出した後、コールバックされます。</p> <p>サンプルコード:</p> <pre> - (void)AsReaderWiredBOXType:(AsReaderWiredBOXType *)AsReaderWiredBOXType receivedOutputPowerLevel:(uint16_t)powerLevel maxPower:(uint16_t)maxPower minPower:(uint16_t)minPower { // powerLevel 現在 RFID の出力 // maxPower 設定可能な最大出力 // minPower 設定可能な最小出力 } </pre>				

2.3.14. didSetOutputPowerLevelStatus

-(void)AsReaderWiredBOXType:(AsReaderWiredBOXType *)AsReaderWiredBOXType didSetOutputPowerLevelStatus:(AsReaderWiredBOXType SDKBOXStatusCode)statusCode;				
	引数名	In/Out	型	説明
引数	AsReaderWiredBOXType	Out	AsReaderWiredBOXType	AsReaderWiredBOXType オブジェクト
引数	statusCode	Out	AsReaderWiredBOXType SDKBOXStatusCode	AsReaderWiredBOXType SDKBOXStatusCode 列挙 (4.1.9 を参照)
<p>メソッド説明: アンテナの共通出力設定の実行結果を受け取ります。 setOutputPowerLevel(2.2.20を参照)メソッドを呼び出した後、コールバックされます。</p> <p>サンプルコード:</p>				

```
- (void)AsReaderWiredBOXType:(AsReaderWiredBOXType *)AsReaderWiredBOXType
didSetOutputPowerLevelStatus:(AsReaderWiredBOXTypeSDKBOXStatusCode)statusCode {
// statusCode AsReaderWiredBOXTypeSDKBOXStatusCode 列挙(4.1.9を参照)
}
```

2.3.15. receivedFHLBTData

```
-(void)AsReaderWiredBOXType:(AsReaderWiredBOXType *)AsReaderWiredBOXType
receivedFHLBTData:(NSData*)data
readTime:(int)readTime idleTime:(int)idleTime;
```

	引数名	In/Out	型	説明
引数	AsReaderWiredBOXType	Out	AsReaderWiredBOXType	AsReaderWiredBOXType オブジェクト
引数	data	Out	NSData	読み取り時間(16bit) アイドル時間(16bit) キャリアセンス時間(16bit) キャリアセンス RF レベル(16bit) FH(8bit) LBT(8bit) CW(8bit)
引数	readTime	Out	int	読み取り時間
引数	idleTime	Out	int	アイドル時間

メソッド説明:

FH と LBT 値を受け取ります。

getFHLBTParam(2.2.21を参照)メソッドを呼び出した後、コールバックされます。

サンプルコード:

```
- (void)AsReaderWiredBOXType:(AsReaderWiredBOXType *)AsReaderWiredBOXType
receivedFHLBTData:(NSData *)data readTime:(int)readTime idleTime:(int)idleTime {
// data
読み取り時間(16bit)
アイドル時間(16bit)
キャリアセンス時間(16bit)
キャリアセンス RF レベル(16bit)
FH(8bit)
LBT(8bit)
CW(8bit)
// readTime
```

```
読み取り時間
// idleTime
アイドル時間
}
```

2.3.16. didSetFHLBTStatus

```
-(void)AsReaderWiredBOXType:(AsReaderWiredBOXType
*)AsReaderWiredBOXTypedidSetFHLBTStatus:(AsReaderWiredBOXTypeSDKBOXSt
atusCode)statusCode;
```

	引数名	In/Out	型	説明
引数	AsReaderWiredBOXType	Out	AsReaderWiredBOXType	AsReaderWiredBOXType オブジェクト
引数	statusCode	Out	AsReaderWiredBOXTypeSDKBOXStatus Code	AsReaderWiredBOXType SDKBOXStatusCode 列挙 (4.1.9 を参照)

メソッド説明:

FH と LBT 設定の実行結果を受け取ります。

setFHLBTParamReadTime ([2.2.22](#)を参照)メソッドを呼び出した後、コールバックされます。

サンプルコード:

```
- (void)AsReaderWiredBOXType:(AsReaderWiredBOXType *)AsReaderWiredBOXType
didSetFHLBTStatus:(AsReaderWiredBOXTypeSDKBOXStatusCode)statusCode {
// statusCode AsReaderWiredBOXTypeSDKBOXStatusCode 列挙(4.1.9を参照)
}
```

