

StarPRNT Android SDK ユーザーズマニュアル

2018 年 06 月 29 日

要約

このマニュアルは StarPRNT Android SDK に関する情報を提供するものです。

ユーザー向けに StarPRNT アプリケーションの構築に関するガイドラインを示します。

ドキュメントの履歴

日付	変更内容
2016 年 08 月 11 日	発行
2016 年 11 月 18 日	AllReceipts クラスに uploadData メソッドを追加 onAccessoryConnectSuccess と onAccessoryConnectFailure メソッドの説明文を改訂 BSC10 の USB 接続時のプリンタビジー条件の設定方法を追加
2017 年 04 月 17 日	ブラックマーク、ページモードに対応
2017 年 09 月 06 日	StarIO ライブラリに含まれるクラスの API リファレンスを追加 StarBluetoothManagerFactory クラスを追加 外部機器（バーコードリーダー・スケール・カスタマーディスプレイ）の API リファレンスを追加
2018 年 01 月 26 日	ICommandBuilder インターフェイスに appendHorizontalTabPosition メソッドを追加
2018 年 05 月 21 日	mC-Print2、mC-Print3 対応追加 対応外部機器一覧表を追加 プリンタ設定用ソフトウェア一覧表を追加 portSettings の I オプション対応インターフェイスに Ethernet を追加 StarPrinterStatus クラスに connectedInterface フィールドを追加 ICommandBuilder インターフェイスの appendSound メソッドに driveTime パラメータと delayTime パラメータを追加 ICommandBuilder インターフェイスに appendTopMargin メソッドを追加 ICommandBuilder インターフェイスに appendPrintableArea メソッドを追加
2018 年 06 月 29 日	APS シリーズのサポート終了

本書に関して

本マニュアルはStarPRNT SDK for Androidアプリに関する情報を解説しています。

またこのマニュアルはアプリケーション開発者を対象に作成しており、利用者はJava言語の基礎を理解していることを前提としています。

スター精密グローバルサポートサイトのDevelopersセクションには、その他のオペレーティングシステムとプログラミング言語に利用可能なSDKが用意されています。最新のSDK、テクニカルドキュメント、FAQ及びその他の追加情報についてはDevelopersセクションをご確認ください。

注意事項:

- Android は、Google Inc. の商標です。
- Windows は、米国 Microsoft Corporation の、米国、日本およびその他の国における登録商標です。
- Bluetooth® のワードマークおよびロゴは、Bluetooth SIG, Inc. が所有する登録商標です。
- 本マニュアルの内容は、予告無く変更する場合があります。
- スター精密株式会社は、正確な情報を提供するためにあらゆる措置を取っていますが、誤りや不作為について責任を負うものではありません。
- スター精密株式会社は、このマニュアルに記載されている情報の使用に起因するいかなる損害に対しても責任を負うものではありません。
- 本マニュアルの一部、あるいは全部を無断で複写・複製・転載することは、固くお断りします。

目次

1	はじめに	9
1.1	Android Studio で StarPRNT SDK プロジェクトをビルドする	9
1.2	対応プリンタ	12
1.3	対応外部機器	14
1.4	プリンタを Android デバイスに接続する	16
1.5	プリンタ設定用ソフトウェア	20
2	StarPRNT アプリケーション構成方法	23
2.1	ライブラリを追加する	23
2.2	AndroidManifest.xml へ項目を設定する	24
2.3	device_filter.xml へ項目を設定する	26
2.4	build.gradle (Module: app)へ項目を設定する	27
2.5	build.gradle (Project: SDK)へ項目を設定する	29
3	StarIOPort クラス (StarIOPort3.1.jar).....	30
3.1	StarIOPort クラスを使った印刷処理フロー	31
3.2	モデル: StarIOPort クラス.....	32
3.3	getPort メソッド	33
3.4	releasePort メソッド	38
3.5	searchPrinter メソッド	39
3.6	writePort メソッド	40
3.7	readPort メソッド	41
3.8	retrieveStatus メソッド	42
3.9	beginCheckedBlock メソッド	43
3.10	endCheckedBlock メソッド	44
3.11	setEndCheckedBlockTimeoutMillis メソッド	45
3.12	getFirmwareInformation メソッド	46
3.13	getPortName メソッド	47
3.14	getPortSettings メソッド	47
3.15	getStarIOVersion メソッド	47
4	StarPrinterStatus クラス (StarIOPort3.1.jar).....	48
4.1	モデル: StarPrinterStatus クラス	49
5	PortInfo クラス (StarIOPort3.1.jar)	50
5.1	getPortName メソッド	50
5.2	getMacAddress メソッド	50
5.3	getModelName メソッド	51
5.4	getUSBSerialNumber メソッド	51
6	StarBluetoothManager クラス (StarIOPort3.1.jar)	52
6.1	StarBluetoothManager を使った Bluetooth 設定変更フロー	53
6.2	モデル: StarBluetoothManager クラス	54
6.3	StarBluetoothManager コンストラクタ	56
6.4	open メソッド	57
6.5	loadSetting メソッド	57
6.6	close メソッド	58
6.7	apply メソッド	58
6.8	getBluetoothDeviceNameCapability メソッド	59
6.9	getBluetoothDeviceName メソッド	59
6.10	setBluetoothDeviceName メソッド	60

6.11	getiOSPortNameCapability メソッド	60
6.12	getiOSPortName メソッド	61
6.13	setiOSPortName メソッド	61
6.14	getPinCodeCapability メソッド	62
6.15	getPinCode メソッド	62
6.16	setPinCode メソッド	63
6.17	getAutoConnectCapability メソッド	63
6.18	getAutoConnect メソッド	64
6.19	setAutoConnect メソッド	64
6.20	getSecurityTypeCapability メソッド	65
6.21	getSecurityType メソッド	65
6.22	setSecurityType メソッド	65
6.23	getPortName メソッド	66
6.24	getPortSettings メソッド	66
6.25	getTimeoutMillis メソッド	66
6.26	getDeviceType メソッド	67
6.27	isOpened メソッド	67
6.28	StarDeviceType コンスタント	68
6.29	StarBluetoothSecurity コンスタント	69
6.30	StarBluetoothSettingCapability コンスタント	69
7	StarIoExt クラス (starioextension.jar)	70
7.1	createCommandBuilder メソッド	71
7.2	createDisplayCommandBuilder メソッド	72
7.3	createBcrConnectParser メソッド	73
7.4	createDisplayConnectParser メソッド	74
7.5	Emulation コンスタント	75
7.6	CharacterCode コンスタント	76
7.7	BcrModel コンスタント	77
7.8	DisplayModel コンスタント	77
8	ICommandBuilder インターフェイス (starioextension.jar)	78
8.1	モデル : ICommandBuilder interface メソッド	80
8.2	beginDocument メソッド	83
8.3	endDocument メソッド	84
8.4	appendInitialization メソッド	85
8.5	append メソッド	86
8.6	appendRaw メソッド	87
8.7	appendFontStyle メソッド	88
8.8	appendCodePage メソッド	89
8.9	appendInternational メソッド	91
8.10	appendLineFeed メソッド	92
8.11	appendUnitFeed メソッド	93
8.12	appendCharacterSpace メソッド	94
8.13	appendLineSpace メソッド	95
8.14	appendTopMargin メソッド	96
8.15	appendEmphasis メソッド	97
8.16	appendInvert メソッド	98
8.17	appendMultiple メソッド	99
8.18	appendUnderLine メソッド	101

8.19	appendLogo メソッド.....	102
8.20	appendAbsolutePosition メソッド	103
8.21	appendAlignment メソッド.....	104
8.22	appendHorizontalTabPosition メソッド.....	105
8.23	appendCutPaper メソッド.....	106
8.24	appendPeripheral メソッド	107
8.25	appendSound メソッド	108
8.26	appendBarcode メソッド	109
8.27	appendBarcodeWithAbsolutePosition メソッド	111
8.28	appendBarcodeWithAlignment メソッド.....	113
8.29	appendPdf417 メソッド	115
8.30	appendPdf417WithAbsolutePosition メソッド.....	116
8.31	appendPdf417WithAlignment メソッド.....	117
8.32	appendQrCode メソッド	119
8.33	appendQrCodeWithAbsolutePosition メソッド.....	120
8.34	appendQrCodeWithAlignment メソッド	121
8.35	appendBitmap メソッド	122
8.36	appendBitmapWithAbsolutePosition メソッド	124
8.37	appendBitmapWithAlignment メソッド.....	126
8.38	appendBlackMark メソッド	128
8.39	beginPageMode メソッド.....	129
8.40	endPageMode メソッド	130
8.41	appendPageModeVerticalAbsolutePosition メソッド	131
8.42	appendPageModeRotation メソッド	132
8.43	appendPrintableArea メソッド	133
8.44	getCommands メソッド	136
8.45	InitializationType コンスタント	137
8.46	FontStyleType コンスタント	137
8.47	CodePageType コンスタント.....	138
8.48	InternationalType コンスタント	140
8.49	LogoSize コンスタント	141
8.50	AlignmentPosition コンスタント	141
8.51	CutPaperAction コンスタント	142
8.52	PeripheralChannel コンスタント	142
8.53	SoundChannel コンスタント.....	143
8.54	BarcodeSymbology コンスタント.....	143
8.55	BarcodeWidth コンスタント	144
8.56	Pdf417Level コンスタント.....	145
8.57	QrCodeModel コンスタント	145
8.58	QrCodeLevel コンスタント	146
8.59	BitmapConverterRotation コンスタント.....	146
8.60	BlackMarkType コンスタント	147
8.61	PrintableAreaType コンスタント	147
9	IDisplayCommandBuilder インターフェイス (starioextension.jar)	148
9.1	モデル : IDisplayCommandBuilder インターフェイス.....	149
9.2	append メソッド	149
9.3	appendBackSpace メソッド.....	150
9.4	appendHorizontalTab メソッド.....	150

9.5	appendLineFeed メソッド	150
9.6	appendCarriageReturn メソッド	151
9.7	appendBitmap メソッド	152
9.8	appendInternational メソッド	153
9.9	appendCodePage メソッド	154
9.10	appendDeleteToEndOfLine メソッド	155
9.11	appendClearScreen メソッド	155
9.12	appendHomePosition メソッド	156
9.13	appendTurnOn メソッド	157
9.14	appendSpecifiedPosition メソッド	158
9.15	appendCursorMode メソッド	159
9.16	appendContrastMode メソッド	160
9.17	appendUserDefinedCharacter メソッド	161
9.18	appendUserDefinedDbcsCharacter メソッド	163
9.19	getCommands メソッド	165
9.20	getPassThroughCommands メソッド	165
9.21	InternationalType コンスタント	166
9.22	CodePageType コンスタント	167
9.23	CursorMode コンスタント	168
9.24	ContrastMode コンスタント	168
10	IPeripheralCommandParser インターフェイス (starioextension.jar)	169
10.1	モデル : IPeripheralCommandParser インターフェイス	169
10.2	createSendCommands メソッド	169
10.3	parse メソッド	171
10.4	ParseResult コンスタント	172
11	IPeripheralConnectParser インターフェイス (starioextension.jar)	173
11.1	モデル : IPeripheralConnectParser インターフェイス	173
11.2	isConnected メソッド	174
12	StarPRNT Android SDK Sample	175
12.1	Communication	175
13	StarIoExtManager クラス (starioextension.jar)	177
13.1	StarIoExtManager コンストラクタ	178
13.2	connect メソッド	179
13.3	disconnect メソッド	180
13.4	getPort メソッド	181
13.5	getPrinterOnlineStatus メソッド	181
13.6	getPrinterPaperStatus メソッド	182
13.7	getPrinterCoverStatus メソッド	182
13.8	getCashDrawerStatus メソッド	183
13.9	getBarcodeReaderStatus メソッド	183
13.10	getCashDrawerOpenActiveHigh メソッド	184
13.11	setCashDrawerOpenActiveHigh メソッド	184
13.12	setListener メソッド	185
13.13	Type コンスタント	186
13.14	PrinterStatus コンスタント	186
13.15	PrinterPaperStatus コンスタント	187
13.16	PrinterCoverStatus コンスタント	187
13.17	CashDrawerStatus コンスタント	188

13.18	BarcodeReaderStatus コンスタント	188
14	IConnectionCallback インターフェイス (starioextension.jar)	189
14.1	onConnected メソッド	189
14.2	onDisconnected メソッド	190
14.3	ConnectResult コンスタント	190
15	StarIoExtManagerListener (starioextension.jar)	191
15.1	onPrinterImpossible メソッド	191
15.2	onPrinterOnline メソッド	192
15.3	onPrinterOffline メソッド	192
15.4	onPrinterPaperReady メソッド	193
15.5	onPrinterPaperNearEmpty メソッド	193
15.6	onPrinterPaperEmpty メソッド	194
15.7	onPrinterCoverOpen メソッド	194
15.8	onPrinterCoverClose メソッド	195
15.9	onCashDrawerOpen メソッド	195
15.10	onCashDrawerClose メソッド	196
15.11	onBarcodeReaderImpossible メソッド	196
15.12	onBarcodeReaderConnect メソッド	197
15.13	onBarcodeReaderDisconnect メソッド	197
15.14	onBarcodeDataReceive メソッド	198
15.15	onAccessoryConnectSuccess メソッド	198
15.16	onAccessoryConnectFailure メソッド	199
15.17	onAccessoryDisconnect メソッド	199
15.18	onStatusUpdate メソッド	200
16	StarBluetoothManagerFactory クラス (starioextension.jar)	201
16.1	getManager メソッド	201
17	CloudServices クラス (smcloudservices.aar)	202
17.1	showRegistrationView メソッド	202
17.2	isRegistered メソッド	203
18	AllReceipts クラス (smcloudservices.aar)	204
18.1	モデル : AllReceipts class メソッド	205
18.2	uploadBitmap メソッド	206
18.3	uploadData メソッド	207
18.4	updateStatus メソッド	209
18.5	generateAllReceipts メソッド	210
19	RequestCallback クラス (smcloudservices.aar)	212
19.1	onRequestResult メソッド	212
20	RequestError クラス (smcloudservices.aar)	213
20.1	getMessage メソッド	213
20.2	getInternalException メソッド	214
	Appendix A. AllReceipt™ 利用方法案内 (配布用)	215

1 はじめに

1.1 Android Studio で StarPRNT SDK プロジェクトをビルドする

Android のプロジェクトをビルドするには Android Studio が必要です。これらのツールは [Android Studio – Official Site](#) から入手可能です。

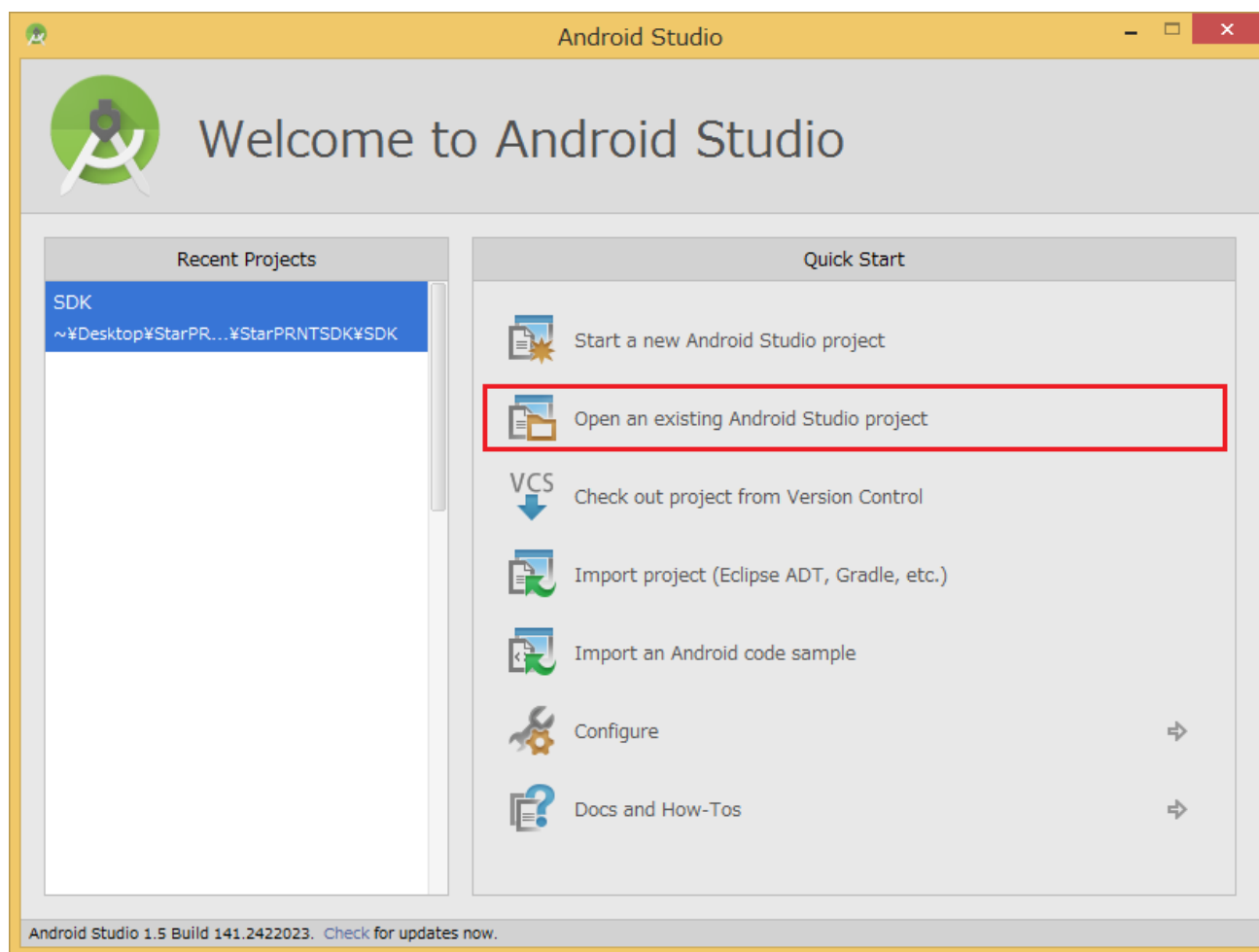
事前に、開発を行う PC に Android Studio のインストールを行ってください。サポートまたは追加情報が必要な場合は [Android Studio – Official Site](#) を参照してください。

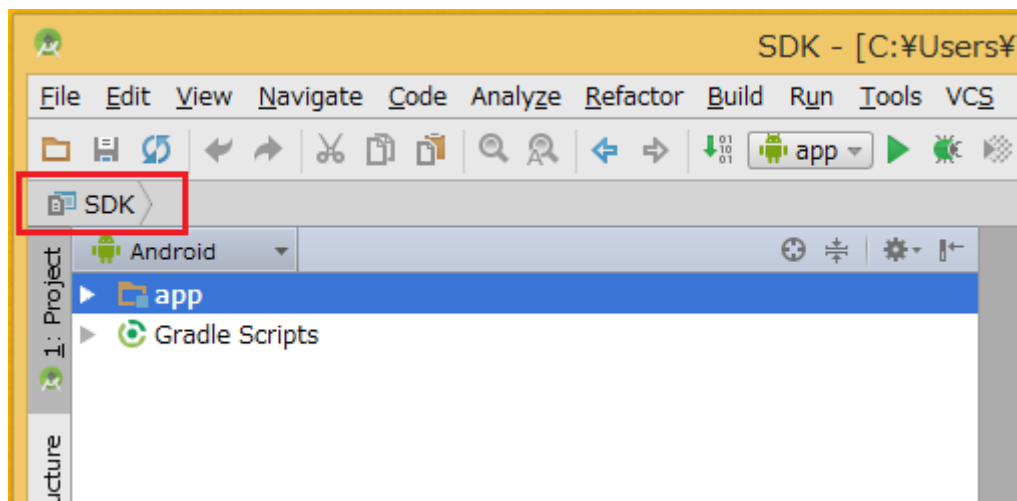
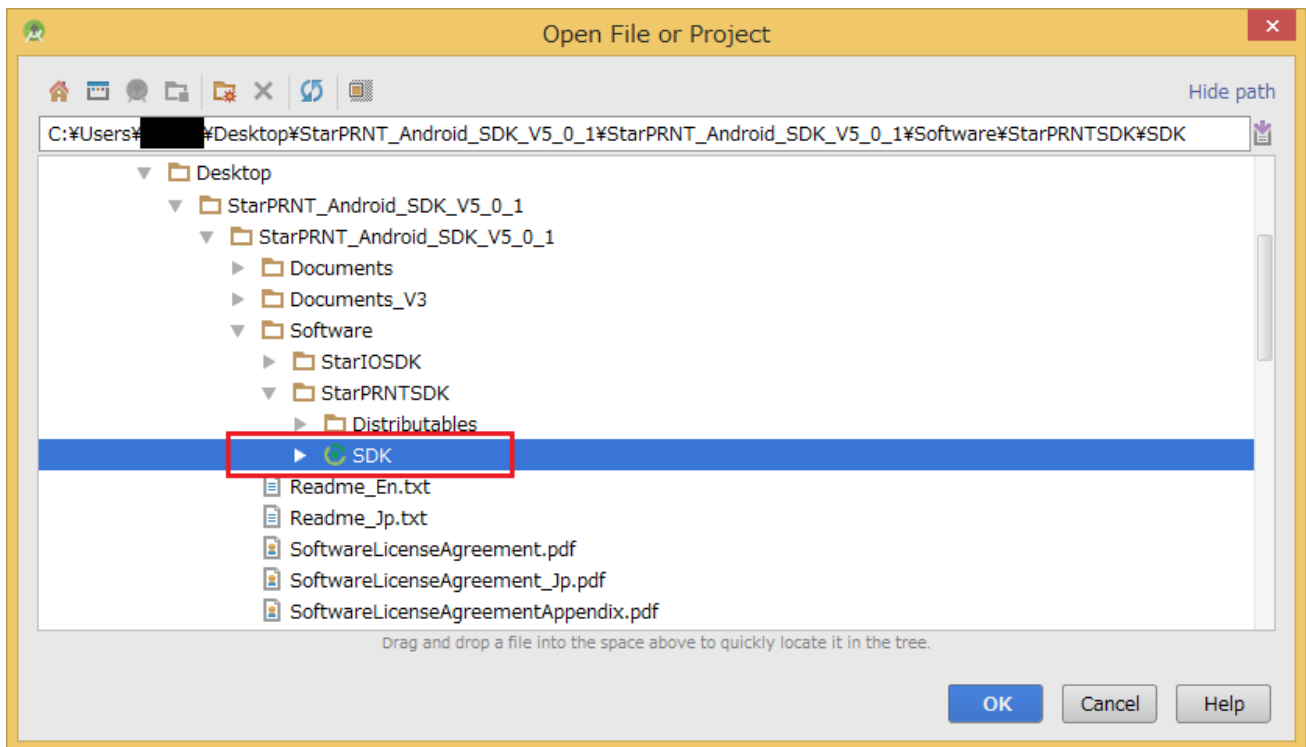
Android Studio で StarPRNT SDK プロジェクトを開く方法