2.3.17. readTimeForMultiAntennaReceivedAnt1

```
- (void)AsReaderWiredBOXType:(AsReaderWiredBOXType
*)AsReaderWiredBOXTypedreadTimeForMultiAntennaReceivedAnt1:(uint16_t)ant1An
t2:(uint16_t)ant2Ant3:(uint16_t)ant3Ant4:(uint16_t)ant4Ant5:(uint16_t)ant5Ant6:(uint
16_t)ant6 Ant7:(uint16_t)ant7 Ant8:(uint16_t)ant8;
```

	引数名	In/Out	型	説明
引数	AsReaderWiredBOXType	Out	AsReaderWiredBOXType	AsReaderWiredBOXType オブジェクト
引数	ant1	Out	uint16_t	アンテナ 1 の読み取り時間
引数	ant2	Out	uint16_t	アンテナ 2 の読み取り時間
引数	ant3	Out	uint16_t	アンテナ 3 の読み取り時間

引数	ant4	Out	uint16_t	アンテナ 4 の読み取り時間
引数	ant5	Out	uint16_t	アンテナ 5 の読み取り時間
引数	ant6	Out	uint16_t	アンテナ 6 の読み取り時間
引数	ant7	Out	uint16_t	アンテナ 7 の読み取り時間
引数	ant8	Out	uint16_t	アンテナ 8 の読み取り時間

メソッド説明:
 各アンテナの読み取り時間を受け取ります。
 getReadTimeForMultiAntenna ([2.2.23](#)を参照)メソッドを呼び出した後、コールバックされます。
サンプルコード:
 - (void)AsReaderWiredBOXType:(AsReaderWiredBOXType *)AsReaderWiredBOXType
 readTimeForMultiAntennaReceivedAnt1:(uint16_t)ant1 Ant2:(uint16_t)ant2
 Ant3:(uint16_t)ant3 Ant4:(uint16_t)ant4 Ant5:(uint16_t)ant5 Ant6:(uint16_t)ant6
 Ant7:(uint16_t)ant7 Ant8:(uint16_t)ant8 {
 // ant1 アンテナ 1 の読み取り時間
 // ant2 アンテナ 2 の読み取り時間
 // ant3 アンテナ 3 の読み取り時間
 // ant4 アンテナ 4 の読み取り時間
 // ant5 アンテナ 5 の読み取り時間
 // ant6 アンテナ 6 の読み取り時間
 // ant7 アンテナ 7 の読み取り時間
 // ant8 アンテナ 8 の読み取り時間
 }

2.3.18. didSetReadTimeForMultiAntenna

- (void)AsReaderWiredBOXType:(AsReaderWiredBOXType *)AsReaderWiredBOXTypedidSetReadTimeForMultiAntenna:(AsReaderWiredBOXTypeSDKCommonStatus)status;				
	引数名	In/Out	型	説明
引数	AsReaderWiredBOXType	Out	AsReaderWiredBOXType	AsReaderWiredBOXType オブジェクト
引数	status	Out	AsReaderWiredBOXTypeSDKCommonStatus	AsReaderWiredBOXType SDKCommonStatus 列挙 (4.1.10 を参照)

メソッド説明:
 各アンテナ読み取り時間設定の実行結果を受け取ります。

setReadTimeForMultiAntennaReceivedAnt1 ([2.2.24](#)を参照)メソッドを呼び出した後、コールバックされます。

サンプルコード:

```
- (void)AsReaderWiredBOXType:(AsReaderWiredBOXType *)AsReaderWiredBOXType
didSetReadTimeForMultiAntenna:(AsReaderWiredBOXTypeSDKCommonStatus)status {
// status AsReaderWiredBOXTypeSDKCommonStatus 列挙(4.1.10を参照)
}
```

2.3.19. setRecharging

- (void)AsReaderWiredBOXType:(AsReaderWiredBOXType *)AsReaderWiredBOXType
setRecharging:(AsReaderWiredBOXTypeSDKBOXStatusCode)statusCode;

	引数名	In/Out	型	説明
引数	AsReaderWiredBOXType	Out	AsReaderWiredBOXType	AsReaderWiredBOXType オブジェクト
引数	statusCode	Out	AsReaderWiredBOXTypeSDKBOXStatusCode	AsReaderWiredBOXType SDKBOXStatusCode 列挙 (4.1.9 を参照)

メソッド説明:

WiredBOX-Type デバイスに接続された iOS デバイスの充電有無設定の実行結果を受け取ります。

setRecharging ([2.2.13](#)を参照)メソッドを呼び出した後、コールバックされます。

サンプルコード:

```
- (void)AsReaderWiredBOXType:(AsReaderWiredBOXType *)AsReaderWiredBOXType
setRecharging:(AsReaderWiredBOXTypeSDKBOXStatusCode)statusCode {
// statusCode AsReaderWiredBOXTypeSDKBOXStatusCode 列挙(4.1.9を参照)
}
```

2.3.20. rfidModuleVersionReceived

- (void)AsReaderWiredBOXType:(AsReaderWiredBOXType *)AsReaderWiredBOXType
rfidModuleVersionReceived:(NSString *)rfidModuleVersion;

	引数名	In/Out	型	説明
引数	AsReaderWiredBOXType	Out	AsReaderWiredBOXType	AsReaderWiredBOXType オブジェクト

引数	rfidModuleVersion	Out	NSString	Wired BOX RFID モジュールのファームウェアバージョン
<p>メソッド説明: Wired BOX-Type デバイスの RFID モジュールのファームウェアバージョンを受け取ります。 getFirmwareVersion (2.2.28 を参照) メソッドを呼び出した後、コールバックされます。</p> <p>サンプルコード:</p> <pre>- (void)AsReaderWiredBOXType:(AsReaderWiredBOXType *)AsReaderWiredBOXType rfidModuleVersionReceived:(NSString *)rfidModuleVersion { // rfidModuleVersion Wired BOX RFID モジュールファームウェアバージョン }</pre>				

2.3.21. receivedRechargedStatus

<p>- (void)AsReaderWiredBOXType:(AsReaderWiredBOXType *)AsReaderWiredBOXType receivedRechargedStatus:(AsReaderWiredBOXTypeSDKBOXStatusCode)statusCode;</p>				
	引数名	In/Out	型	説明
引数	AsReaderWiredBOXType	Out	AsReaderWiredBOXType	AsReaderWiredBOXType オブジェクト
引数	statusCode	Out	AsReaderWiredBOXTypeSDKBOXStatusCode	AsReaderWiredBOXType SDKBOXStatusCode 列挙 (4.1.9 を参照)
<p>メソッド説明: WiredBOX-Type デバイスに接続された iOS デバイスの充電有無ステータスを受け取ります。 getRechargedStatus (2.2.14 を参照) メソッドを呼び出した後、コールバックされます。</p> <p>サンプルコード:</p> <pre>- (void)AsReaderWiredBOXType:(AsReaderWiredBOXType *)AsReaderWiredBOXType receivedRechargedStatus:(AsReaderWiredBOXTypeSDKBOXStatusCode)statusCode { // statusCode AsReaderWiredBOXTypeSDKBOXStatusCode 列挙(4.1.9 を参照) }</pre>				

3 AsWiredBOXPrinter クラス

AsWiredBOXPrinter オブジェクトはプリンタステータスの取得、テキスト・画像・QR Code の印刷、紙の切り取り、キャッシュドローワーの開き、ユーザーカスタマイズなどの機能があります。

3.1. 属性

属性名	属性	型	説明
printDelegate	nonatomic, weak	AsReaderWiredBOXTypePrinterDelegate	AsReaderWiredBOXTypePrinterDelegate (3.3 を参照)
isConnected	nonatomic, assign, readonly	BOOL	接続状態