1.StarPRNT SDK パッケージを解凍します。



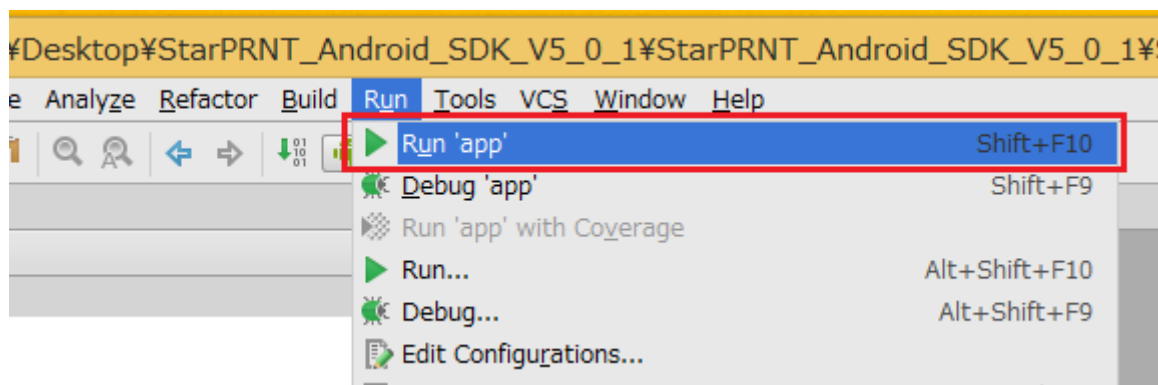
2.Android Studio を実行し、StarPRNT SDK を開きます。



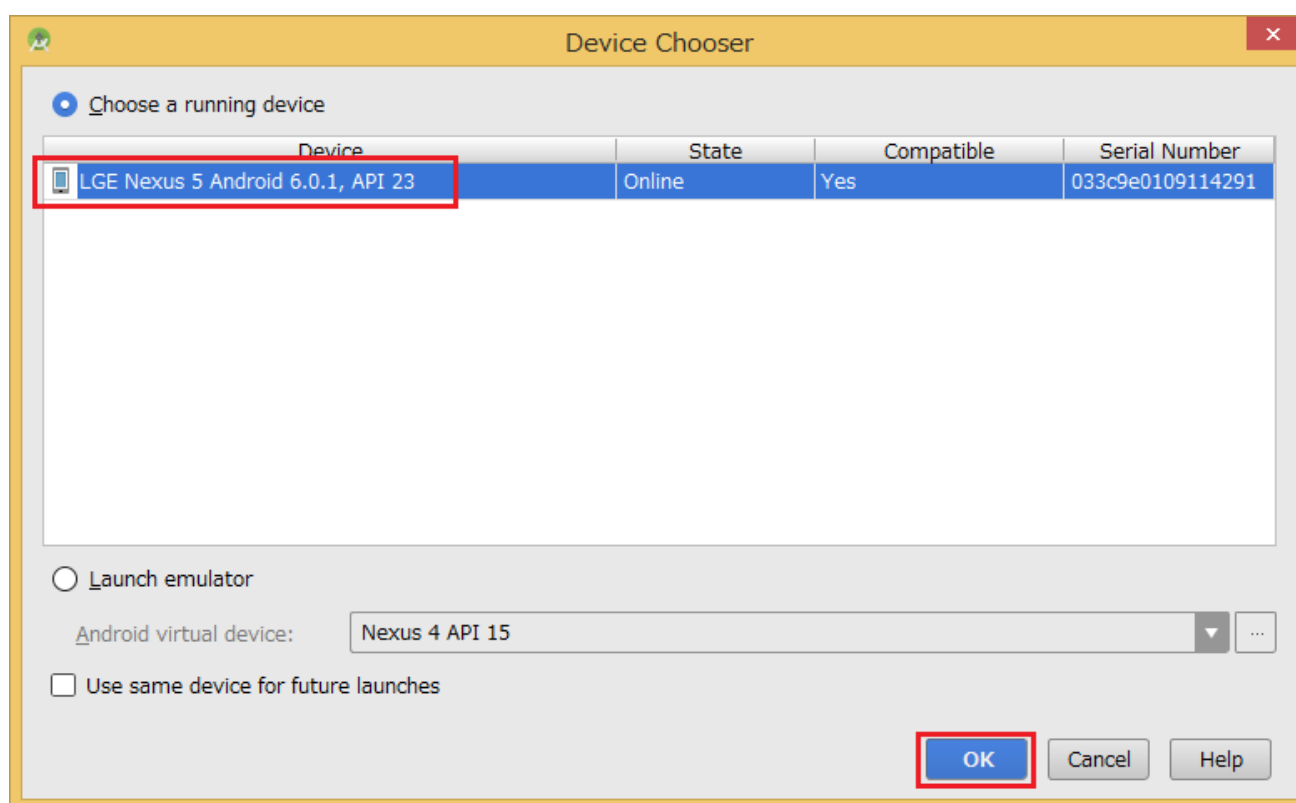


プロジェクトを実行する

1. 上部メニューバーの"Run"をクリックして"Run 'app'"を選択します。



2. デバイスを選択します。



1.2 対応プリンタ

対応プリンタ		F/W バージョン	対応インターフェイス	対応エミュレーション
POS プリンタ	mC-Print2	1.0 以降	Bluetooth Ethernet USB	StarPRNT
	mC-Print3	1.0 以降	Bluetooth Ethernet USB	StarPRNT
	mPOP	1.0 以降	Bluetooth USB	StarPRNT
	FVP10	1.2 以降	Bluetooth Ethernet USB	StarLine
	TSP100IIIW	1.0 以降	Wireless LAN	StarGraphic
	TSP100IIILAN	1.0 以降	Ethernet	StarGraphic
	TSP100IIIBI	1.0 以降	Bluetooth *2	StarGraphic
	TSP100IIIU	1.0 以降	USB	StarGraphic
	TSP100ECO	2.0 以降	USB	StarGraphic
	TSP100U	2.0 以降	USB	StarGraphic
	TSP100GT	2.0 以降	USB	StarGraphic
	TSP100LAN	2.0 以降	Ethernet	StarGraphic
	TSP650II	1.0 以降	Bluetooth *2 Ethernet USB	StarLine
	TSP700II	3.0 以降	Bluetooth *2 Ethernet USB	StarLine
	TSP800II	1.2 以降	Bluetooth *2 Ethernet USB	StarLine
	BSC10 LAN	1.0 以降	Ethernet	ESC/POS
	BSC10 *1	1.0 以降	USB	
	SP700	3.0 以降	Bluetooth(欧米のみ)	StarDotImpact

	対応プリンタ	F/W バージョン	対応インターフェイス	対応エミュレーション
			Ethernet USB	
モバイル プリンタ	SM-S210i	2.4 以降	Bluetooth	StarPRNT *3 / EscPosMobile
	SM-S220i	2.0 以降	Bluetooth	StarPRNT *3 / EscPosMobile
	SM-S230i	1.0 以降	Bluetooth USB	StarPRNT / EscPosMobile StarPRNT
	SM-T300i	2.4 以降	Bluetooth	StarPRNT *3 / EscPosMobile
	SM-T300	1.1 以降	Bluetooth	StarPRNT *3 / EscPosMobile
	SM-T400i	2.4 以降	Bluetooth	StarPRNT *3 / EscPosMobile
	SM-L200	1.0 以降	Bluetooth	StarPRNT / StarPRNTL*4
	SM-L300	1.0 以降	Bluetooth	StarPRNT / StarPRNTL

*1 : BSC10 の USB 接続においてはプリンタビジー条件を"受信バッファフル"（工場出荷時設定は"受信バッファフル、オフライン"）に設定してください。（メモリスイッチ 7 のビット 4 を 1 に設定）

*2 : データタイムアウト機能のタイムアウト値設定をサポート。TSP100III BI はファームウェアバージョン 1.0 以降、TSP650II, TSP800II はファームウェアバージョン 2.0 以降、TSP700II はファームウェアバージョン 5.0 以降が必要です。

*3 : StarPRNT エミュレーションのサポートは SM-S210i, SM-S220i, SM-T300i, SM-T400i がファームウェアバージョン 3.0 以降、SM-T300(DB)がファームウェアバージョン 4.0 以降

*4 : StarPRNTL エミュレーションのサポートはファームウェアバージョン 2.0 以降

1.3 対応外部機器

対応している外部機器および使用可能なモデルは以下のとおりです。

外部機器		説明																			
BCR-POP1		バーコードリーダー																			
SCD222U		カスタマーディスプレイ																			

外部機器	mC-Print2	mC-Print3	mPOP	FVP40	TSP100 (TSP100IIIUのみ)	TSP650H	TSP700H	TSP800H	SM-S210i	SM-S220i	SM-S230i	SM-T300i/T300	SM-T400i	BSC10	SM-S240i-StarPRNT	SM-S220i-StarPRNT	SM-S230i-StarPRNT	SM-T300i/T300 StarPRNT	SM-T400i-StarPRNT	SM-L200	SM-L300	SP700
BCR-POP1	✓	✓	✓	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
SCD222U	✓	✓	✓	-	✓	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

-: 対象外

❖ モバイルプリンタ

StarPRNT エミュレーションを使用するには

StarPRNT エミュレーションを使用するには、プリンタ本体のエミュレーションを “StarPRNT” に設定する必要があります。（SM-L200、L300 は切り替え不要です。）

エミュレーションの切り替えは以下の手順で行ってください。

◆ ESC/POS ⇄ StarPRNT エミュレーション切り替え方法

1. プリンタの電源を入れ、プリンタカバーを開きます。
2. 電源ボタンと FEED ボタンを同時に長押しした後、ERROR ランプが 5 回点滅したことを確認しすぐに電源ボタンと FEED ボタンから指を放します。エミュレーションの切り替えが自動的に行われます。
3. 用紙をセット後、プリンタカバーを閉めると設定されたエミュレーションが印字されます。
ESC/POS の場合 : EMU = ESC/POS
StarPRNT の場合 : EMU = StarPRNT

エミュレーションが正しく切り替わっていない場合、再度 1～3 の手順を行ってください。

その際、2 の手順においては、点滅中に指を放さず、点滅が 5 回完了したことを確認してから指を放すように注意してください。

4. エミュレーションの切り替え後は、プリンタの電源を一度オフにしてから再投入してください。
選択したエミュレーションは、プリンタの電源を再投入することで有効になります。

1.4 プリンタを Android デバイスに接続する

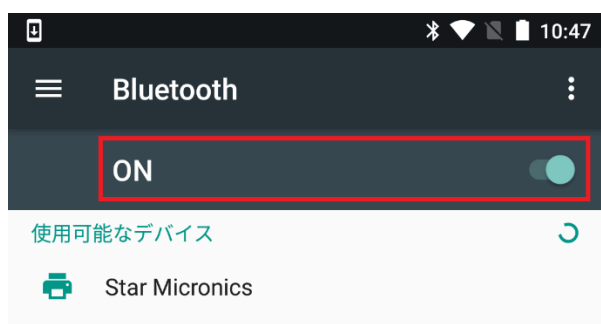
Bluetooth インターフェイス

Star POS デバイスは、工場出荷時の初期設定では機種ごとに"Star Micronics"等、共通の Bluetooth デバイス名が設定されている場合があります。同じ Bluetooth デバイス名の機種を複数台配置して運用される場合、Bluetooth デバイス名の変更を行うと Star POS デバイスの判別が付けやすく便利です。

Bluetooth デバイス名の変更等、Star POS デバイスの LAN/Bluetooth 設定値は専用のソフトウェアを使用して変更することができます。各プリンタで使用するソフトウェアは[プリンタ設定用ソフトウェア](#)を参照ください。

◆Bluetooth プリンタとのペアリング

1. 設定を行う Star POS デバイスの Bluetooth が有効になっていることを確認して、デバイスの電源を投入します。Star POS デバイスのセキュリティ方式が SSP の場合は、PAIR ボタンを 5 秒以上押してペアリング可能にします。
2. Android デバイスにて「設定」をタップします。
3. 「Bluetooth」を ON に設定します。

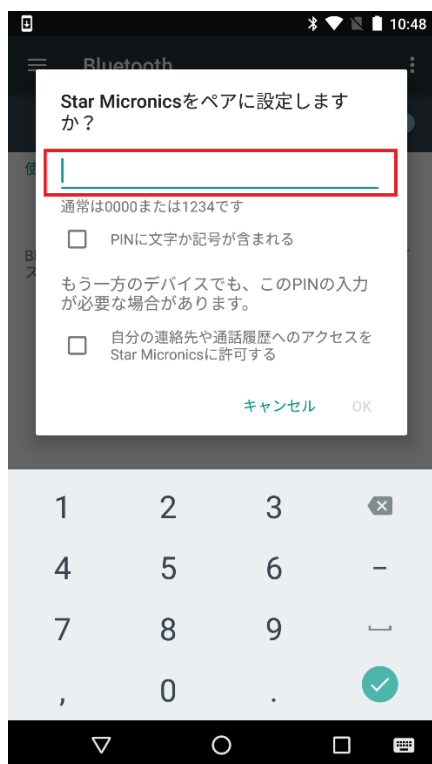


4. デバイスの検索を実行します。

Bluetooth デバイスのスキャンが行われ、見つかったデバイスが表示されます。設定を行う Star POS デバイスのデバイス名をタップします。



5. PIN Code を使用する場合、PIN 番号（パスキー）を入力します。工場出荷時の初期値は“1234”です。（SSP の場合はこの操作は必要ありません。）



Ethernet インターフェイス

Star POS プリンタは、工場出荷時の初期設定は DHCP が有効になっています。使用するネットワークが DHCP をサポートする場合、POS プリンタが自動的に IP アドレスを取得できるように、必要なネットワーク構成を構築してください。

また、プリンタ本体の#9100 Multi Session を無効に設定して使用してください。設定は、FEED ボタンを押しながら電源スイッチを ON することで実行される自己印字で確認できます。

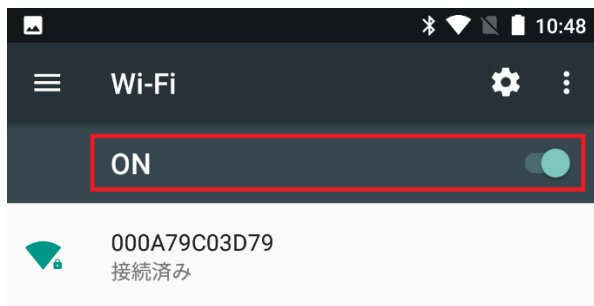
#9100 Multi Session の確認・変更、固定 IP アドレスの設定方法については、[こちらのリンク](#)の「対応 OS・環境一覧 > ユーティリティー一覧」より「イーサネットプリンタ利用手引き」をご参照ください。

POS プリンタの Ethernet 設定は専用のソフトウェアを使用して変更することができます。各プリンタで使用するソフトウェアは[プリンタ設定用ソフトウェア](#)を参照ください。

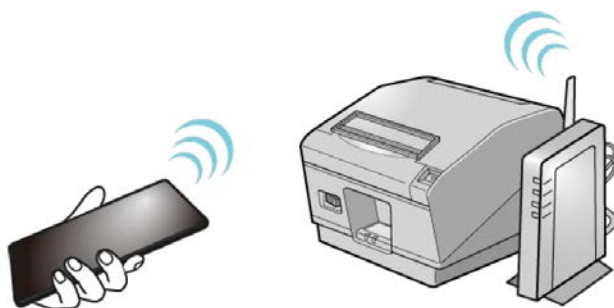
TSP100III Utility、Star Setting Utility は IP アドレスの割り当てられていない（IP アドレス：0.0.0.0）プリンタでは使用できません。

◆Ethernet プリンタと Android デバイスの接続設定

1. Star POS デバイスに IP アドレスを割り当て、ネットワークに接続します。
2. 「設定」をタップします。
3. 「Wi-Fi」を ON に設定します。



4. Star POS デバイスと同じネットワークに接続します。



Wireless LAN インターフェイス

TSP100IIIW をタブレット／PC と接続する方法については、オンラインマニュアル (設定する-タブレット/PC と接続する) を参照ください。

その他の点は Ethernet インターフェイスの項目を参照ください。

USB インターフェイス

- Android を使用しているデバイスに、プリンタドライバ等のソフトウェアがインストールされることはありません。
- 接続に使用する USB ケーブルはタブレットによって異なります。ほとんどの Android タブレットは、共通の USB ケーブルをサポートしていません。マイクロ USB ケーブルもしくはアダプタが必要になります。正しいケーブルを使用する為に、タブレットの仕様をご確認ください。
- USB ハブを用いて複数デバイスの接続を行う場合、カスケード接続は 1 段までを推奨します。また、同時に接続できるデバイス数は 4 デバイスを上限とします。



◆USB プリンタと Android デバイスの接続

AndroidManifest.xml へ項目を設定する、device_filter.xml へ項目を設定するをご覧ください。

1.5 プリンタ設定用ソフトウェア

プリンタ本体の設定を変更する場合は以下のソフトウェアをご利用ください。

Android 版

ソフトウェア	mC-Print2	mC-Print3	mPOP	FVP10	TSP100	TSP650II	TSP700II	TSP800II	SM-S210i	SM-S220i	SM-S230i	SM-T300i/T300	SM-T400i	BSC10	SM-S210i StarPRNT	SM-S220i StarPRNT	SM-S230i StarPRNT	SM-T300i/T300 StarPRNT	SM-T400i StarPRNT	SM-L200	SM-L300	SP700
Star Setting Utility * 1	-	-	-	✓	-	✓	✓	✓	✓	✓	✓	✓	✓	-	✓	✓	✓	✓	✓	✓	✓	✓
mPOP Utility	-	-	✓	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
mC-Print Utility	✓	✓	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
TSP100III Utility	-	-	-	-	✓	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

[Google play](#) よりダウンロードしてください。

- : 対象外

*1 : メモリスイッチ設定の変更はできません。

メモリスイッチ設定を変更する場合は Windows 版ソフトウェアをご利用ください。

iOS 版

ソフトウェア	mC-Print2	mC-Print3	mPOP	FVP10	TSP100	TSP650II	TSP700II	TSP800II	SM-S210i	SM-S220i	SM-S230i	SM-T300i/T300	SM-T400i	BSC10	SM-S210i StarPRNT	SM-S220i StarPRNT	SM-S230i StarPRNT	SM-T300i/T300 StarPRNT	SM-T400i StarPRNT	SM-L200	SM-L300	SP700
Star Setting Utility * 1	-	-	-	✓	-	✓	✓	✓	✓	✓	✓	✓	✓	-	✓	✓	✓	✓	✓	✓	✓	✓
mPOP Utility	-	-	✓	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
mC-Print Utility	✓	✓	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
TSP100III Utility	-	-	-	-	✓	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