3.2. メソッド

3.2.1. initWithAsReaderWiredBOXType

-(id)initWithAsReaderWiredBOXType:(AsReaderWiredBOXType *)asReaderWiredBOXType;				
	引数名	In/Out	型	説明
引数	asReaderWiredBOXType	In	AsReaderWiredBOXType	AsReaderWiredBOXType オブジェクト
返回值	-	Out	AsWiredBOXPrinter	AsWiredBOXPrinter オブジェクト
メソッド説明: AsWiredBOXPrinter オブジェクトを生成します。 サンプルコード: (注: asReaderWiredBOXType は AsReaderWiredBOXType クラスのインスタンスオブジェクトです。) AsWiredBOXPrinter *asWiredBOXPrinter = [[AsWiredBOXPrinter alloc] initWithAsReaderWiredBOXType:asReaderWiredBOXType];				

3.2.2. getPrinterStatus

-(void)getPrinterStatus;				
メソッド説明:				

プリンターのステータスを取得します。

このメソッドを実行した後、「AsReaderWiredBOXTypePrinterStatus」([3.3.1](#)を参照)デリゲートがコールバックされます。

サンプルコード:(注: asPrinter は AsWiredBOXPrinter クラスのインスタンスオブジェクトです。)

```
[asPrinter getPrinterStatus];
```

3.2.3. addPrintText

```
-(void)addPrintText:(NSString*)text
font:(AsReaderWiredBOXTypeSDKMNPrintFont)font
printerAlignment:(AsReaderWiredBOXTypeSDKPrinterAlignment)printerAlignment;
```

	引数名	In/Out	型	説明
引数	text	In	NSString	データ内容を印刷する
引数	font	In	AsReaderWiredBOXTypeSDKMNPrintFont	AsReaderWiredBOXTypeSDKMNPrintFont 列挙型 (4.1.6 を参照)
引数	printerAlignment	In	AsReaderWiredBOXTypeSDKPrinterAlignment	AsReaderWiredBOXTypeSDKPrinterAlignment 列挙型 (4.1.5 を参照)

メソッド説明:

各行の印刷データを追加します。

サンプルコード:(注: asPrinter は AsWiredBOXPrinter クラスのインスタンスオブジェクトです。)

```
[asPrinter addPrintText:@"123456" font:AsReaderWiredBOXTypeSDKMNPrintFont_1
printerAlignment:AsReaderWiredBOXTypeSDKPrinterAlignmentRight];
```

3.2.4. addPrintQRCode

```
-(void)addPrintQRCode:(NSString*)qrCode
size:(AsReaderWiredBOXTypeSDKQrDotSize)size
printerAlignment:(AsReaderWiredBOXTypeSDKPrinterAlignment)printerAlignment;
```

	引数名	In/Out	型	説明
引数	qrCode	In	NSString	印刷する QRCode コンテンツ
引数	size	In	AsReaderWiredBOXTypeSDKQrDotSize	AsReaderWiredBOXTypeSDKQrDotSize 列挙型 (4.1.8 を参照)

引数	printerAlignment	In	AsReaderWiredBOXTypeSDKPrinterAlignment	AsReaderWiredBOXTypeSDKPrinterAlignment 列挙型 (4.1.5 を参照)
メソッド説明: 印刷する QRCode コンテンツを追加します。 サンプルコード: (注: asPrinter は AsWiredBOXPrinter クラスのインスタンスオブジェクトです。) [asPrinter addPrintQRCode:@"123456" size: AsReaderWiredBOXTypeSDKQrDotSize_1 printerAlignment: AsReaderWiredBOXTypeSDKPrinterAlignmentLeft];				

3.2.5. addPrintImage

-(void)addPrintImage:(UIImage *)image width:(float)width height:(float)height printerAlignment:(AsReaderWiredBOXTypeSDKPrinterAlignment)printerAlignment;				
	引数名	In/Out	型	説明
引数	image	In	UIImage	印刷する画像
引数	width	In	float	印刷する画像の幅
引数	height	In	float	印刷する画像の高さ
引数	printerAlignment	In	AsReaderWiredBOXTypeSDKPrinterAlignment	AsReaderWiredBOXTypeSDKPrinterAlignment 列挙型 (4.1.5 を参照)
メソッド説明: 印刷する画像を追加します。 サンプルコード: (注: asPrinter は AsWiredBOXPrinter クラスのインスタンスオブジェクトで、pic はローカル画像の名前です。) UIImage *image = [UIImage imageNamed:@"pic"]; [asPrinter addPrintImage:image width:200 height:200 printerAlignment: AsReaderWiredBOXTypeSDKPrinterAlignmentCenter];				

3.2.6. addCutCommand

-(void)addCutCommand:(AsReaderWiredBOXTypeSDKCutType)cutType;				
	引数名	In/Out	型	説明
引数	cutType	In	AsReaderWiredBOXTypeSDKCutType	AsReaderWiredBOXTypeSDKCutType 列挙型 (4.1.4 を参照)

メソッド説明:

紙の切り取りコマンドを追加します。

サンプルコード:(注: asPrinter は AsWiredBOXPrinter クラスのインスタンスオブジェクトです。)

```
[asPrinter addCutCommand:AsReaderWiredBOXTypeSDKCutTypeHalf];
```

3.2.7. openDrawer

-(void)openDrawer;

メソッド説明:

キャッシュドローワーを開きます。

サンプルコード:(注: asPrinter は AsWiredBOXPrinter クラスのインスタンスオブジェクトです。)

```
[asPrinter openDrawer];
```

3.2.8. cleanPrintData

-(void)cleanPrintData;

メソッド説明:

追加した印刷データをクリアします。

サンプルコード:(注: asPrinter は AsWiredBOXPrinter クラスのインスタンスオブジェクトです。)

```
[asPrinter cleanPrintData];
```

3.2.9. doPrint

-(void)doPrint;

メソッド説明:

以前に追加した印刷データを印刷します。印刷が成功すると、追加した印刷データはクリアされます。

このメソッドを実行した後、「AsReaderWiredBOXTypePrinterConnected」([3.3.2](#)を参照)、「retryPrintOnError」([3.3.3](#)を参照)、「whenPrintSucceeded」([3.3.4](#)を参照)デリゲートがコールバックされます。

サンプルコード:(注: asPrinter は AsWiredBOXPrinter クラスのインスタンスオブジェクトです。)

```
[asPrinter doPrint];
```

3.2.10. txRawData

-(void)txRawData:(NSData *)txRawData;				
	引数名	In/Out	型	説明
引数	txRawData	In	NSData	プリンタにあり、SDK ではサポートされていない機能
<p>メソッド説明: ユーザーはこのメソッドでプリンタと直接通信できます。SDK でサポートされていない機能を実装できます。 このメソッドを実行した後、プリンターから応答がある場合、「rxRawData」(3.3.5を参照)デリゲートがコールバックされます。 サンプルコード:(注: asPrinter は AsWiredBOXPrinter クラスのインスタンスオブジェクトで、data は NSData 型のデータです。) [asPrinter txRawData:data];</p>				

3.3. デリゲート

3.3.1. AsReaderWiredBOXTypePrinterStatus

-(void)AsReaderWiredBOXTypePrinterStatus:(AsReaderWiredBOXTypeSDKPrinterStatus)printerStatus;				
	引数名	In/Out	型	説明
引数	printerStatus	Out	AsReaderWiredBOXTypeSDKPrinterStatus	AsReaderWiredBOXTypeSDKPrinterStatus 列挙型(4.1.1 を参照)
<p>メソッド説明: プリンターのステータスを返します。 getPrinterStatus(3.2.2を参照)メソッドを呼び出した後、コールバックされます。 サンプルコード: - (void)AsReaderWiredBOXTypePrinterStatus:(AsReaderWiredBOXTypeSDKPrinterStatus)printerStatus { // printerStatus AsReaderWiredBOXTypeSDKPrinterStatus 列挙型(4.1.1を参照) }</p>				

3.3.2. AsReaderWiredBOXTypePrinterConnected

- (void)AsReaderWiredBOXTypePrinterConnected:(BOOL)isConnected;				
	引数名	In/Out	型	説明
引数	isConnected	Out	BOOL	YES: 接続 NO: 切断
メソッド説明: プリンターの接続状態を返します。 doPrint(3.2.9 を参照)メソッドを呼び出した後、コールバックされます。 サンプルコード: - (void)AsReaderWiredBOXTypePrinterConnected:(BOOL)isConnected { // isConnected YES: 接続 NO: 切断 }				