[App Store](#) よりダウンロードしてください。

- : 対象外

*1 : メモリスイッチ設定の変更はできません。

メモリスイッチ設定を変更する場合は Windows 版ソフトウェアをご利用ください。

Windows 版

ソフトウェア	mC-Print2	mC-Print3	mPOP	FVP10	TSP100	TSP650II	TSP700II	TSP800II	SM-S210i	SM-S220i	SM-S230i	SM-T300i/T300	SM-T400i	BSC10	SM-S210i StarPRNT	SM-S220i StarPRNT	SM-S230i StarPRNT	SM-T300i/T300 StarPRNT	SM-T400i StarPRNT	SM-L200	SM-L300	SP700
StarPRNT Intelligence	-	-	-	✓	-	✓	✓	✓	-	-	-	-	-	-	-	-	-	-	-	-	-	✓
StarPRNT Intelligence for mCollection	✓	✓	✓	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
StarPRNT Intelligence for Portable	-	-	-	-	-	-	-	-	-	-	-	-	-	-	✓	✓	✓	✓	✓	✓	✓	-
StarPRNT Intelligence for BSC10	-	-	-	-	-	-	-	-	-	-	-	-	-	✓	-	-	-	-	-	-	-	-
Star TSP100 futurePRNT *1	-	-	-	-	✓	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

こちらのリンクよりダウンロードしてください。

-: 対象外

*1: メモリスイッチ設定の変更はできません。

メモリスイッチ設定を変更する場合は Android 版もしくは iOS 版ソフトウェアをご利用ください。

2 StarPRNT アプリケーション構成方法

StarPRNT SDK プロジェクトには、既にライブラリ（StarIOPort3.1.jar / starioextension.jar / smcloudservices.aar）は含まれています。（この SDK をテストする場合、そのまま使用できます）新規のアプリケーションを作成する場合、プロジェクトに必要なライブラリを追加する必要があります。

2.1 ライブラリを追加する

1. StarPRNT SDK の libs フォルダにあるライブラリを選択します。



–smcloudservices.aar について–

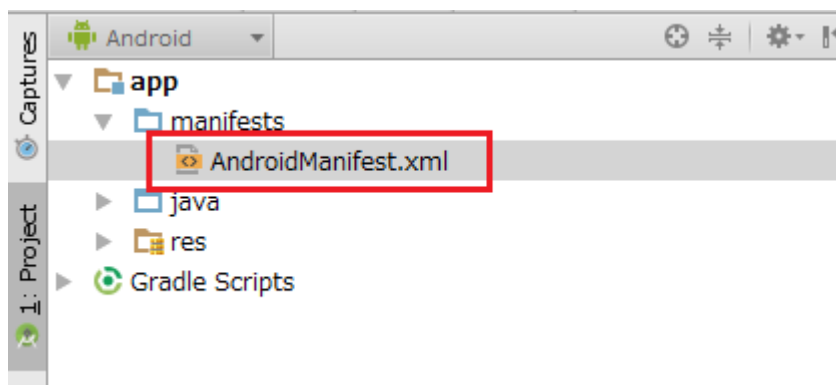
スター精密クラウドサービス（AllReceipts 等）をアプリケーションで用いる際に追加する必要があります。

2. プロジェクトの libs フォルダにライブラリをコピーします。

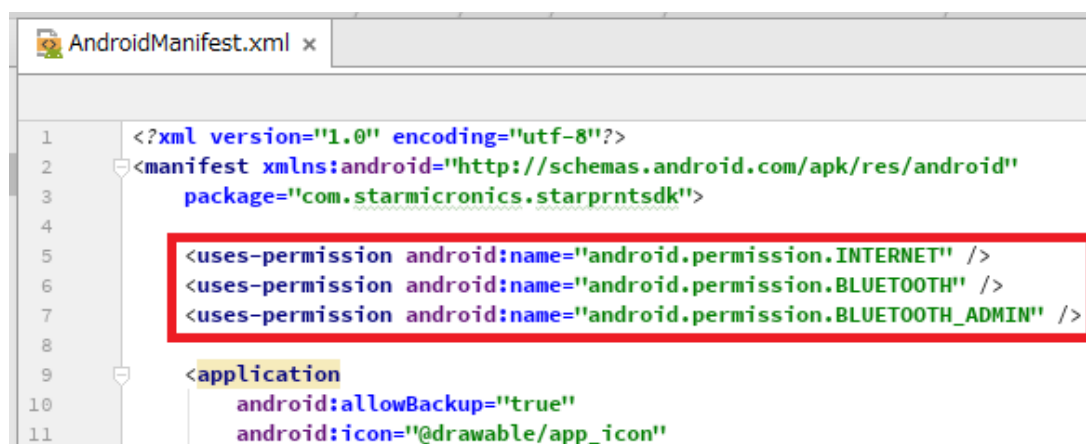


2.2 AndroidManifest.xml へ項目を設定する

1. AndroidManifest.xml ファイルをダブルクリックします。



2. TCP/IP 接続の場合、TCP/IP 接続の利用に関する追記を行います。
Bluetooth 接続の場合、Bluetooth 接続の利用に関する追記を行います。



```
<uses-permission android:name="android.permission.INTERNET" />
<uses-permission android:name="android.permission.BLUETOOTH" />
<uses-permission android:name="android.permission.BLUETOOTH_ADMIN" />
```


3. USB 接続の場合、USB 接続の利用に関する追記を行います。

```

13
14
15 <activity
16     android:name=".TitleActivity"
17     android:theme="@style/TitleTheme"
18     android:hardwareAccelerated="false"
19     android:configChanges="orientation|screenSize"
20     android:screenOrientation="portrait">
21     <meta-data android:name="android.hardware.usb.action.USB_DEVICE_ATTACHED" android:resource="@xml/device_filter" />
22
23     <intent-filter>
24         <action android:name="android.intent.action.MAIN" />
25         <category android:name="android.intent.category.LAUNCHER" />
26         <action android:name="android.hardware.usb.action.USB_DEVICE_ATTACHED" />
27     </intent-filter>
28 </activity>
29

```

```

<meta-data android:name="android.hardware.usb.action.USB_DEVICE_ATTACHED"
android:resource="@xml/device_filter" />

```

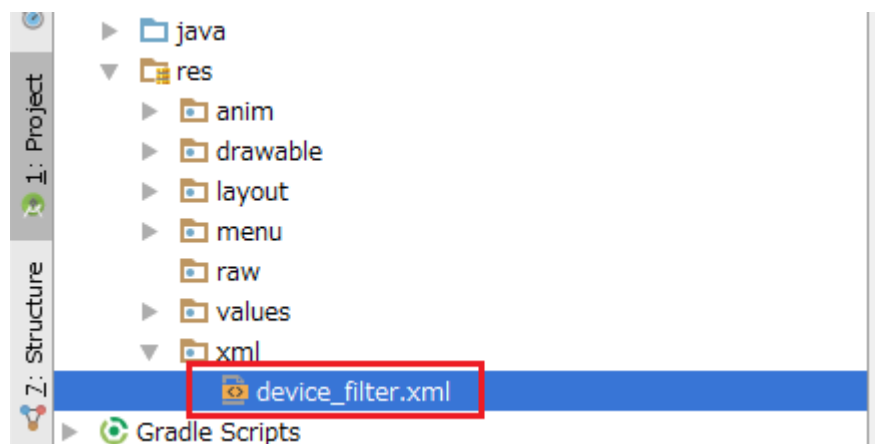
```

<action android:name="android.hardware.usb.action.USB_DEVICE_ATTACHED" />

```

2.3 device_filter.xml へ項目を設定する

1. device_filter.xml ファイルをダブルクリックします。



2. USB 接続の場合、USB 接続の利用に関する追記を行います。



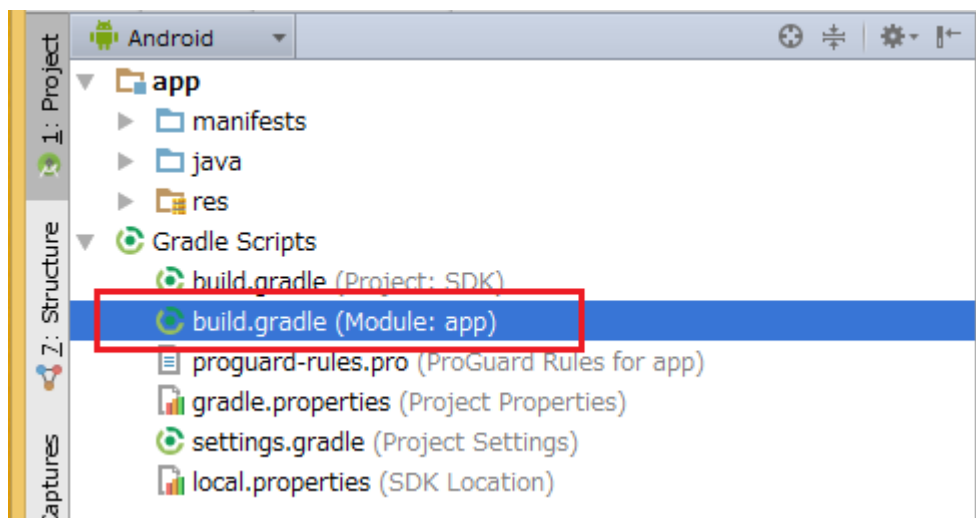
```
<usb-device class="255" subclass="66" protocol="1" />
<usb-device vendor-id="1305" product-id="****" /> <!--Target Printer -->
```

device_filter.xml に記述された Star USB プリンタについては、接続許可時にチェックボックスにチェックをいれることで、ケーブルを挿抜してもアプリケーションを再インストールしない限り許可ダイアログが表示されなくなります。また、挿抜時サンプルアプリケーションが自動で起動するようになります。

TSP100IIIU はメモリスイッチの設定により、product-id を TSP100U/ECO 互換、もしくは TSP100GT 互換に変更できます。メモリスイッチの設定は専用ソフトウェアから行えます。[プリンタ設定用ソフトウェア](#)を参照ください。

2.4 build.gradle (Module: app)へ項目を設定する

1. build.gradle (Module: app)ファイルをダブルクリックします。



2. ライブラリの利用に関する追記を行います。

```

21
22 dependencies {
23     compile files('libs/StarIOPort3.1.jar')
24     compile files('libs/starioextension.jar')
25     compile (name:'smcloudservices', ext:'aar')
26     compile 'com.android.support:appcompat-v7:23.4.0'
27     compile 'com.android.support:support-v4:23.4.0'
28     compile 'com.google.zxing:core:3.2.1'
29 }

```

```

compile files('libs/StarIOPort3.1.jar')
compile files('libs/starioextension.jar')

```

3. TSP100 シリーズにおいて QR コード / PDF417 の印刷を行う場合、ライブラリの利用に関する追記を行います。

```

22 dependencies {
23     compile files('libs/StarIOPort3.1.jar')
24     compile files('libs/starioextension.jar')
25     compile (name:'smcloudservices', ext:'aar')
26     compile 'com.android.support:appcompat-v7:23.4.0'
27     compile 'com.android.support:support-v4:23.4.0'
28     compile 'com.google.zxing:core:3.2.1'
29 }

```

```
compile 'com.google.zxing:core:3.2.1'
```

4. スター精密クラウドサービス（AllReceipts 等）をアプリケーションで用いる場合、ライブラリ
の利用に関する追記を行います。加えて、TSP100 シリーズにおいて AllReceipts の QR コード印刷
を行う場合、QR コード生成ライブラリの利用に関する追記を行います。

```

22 dependencies {
23     compile files('libs/StarIOPort3.1.jar')
24     compile files('libs/starioextension.jar')
25     compile (name:'smcloudservices', ext:'aar')
26     compile 'com.android.support:appcompat-v7:23.4.0'
27     compile 'com.android.support:support-v4:23.4.0'
28     compile 'com.google.zxing:core:3.2.1'
29 }

```

```
compile (name:'smcloudservices', ext:'aar')
```

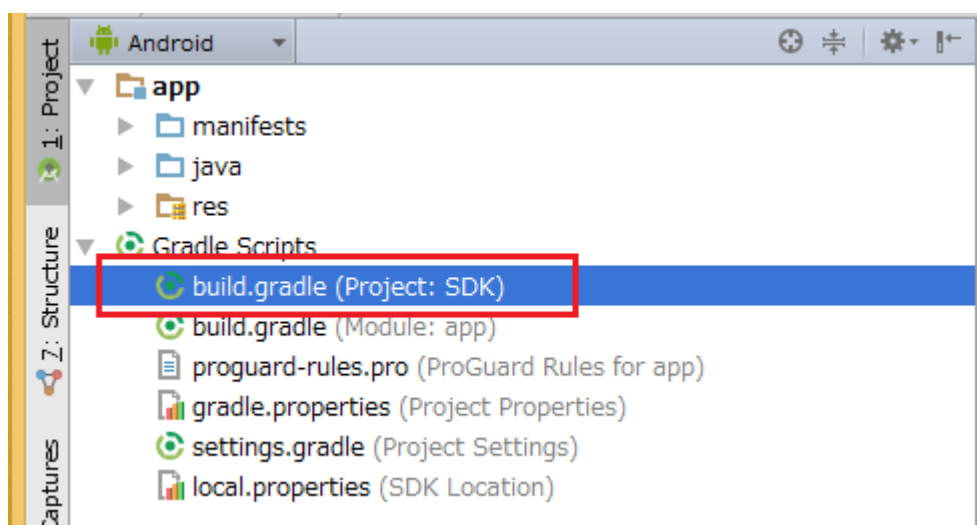
```
compile 'com.android.support:support-v4:23.4.0'
```

TSP100 シリーズにおいて AllReceipts の QR コード印刷を行う場合

```
compile 'com.google.zxing:core:3.2.1'
```

2.5 build.gradle (Project: SDK)へ項目を設定する

1. build.gradle (Project: SDK)ファイルをダブルクリックします。



2. スター精密クラウドサービス（AllReceipts等）をアプリケーションで用いる場合、ライブラリ
の利用に関する追記を行います。

```

15  allprojects {
16      repositories {
17          jcenter()
18          flatDir{
19              dirs 'libs'
20          }
21      }
22  }

```

```

flatDir{
    dirs 'libs'
}

```

3 StarIOPort クラス (StarIOPort3.1.jar)

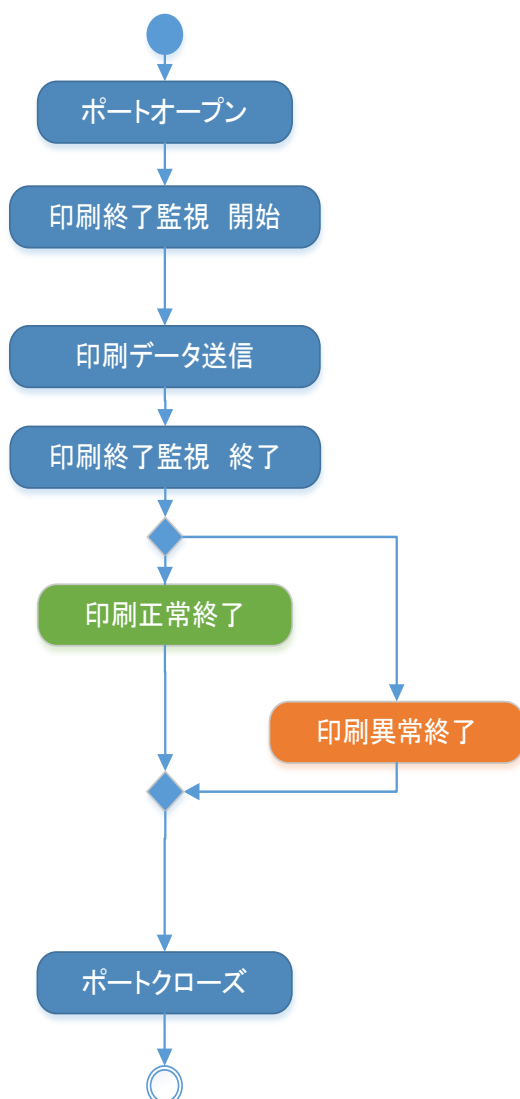
StarIOPort クラスはプリンタとの通信を行うためのクラスです。

メソッド

名称	説明
getPort	StarIOPort オブジェクトを生成し、プリンタと通信するためのポートをオープンします。
releasePort	StarIOPort オブジェクトを破棄し、プリンタと通信するためのポートをクローズします。
searchPrinter	Android デバイスと接続できるプリンタを検索します。
writePort	プリンタにデータを書き込みます。
readPort	プリンタからデータを読み込みます。
retreiveStatus	プリンタのステータスを取得します。
beginCheckedBlock	印刷終了監視を開始します。
endCheckedBlock	印刷終了監視を終了します。
setEndCheckedBlockTimeoutMillis	印刷終了監視のタイムアウト時間を設定します。
getFirmwareInformation	プリンタのモデル名とファームウェアバージョンを取得します。
getPortName	ポートをオープンしたときの portName を取得します。
getPortSettings	ポートをオープンしたときの portSettings を取得します。
getStarIOVersion	StarIO ライブラリのバージョン番号を取得します。

3.1 StarIOPort クラスを使った印刷処理フロー

StarIOPort クラスを利用し、以下の手順に沿って印刷を行います。



Communication.java を参照ください。

```

StarIOPort port = null;

try {
    // ポートオープン
    port = StarIOPort.getPort(portName, portSettings, 10000,
    context);

    // 印刷終了監視 開始
    StarPrinterStatus status = port.beginCheckedBlock();

    byte[] command =
    PrinterFunctions.createTextReceiptData(emulation, localizeReceipts,
    false);

    // 印刷データ送信
    port.writePort(command, 0, command.length);

    // 印刷終了監視 終了
    status = port.endCheckedBlock();

    // 印刷終了監視時のステータス判定
    if (status.offline == false) {
        // 印刷正常終了（プリンタオンライン）
    }
    else {
        // 印刷異常終了（紙無し、プリンタカバーオープンなど）
        // ユーザーに通知
    }
}
catch (StarIOPortException e) {
    // エラー発生
}
finally {
    try {
        // ポートクローズ
        StarIOPort.releasePort(port);
    }
    catch (StarIOPortException e) {}
}
  
```

3.2 モデル: StarIOPort クラス

各モデルにおいて機能するメソッドは以下のとおりです。

Method	mC-Print2	mC-Print3	mPOP	FVP10	TSP100	TSP650II	TSP700II	TSP800II	SM-S210i	SM-S220i	SM-S230i	SM-T300i/T300	SM-T400i	BSC10	SM-S210i StarPRNT	SM-S220i StarPRNT	SM-S230i StarPRNT	SM-T300i/T300 StarPRNT	SM-T400i StarPRNT	SM-L200	SM-L300	SP700
getPort	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
releasePort	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
searchPrinter	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
writePort	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
readPort	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
retrieveStatus	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
beginCheckedBlock	✓	✓	✓	✓	✓	✓	*1	✓	✓	*3	✓	*3	✓	✓	✓	✓	✓	✓	✓	✓	✓	*1
endCheckedBlock	✓	✓	✓	✓	✓	✓	*1	✓	✓	*3	✓	*3	✓	✓	✓	✓	✓	✓	✓	✓	✓	*1
setEndCheckedBlockTimeoutMillis	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
getFirmwareInformation	✓	✓	✓	*4	*2 *4	*4	*4	*4	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	*4
getPortName	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
getPortSetting	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
getStarIOVersion	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓

-: 無視

*1: ファームウェアバージョン 3.0 以降が必要です。

*2: TSP100IIIU, TSP100U, TSP100GT, TSP100ECO の場合、モデル名は TSP100 になります。

TSP100U, TSP100GT, TSP100ECO, TSP100LAN はファームウェアバージョンを取得できません。

*3: SM-220i はファームウェアバージョン 2.1 未満、SM-T300 はファームウェアバージョン 2.4 未満では印刷データ送信完了の確認はできますが、印刷終了監視はできません。

*4: Apple AirMac シリーズを使用する場合、USB モデルは空文字を返します。

3.3 getPort メソッド

StarIOPort オブジェクトを生成し、プリンタと通信するためのポートをオープンします。

宣言

```
public static StarIOPort getPort(String portName, String portSettings, int ioTimeoutMillis, Context context) throws StarIOPortException;
```

@Deprecated

```
public static StarIOPort getPort(String portName, String portSettings, int ioTimeoutMillis) throws StarIOPortException;
```

引数

名称	説明	型
portName	プリンタと通信するためのポートを特定する文字列	String
portSettings	プリンタとの接続設定情報 <ul style="list-style-type: none"> エミュレーションの指定 リトライの指定 データタイムアウト機能の設定 	String
ioTimeoutMillis	内部制御および API のタイムアウト値（ミリ秒指定）	int
context	コンテキスト	android.content.Context

戻り値

説明	型
StarIOPort オブジェクト	com.starmicronics.stario.StarIOPort

例外

説明	型
ポートオープンに失敗したとき 誤った portName が渡されたとき	com.starmicronics.stario.StarIOPortException

印刷の手順については [StarIOPort クラスを使った印刷処理フロー](#) を参照ください。

1. portName 引数

インターフェイス	portName	説明	例
Bluetooth	BT:	最初に見つかったスタープリンタと通信します。 スタープリンタを1台のみペアリングして使用してください。	BT:
	BT:デバイス名	Bluetooth デバイス名で指定する場合	BT:Star Micronics
	BT:MAC アドレス	MAC アドレスで指定する場合	BT:00:12:F3:1E:2B:72
Ethernet/ Wireless LAN	TCP:IP アドレス	IP アドレスで指定してください。	TCP:192.168.1.130
USB	USB:	最初に見つかったスタープリンタと通信します。 スタープリンタを USB で1台のみ接続して使用してください。	USB:
	USB:SN:シリアル ナンバー	シリアルナンバーを設定され、有効になっている USB プリンタを指定したい場合	USB:SN:1234ABCD
	USB:検索結果	searchPrinter メソッドで検索した結果得られる PortName。PortInfo クラスの getPortName メソッドで取得できます。	USB:1-3