3.3.3. retryPrintOnError

- (BOOL)retryPrintOnError;				
	引 数 名	In/Out	型	説明
返回值	-	In	BOOL	YES: 再印刷する NO: 再印刷しない
メソッド説明: データをクリアせずにもう一度印刷します。 doPrint(3.2.9 を参照)メソッドを呼び出した後、コールバックされます。 サンプルコード: - (BOOL)retryPrintOnError{ return YES; }				

3.3.4. whenPrintSucceeded

- (void)whenPrintSucceeded;				
メソッド説明: データの印刷が完了した際に呼び出されます。 doPrint(3.2.9 を参照)メソッドを呼び出した後、コールバックされます。 サンプルコード: - (void)whenPrintSucceeded {				

```
//データの印刷に成功。
}
```

3.3.5. rxRawData

- (void)rxRawData:(NSData *)rxRawData;				
	引数名	In/Out	型	説明
引数	rxRawData	Out	NSData	プリンターから返された実行結果を受け取る
<p>メソッド説明: プリンターにあり、SDK ではサポートされていない機能に関するメソッドの実行結果を受け取ります。</p> <p>txRawData (3.2.10 を参照) メソッドを呼び出した後、コールバックされます。</p> <p>サンプルコード:</p> <pre>- (void)rxRawData:(NSData *)rxRawData { // rxRawData プリンターから返された実行結果を受け取る }</pre>				

4 AsReaderWiredBOXTypeSDKDefine クラス

AsReaderWiredBOXTypeSDKDefine オブジェクトは各パラメータの列挙クラスです。

4.1. 列挙

4.1.1. AsReaderWiredBOXTypeSDKPrinterStatus

プリンターのステータス

定義	値	説明
AsReaderWiredBOXTypeSDKPrinterStatus_NOERROR	0	エラーなし
AsReaderWiredBOXTypeSDKPrinterStatus_NOPAPER	1	紙切れ
AsReaderWiredBOXTypeSDKPrinterStatus_InstitutionalFailure	2	機械の故障
AsReaderWiredBOXTypeSDKPrinterStatus_CoverOpen	3	機械の蓋オープン

4.1.2. AsReaderWiredBOXTypeSDKBOXNetworkState

ネットワークのステータス

定義	値	説明
AsReaderWiredBOXTypeSDKBOXNetworkStateConnected_	0	接続
AsReaderWiredBOXTypeSDKBOXNetworkStateDisconnected	1	切断
AsReaderWiredBOXTypeSDKBOXNetworkStateConnecting	2	接続処理中
AsReaderWiredBOXTypeSDKBOXNetworkStateRecievedData	3	データを受信
AsReaderWiredBOXTypeSDKBOXNetworkStateError	4	エラー発生

4.1.3. AsReaderWiredBOXTypeSDKBOXSelectAntenna

選択したアンテナ

定義	値	説明
AsReaderWiredBOXTypeSDKBOXSelectAntennaNone_	0	アンテナなし
AsReaderWiredBOXTypeSDKBOXSelectAntenna1	1	アンテナ 1
AsReaderWiredBOXTypeSDKBOXSelectAntenna2	2	アンテナ 2
AsReaderWiredBOXTypeSDKBOXSelectAntenna3	3	アンテナ 3
AsReaderWiredBOXTypeSDKBOXSelectAntenna4	4	アンテナ 4
AsReaderWiredBOXTypeSDKBOXSelectAntenna5	5	アンテナ 5
AsReaderWiredBOXTypeSDKBOXSelectAntenna6	6	アンテナ 6
AsReaderWiredBOXTypeSDKBOXSelectAntenna7	7	アンテナ 7
AsReaderWiredBOXTypeSDKBOXSelectAntenna8	8	アンテナ 8

4.1.4. AsReaderWiredBOXTypeSDKCutType

紙のカットステータス

定義	値	説明
AsReaderWiredBOXTypeSDKCutTypeFull_	0x69	全切り
AsReaderWiredBOXTypeSDKCutTypeHalf	0x6D	半切り

4.1.5. AsReaderWiredBOXTypeSDKPrinterAlignment

印刷コンテンツの位置

定義	値	説明
AsReaderWiredBOXTypeSDKPrinterAlignmentLeft	0	左寄せ
AsReaderWiredBOXTypeSDKPrinterAlignmentCenter	1	中央寄せ
AsReaderWiredBOXTypeSDKPrinterAlignmentRight	2	右寄せ

4.1.6. AsReaderWiredBOXTypeSDKMNPrintFont

印刷コンテンツのフォントサイズ

定義	値	説明
AsReaderWiredBOXTypeSDKMNPrintFont_1	0x00	No.1 フォント
AsReaderWiredBOXTypeSDKMNPrintFont_2	0x11	No.2フォント
AsReaderWiredBOXTypeSDKMNPrintFont_3	0x22	No.3フォント
AsReaderWiredBOXTypeSDKMNPrintFont_4	0x33	No.4フォント
AsReaderWiredBOXTypeSDKMNPrintFont_5	0x44	No.5フォント
AsReaderWiredBOXTypeSDKMNPrintFont_6	0x55	No.6フォント
AsReaderWiredBOXTypeSDKMNPrintFont_7	0x66	No.7フォント
AsReaderWiredBOXTypeSDKMNPrintFont_8	0x77	No.8フォント

4.1.7. AsReaderWiredBOXTypeSDKQrModel

QR モデル

定義	値	説明
AsReaderWiredBOXTypeSDKQrModel_1	49	QR モデル 1
AsReaderWiredBOXTypeSDKQrModel_2	50	QR モデル 2

4.1.8. AsReaderWiredBOXTypeSDKQrDotSize

QR ドットサイズ

定義	値	説明
AsReaderWiredBOXTypeSDKQrDotSize_1	1	QR ドットサイズ 1x1
AsReaderWiredBOXTypeSDKQrDotSize_2	2	QR ドットサイズ 2x2
AsReaderWiredBOXTypeSDKQrDotSize_3	3	QR ドットサイズ 3x3
AsReaderWiredBOXTypeSDKQrDotSize_4	4	QR ドットサイズ 4x4

AsReaderWiredBOXTypeSDKQrDotSize_5	5	QR ドットサイズ 5x5
AsReaderWiredBOXTypeSDKQrDotSize_6	6	QR ドットサイズ 6x6
AsReaderWiredBOXTypeSDKQrDotSize_7	7	QR ドットサイズ 7x7

4.1.9. AsReaderWiredBOXTypeSDKBOXStatusCode

設定ステータス。

定義	値	説明
AsReaderWiredBOXTypeSDKBOXStatusCodeUnknown	-1	不明
AsReaderWiredBOXTypeSDKBOXStatusCodeSuccess	0	成功
AsReaderWiredBOXTypeSDKBOXStatusCodeFailed	1	失敗

4.1.10. AsReaderWiredBOXTypeSDKCommonStatus

設定ステータス

定義	値	説明
AsReaderWiredBOXTypeSDKCommonStatusNone	0	不明
AsReaderWiredBOXTypeSDKCommonStatusSuccess	1	成功
AsReaderWiredBOXTypeSDKCommonStatusFailure	2	失敗

4.1.11. AsReaderWiredBOXTypeSDKResultSetRecharging

充電ステータス

定義	値	説明
AsReaderWiredBOXTypeSDKResultSetRechargingSucceed	0	充電成功
AsReaderWiredBOXTypeSDKResultSetRechargingFailed _Type_C_Busy	1	インベントリ中、充電 できない
AsReaderWiredBOXTypeSDKResultSetRechargingFailed	2	充電失敗

4.1.12. AsReaderWiredBOXTypeConnectMethod

接続方法

定義	値	説明
AsReaderWiredBOXTypeConnectMethod_TCP	0	TCP 接続
AsReaderWiredBOXTypeConnectMethod_MQTT	1	MQTT 接続