2. portSettings 引数

portSettings 文字列は以下のフォーマットで指定してください。

- ・各モデルで適用すべき識別子+";"で区切ったオプション指定文字列

各モデルで適用すべき識別子は以下のとおりです。

Model	識別子
mC-Print2	"mC"
mC-Print3	"mC"
mPOP	"mC"
FVP10	"mC"
TSP100	"mC"
TSP650II	"mC"
TSP700II	"mC"
TSP800II	"mC"
SM-S210i	"mini"
SM-S220i	"mini"
SM-S230i	"mini"
SM-T300i/T300	"mini"
SM-T400i	"mini"
BSC10	"escpos"
SM-S210i StarPRNT	"Portable"
SM-S220i StarPRNT	"Portable"
SM-S230i StarPRNT	"Portable"
SM-T300i/T300 StarPRNT	"Portable"
SM-T400i StarPRNT	"Portable"
SM-L200	"Portable"
SM-L300	"Portable"
SP700	"mC"

オプション指定文字列は以下のとおりです。

インターフェイス	オプション指定文字列	接続オプション
Bluetooth	"[数値]" ("l"の小文字) 例: "l10000"	<p>接続のリトライをおこないます。 リトライ時間の指定方法は以下のとおりです。</p> <ul style="list-style-type: none"> ① 数値を指定しない場合 ("l"のみ) : getPort メソッドで指定する timeout 時間 ② [数値]に 0 から 300000 を指定した場合 : [数値]に指定した時間 (単位: ミリ秒) ③ [数値]に 300001 以上の値を指定した場合 : getPort メソッドで指定する timeout 時間 ④ 上記以外の場合 : リトライしない設定となります
	"d[数値]" 例: "d10"	<p>データタイムアウト機能のタイムアウト値設定を行います。 このオプションをサポートするモデルはモデル: StarIOPort クラスを参照ください。 データタイムアウト機能とは、印刷中に指定した時間プリンタに対してデータが送られなかった場合に、残りの印字データを無視する機能です。この機能により、データ送信中に Bluetooth 接続が切断された場合に、次の印刷が不正な内容になることを防ぐことができます。 タイムアウト時間の指定方法は以下のとおりです。</p> <ul style="list-style-type: none"> ① [数値]に 1 から 255 を指定した場合 : [数値]に指定した時間 (単位: 秒) ② [数値]に 0 を指定した場合 : データタイムアウト機能を無効にする ③ 上記以外の場合 : 3 秒 <p>データタイムアウト機能をサポートするモデルで d オプションを指定しない場合、データタイムアウト機能は有効で、タイムアウト値は 3 秒に設定されます。</p>
Ethernet/ Wireless LAN	"[数値]" ("l"の小文字) 例: "l10000"	<p>対象とするプリンタが他ホストから使用中の場合に接続のリトライをおこないます。 リトライ時間の指定方法は以下のとおりです。</p> <ul style="list-style-type: none"> ① 数値を指定しない場合 ("l"のみ) : getPort メソッドで指定する timeout 時間 ② [数値]に 0 から 300000 を指定した場合 : [数値]に指定した時間 (単位: ミリ秒) ③ [数値]に 300001 以上の値を指定した場合 : getPort メソッドで指定する timeout 時間 ④ 上記以外の場合 : リトライしない設定となります
	"[ポート番号]" 例: "9100"	ポート番号(AirMac 使用時のみ)

portSettings 引数指定例

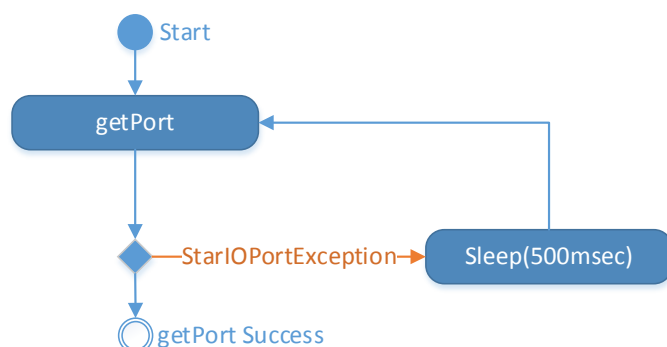
プリンタ接続環境	指定すべき portSettings
TSP650II をデフォルト設定で使用する	""
mC-Print3 を Ethernet インターフェイスでリトライ有効で使用する	";I10000"
SM-L200 を Bluetooth インターフェイスでリトライ有効で使用する	"Portable;I10000"
TSP650II の Bluetooth インターフェイスで、印刷中に 10 秒間データが送られなかった場合にデータキャンセル機能を動作させる	";d10"

3. getPort リトライフロー

Bluetooth プリンタを使用する場合、端末、使用環境によって、getPort が失敗し、Bluetooth デバイスとの接続ができず

印刷、ステータス取得ができない場合があります。

getPort が失敗した場合(StarIOException が発生します)、下記手順 II、III を繰り返す処理を行ってください。



I. getPort が失敗する (StarIOException が発生する)

II. 一定時間(例：500msec)おく

III. getPort を実行する

IV. getPort が成功した場合、印刷またステータス取得などの処理に進んでください。

* 一定時間の値や getPort の繰り返し回数は、

端末や使用環境によって

異なるため、十分な検証を行い決定してください。

4. ioTimeoutMillis 引数

ioTimeoutMillis は、ミリ秒のタイムアウト値で、内部制御と API での通信に使用されるものです。このパラメータは、API が制限された時間内で完了することを保証しますが、正確なタイムアウトの長さを保証するものではありません。

portSettings パラメータで[データタイムアウト機能のタイムアウト値設定]を行う場合、

ioTimeoutMillis 引数の値をデータタイムアウト機能のタイムアウト値の指定時間より 3 秒以上長くなるように設定してください。3 秒未満に設定した場合、内部制御により endCheckedBlock メソッドのタイムアウト時間は自動的に 3 秒長くなるように設定されます。

5. Apple AirMac シリーズのプリンタ共有機能をご使用の場合

portName には、AirMac シリーズの IP アドレスを指定してください。

例): TCP:192.168.1.2

portSettings には、ポート番号を指定します。

“9100” ~ “9109”を順に指定し、接続に成功した値をご使用ください。

3.4 releasePort メソッド

StarIOPort オブジェクトを破棄し、プリンタと通信するためのポートをクローズします。

宣言

```
public static void releasePort(StarIOPort port);
```

引数

名称	説明	型
port	以前に getPort メソッドにより生成された StarIOPort オブジェクト	com.starmicronics.stario.StarIOPort

戻り値

説明	型
-	-

印刷の手順については [StarIOPort クラスを使った印刷処理フロー](#) を参照ください。

1. 注意事項

プリンタとのデータ通信を行わない時は、releasePort メソッドを呼び出し、ポートをクローズしてください。

ポートをオープンしたままにしておくと、今後のポートオープンの呼び出し要求が失敗する可能性があります。

2. Ethernet インターフェイスを使用する場合の注意事項

releasePort メソッド実行後すぐに、getPort メソッドを実行すると失敗する場合があります。

releasePort メソッド実行後、500 ミリ秒の時間をおいてから getPort メソッドを実行してください。

3.5 searchPrinter メソッド

Star プリンタを検索し、検索結果を ArrayList<PortInfo>にて返します。

宣言

```
public static ArrayList<PortInfo> searchPrinter(String target, Context context) throws
StarIOPortException;
public static ArrayList<PortInfo> searchPrinter(String target) throws StarIOPortException;
```

引数

名称	説明	型
target	検索したい Star プリンタのインターフェイスタイプを指定する文字列	com.starmicronics.stario.StarIOPort
	Bluetooth I/F の場合	
	Ethernet / Wireless LAN I/F の場合	
	USB I/F の場合	
context	コンテキスト target に"USB:"を指定する場合に必要です。	android.content.Context

戻り値

説明	型
Star プリンタの検索結果 取得できる情報については PortInfo クラス を参照ください。	ArrayList<com.starmicronics.stario.PortInfo>

例外

説明	型
ポートオープンに失敗した 誤った target が渡された	com.starmicronics.stario.StarIOPortException

1. USB プリンタを接続しなおした場合

USB プリンタを接続し直した際は、再度 searchPrinter メソッドを実行し、最新のプリンタ情報を取得してください。

2. 制限事項

本 API はデバイスを確実に検出する事を保証するものではありません。

3. Android7.0 以降での制限事項

Android 7.0 以降では、searchPrinter メソッドにより得られるプリンタ情報のうち、PortName（例 USB:1-3）は USB プリンタの再接続によって変わります。

Example

```
try {
    // 同じネットワーク内の Ethernet / Wireless LAN インターフェイスのスタープリンタを検索します。
    List<PortInfo> portList = StarIOPort.searchPrinter("TCP:");

    for (PortInfo port : portList) {
        Log.i("LOG", "Port Name: " + port.getPortName());
        Log.i("LOG", "MAC Address: " + port.getMacAddress());
        Log.i("LOG", "Model Name: " + port.getModelName());
    }
} catch (StarIOPortException e) {
    // エラー発生
}
```

SearchPortFragment.java を参照ください。

3.6 writePort メソッド

プリンタにデータを書き込みます。

宣言

```
public void writePort(byte[] writeBuffer, int offset, int size) throws StarIOPortException;
```

引数

名称	説明	型
writeBuffer	書き込みたいデータを格納する byte 配列	byte[]
offset	writeBuffer のどの位置から書き込み始めるかを指定するインデックス	int
size	書き込む byte 数	int

戻り値

説明	型
-	-

例外

説明	型
通信エラーが発生したとき	com.starmicronics.stario.StarIOPortException

印刷の手順については [StarIOPort クラスを使った印刷処理フロー](#) を参照ください。

3.7 readPort メソッド

プリンタからデータを読み込みます。プリンタから Raw byte を読み取る必要のある場合のみ、ご使用ください。

宣言

```
public int readPort(byte[] readBuffer, int offset, int size) throws StarIOPortException;
```

引数

名称	説明	型
readBuffer	データが読み込まれる byte 配列	byte[]
offset	readBuffer にデータを書き込み始めるインデックス	int
size	読み込む byte 数	int

戻り値

説明	型
実際に読み込まれたバイト数。 データが全て読み取れなかった時でも、readPort メソッドは成功します。アプリケーションは期待されるデータが全て読み取れるまで、readPort メソッドを複数回呼び出す必要があります。または、しきい値に達するまで再試行をするようにします。	int

例外

説明	型
通信エラーが発生したとき	com.starmicronics.stario.StarIOPortException

1. 制限事項

Raw Status の取得にこのメソッドを使用しないでください。ステータスの取得には retrieveStatus メソッドを使用してください。

3.8 retrieveStatus メソッド

プリンタのステータスを取得します。

宣言

```
public StarPrinterStatus retrieveStatus() throws StarIOPortException;
```

引数

名称	説明	型
-	-	-

戻り値

説明	型
現在のプリンタのステータス 取得できるステータスの種類は StarPrinterStatus クラス を参照 ください。	com.starmicronics.stario.StarPrinterStatus

例外

説明	型
通信エラーが発生したとき	com.starmicronics.stario.StarIOPortException

Example

```
StarPrinterStatus status = port.retrieveStatus();
```

```
if (status.offline == false) {
    // プリンタオンライン
    if (status.blackMarkError == true) {
        // ブラックマークエラー発生
    }
    if (status.compulsionSwitch == true) {
        // キャッシュドロアオープン
    }
    else {
        // キャッシュドロアクローズ
    }
}
else {
    // プリンタオフライン
}
```

Communication.java を参照ください。

3.9 beginCheckedBlock メソッド

endCheckedBlock メソッドとセットで使用して印字終了の監視を行います。

最後まで正しく印刷されたか確認したい場合に、印刷データ送信の直前に本メソッドを、印刷データ送信の直後に endCheckedBlock メソッドを実行します。

宣言

```
public StarPrinterStatus beginCheckedBlock() throws StarIOPortException;
```

引数

名称	説明	型
-	-	-

戻り値

説明	型
現在のプリンタのステータス 取得できるステータスの種類は StarPrinterStatus クラス を参照 ください。	com.starmicronics.stario.StarPrinterStatus

例外

説明	型
通信エラーが発生したとき プリンタがオフライン状態のとき	com.starmicronics.stario.StarIOPortException

beginCheckedBlock/endCheckedBlock による印刷終了監視処理の手順は [StarIOPort クラスを使った印刷処理フロー](#) を参照ください。

3.10 endCheckedBlock メソッド

beginCheckedBlock メソッドとセットでを使用して印字終了の監視を行います。
 プリンタの状態を監視し、送信した印刷データの印刷が完了すると制御を返します。
 印刷データ以外を送信した場合は、そのコマンドがプリンタに処理されると制御を返します。
 setEndCheckedBlockTimeoutMillis メソッドで指定したタイムアウト時間内に印刷が完了しなかった場合や、印刷中に通信エラーが発生した場合は、例外 StarIOException をスローします。

宣言

```
public StarPrinterStatus endCheckedBlock() throws StarIOException;
```

引数

名称	説明	型
-	-	-

戻り値

説明	型
現在のプリンタのステータス 取得できるステータスの種類は StarPrinterStatus クラス を参照ください。	com.starmicronics.stario.StarPrinterStatus

例外

説明	型
通信エラーが発生したとき タイムアウト時間内にプリンタからの応答がないとき	com.starmicronics.stario.StarIOException

beginCheckedBlock/endCheckedBlock による印刷終了監視処理の手順は [StarIOException クラスを使った印刷処理フロー](#) を参照ください。

3.11 setEndCheckedBlockTimeoutMillis メソッド

endCheckedBlock メソッドのタイムアウト時間を設定します。[単位: ミリ秒]

印刷に時間がかかる場合、この値を大きくする事で endCheckedBlock メソッドの印刷完了待ち時間を伸ばす事ができます。

宣言

```
public void setEndCheckedBlockTimeoutMillis(int endCheckedBlockTimeoutMillis);
```

引数

名称	説明	型
endCheckedBlockTimeoutMillis	endCheckedBlock メソッドのタイムアウト値（ミリ秒指定）	int

戻り値

説明	型
-	-

1. endCheckedBlockTimeoutMillis 引数

endCheckedBlock メソッドのタイムアウト時間の初期値は、getPort メソッドの引数で指定したタイムアウト時間と同じになります。

指定した時間が 10 秒未満の場合、タイムアウトは 10 秒になります。

また、getPort メソッドの portSettings パラメータで[データタイムアウト機能の設定]を使用する場合、このメソッドで設定するタイムアウト時間には、データタイムアウト機能の指定時間より 3 秒以上長くなるように設定してください。3 秒未満に設定した場合、内部制御により自動的に 3 秒長くなるように設定されます。

3.12 getFirmwareInformation メソッド

プリンタのモデル名とファームウェアバージョンを取得します。

宣言

```
public Map<String, String> getFirmwareInformation() throws StarIOPortException;
```

引数

名称	説明	型
-	-	-

戻り値

説明	型
ファームウェア情報取得結果の Map オブジェクト 戻り値の Map オブジェクトのキーに"ModelName"を設定することでモデル名を、"FirmwareVersion"を設定することでファームウェアバージョンを取得できます。	Map

例外

説明	型
通信エラーが発生したとき	com.starmicronics.stario.StarIOPortException

Example

```
try {
    Map<String, String> firmwareInformationMap = port.getFirmwareInformation();

    Log.i ("LOG", "Model Name: " + firmwareInformationMap.get("ModelName"));
    Log.i ("LOG", "Firmware Version: " + firmwareInformationMap.get("FirmwareVersion"));
}
catch (StarIOPortException e) {
    // エラー発生
}
```

DeviceStatusFragment.java / Communication.java を参照ください。

3.13 getPortName メソッド

getPort メソッドで指定した portName を取得します。

宣言

```
public String getPortName();
```

引数

名称	説明	型
-	-	-

戻り値

説明	型
getPort メソッドに渡した portName 引数	String

3.14 getPortSettings メソッド

getPort メソッドで指定した portSettings を取得します。

宣言

```
public String getPortSettings();
```

引数

名称	説明	型
-	-	-

戻り値

説明	型
getPort メソッドに渡した portSettings 引数	String

3.15 getStarIOVersion メソッド

StarIO ライブラリのバージョンを取得します。

宣言

```
public String getStarIOVersion();
```

引数

名称	説明	型
-	-	-

戻り値

説明	型
StarIO ライブラリのバージョン文字列	String

4 StarPrinterStatus クラス (StarIOPort3.1.jar)

プリンタのステータスを保持するクラスです。

boolean 型とバイナリ(raw フィールド)の両方の形式でプリンタステータスを保持します。取得できるステータスの種類はフィールド一覧をご覧ください。

フィールド

名称	説明	型	詳細
blackMarkError	ブラックマークエラー	boolean	ブラックマークエラー(ブラックマーク設定時に非ブラックマーク用紙を使って印刷した場合等に発生)の時に true となる。通常時は false。
compulsionSwitch	コンパルジョンスイッチ	boolean	ドロワのコンパルジョン SW が押されていると true となる。通常は false。
coverOpen	カバーオープン	boolean	カバーの状態 boolean カバーが開いている場合に true となる。閉じている場合は false。
cutterError	オートカッターエラー	boolean	カッターエラー発生時に true となる。
etbAvailable	ETB 使用可否	boolean	ETB が使用可能な場合に true となる。使用できない場合は false。
etbCounter	ETB カウンタ	int	現在の ETB カウンタの値。
headThermistorError	ヘッドサーミスタエラー	boolean	ヘッドサーミスタ異常値検出時に true となる。通常は false。
offline	オフライン	boolean	オフラインの場合に true となる。オンライン時は false。
overTemp	印字ヘッド高温による停止中	boolean	ヘッドが高温になり印刷停止している状態で true となる。通常は false。
raw	ステータスバイト列	byte[]	ステータスのバイト列 (例: HEX 23 86 00 00 00 00 00 00)
rawLength	raw の長さ	int	raw の長さ
receiptPaperEmpty	用紙エンド	boolean	用紙切れの場合は true となる。通常は false。
receiptPaperNearEmptyInner	用紙ニアエンド(内側)	boolean	用紙ニアエンド状態の時に true となる。通常は false。
receiveBufferOverflow	受信バッファオーバーフロー	boolean	受信バッファフルの時に true となる。通常は false。
unrecoverableError	復帰不可能エラー	boolean	復帰不可能エラー(ヘッドサーミスタエラー、オートカッターエラー、電源電圧エラー等)が発生した場合に true となる。通常は false。
voltageError	電源電圧エラー	boolean	電源電圧異常値検出時に true となる。通常は false。
connectedInterface	現在通信しているインターフェイスの種類。	int	以下のいずれかの数値。 0: USB-B 1: USB-A 2: Bluetooth 3: Ethernet

4.1 モデル: StarPrinterStatus クラス

各モデルにおいて機能するフィールドは以下のとおりです。

Field	mC-Print2	mC-Print3	mPOP	FVP10	TSP100	TSP650II	TSP700II	TSP800II	SM-S210i	SM-S220i	SM-S230i	SM-T300i/T300	SM-T400i	BSC10	SM-S210i StarPRNT	SM-S220i StarPRNT	SM-S230i StarPRNT	SM-T300i/T300 StarPRNT	SM-T400i StarPRNT	SM-L200	SM-L300	SP700
blackMarkError	-	-	-	✓	-	-	✓	✓	-	-	-	-	-	-	✓	✓	✓	✓	✓	✓	✓	✓
compulsionSwitch	✓	✓	✓	✓	✓	✓	✓	✓	-	-	-	-	-	✓	-	-	-	-	-	-	-	✓
coverOpen	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
cutterError	✓	✓	✓	✓	✓	✓	✓	✓	-	-	-	-	-	-	-	-	-	-	-	-	-	✓
etbAvailable	✓	✓	✓	✓	✓	✓	✓	✓	-	-	-	-	-	-	✓	✓	✓	✓	✓	✓	✓	✓
etbCounter	✓	✓	✓	✓	✓	✓	✓	✓	-	-	-	-	-	-	✓	✓	✓	✓	✓	✓	✓	✓
headThermistorError	✓	✓	✓	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
offline	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
overTemp	✓	✓	✓	✓	✓	✓	✓	✓	-	-	-	-	-	-	✓	✓	✓	✓	✓	✓	✓	✓
raw	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	-	✓	✓	✓	✓	✓	✓	✓	✓
rawLength	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	-	✓	✓	✓	✓	✓	✓	✓	✓
receiptPaperEmpty	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
receiptPaperNearEmptyInner	-	✓	-	✓	-	✓	✓	✓	-	-	-	-	-	-	-	-	-	-	-	-	-	✓
receiveBufferOverflow	-	-	-	✓	-	✓	✓	✓	-	-	-	-	-	-	-	-	-	-	-	-	-	✓
unrecoverableError	✓	✓	✓	✓	✓	✓	✓	✓	-	-	-	-	-	-	-	-	-	-	-	-	-	✓
voltageError	✓	✓	✓	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
connectedInterface	✓	✓	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

-: 機能しない

5 PortInfo クラス (StarIOPort3.1.jar)

Star プリンタと接続するための情報やモデル情報を保持するクラスです。

メソッド

名称	説明
getPortName	プリンタの PortName を取得します。
getMacAddress	プリンタの MAC アドレスを取得します。
getModelName	プリンタのモデル名を取得します。
getUSBSerialNumber	USB インターフェイスのプリンタの USB シリアルナンバーを取得します。

5.1 getPortName メソッド

プリンタの PortName を取得します。

宣言

```
public String getPortName();
```

引数

名称	説明	型
-	-	-

戻り値

説明	型
プリンタと通信するためのポートを特定する文字列 getPort メソッド で接続先プリンタを指定するのに使用できます。	String

5.2 getMacAddress メソッド

プリンタの MAC アドレスを取得します。

宣言

```
public String getMacAddress();
```

引数

名称	説明	型
-	-	-

戻り値

説明	型
プリンタの MAC アドレスを表す文字列 プリンタと Bluetooth インターフェイスで接続する場合、 getPort メソッド で接続先プリンタを指定するのに使用できます。 プリンタと USB インターフェイスで接続する場合、空文字となります。	String

5.3 getModelName メソッド

プリンタのモデル名を取得します。

宣言

```
public String getModelName();
```

引数

名称	説明	型
-	-	-

戻り値

説明	型
プリンタのモデル名を表す文字列 プリンタと Bluetooth インターフェイスで接続する場合、空文字となります。	String

5.4 getUSBSerialNumber メソッド

USB インターフェイスのプリンタの USB シリアルナンバーを取得します。

宣言

```
public String getUSBSerialNumber();
```

引数

名称	説明	型
-	-	-

戻り値

説明	型
USB プリンタのシリアルナンバーを表す文字列 プリンタと USB インターフェイスで接続する場合、 getPort メソッド で接続先プリンタを指定するのに使用できます。	String

6 StarBluetoothManager クラス (StarIOPort3.1.jar)

StarBluetoothManager クラスは、Bluetooth インターフェイスの各種設定を行うためのクラスです。StarIOPort クラスと同時に使用しないでください。

コンストラクタ

名称	説明
StarBluetoothManager	StarBluetoothManager を初期化します。

メソッド

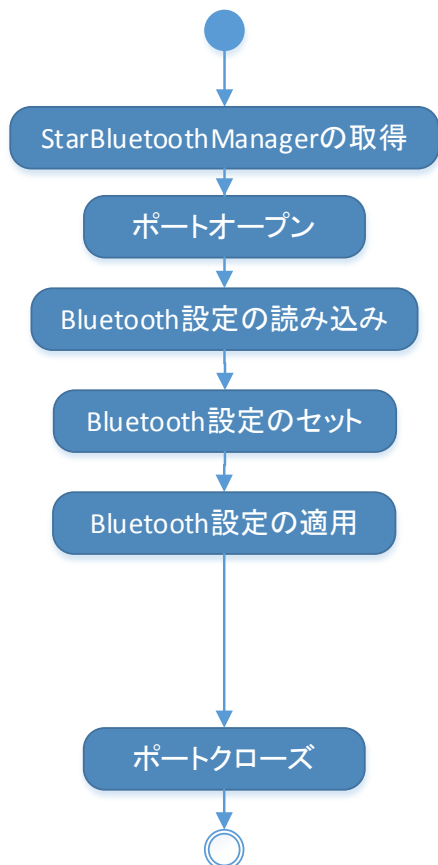
名称	説明
open	StarBluetooth デバイスとの接続を開きます。
loadSetting	接続した StarBluetooth デバイスから設定を取得します。
close	StarBluetooth デバイスとの接続を閉じます。
apply	StarBluetooth デバイスに指定した値を設定します。
getBluetoothDeviceNameCapability	Bluetooth デバイス名の設定可否情報を取得します。
getBluetoothDeviceName	Bluetooth デバイス名を取得します。
setBluetoothDeviceName	Bluetooth デバイス名を設定します。
getIOPortNameCapability	iOSPort 名の設定可否情報を取得します。
getIOPortName	iOSPort 名を取得します。
setIOPortName	iOSPort 名を設定します。
getPinCodeCapability	PIN コード設定可否情報を取得します。
getPinCode	PIN コードを取得します。
setPinCode	PIN コードを設定します。
getAutoConnectCapability	AutoConnection の設定可否情報を取得します。
getAutoConnect	AutoConnection のオン／オフを取得します。
setAutoConnect	AutoConnection のオン／オフを設定します。
getSecurityTypeCapability	Bluetooth セキュリティタイプの設定可否情報を取得します。
getSecurityType	Bluetooth セキュリティタイプを取得します。
setSecurityType	Bluetooth セキュリティタイプを設定します。
getPortName	StarBluetoothManager コンストラクタで指定した portName を取得します。
getPortSettings	StarBluetoothManager コンストラクタで指定した portSetting を取得します。
getTimeoutMillis	StarBluetoothManager コンストラクタで指定した ioTimeoutMillis を取得します。
getDeviceType	StarBluetoothManager コンストラクタで指定した deviceType を取得します。
isOpened	ポートの状態を取得します。

コンスタント

名称	説明
StarDeviceType	プリンタタイプ定数。
StarBluetoothSecurity	Bluetooth セキュリティタイプ定数。
StarBluetoothSettingCapability	Bluetooth 設定可否定数。

6.1 StarBluetoothManager を使った Bluetooth 設定変更フロー

StarBluetoothManager クラスを利用し、以下の手順に沿ってプリンタの Bluetooth 設定変更を行います。



```

StarBluetoothManager manager = null;

try {
    // StarBluetoothManager の取得
    manager = StarBluetoothManagerFactory.getManager(portName,
        portSettings, 10000, emulation);

    // ポートオープン
    manager.open();

    // Bluetooth 設定の読み込み
    manager.loadSetting();

    // Bluetooth 設定のセット
    // 例) デバイス名を"Star Micronics"に設定したい場合
    manager.setBluetoothDeviceName("Star Micronics");

    // Bluetooth 設定の適用
    manager.apply();
}
catch (StarIOException e) {
    // エラー発生
}
finally {
    if (manager.isOpened()) {
        try {
            // ポートクローズ
            manager.close();
        }
        catch (StarIOException e) {}
    }
}
  
```

BluetoothSettingFragment.java を参照ください。

6.2 モデル: StarBluetoothManager クラス

各モデルにおいて機能するメソッドは以下のとおりです。

Method	mC-Print2	mC-Print3	mPOP	FVP10	TSP100 (TSP100IIIのみ)	TSP650II	TSP700II	TSP800II	SM-S210i *1	SM-S220i *1	SM-S230i *1	SM-T300i/T300 *1	SM-T400i *1	BSC40	SM-S210i StarPRNT	SM-S220i StarPRNT	SM-S230i StarPRNT	SM-T300i/T300 StarPRNT	SM-T400i StarPRNT	SM-L200	SM-L300	SP700
open	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	-	✓	✓	✓	✓	✓	✓	✓	✓
loadSetting	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	-	✓	✓	✓	✓	✓	✓	✓	✓
close	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	-	✓	✓	✓	✓	✓	✓	✓	✓
apply	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	-	✓	✓	✓	✓	✓	✓	✓	✓
getBluetoothDeviceNameCapability	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	-	✓	✓	✓	✓	✓	✓	✓	✓
getBluetoothDeviceName	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	-	✓	✓	✓	✓	✓	✓	✓	✓
setBluetoothDeviceName	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	-	✓	✓	✓	✓	✓	✓	✓	✓
getiOSPortNameCapability	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	-	✓	✓	✓	✓	✓	✓	✓	✓
getiOSPortName	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	-	✓	✓	✓	✓	✓	-	-	✓
setiOSPortName	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	-	✓	✓	✓	✓	✓	-	-	✓
getPinCodeCapability	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	-	✓	✓	✓	✓	✓	✓	✓	✓
getPinCode	-	-	-	✓	-	✓	✓	✓	✓	✓	✓	✓	✓	-	✓	✓	✓	✓	✓	✓	✓	✓
setPinCode	-	-	-	✓	-	✓	✓	✓	✓	✓	✓	✓	✓	-	✓	✓	✓	✓	✓	✓	✓	✓
getAutoConnectCapability	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	-	✓	✓	✓	✓	✓	✓	✓	✓
getAutoConnect	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	-	✓	✓	✓	✓	✓	-	-	✓
setAutoConnect	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	-	✓	✓	✓	✓	✓	-	-	✓
getSecurityTypeCapability	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	-	✓	✓	✓	✓	✓	✓	✓	✓
getSecurityType	-	-	-	*2	-	*2	*2	*2	*3	*3	*3	*3	*3	-	*3	*3	*3	*3	*3	*3	*3	*2
setSecurityType	-	-	-	*2	-	*2	*2	*2	*3	*3	*3	*3	*3	-	*3	*3	*3	*3	*3	*3	*3	*2
getPortName	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	-	✓	✓	✓	✓	✓	✓	✓	✓
getPortSetting	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	-	✓	✓	✓	✓	✓	✓	✓	✓
getTimeoutMillis	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	-	✓	✓	✓	✓	✓	✓	✓	✓
getDeviceType	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	-	✓	✓	✓	✓	✓	✓	✓	✓
isOpened	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	-	✓	✓	✓	✓	✓	✓	✓	✓

-: 機能しない

*1: ファームウェアバージョン 3.0 以降が必要です

*2: 設定・取得できる SecurityType は PIN コードもしくは SSP

*3: 設定・取得できる SecurityType は PIN コードもしくは Disable

6.3 StarBluetoothManager コンストラクタ

StarBluetoothManager を初期化します。

宣言

```
StarBluetoothManager(String portName, String portSetting, int ioTimeoutMillis,
StarBluetoothManager.StarDeviceType starDeviceType);
```

引数

名称	説明	型
portName	プリンタポート名 StarIOPort クラスの getPort メソッド の portName と同じです。	String
portSettings	ポートセッティング StarIOPort クラスの getPort メソッド の portSetting と同じです。	String
ioTimeoutMillis	内部制御および API のタイムアウト値	int
starDeviceType	プリンタタイプ定数	com.starmicronics.stario.StarBluetoothManager.StarDeviceType

戻り値

説明	型
StarBluetoothManager オブジェクト	com.starmicronics.stario.StarBluetoothManager

例外

説明	型
ポートオープンに失敗したとき	com.starmicronics.stario.StarIOPortException

Bluetooth 設定変更の手順は [StarBluetoothManager を使った Bluetooth 設定変更フロー](#)を参照ください。

各モデルに対して適用すべき StarDeviceType コンスタント定数は [StarDeviceType コンスタント](#)を参照ください。

1. StarBluetoothManagerFactory クラス

[StarBluetoothManagerFactory クラス](#)は、プリンタのエミュレーションに対して適切な StarBluetoothManager オブジェクトを取得することができます。

6.4 open メソッド

StarBluetooth デバイスとの接続を開きます。

open メソッド実行後は、必ず loadSetting メソッドを実行し、現在の設定を取得してください。

宣言

```
public void open();
```

引数

名称	説明	型
-	-	-

戻り値

説明	型
-	-

例外

説明	型
通信エラーが発生したとき	com.starmicronics.stario.StarIOException

Bluetooth 設定変更の手順は [StarBluetoothManager を使った Bluetooth 設定変更フロー](#) を参照ください。

6.5 loadSetting メソッド

接続した StarBluetooth デバイスから設定を取得します。

宣言

```
public void loadSetting();
```

引数

名称	説明	型
-	-	-

戻り値

説明	型
-	-

例外

説明	型
通信エラーが発生したとき	com.starmicronics.stario.StarIOException

Bluetooth 設定変更の手順は [StarBluetoothManager を使った Bluetooth 設定変更フロー](#) を参照ください。

6.6 close メソッド

StarBluetooth デバイスとの接続を閉じます。

宣言

```
public void close();
```

引数

名称	説明	型
-	-	-

戻り値

説明	型
-	-

例外

説明	型
通信エラーが発生したとき	com.starmicronics.stario.StarIOException

Bluetooth 設定変更の手順は [StarBluetoothManager を使った Bluetooth 設定変更フロー](#) を参照ください。

6.7 apply メソッド

StarBluetooth デバイスに以下のメソッドで指定した値を設定します。

- setBluetoothDevice メソッド
- setiOSPortName メソッド
- setAutoConnect メソッド
- setPinCode メソッド
- setSecurityType メソッド

宣言

```
public void apply();
```

引数

名称	説明	型
-	-	-

戻り値

説明	型
-	-

例外

説明	型
通信エラーが発生したとき	com.starmicronics.stario.StarIOException

Bluetooth 設定変更の手順は [StarBluetoothManager を使った Bluetooth 設定変更フロー](#) を参照ください。

6.8 getBluetoothDeviceNameCapability メソッド

Bluetooth デバイス名の設定可否情報を取得します。

宣言

```
public StarBluetoothManager.StarBluetoothSettingCapability getBluetoothDeviceNameCapability();
```

引数

名称	説明	型
-	-	-

戻り値

説明	型
Bluetooth デバイス名の設定可否情報	StarBluetoothManager.StarBluetoothSettingCapability

6.9 getBluetoothDeviceName メソッド

loadSetting メソッドで取得した、または SetBluetoothDeviceName メソッドで設定した Bluetooth デバイス名を取得します。

loadSetting メソッド、または setBluetoothDeviceName メソッドを呼び出した後に使用してください。

宣言

```
public String getBluetoothDeviceName();
```

引数

名称	説明	型
-	-	-

戻り値

説明	型
Bluetooth デバイス名	String

6.10 setBluetoothDeviceName メソッド

Bluetooth デバイス名を設定します。

Bluetooth デバイス名を変更するには、setBluetoothDeviceName メソッド後に apply メソッドを実行してください。

宣言

```
public void setBluetoothDeviceName(String bluetoothDeviceName);
```

引数

名称	説明	型
bluetoothDeviceName	Bluetooth デバイス名	String

戻り値

説明	型
-	-

例外

説明	型
使用可能文字以外を含む文字列を設定したとき 文字数制限を満たさない文字列を設定したとき	com.starmicronics.stario.StarIOException

1. 使用可能文字

0-9 a-z A-Z ; : ! ? # \$ % & , . @ _ - = Space / * + ~ ^ [{ () }] \

2. 文字数制限

1 文字以上かつ 16 文字以下

6.11 getiOSPortNameCapability メソッド

iOSPort 名の設定可否情報を取得します。

宣言

```
public StarBluetoothManager.StarBluetoothSettingCapability getiOSPortName();
```

引数

名称	説明	型
-	-	-

戻り値

説明	型
iOS ポート名の設定可否情報	StarBluetoothManager.StarBluetoothSettingCapability

6.12 getiOSPortName メソッド

loadSetting メソッドで取得した、または setiOSPortName メソッドで設定した iOSPort 名を取得します。

loadSetting メソッド、または setiOSPortName メソッドを呼び出した後に使用してください。

宣言

```
public String getiOSPortName();
```

引数

名称	説明	型
-	-	-

戻り値

説明	型
iOS ポート名	String

6.13 setiOSPortName メソッド

iOSPort 名を設定します。

iOSPort 名を変更するには、setiOSPortName メソッド後に apply メソッドを実行してください。

宣言

```
public void setiOSPortName(String iOSPortName);
```

引数

名称	説明	型
iOSPortName	iOS ポート名	String

戻り値

説明	型
-	-

例外

説明	型
使用可能文字以外を含む文字列を設定したとき 文字数制限を満たさない文字列を設定したとき	com.starmicronics.stario.StarIOPortException

1. 使用可能文字

0-9 a-z A-Z ; : ! ? # \$ % & , . @ _ - = Space / * + ~ ^ [{ () }] \

2. 文字数制限

1 文字以上かつ 16 文字以下

6.14 getPinCodeCapability メソッド

PIN コード設定可否情報を取得します。

宣言

```
public StarBluetoothManager.StarBluetoothSettingCapability getPinCodeCapability();
```

引数

名称	説明	型
-	-	-

戻り値

説明	型
PIN コード設定可否情報	StarBluetoothManager.StarBluetoothSettingCapability

6.15 getPinCode メソッド

setPinCode メソッドで設定した PIN コードの設定を取得します。

setPinCode メソッドを呼び出した後に使用してください。

宣言

```
public String getPinCode();
```

引数

名称	説明	型
-	-	-

戻り値

説明	型
PIN コード	String

6.16 setPinCode メソッド

StarBluetooth デバイスの PIN コードを設定します。

PIN コードを変更するには、setPinCode メソッド後に apply メソッドを実行してください。

宣言

```
public void setPinCode(String pinCode);
```

引数

名称	説明	型
	-	String

戻り値

説明	型
-	-

例外

説明	型
使用可能文字以外を含む文字列を設定したとき 文字数制限を満たさない文字列を設定したとき	com.starmicronics.stario.StarIOException

1. 使用可能文字

0-9 (SM-L200, SM-L300)

0-9 a-z A-Z (その他のモデル)

2. 文字数制限

4 文字 (SM-L200, SM-L300)

4 文字以上かつ 16 文字以下 (その他のモデル)

6.17 getAutoConnectCapability メソッド

AutoConnection の設定可否情報を取得します。

宣言

```
public StarBluetoothManager.StarBluetoothSettingCapability getAutoConnectCapability();
```

引数

名称	説明	型
-	-	-

戻り値

説明	型
AutoConnection の設定可否情報	StarBluetoothManager.StarBluetoothSettingCapability

6.18 getAutoConnect メソッド

loadSetting メソッドで取得した、または setAutoConnect メソッドで設定した AutoConnection の設定を取得します。

loadSetting メソッド、または setAutoConnect メソッドを呼び出した後に使用してください。

宣言

```
public boolean getAutoConnect();
```

引数

名称	説明	型
-	-	-

戻り値

説明	型
AutoConnection 設定 true - AutoConnection 設定がオン false - AutoConnection 設定がオフ	boolean

6.19 setAutoConnect メソッド

StarBluetooth デバイスの AutoConnection を設定します。

AutoConnection の設定を変更するには、setAutoConnect メソッド後に apply メソッドを実行してください。

宣言

```
public void setAutoConnect(boolean autoConnect);
```

引数

名称	説明	型
autoConnect	AutoConnection 設定 true - AutoConnection 設定をオンにする false - AutoConnection 設定をオフにする	boolean

戻り値

説明	型
-	-

6.20 getSecurityTypeCapability メソッド

Bluetooth セキュリティタイプの設定可否情報を取得します。

宣言

```
public StarBluetoothManager.StarBluetoothSettingCapability getSecurityTypeCapability();
```

引数

名称	説明	型
-	-	-

戻り値

説明	型
Bluetooth セキュリティタイプの設定可否情報	StarBluetoothManager.StarBluetoothSettingCapability

6.21 getSecurityType メソッド

loadSetting メソッドで取得した、または setSecurityType で設定した Bluetooth セキュリティタイプを取得します。

loadSetting メソッドまたは setSecurityType メソッドを呼び出した後に使用してください。

宣言

```
public StarBluetoothManager.StarBluetoothSecurity getSecurityType();
```

引数

名称	説明	型
-	-	-

戻り値

説明	型
Bluetooth セキュリティタイプ	com.starmicronics.stario.StarBluetoothManager.StarBluetoothSecurity

6.22 setSecurityType メソッド

StarBluetooth デバイスのセキュリティタイプを設定します。

セキュリティタイプを変更するには、setSecurityType メソッド後に apply メソッドを実行してください。

宣言

```
public void setSecurityType(StarBluetoothManager.StarBluetoothSecurity securityType);
```

引数

名称	説明	型
securityType	Bluetooth セキュリティタイプ	com.starmicronics.stario.StarBluetoothManager.StarBluetoothSecurity

戻り値

説明	型
-	-

6.23 getPortName メソッド

ポートをオープンしたときの portName を取得します。

宣言

```
public String getPortName();
```

引数

名称	説明	型
-	-	-

戻り値

説明	型
getPort メソッドで指定した portName	String

6.24 getPortSettings メソッド

ポートをオープンしたときの portSettings を取得します。

宣言

```
public String getPortSettings();
```

引数

名称	説明	型
-	-	-

戻り値

説明	型
getPort メソッドで指定した portSettings	String

6.25 getTimeoutMillis メソッド

StarBluetoothManager コンストラクタで指定した ioTimeoutMillis を取得します。

宣言

```
public int getTimeoutMillis();
```

引数

名称	説明	型
-	-	-

戻り値

説明	型
StarBluetoothManager コンストラクタで指定した ioTimeoutMillis	int

6.26 getDeviceType メソッド

StarBluetoothManager コンストラクタで指定した deviceType を取得します。

宣言

```
public StarBluetoothManager.StarDeviceType getDeviceType();
```

引数

名称	説明	型
-	-	-

戻り値

説明	型
StarBluetoothManager コンストラクタで指定した deviceType	StarBluetoothManager.StarDeviceType

6.27 isOpened メソッド

ポートの状態を取得します。

宣言

```
public boolean isOpened();
```

引数

名称	説明	型
-	-	-

戻り値

説明	型
ポートの状態 true - ポートがオープンされている false - ポートがクローズされている	boolean

6.28 StarDeviceType コンスタント

プリンタタイプ定数。

宣言

```
public enum StarDeviceType {
    StarDeviceTypeDesktopPrinter,
    StarDeviceTypePortablePrinter
}
```

定数

名称	説明
StarDeviceTypeDesktopPrinter	デスクトッププリンタ
StarDeviceTypePortablePrinter	モバイルプリンタ

各モデルに対して適用すべき StarDeviceType コンスタント定数は以下のとおりです。

Model	StarDeviceType
TSP100	StarDeviceTypeDesktopPrinter
FVP10	
TSP650II	
TSP700II	
TSP800II	
BSC10	
SP700	
mC-Print2	StarDeviceTypePortablePrinter
mC-Print3	
mPOP	
SM-S210i	
SM-S220i	
SM-S230i	
SM-T300i	
SM-T400i	
SM-S210i StarPRNT	
SM-S220i StarPRNT	
SM-S230i StarPRNT	
SM-T300i StarPRNT	
SM-T400i StarPRNT	
SM-L200	
SM-L300	

6.29 StarBluetoothSecurity コンスタント

Bluetooth セキュリティタイプ定数。

宣言

```
public enum StarBluetoothSecurity {
    PINCODE,
    SSP,
    DISABLE
}
```

定数

名称	説明
PINCODE	PIN コード入力によるセキュリティ
SSP	インターフェイスカードのボタン押下によるセキュリティ
DISABLE	セキュリティなし

6.30 StarBluetoothSettingCapability コンスタント

Bluetooth 設定可否定数。

宣言

```
public enum StarBluetoothSecurity {
    SUPPORT,
    NOSUPPORT
}
```

定数

名称	説明
SUPPORT	現在接続中のプリンタではその項目・機能が設定できることを示す。
NOSUPPORT	現在接続中のプリンタではその項目・機能が設定できないことを示す。

7 StarloExt クラス (starloextension.jar)

メソッド

名称	説明
createCommandBuilder	コマンド生成オブジェクトを生成します。
createDisplayCommandBuilder	カスタマーディスプレイコマンド生成オブジェクトを生成します。
createBcrConnectParser	バーコードリーダー接続確認コマンド応答解析オブジェクトを生成します。
createDisplayConnectParser	カスタマーディスプレイ接続確認コマンド応答解析オブジェクトを生成します。

コンスタント

名称	説明
Emulation	エミュレーション指定定数。
CharacterCode	文字コード指定定数。
BcrModel	バーコードリーダーモデル指定定数。
DisplayModel	カスタマーディスプレイモデル指定定数。

7.1 createCommandBuilder メソッド

コマンド生成オブジェクトを生成します。

宣言

```
public static ICommandBuilder createCommandBuilder(Emulation emulation);
```

引数

名称	説明	型
emulation	エミュレーション指定 <ul style="list-style-type: none"> StarPRNT ... StarPRNT エミュレーション StarPRNTL ... StarPRNTL エミュレーション StarLine ... STAR Line Mode エミュレーション StarGraphic ... STAR Graphic Mode エミュレーション EscPos ... ESC/POS エミュレーション EscPosMobile ... ESC/POS Mobile エミュレーション StarDotImpact ... STAR ドットインパクトエミュレーション 	com.starmicronics.starioextension.StarloExt.Emulation

戻り値

説明	型
ICommandBuilder オブジェクト	com.starmicronics.starioextension.ICommandBuilder

各モデルに対して適用すべき Emulation コンスタント定数は [Emulation コンスタント](#) を参照ください。

Example

```
public static byte[] createCommandsImage(Emulation emulation, Bitmap bitmap) {
    ICommandBuilder builder = StarloExt.createCommandBuilder(emulation);

    builder.beginDocument();

    builder.appendBitmap(bitmap, false);

    builder.appendCutPaper(CutPaperAction.PartialCutWithFeed);

    builder.endDocument();

    return builder.getCommands();
}
```

PrinterFunctions.java を参照ください。

7.2 createDisplayCommandBuilder メソッド

カスタマーディスプレイコマンド生成オブジェクトを生成します。

宣言

```
public static IDisplayCommandBuilder createDisplayCommandBuilder(DisplayModel model);
```

引数

名称	説明	型
model	カスタマーディスプレイモデル指定定数 ● SCD222	com.starmicronics.starioextension. StarloExt.DisplayModel

戻り値

説明	型
IDisplayCommandBuilder オブジェクト	com.starmicronics.starioextension. IDisplayCommandBuilder

Example

```
public static byte[] createTextPattern(int number) {
    IDisplayCommandBuilder builder = StarloExt.createDisplayCommandBuilder(DisplayModel.SCD222);

    builder.appendClearScreen();
    builder.appendCursorMode(CursorMode.Off);
    builder.appendHomePosition();

    byte[] pattern;

    ...

    builder.append(pattern);

    return builder.getPassThroughCommands();
}
```

DisplayFunctions.java を参照ください。

7.3 createBcrConnectParser メソッド

バーコードリーダー接続確認コマンド応答解析オブジェクトを生成します。

宣言

```
public static IPeripheralConnectParser createBcrConnectParser(BcrModel model);
```

引数

名称	説明	型
model	バーコードリーダーモデル指定定数 ● POP1	com.starmicronics.starioextension. StarIoExt.BcrModel

戻り値

説明	型
IPeripheralConnectParser オブジェクト	com.starmicronics.starioextension. IPeripheralConnectParser

7.4 createDisplayConnectParser メソッド

カスタマーディスプレイ接続確認コマンド応答解析オブジェクトを生成します。

宣言

```
public static IPeripheralConnectParser createDisplayConnectParser(DisplayModel model);
```

引数

名称	説明	型
model	カスタマーディスプレイモデル指定定数 ● SCD222	com.starmicronics.starloextension.StarloExt.DisplayModel

戻り値

説明	型
IPeripheralConnectParser オブジェクト	com.starmicronics.starloextension.IPeripheralConnectParser

Example

```
public void onItemClickListener(final AdapterView<?> parent, View view, final int position, long id) {
    super.onItemClickListener(parent, view, position, id);

    CommonAlertDialogFragment dialog = null;

    if (position == 1) {
        mProgressDialog.show();

        PrinterSetting setting = new PrinterSetting(getActivity());

        final IPeripheralConnectParser parser = StarloExt.createDisplayConnectParser(DisplayModel.SCD222);

        Communication.parseDoNotCheckCondition(DisplayFragment.class, parser, setting.getPortName(),
        setting.getPortSettings(), 10000, getActivity(), new Communication.SendCallback() {
            @Override
            public void onStatus(boolean result, Communication.Result communicateResult) {
                ...

                String msg;
                String title;

                if (result) {
                    title = "Check Status";

                    if (parser.isConnected()) {
                        msg = "Display Connect";
                    }
                    else {
                        msg = "Display Disconnect";
                    }
                }
                else {
                    title = "Communication Result";
                    msg = "Printer Impossible";
                }
            }
        });
    }
}
```

DisplayFragment.java/DisplayExtFragment.java を参照ください。

7.5 Emulation コンスタント

エミュレーション指定定数。

宣言

```
public enum Emulation {
    None,
    StarPRNT,
    StarPRNTL,
    StarLine,
    StarGraphic,
    EscPos,
    EscPosMobile,
    StarDotImpact
}
```

定数

名称	説明
StarPRNT	StarPRNT エミュレーション
StarPRNTL	StarPRNTL エミュレーション
StarLine	STAR Line Mode エミュレーション
StarGraphic	STAR Graphic Mode エミュレーション
EscPos	ESC/POS エミュレーション
EscPosMobile	ESC/POS Mobile エミュレーション
StarDotImpact	STAR ドットインパクトエミュレーション

1. StarPRNTL エミュレーションについて

ラスタイメージ印字コマンドを生成する [appendBitmap](#) 系メソッドでコマンドサイズを削減した StarPRNT エミュレーションです。ラスタイメージコマンド送信に掛かる時間が短縮されます。StarPRNT エミュレーションをサポートするモデルのうち、一部モデルのみ選択可能です。使用可能なモデルは[対応プリンタ](#)を参照ください。

各モデルに対して適用すべき Emulation コンスタント定数は以下のとおりです。

Model	Emulation
mC-Print2	StarPRNT
mC-Print3	StarPRNT
mPOP	StarPRNT
FVP10	StarLine
TSP100	StarGraphic
TSP650II	StarLine
TSP700II	StarLine
TSP800II	StarLine
SM-S210i	EscPosMobile
SM-S220i	EscPosMobile
SM-S230i	EscPosMobile
SM-T300i/T300	EscPosMobile
SM-T400i	EscPosMobile
BSC10	EscPos
SM-S210i StarPRNT	StarPRNT
SM-S220i StarPRNT	StarPRNT
SM-S230i StarPRNT	StarPRNT
SM-T300i/T300 StarPRNT	StarPRNT
SM-T400i StarPRNT	StarPRNT
SM-L200	StarPRNT
	StarPRNTL
SM-L300	StarPRNT
	StarPRNTL
SP700	StarDotImpact

7.6 CharacterCode コンスタント

文字コード指定定数。

宣言

```
public enum CharacterCode {
    None,
    Standard,
    Japanese,
    SimplifiedChinese,
    TraditionalChinese
}
```

定数

名称	説明
Standard	スタンダード
Japanese	日本語
SimplifiedChinese	簡体字中国語
TraditionalChinese	繁体字中国語

7.7 BcrModel コンスタント

バーコードリーダーモデル指定定数。

宣言

```
public enum BcrModel {
    None,
    POP1,
}
```

定数

名称	説明
POP1	バーコードリーダー BCR-POP1

7.8 DisplayModel コンスタント

カスタマーディスプレイモデル指定定数。

宣言

```
public enum CharacterCode {
    None,
    SCD222,
}
```

定数

名称	説明
SCD222	カスタマーディスプレイ SCD222U

8 ICommandBuilder インターフェイス (starioextension.jar)

メソッド

名称	説明
beginDocument	ドキュメント開始コマンドを生成し、コマンドバッファに追加します。
endDocument	ドキュメント終了コマンドを生成し、コマンドバッファに追加します。
appendInitialization	初期化コマンドを生成し、コマンドバッファに追加します。
append	データ（テキストやコマンド）をコマンドバッファに追加します。
appendRaw	生のデータ（テキストやコマンド）をコマンドバッファに追加します。
appendFontStyle	フォントスタイル指定コマンドを生成し、コマンドバッファに追加します。
appendCodePage	コードページ指定コマンドを生成し、コマンドバッファに追加します。
appendInternational	国際文字指定コマンドを生成し、コマンドバッファに追加します。
appendLineFeed	行単位の用紙送りコマンドを生成し、コマンドバッファに追加します。
appendUnitFeed	1 ドットライン単位の用紙送りコマンドを生成し、コマンドバッファに追加します。
appendCharacterSpace	文字間スペース設定コマンドを生成し、コマンドバッファに追加します。
appendLineSpace	行間スペース設定コマンドを生成し、コマンドバッファに追加します。
appendTopMargin	トップマージン設定コマンドを生成し、コマンドバッファに追加します。
appendEmphasis	強調指定コマンドを生成し、コマンドバッファに追加します。
appendInvert	白黒反転指定コマンドを生成し、コマンドバッファに追加します。
appendMultiple	拡大指定コマンドを生成し、コマンドバッファに追加します。
appendMultipleHeight	
appendMultipleWidth	
appendUnderLine	アンダーライン指定コマンドを生成し、コマンドバッファに追加します。
appendLogo	ロゴ印刷コマンドを生成し、コマンドバッファに追加します。
appendAbsolutePosition	絶対位置指定コマンドを生成し、コマンドバッファに追加します。
appendAlignment	位置揃え指定コマンドを生成し、コマンドバッファに追加します。
appendHorizontalTabPosition	水平タブ設定／クリアコマンドを生成し、コマンドバッファに追加します。
appendCutPaper	用紙カットコマンドを生成し、コマンドバッファに追加します。
appendPeripheral	キャッシュドローア駆動コマンドを生成し、コマンドバッファに追加します。
appendSound	サウンド/ブザー駆動コマンドを生成し、コマンドバッファに追加します。
appendBarcode	バーコード印刷コマンドを生成し、コマンドバッファに追加します。
appendBarcodeWithAbsolutePosition	絶対位置指定バーコード印刷コマンドを生成し、コマンドバッファに追加します。
appendBarcodeWithAlignment	位置揃えバーコード印刷コマンドを生成し、コマンドバッファに追加します。
appendPdf417	PDF417 印刷コマンドを生成し、コマンドバッファに追加します。
appendPdf417WithAbsolutePosition	絶対位置指定 PDF417 印刷コマンドを生成し、コマンドバッファに追加します。
appendPdf417WithAlignment	位置揃え PDF417 印刷コマンドを生成し、コマンドバッファに追加します。
appendQrCode	QR コード印刷コマンドを生成し、コマンドバッファに追加します。
appendQrCodeWithAbsolutePosition	絶対位置指定 QR コード印刷コマンドを生成し、コマンドバッファに追加します。

名称	説明
appendQrCodeWithAlignment	位置揃え QR コード印刷コマンドを生成し、コマンドバッファに追加します。
appendBitmap	ビットマップ印刷コマンドを生成し、コマンドバッファに追加します。
appendBitmapWithAbsolutePosition	絶対位置指定ビットマップ印刷コマンドを生成し、コマンドバッファに追加します。
appendBitmapWithAlignment	位置揃えビットマップ印刷コマンドを生成し、コマンドバッファに追加します。
appendBlackMark	ブラックマークコマンドを生成し、コマンドバッファに追加します。
beginPageMode	ページモード開始コマンドを生成し、コマンドバッファに追加します。
endPageMode	ページモード終了コマンドを生成し、コマンドバッファに追加します。
appendPageModeVerticalAbsolutePosition	ページモードの縦方向絶対位置指定コマンドを生成し、コマンドバッファに追加します。
appendPageModeRotation	ページモード印刷方向コマンドを生成し、コマンドバッファに追加します。
appendPrintableArea	印字領域設定コマンドを生成し、コマンドバッファに追加します。
getCommands	コマンドバッファを取得します。

コンスタント

名称	説明
InitializationType	初期化指定定数。
FontStyleType	フォントスタイル指定定数。
CodePageType	コードページ指定定数。
InternationalType	国際文字指定定数。
LogoSize	ロゴサイズ指定定数。
AlignmentPosition	位置揃え指定定数。
CutPaperAction	用紙カット指定定数。
PeripheralChannel	キャッシュドロアーチャネル指定定数。
SoundChannel	サウンド/ブザーチャネル指定定数。
BarcodeSymbology	バーコードシンボル指定定数。
BarcodeWidth	バーコード幅指定定数。
Pdf417Level	PDF417 ECC（セキュリティレベル）指定定数。
QrCodeModel	QR コードモデル指定定数
QrCodeLevel	QR コード誤り訂正レベル指定定数。
BitmapConverterRotation	ビットマップ回転指定定数。
BlackMarkType	ブラックマーク指定定数。
PrintableAreaType	印字領域タイプ指定定数。

8.1 モデル : ICommandBuilder interface メソッド

各モデルにおいて機能するメソッドは以下のとおりです。

Function	Method	mC-Print2	mC-Print3	mPOP	FVP10	TSP100	TSP650II	TSP700II	TSP800II	SM-S210i	SM-S220i	SM-S230i	SM-T300i/T300	SM-T400i	BSC10	SM-S210i StarPRNT	SM-S220i StarPRNT	SM-S230i StarPRNT	SM-T300i/T300 StarPRNT	SM-T400i StarPRNT	SM-L200	SM-L300	SP700
Document control	beginDocument	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	endDocument	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Initialization	appendInitialization	✓	✓	✓	✓	-	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Data (Text and Command)	append	✓	✓	✓	✓	-	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Raw data (Text and Command)	appendRaw	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Font style	appendFontStyle	✓	✓	✓	✓	-	✓	✓	✓	-	-	-	-	-	✓	✓	✓	✓	✓	✓	✓	✓	✓
Code page	appendCodePage	✓	✓	✓	✓	-	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
International	appendInternational	✓	✓	✓	✓	-	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Line feed	appendLineFeed	✓	✓	✓	✓	-	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Unit feed	appendUnitFeed	✓	✓	✓	✓	*1	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Character space	appendCharacterSpace	✓	✓	✓	✓	-	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Line space	appendLineSpace	✓	✓	✓	✓	-	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Top margin	appendTopMargin	-	*7	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Emphasis	appendEmphasis	✓	✓	✓	✓	-	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Invert	appendInvert	✓	✓	✓	✓	-	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Multiple	appendMultiple	✓	✓	✓	✓	-	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	appendMultipleHeight	✓	✓	✓	✓	-	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	appendMultipleWidth	✓	✓	✓	✓	-	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Under line	appendUnderLine	✓	✓	✓	✓	-	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Logo	appendLogo	✓	✓	✓	✓	-	✓	✓	✓	*2	*2	*2	*2	*2	✓	✓	✓	✓	✓	✓	✓	✓	✓

Function	Method	mC-Print2	mC-Print3	mPOP	FVP10	TSP100	TSP650II	TSP700II	TSP800II	SM-S210i	SM-S220i	SM-S230i	SM-T300i/T300	SM-T400i	BSC10	SM-S210i StarPRNT	SM-S220i StarPRNT	SM-S230i StarPRNT	SM-T300i/T300 StarPRNT	SM-T400i StarPRNT	SM-L200	SM-L300	SP700
Absolute position	appendAbsolutePosition	✓	✓	✓	✓	-	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Alignment	appendAlignment	✓	✓	✓	✓	-	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Horizontal Tab position	appendHorizontalTabPosition	✓	✓	✓	✓	-	✓	✓	✓	-	-	-	-	-	✓	✓	✓	✓	✓	✓	✓	✓	✓
Cut paper	appendCutPaper	✓	✓	✓	✓	✓	✓	✓	✓	*3	*3	*3	*3	*3	✓	*3	*3	*3	*3	*3	*3	*3	✓
Peripheral	appendPeripheral	✓	✓	✓	✓	✓	✓	✓	✓	-	-	-	-	-	✓	-	-	-	-	-	-	-	✓
Sound	appendSound	✓	✓	✓	✓	✓	✓	✓	✓	-	-	-	-	-	✓	-	-	-	-	-	-	-	✓
Barcode	appendBarcode	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	-
	appendBarcodeWithAbsolutePosition	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	-
	appendBarcodeWithAlignment	✓	✓	✓	✓	*4	✓	✓	✓	*4	*4	*4	*4	*4	✓	✓	✓	✓	✓	✓	✓	✓	-
PDF417	appendPdf417	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	-
	appendPdf417WithAbsolutePosition	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	-
	appendPdf417WithAlignment	✓	✓	✓	✓	*4	✓	✓	✓	*4	*4	*4	*4	*4	✓	✓	✓	✓	✓	✓	✓	✓	-
QR code	appendQrCode	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	-
	appendQrCodeWithAbsolutePosition	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	-
	appendQrCodeWithAlignment	✓	✓	✓	✓	*4	✓	✓	✓	*4	*4	*4	*4	*4	✓	✓	✓	✓	✓	✓	✓	✓	-
Bitmap	appendBitmap	✓	✓	✓	✓	✓	✓	✓	✓	*5	*5	*5	*5	*5	✓	✓	✓	✓	✓	✓	✓	✓	✓
	appendBitmapWithAbsolutePosition	✓	✓	✓	✓	✓	✓	✓	✓	*5	*5	*5	*5	*5	✓	✓	✓	✓	✓	✓	✓	✓	✓
	appendBitmapWithAlignment	✓	✓	✓	✓	*4	✓	✓	✓	*4*5	*4*5	*4*5	*4*5	*4*5	✓	✓	✓	✓	✓	✓	✓	✓	✓
Black mark	appendBlackMark	-	-	-	✓	-	-	✓	✓	*6	*6	*6	*6	*6	-	*6	*6	*6	*6	*6	*6	*6	✓
Page mode	beginPageMode	✓	✓	✓	✓	-	✓	-	-	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	-
	endPageMode	✓	✓	✓	✓	-	✓	-	-	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	-
	appendPageModeVerticalAbsolutePosition	✓	✓	✓	✓	-	✓	-	-	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	-
	appendPageModeRotation	✓	✓	✓	✓	-	✓	-	-	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	-
Printable area	appendPrintableArea	✓	✓	✓	✓	✓	✓	✓	✓	-	-	-	-	-	✓	-	-	-	-	-	-	✓	✓

-: 無視

*1: データ（テキストやコマンド）の追加は行わず、1 ドットライン単位用の紙送りコマンドの追加のみ行う

*2: 常にノーマルサイズ

*3: 用紙カットを行わない

*4: 常に左揃え

*5: 縦方向の最大印刷可能領域は 30cm (2400dot)まで

*6: プリンタ側で、BM 設定を有効にする必要有り。

設定方法はユーザマニュアル参照のこと。

*7: トップマージン設定が 10mm 以下の場合は、印字用紙長さ（カット長）が 50mm を超えないようにすること。

（カットした用紙が残っている場合は、用紙詰まりが発生する可能性がある。カットした用紙を取り除いた状態では印字用紙長さの制限はなし）

8.2 beginDocument メソッド

ドキュメント開始コマンドを生成し、コマンドバッファに追加します。

宣言

```
void beginDocument();
```

引数

名称	説明	型
-	-	-

戻り値

説明	型
-	-

Example

```
public static byte[] createData(Emulation emulation) {
    byte[] data = "Hello World.\n".getBytes();

    ICommandBuilder builder = StarIoExt.createCommandBuilder(emulation);

    builder.beginDocument();

    builder.append(data);

    builder.appendCutPaper(CutPaperAction.PartialCutWithFeed);

    builder.endDocument();

    return builder.getCommands();
}
```

ApiFunctions.java を参照ください。

8.3 endDocument メソッド

ドキュメント終了コマンドを生成し、コマンドバッファに追加します。

宣言

```
void endDocument();
```

引数

名称	説明	型
-	-	-

戻り値

説明	型
-	-

Example

```
public static byte[] createData(Emulation emulation) {
    byte[] data = "Hello World.\n".getBytes();

    ICommandBuilder builder = StarIoExt.createCommandBuilder(emulation);

    builder.beginDocument();

    builder.append(data);

    builder.appendCutPaper(CutPaperAction.PartialCutWithFeed);

    builder.endDocument();

    return builder.getCommands();
}
```

ApiFunctions.java を参照ください。

8.4 appendInitialization メソッド

初期化コマンドを生成し、コマンドバッファに追加します。

宣言

```
void appendInitialization(InitializationType type);
```

引数

名称	説明	型
type	初期化タイプ <ul style="list-style-type: none"> Command ... コマンド初期化 Reset ... プリンタリセット ResetWithPrint ... プリンタリセット (自己印字実行) 	com.starmicronics.starioextension.ICommandBuilder.InitializationType

戻り値

説明	型
-	-

Example

```
public static byte[] createData(Emulation emulation) {
    byte[] data = "Hello World.\n".getBytes();

    ICommandBuilder builder = StarIoExt.createCommandBuilder(emulation);

    builder.beginDocument();

    builder.appendUnderLine(true);
    builder.appendMultiple(2, 2);
    builder.append(data);

    builder.appendInitialization(InitializationType.Command);

    builder.append(data);

    builder.appendCutPaper(CutPaperAction.PartialCutWithFeed);

    builder.endDocument();

    return builder.getCommands();
}
```

ApiFunctions.java を参照ください。

8.5 append メソッド

データ（テキストやコマンド）をコマンドバッファに追加します。

宣言

```
void append(byte data);
```

```
void append(byte[] data);
```

引数

名称	説明	型
data	データ（テキストやコマンド）	byte
		byte[]

戻り値

説明	型
-	-

Example

```
public static byte[] createData(Emulation emulation) {
    byte[] data = "Hello World.".getBytes();

    byte[] data = new byte[] {0x48, 0x65, 0x6c, 0x6c, 0x6f, 0x20, 0x57, 0x6f, 0x72, 0x6c, 0x64, 0x2e};

    ICommandBuilder builder = StarIoExt.createCommandBuilder(emulation);

    builder.beginDocument();

    builder.append(data);
    builder.append((byte) 0x0a);

    builder.appendCutPaper(CutPaperAction.PartialCutWithFeed);

    builder.endDocument();

    return builder.getCommands();
}
```

ApiFunctions.java /各コマンド仕様書を参照ください。

8.6 appendRaw メソッド

生のデータ（テキストやコマンド）をコマンドバッファに追加します。

宣言

```
void appendRaw(byte data);
```

```
void appendRaw(byte[] data);
```

引数

名称	説明	型
data	生のデータ（テキストやコマンド）	byte
		byte[]

戻り値

説明	型
-	-

Example

```
public static byte[] createData(Context context, Emulation emulation, Bitmap bitmap, RequestCallback callback) {
    String urlString = AllReceipts.uploadBitmap(context, bitmap, callback);

    ICommandBuilder builder = StarIoExt.createCommandBuilder(emulation);

    builder.beginDocument();

    builder.appendBitmap(bitmap, false);

    byte[] data = AllReceipts.generateAllReceipts(context, urlString, emulation, true, true);

    builder.appendRaw(data);

    builder.appendCutPaper(CutPaperAction.PartialCutWithFeed);

    builder.endDocument();

    return builder.getCommands();
}
```

AllReceiptsFunctions.java /各コマンド仕様書を参照ください。

8.7 appendFontStyle メソッド

フォントスタイル指定コマンドを生成し、コマンドバッファに追加します。

宣言

```
void appendFontStyle(FontStyleType type);
```

引数

名称	説明	型
type	フォントスタイル <ul style="list-style-type: none"> A ... フォント A (12 x 24 ドット) / 7 x 9 フォント (ハーフドット) B ... フォント B (9 x 24 ドット) / 5 x 9 フォント (2P-1) 	com.starmicronics.starioextension.ICommandBuilder.FontStyleType

戻り値

説明	型
-	-

Example

```
public static byte[] createData(Emulation emulation) {
    byte[] data = "Hello World.\n".getBytes();

    ICommandBuilder builder = StarIoExt.createCommandBuilder(emulation);

    builder.beginDocument();

    builder.appendFontStyle(FontStyleType.A);
    builder.append(data);
    builder.appendFontStyle(FontStyleType.B);
    builder.append(data);

    builder.appendCutPaper(CutPaperAction.PartialCutWithFeed);

    builder.endDocument();

    return builder.getCommands();
}
```

ApiFunctions.java /各コマンド仕様書を参照ください。

8.8 appendCodePage メソッド

コードページ指定コマンドを生成し、コマンドバッファに追加します。

宣言

```
void appendCodePage(CodePageType type);
```

引数

名称	説明	型
type	<p>コードページ</p> <ul style="list-style-type: none"> CP437 ... CodePage437 (USA, Std. Europe). CP737 ... Codepage 737 (Greek). CP772 ... Codepage 772 (Lithuanian). CP774 ... Codepage 774 (Lithuanian). CP851 ... Codepage 851 (Greek). CP852 ... Codepage 852 (Latin-2). CP855 ... Codepage 855 (Cyrillic Bulgarian). CP857 ... Codepage 857 (Turkey). CP858 ... Codepage 858 (Multilingual). CP860 ... Codepage 860 (Portuguese). CP861 ... Codepage 861 (Icelandic). CP862 ... Codepage 862 (Israel (Hebrew)). CP863 ... Codepage 863 (Canadian French). CP864 ... Codepage 864 (Arabic). CP865 ... Codepage 865 (Nordic). CP866 ... Codepage 866 (Cyrillic Russian). CP869 ... Codepage 869 (Greek). CP874 ... Codepage 874 (Thai). CP928 ... Codepage 928 (Greek). CP932 ... Katakana. CP998 ... Normal. CP999 ... Codepage 1252 (Windows Latin-1). CP1001 ... Codepage 1001 (Arabic). CP1250 ... Codepage 1250 (Windows Latin-2). CP1251 ... Codepage 1251 (Windows Cyrillic). CP1252 ... Codepage 1252 (Windows Latin-1). CP2001 ... Codepage 2001 (Lithuanian-KBL). CP3001 ... Codepage 3001 (Estonian-1). CP3002 ... Codepage 3002 (Estonian-2). CP3011 ... Codepage 3011 (Latvian-1). CP3012 ... Codepage 3012 (Latvian-2). CP3021 ... Codepage 3021 (Bulgarian). CP3041 ... Codepage 3041 (Maltese). CP3840 ... Codepage 3840 (IBM-Russian). 	com.starmicronics.starextension.ICommandBuilder.CodePageType

名称	説明	型
type	<ul style="list-style-type: none"> CP3841 ... Codepage 3841 (Gost). CP3843 ... Codepage 3843 (Polish). CP3844 ... Codepage 3844 (CS2). CP3845 ... Codepage 3845 (Hungarian). CP3846 ... Codepage 3846 (Turkish). CP3847 ... Codepage 3847 (Brazil-ABNT). CP3848 ... Codepage 3848 (Brazil-ABICOMP). UTF8 ... UTF-8. Blank ... User Setting Blank Code Page. 	com.starmicronics.starioextension.ICommandBuilder.CodePageType

戻り値

説明	型
-	-

Example

```

public static byte[] createData(Emulation emulation) {
    byte[] data8 = new byte[] {0x80, 0x81, 0x82, 0x83, 0x84, 0x85, 0x86, 0x87, 0x88, 0x89, 0x8a, 0x8b, 0x8c, 0x8d, 0x8e, 0x8f, 0x0a};
    byte[] data9 = new byte[] {0x90, 0x91, 0x92, 0x93, 0x94, 0x95, 0x96, 0x97, 0x98, 0x99, 0x9a, 0x9b, 0x9c, 0x9d, 0x9e, 0x9f, 0x0a};
    byte[] dataA = new byte[] {0xa0, 0xa1, 0xa2, 0xa3, 0xa4, 0xa5, 0xa6, 0xa7, 0xa8, 0xa9, 0xaa, 0xab, 0xac, 0xad, 0xae, 0xaf, 0x0a};
    byte[] dataB = new byte[] {0xb0, 0xb1, 0xb2, 0xb3, 0xb4, 0xb5, 0xb6, 0xb7, 0xb8, 0xb9, 0xba, 0xbb, 0xbc, 0xbd, 0xbe, 0xbf, 0x0a};
    byte[] dataC = new byte[] {0xc0, 0xc1, 0xc2, 0xc3, 0xc4, 0xc5, 0xc6, 0xc7, 0xc8, 0xc9, 0xca, 0xcb, 0xcc, 0xcd, 0xce, 0xcf, 0x0a};
    byte[] dataD = new byte[] {0xd0, 0xd1, 0xd2, 0xd3, 0xd4, 0xd5, 0xd6, 0xd7, 0xd8, 0xd9, 0xda, 0xdb, 0xdc, 0xdd, 0xde, 0xdf, 0x0a};
    byte[] dataE = new byte[] {0xe0, 0xe1, 0xe2, 0xe3, 0xe4, 0xe5, 0xe6, 0xe7, 0xe8, 0xe9, 0xea, 0xeb, 0xec, 0xed, 0xee, 0xef, 0x0a};
    byte[] dataF = new byte[] {0xf0, 0xf1, 0xf2, 0xf3, 0xf4, 0xf5, 0xf6, 0xf7, 0xf8, 0xf9, 0xfa, 0xfb, 0xfc, 0xfd, 0xfe, 0xff, 0x0a};

    ICommandBuilder builder = StarIoExt.createCommandBuilder(emulation);

    builder.beginDocument();

    builder.appendCodePage(CodePageType.CP932);

    builder.append(data8);
    builder.append(data9);
    builder.append(dataA);
    builder.append(dataB);
    builder.append(dataC);
    builder.append(dataD);
    builder.append(dataE);
    builder.append(dataF);

    builder.appendCutPaper(CutPaperAction.PartialCutWithFeed);

    builder.endDocument();

    return builder.getCommands();
}

```

ApiFunctions.java /各コマンド仕様書を参照ください。

8.9 appendInternational メソッド

国際文字指定コマンドを生成し、コマンドバッファに追加します。

宣言

```
void appendInternational(InternationalType type);
```

引数

名称	説明	型
type	国際文字 <ul style="list-style-type: none"> • USA ... USA • France ... フランス • Germany ... ドイツ • UK ... イギリス • Denmark ... デンマーク • Sweden ... スウェーデン • Italy ... イタリア • Spain ... スペイン • Japan ... 日本 • Norway ... ノルウェー • Denmark2 ... デンマーク II • Spain2 ... スペイン II • LatinAmerica ... ラテンアメリカ • Korea ... 韓国 • Ireland ... アイルランド • Legal ... Legal 	com.starmicronics.starioextension.ICommandBuilder.InternationalType

戻り値

説明	型
-	-

Example

```
public static byte[] createData(Emulation emulation) {
    byte[] data = new byte[] {0x23, 0x24, 0x40, 0x58, 0x5a, 0x5b, 0x5c, 0x5d, 0x5e, 0x60, 0x7b, 0x7c, 0x7d, 0x7e, 0x0a};

    ICommandBuilder builder = StarIoExt.createCommandBuilder(emulation);

    builder.beginDocument();

    builder.appendInternational(InternationalType.USA);
    builder.append(data);

    builder.appendInternational(InternationalType.Japan);
    builder.append(data);

    builder.appendCutPaper(CutPaperAction.PartialCutWithFeed);

    builder.endDocument();

    return builder.getCommands();
}
```

ApiFunctions.java /各コマンド仕様書を参照ください。

8.10 appendLineFeed メソッド

行単位用の紙送りコマンドを生成し、コマンドバッファに追加します。

宣言

```
void appendLineFeed();
```

```
void appendLineFeed(byte[] data);
```

```
void appendLineFeed(int line);
```

```
void appendLineFeed(byte[] data, int line);
```

引数

名称	説明	型
line	紙送り量（行数単位）	int
data	データ（テキストやコマンド）	byte[]

戻り値

説明	型
-	-

Example

```
public static byte[] createData(Emulation emulation) {
    byte[] data = "Hello World.".getBytes();

    ICommandBuilder builder = StarIoExt.createCommandBuilder(emulation);

    builder.beginDocument();

    builder.append(data);
    builder.appendLineFeed();
    builder.appendLineFeed(data);

    builder.append(data);
    builder.appendLineFeed(2);
    builder.appendLineFeed(data, 2);

    builder.appendCutPaper(CutPaperAction.PartialCutWithFeed);

    builder.endDocument();

    return builder.getCommands();
}
```

ApiFunctions.java を参照ください。

8.11 appendUnitFeed メソッド

1 ドットライン単位用の紙送りコマンドを生成し、コマンドバッファに追加します。

宣言

```
void appendUnitFeed(int unit);
```

```
void appendUnitFeed(byte[] data, int unit);
```

引数

名称	説明	型
unit	紙送り量（ドット数単位）	int
data	データ（テキストやコマンド）	byte[]

戻り値

説明	型
-	-

Example

```
public static byte[] createData(Emulation emulation) {
    byte[] data = "Hello World.".getBytes();

    ICommandBuilder builder = StarIoExt.createCommandBuilder(emulation);

    builder.beginDocument();

    builder.append(data);
    builder.appendUnitFeed(64);

    builder.appendUnitFeed(data, 64);

    builder.appendCutPaper(CutPaperAction.PartialCutWithFeed);

    builder.endDocument();

    return builder.getCommands();
}
```

ApiFunctions.java を参照ください。

8.12 appendCharacterSpace メソッド

文字間スペース設定コマンドを生成し、コマンドバッファに追加します。

宣言

```
void appendCharacterSpace(int space);
```

引数

名称	説明	型
space	文字間スペース（ドット数単位）	int

戻り値

説明	型
-	-

1. 注意事項

日本語、中国語簡体字、繁体字、韓国語(DBCS)では文字間スペースが設定値の 2 倍になります。

Example

```
public static byte[] createData(Emulation emulation) {
    byte[] data = "Hello World.\n".getBytes();

    ICommandBuilder builder = StarIoExt.createCommandBuilder(emulation);

    builder.beginDocument();

    builder.appendCharacterSpace(0);
    builder.append(data);
    builder.appendCharacterSpace(4);
    builder.append(data);

    builder.appendCutPaper(CutPaperAction.PartialCutWithFeed);

    builder.endDocument();

    return builder.getCommands();
}
```

ApiFunctions.java を参照ください。

8.13 appendLineSpace メソッド

行間スペース設定コマンドを生成し、コマンドバッファに追加します。

宣言

```
void appendLineSpace(int lineSpace);
```

引数

名称	説明	型
lineSpace	行間スペース（ドット数単位）	int

戻り値

説明	型
-	-

Example

```
public static byte[] createData(Emulation emulation) {
    byte[] data = "Hello World.\n".getBytes();

    ICommandBuilder builder = StarIoExt.createCommandBuilder(emulation);

    builder.beginDocument();

    builder.appendLineSpace(32);
    builder.append(data);
    builder.append(data);
    builder.append(data);
    builder.appendLineSpace(24);
    builder.append(data);
    builder.append(data);
    builder.append(data);

    builder.appendCutPaper(CutPaperAction.PartialCutWithFeed);

    builder.endDocument();

    return builder.getCommands();
}
```

ApiFunctions.java /各コマンド仕様書を参照ください。

8.14 appendTopMargin メソッド

トップマージン設定コマンドを生成し、コマンドバッファに追加します。

宣言

```
void appendTopMargin(int margin);
```

引数

名称	説明	型
margin	トップマージン (1 ミリメートル単位)	int

戻り値

説明	型
-	-

Example

```
public static byte[] createTopMarginData(Emulation emulation) {
    byte[] data = "Hello World.\n".getBytes();

    ICommandBuilder builder = StarIoExt.createCommandBuilder(emulation);

    builder.beginDocument();

    builder.appendTopMargin(2);
    builder.append("**Top margin:2mm*\n".getBytes());
    builder.append(data);
    builder.append(data);
    builder.append(data);

    builder.appendCutPaper(CutPaperAction.PartialCutWithFeed);

    builder.appendTopMargin(6);
    builder.append("**Top margin:6mm*\n".getBytes());
    builder.append(data);
    builder.append(data);
    builder.append(data);

    builder.appendCutPaper(CutPaperAction.PartialCutWithFeed);

    builder.appendTopMargin(11); // Default
    builder.append("**Top margin:11mm*\n".getBytes());
    builder.append(data);
    builder.append(data);
    builder.append(data);

    builder.appendCutPaper(CutPaperAction.PartialCutWithFeed);

    builder.endDocument();

    return builder.getCommands();
}
```

ApiFunctions.java /各コマンド仕様書を参照ください。

8.15 appendEmphasis メソッド

強調指定コマンドを生成し、コマンドバッファに追加します。

宣言

```
void appendEmphasis(boolean emphasis);
```

```
void appendEmphasis(byte[] data);
```

引数

名称	説明	型
emphasis	強調 <ul style="list-style-type: none"> • true ... 強調設定 • false ... 強調解除 	boolean
data	データ（テキストやコマンド）	byte[]

戻り値

説明	型
-	-

Example

```
public static byte[] createData(Emulation emulation) {
    byte[] data = "Hello World.\n".getBytes();
    byte[] dataHalf0 = "Hello ".getBytes();
    byte[] dataHalf1 = "World.\n".getBytes();

    ICommandBuilder builder = StarIoExt.createCommandBuilder(emulation);

    builder.beginDocument();

    builder.append(data);
    builder.appendEmphasis(true);
    builder.append(data);
    builder.append(data);
    builder.appendEmphasis(false);
    builder.append(data);

    builder.appendEmphasis(data);
    builder.append(data);

    builder.appendEmphasis(dataHalf0);
    builder.append(dataHalf1);

    builder.appendCutPaper(CutPaperAction.PartialCutWithFeed);

    builder.endDocument();

    return builder.getCommands();
}
```

ApiFunctions.java を参照ください。

8.16 appendInvert メソッド

白黒反転指定コマンドを生成し、コマンドバッファに追加します。

宣言

```
void appendInvert(boolean invert);
```

```
void appendInvert(byte[] data);
```

引数

名称	説明	型
invert	白黒反転 <ul style="list-style-type: none"> • true ... 白黒反転設定 • false ... 白黒反転解除 	boolean
data	データ（テキストやコマンド）	byte[]

戻り値

説明	型
-	-

Example

```
public static byte[] createData(Emulation emulation) {
    byte[] data = "Hello World.\n".getBytes();
    byte[] dataHalf0 = "Hello ".getBytes();
    byte[] dataHalf1 = "World.\n".getBytes();

    ICommandBuilder builder = StarIoExt.createCommandBuilder(emulation);

    builder.beginDocument();

    builder.append(data);
    builder.appendInvert(true);
    builder.append(data);
    builder.append(data);
    builder.appendInvert(false);
    builder.append(data);

    builder.appendInvert(data);
    builder.append(data);

    builder.appendInvert(dataHalf0);
    builder.append(dataHalf1);

    builder.appendCutPaper(CutPaperAction.PartialCutWithFeed);

    builder.endDocument();

    return builder.getCommands();
}
```

ApiFunctions.java を参照ください。

8.17 appendMultiple メソッド

拡大指定コマンドを生成し、コマンドバッファに追加します。

宣言

```
void appendMultiple(int width, int height);
```

```
void appendMultiple(byte[] data, int width, int height);
```

```
void appendMultipleHeight(int height);
```

```
void appendMultipleHeight(byte[] data, int height);
```

```
void appendMultipleWidth(int width);
```

```
void appendMultipleWidth(byte[] data, int width);
```

引数

名称	説明	型
width	横倍率	int
height	縦倍率	int
data	データ（テキストやコマンド）	byte[]

戻り値

説明	型
-	-

Example

```
public static byte[] createData(Emulation emulation) {
    byte[] data = "Hello World.\n".getBytes();
    byte[] dataHalf0 = "Hello ".getBytes();
    byte[] dataHalf1 = "World.\n".getBytes();

    ICommandBuilder builder = StarIoExt.createCommandBuilder(emulation);

    builder.beginDocument();

    builder.append(data);

    builder.appendMultiple(2, 2);
    builder.append(data);
    builder.append(data);
    builder.appendMultiple(1, 1);
    builder.append(data);

    builder.appendMultiple(data, 2, 2);
    builder.append(data);

    builder.appendMultiple(dataHalf0, 2, 2);
    builder.append(dataHalf1);

    builder.append(dataHalf0);
    builder.appendMultiple(dataHalf1 2, 2);

    builder.appendMultipleHeight(2);
    builder.append(data);
    builder.append(data);
    builder.appendMultipleHeight(1);
    builder.append(data);

    builder.appendMultipleHeight(dataHalf0, 2);
    builder.append(dataHalf1);

    builder.append(dataHalf0);
    builder.appendMultipleHeight(dataHalf1, 2);

    builder.appendMultipleWidth(2);
    builder.append(data);
    builder.append(data);
    builder.appendMultipleWidth(1);
    builder.append(data);

    builder.appendMultipleWidth(dataHalf0, 2);
    builder.append(dataHalf1);

    builder.append(dataHalf0);
    builder.appendMultipleWidth(dataHalf1, 2);

    builder.appendCutPaper(CutPaperAction.PartialCutWithFeed);

    builder.endDocument();

    return builder.getCommands();
}
```

ApiFunctions.java /各コマンド仕様書を参照ください。

8.18 appendUnderLine メソッド

アンダーライン指定コマンドを生成し、コマンドバッファに追加します。

宣言

```
void appendUnderLine(boolean underLine);
```

```
void appendUnderLine(byte[] data);
```

引数

名称	説明	型
underLine	アンダーライン <ul style="list-style-type: none"> true ... アンダーライン設定 false ... アンダーライン解除 	boolean
data	データ（テキストやコマンド）	byte[]

戻り値

説明	型
-	-

Example

```
public static byte[] createData(Emulation emulation) {
    byte[] data = "Hello World.\n".getBytes();
    byte[] dataHalf0 = "Hello ".getBytes();
    byte[] dataHalf1 = "World.\n".getBytes();

    ICommandBuilder builder = StarIoExt.createCommandBuilder(emulation);

    builder.beginDocument();

    builder.append(data);

    builder.appendUnderLine(true);
    builder.append(data);
    builder.append(data);
    builder.appendUnderLine(false);
    builder.append(data);

    builder.appendUnderLine(data);
    builder.append(data);

    builder.appendUnderLine(dataHalf0);
    builder.append(dataHalf1);

    builder.appendCutPaper(CutPaperAction.PartialCutWithFeed);

    builder.endDocument();

    return builder.getCommands();
}
```

ApiFunctions.java を参照ください。

8.19 appendLogo メソッド

ロゴ印刷コマンドを生成し、コマンドバッファに追加します。

宣言

```
void appendLogo(LogoSize size, int number);
```

引数

名称	説明	型
number	ロゴサイズ <ul style="list-style-type: none"> Normal ... ノーマルモード DoubleWidth ... 横 2 倍モード DoubleHeight ... 縦 2 倍モード DoubleWidthDoubleHeight ... 縦 2 倍、横 2 倍モード 	com.starmicronics.starioextension.ICommandBuilder.LogoSize
number	ロゴ番号	int

戻り値

説明	型
-	-

Example

```
public static byte[] createData(Emulation emulation) {
    ICommandBuilder builder = StarIoExt.createCommandBuilder(emulation);

    builder.beginDocument();

    builder.appendLogo(LogoSize.Normal, 1);

    builder.appendLogo(LogoSize.DoubleWidth, 1);

    builder.appendLogo(LogoSize.DoubleHeight, 1);

    builder.appendLogo(LogoSize.DoubleWidthDoubleHeight, 1);

    builder.appendCutPaper(CutPaperAction.PartialCutWithFeed);

    builder.endDocument();

    return builder.getCommands();
}
```

ApiFunctions.java /各コマンド仕様書を参照ください。

8.20 appendAbsolutePosition メソッド

絶対位置指定コマンドを生成し、コマンドバッファに追加します。

宣言

```
void appendAbsolutePosition(int position);
```

```
void appendAbsolutePosition(byte[] data, int position);
```

引数

名称	説明	型
position	絶対位置（ドット数単位）	int
data	データ（テキストやコマンド）	byte[]

戻り値

説明	型
-	-

Example

```
public static byte[] createData(Emulation emulation) {
    byte[] data = "Hello World.\n".getBytes();

    ICommandBuilder builder = StarIoExt.createCommandBuilder(emulation);

    builder.beginDocument();

    builder.append(data);

    builder.appendAbsolutePosition(40);
    builder.append(data);
    builder.append(data);

    builder.appendAbsolutePosition(data, 40);
    builder.append(data);

    builder.appendCutPaper(CutPaperAction.PartialCutWithFeed);

    builder.endDocument();

    return builder.getCommands();
}
```

ApiFunctions.java を参照ください。

8.21 appendAlignment メソッド

位置揃え指定コマンドを生成し、コマンドバッファに追加します。

宣言

```
void appendAlignment(AlignmentPosition position);
```

```
void appendAlignment(byte[] data, AlignmentPosition position);
```

引数

名称	説明	型
position	位置揃え <ul style="list-style-type: none"> Left ... 左揃え Center ... 中心揃え Right ... 右揃え 	com.starmicronics.starioextension.ICommandBuilder.AlignmentPosition
data	データ（テキストやコマンド）	byte[]

戻り値

説明	型
-	-

1. 注意事項

appendBitmap メソッドと組み合わせて使用することはできません。ビットマップの位置揃えをする場合は、appendBitmapWithAlignment メソッドをご利用ください。

Example

```
public static byte[] createData(Emulation emulation) {
    byte[] data = "Hello World.\n".getBytes();

    ICommandBuilder builder = StarIoExt.createCommandBuilder(emulation);

    builder.beginDocument();

    builder.append(data);

    builder.appendAlignment(AlignmentPosition.Right);
    builder.append(data);
    builder.append(data);
    builder.appendAlignment(AlignmentPosition.Left);
    builder.append(data);
    builder.append(data);

    builder.appendAlignment(data, AlignmentPosition.Right);
    builder.append(data);

    builder.appendCutPaper(CutPaperAction.PartialCutWithFeed);

    builder.endDocument();

    return builder.getCommands();
}
```

ApiFunctions.java /各コマンド仕様書を参照ください。

8.22 appendHorizontalTabPosition メソッド

水平タブ設定／クリアコマンドを生成し、コマンドバッファに追加します。

宣言

```
void appendHorizontalTabPosition(int[] positions);
```

引数

名称	説明	型
positions	水平タブ位置（ANK 文字ピッチ単位）の配列 null もしくは空の配列を指定した場合、現在設定されている水平タブ位置を全てクリアします。	int[]

戻り値

説明	型
-	-

Example

```
public static byte[] createHorizontalTabData(Emulation emulation) {
    byte[] data1 = "QTY\tITEM\tTOTAL\n".getBytes();
    byte[] data2 = "1\tApple\t1.50\n".getBytes();
    byte[] data3 = "2\tOrange\t2.00\n".getBytes();
    byte[] data4 = "5\tBanana\t3.00\n".getBytes();

    ICommandBuilder builder = StarIoExt.createCommandBuilder(emulation);

    builder.beginDocument();

    builder.appendHorizontalTabPosition(new int[] {5, 27});

    builder.append("**Tab Position:5, 27*\n".getBytes());
    builder.append(data1);
    builder.append(data2);
    builder.append(data3);
    builder.append(data4);

    builder.appendCutPaper(CutPaperAction.PartialCutWithFeed);

    builder.endDocument();

    return builder.getCommands();
}
```

ApiFunctions.java を参照ください。

8.23 appendCutPaper メソッド

用紙カットコマンドを生成し、コマンドバッファに追加します。

宣言

```
void appendCutPaper(CutPaperAction action);
```

引数

名称	説明	型
action	用紙カット <ul style="list-style-type: none"> FullCut ... フルカット PartialCut ... パーシャルカット FullCutWithFeed ... 用紙送り付きフルカット PartialCutWithFeed ... 用紙送り付きパーシャルカット 	com.starmicronics.starioextension.ICommandBuilder.CutPaperAction

戻り値

説明	型
-	-

Example

```
public static byte[] createData(Emulation emulation) {
    byte[] data = "Hello World.\n".getBytes();

    ICommandBuilder builder = StarIoExt.createCommandBuilder(emulation);

    builder.beginDocument();

    builder.append(data);
    builder.append(data);
    builder.append(data);
    builder.append(data);
    builder.append(data);

    builder.appendCutPaper(CutPaperAction.PartialCutWithFeed);

    builder.endDocument();

    return builder.getCommands();
}
```

ApiFunctions.java を参照ください。

8.24 appendPeripheral メソッド

キャッシュドロアー駆動コマンドを生成し、コマンドバッファに追加します。

宣言

```
void appendPeripheral(PeripheralChannel channel);
```

```
void appendPeripheral(PeripheralChannel channel, int time);
```

引数

名称	説明	型
channel	キャッシュドロアーチャンネル <ul style="list-style-type: none"> No1 ... チャンネル 1 No2 ... チャンネル 2 	com.starmicronics.starioextension.ICommandBuilder.PeripheralChannel
time	駆動時間（1 ミリ秒単位） ※チャンネル 1 のみ有効	int

戻り値

説明	型
-	-

Example

```
public static byte[] createData(Emulation emulation) {
    ICommandBuilder builder = StarIoExt.createCommandBuilder(emulation);

    builder.beginDocument();

    builder.appendPeripheral(PeripheralChannel.No1);
    builder.appendPeripheral(PeripheralChannel.No2);
    builder.appendPeripheral(PeripheralChannel.No1, 2000);
    builder.appendPeripheral(PeripheralChannel.No2, 2000);

    builder.endDocument();

    return builder.getCommands();
}
```

ApiFunctions.java /各コマンド仕様書を参照ください。

8.25 appendSound メソッド

サウンド/ブザー駆動コマンドを生成し、コマンドバッファに追加します。

宣言

```
void appendSound(SoundChannel channel);
```

```
void appendSound(SoundChannel channel, int repeat);
```

```
void appendSound(SoundChannel channel, int repeat, int driveTime, int delayTime);
```

引数

名称	説明	型
channel	サウンド/ブザーチャンネル <ul style="list-style-type: none"> No1 ... チャンネル 1 No2 ... チャンネル 2 	com.starmicronics.starioextension.ICommandBuilder.SoundChannel
repeat	繰り返し回数	int
driveTime	駆動時間 (1 ミリ秒単位)	int
delayTime	ディレイ時間 (1 ミリ秒単位)	int

戻り値

説明	型
-	-

Example

```
public static byte[] createData(Emulation emulation) {
    ICommandBuilder builder = StarIoExt.createCommandBuilder(emulation);

    builder.beginDocument();

    builder.appendSound(SoundChannel.No1);
    builder.appendSound(SoundChannel.No2);
    builder.appendSound(SoundChannel.No1, 3);
    builder.appendSound(SoundChannel.No2, 3);
    builder.appendSound(SoundChannel.No1, 1, 1000, 1000);
    builder.appendSound(SoundChannel.No2, 1, 1000, 1000);

    builder.endDocument();

    return builder.getCommands();
}
```

ApiFunctions.java /各コマンド仕様書を参照ください。

8.26 appendBarcode メソッド

バーコード印刷コマンドを生成し、コマンドバッファに追加します。

宣言

```
void appendBarcode(byte[] data, BarcodeSymbology symbology, BarcodeWidth width, int height,
boolean hri);
```

引数

名称	説明	型
data	バーコードデータ	byte[]
symbology	バーコードシンボル <ul style="list-style-type: none"> • UPCE ... UPC-E. • UPCA ... UPC-A. • JAN8 ... JAN/EAN8. • JAN13 ... JAN/EAN13. • Code39 ... Code39. • ITF ... ITF. • Code128 ... Code128. • Code93 ... Code93. • Codabar ... Codabar. • NW7 ... NW7. 	com.starmicronics.starioextension.ICommandBuilder.BarcodeSymbology
width	バーコード幅 <ul style="list-style-type: none"> • Mode1 ... モード 1 • Mode2 ... モード 2 • Mode3 ... モード 3 • Mode4 ... モード 4 • Mode5 ... モード 5 • Mode6 ... モード 6 • Mode7 ... モード 7 • Mode8 ... モード 8 • Mode9 ... モード 9 	com.starmicronics.starioextension.ICommandBuilder.BarcodeWidth
height	バーコードの高さ（ドット数単位）	int
hri	バー下文字列 <ul style="list-style-type: none"> • true ... 印刷あり • false ... 印刷なし 	boolean

戻り値

説明	型
-	-

Example

```
public static byte[] createData(Emulation emulation) {  
    byte[] dataCode128 = "{B0123456789".getBytes();  
  
    ICommandBuilder builder = StarIoExt.createCommandBuilder(emulation);  
  
    builder.beginDocument();  
  
    builder.appendBarcode(dataCode128, BarcodeSymbology.Code128, BarcodeWidth.Mode1, 40, true);  
    builder.appendUnitFeed(32);  
  
    builder.appendCutPaper(CutPaperAction.PartialCutWithFeed);  
  
    builder.endDocument();  
  
    return builder.getCommands();  
}
```

ApiFunctions.java /各コマンド仕様書を参照ください。

8.27 appendBarcodeWithAbsolutePosition メソッド

絶対位置指定バーコード印刷コマンドを生成し、コマンドバッファに追加します。

宣言

```
void appendBarcodeWithAbsolutePosition(byte[] data, BarcodeSymbology symbology,
BarcodeWidth width, int height, boolean hri, int position);
```

引数

名称	説明	型
position	絶対位置（ドット数単位）	int
data	バーコードデータ	byte[]
symbology	バーコードシンボル <ul style="list-style-type: none"> • UPCE ... UPC-E. • UPCA ... UPC-A. • JAN8 ... JAN/EAN8. • JAN13 ... JAN/EAN13. • Code39 ... Code39. • ITF ... ITF. • Code128 ... Code128. • Code93 ... Code93. • Codabar ... Codabar. • NW7 ... NW7. 	com.starmicronics.starioextension.ICommandBuilder.BarcodeSymbology
width	バーコード幅 <ul style="list-style-type: none"> • Mode1 ... モード 1 • Mode2 ... モード 2 • Mode3 ... モード 3 • Mode4 ... モード 4 • Mode5 ... モード 5 • Mode6 ... モード 6 • Mode7 ... モード 7 • Mode8 ... モード 8 • Mode9 ... モード 9 	com.starmicronics.starioextension.ICommandBuilder.BarcodeWidth
height	バーコードの高さ（ドット数単位）	int
hri	バー下文字列 <ul style="list-style-type: none"> • true ... 印刷あり • false ... 印刷なし 	boolean

戻り値

説明	型
-	-

Example

```
public static byte[] createData(Emulation emulation) {  
    byte[] dataCode128 = "B0123456789".getBytes();  
  
    ICommandBuilder builder = StarIoExt.createCommandBuilder(emulation);  
  
    builder.beginDocument();  
  
    builder.appendBarcodeWithAbsolutePosition(dataCode128, BarcodeSymbology.Code128,  
BarcodeWidth.Mode1, 40, true, 40);  
    builder.appendUnitFeed(32);  
  
    builder.appendCutPaper(CutPaperAction.PartialCutWithFeed);  
  
    builder.endDocument();  
  
    return builder.getCommands();  
}
```

ApiFunctions.java /各コマンド仕様書を参照ください。

8.28 appendBarcodeWithAlignment メソッド

位置揃えバーコード印刷コマンドを生成し、コマンドバッファに追加します。

宣言

```
void appendBarcodeWithAlignment(byte[] data, BarcodeSymbology symbology, BarcodeWidth width, int height, boolean hri, AlignmentPosition position);
```

引数

名称	説明	型
position	位置揃え指定 <ul style="list-style-type: none"> Left ... 左揃え Center ... 中心揃え Right ... 右揃え 	com.starmicronics.starioextension.ICommandBuilder.AlignmentPosition
data	バーコードデータ	byte[]
symbology	バーコードシンボル <ul style="list-style-type: none"> UPCE ... UPC-E. UPCA ... UPC-A. JAN8 ... JAN/EAN8. JAN13 ... JAN/EAN13. Code39 ... Code39. ITF ... ITF. Code128 ... Code128. Code93 ... Code93. Codabar ... Codabar. NW7 ... NW7. 	com.starmicronics.starioextension.ICommandBuilder.BarcodeSymbology
width	バーコード幅 <ul style="list-style-type: none"> Mode1 ... モード 1 Mode2 ... モード 2 Mode3 ... モード 3 Mode4 ... モード 4 Mode5 ... モード 5 Mode6 ... モード 6 Mode7 ... モード 7 Mode8 ... モード 8 Mode9 ... モード 9 	com.starmicronics.starioextension.ICommandBuilder.BarcodeWidth
height	バーコードの高さ（ドット数単位）	int
hri	バー下文字列 <ul style="list-style-type: none"> true ... 印刷あり false ... 印刷なし 	boolean

戻り値

説明	型
-	-

Example

```
public static byte[] createData(Emulation emulation) {
    byte[] dataCode128 = "{B0123456789".getBytes();

    ICommandBuilder builder = StarIoExt.createCommandBuilder(emulation);

    builder.beginDocument();

    builder.appendBarcodeWithAlignment(dataCode128, BarcodeSymbology.Code128, BarcodeWidth.Mode1, 40,
true, AlignmentPosition.Center);
    builder.appendUnitFeed(32);

    builder.appendBarcodeWithAlignment(dataCode128, BarcodeSymbology.Code128, BarcodeWidth.Mode1, 40,
true, AlignmentPosition.Right);
    builder.appendUnitFeed(32);

    builder.appendCutPaper(CutPaperAction.PartialCutWithFeed);

    builder.endDocument();

    return builder.getCommands();
}
```

ApiFunctions.java /各コマンド仕様書を参照ください。

8.29 appendPdf417 メソッド

PDF417 印刷コマンドを生成し、コマンドバッファに追加します。

宣言

```
void appendPdf417(byte[] data, int line, int column, Pdf417Level level, int module, int aspect);
```

引数

名称	説明	型
data	PDF417 データ	byte[]
line	ライン数	int
column	カラム数	int
level	PDF417 ECC (セキュリティレベル) <ul style="list-style-type: none"> ECC0 ... セキュリティレベル 0 ECC1 ... セキュリティレベル 1 ECC2 ... セキュリティレベル 2 ECC3 ... セキュリティレベル 3 ECC4 ... セキュリティレベル 4 ECC5 ... セキュリティレベル 5 ECC6 ... セキュリティレベル 6 ECC7 ... セキュリティレベル 7 ECC8 ... セキュリティレベル 8 	com.starmicronics.star.ioextension.ICommandBuilder.Pdf417Level
module	モジュール X 方向サイズ	int
aspect	モジュールアスペクト比	int

戻り値

説明	型
-	-

Example

```
public static byte[] createData(Emulation emulation) {
    byte[] data = "Hello World.".getBytes();

    ICommandBuilder builder = StarIoExt.createCommandBuilder(emulation);

    builder.beginDocument();

    builder.appendPdf417(data, 0, 1, Pdf417Level.ECC0, 2, 2);
    builder.appendUnitFeed(32);

    builder.appendCutPaper(CutPaperAction.PartialCutWithFeed);

    builder.endDocument();

    return builder.getCommands();
}
```

ApiFunctions.java /各コマンド仕様書を参照ください。

8.30 appendPdf417WithAbsolutePosition メソッド

絶対位置指定 PDF417 印刷コマンドを生成し、コマンドバッファに追加します。

宣言

```
void appendPdf417WithAbsolutePosition(byte[] data, int line, int column, Pdf417Level level, int module, int aspect, int position);
```

引数

名称	説明	型
position	絶対位置（ドット数単位）	int
data	PDF417 データ	byte[]
line	ライン数	int
column	カラム数	int
level	PDF417 ECC（セキュリティレベル） <ul style="list-style-type: none"> • ECC0 ... セキュリティレベル 0 • ECC1 ... セキュリティレベル 1 • ECC2 ... セキュリティレベル 2 • ECC3 ... セキュリティレベル 3 • ECC4 ... セキュリティレベル 4 • ECC5 ... セキュリティレベル 5 • ECC6 ... セキュリティレベル 6 • ECC7 ... セキュリティレベル 7 • ECC8 ... セキュリティレベル 8 	com.starmicronics.starioextension.ICommandBuilder.Pdf417Level
module	モジュール X 方向サイズ	int
aspect	モジュールアスペクト比	int

戻り値

説明	型
-	-

Example

```
public static byte[] createData(Emulation emulation) {
    byte[] data = "Hello World.".getBytes();

    ICommandBuilder builder = StarIoExt.createCommandBuilder(emulation);

    builder.beginDocument();

    builder.appendPdf417WithAbsolutePosition(data, 0, 1, Pdf417Level.ECC0, 2, 2, 40);
    builder.appendUnitFeed(32);

    builder.appendCutPaper(CutPaperAction.PartialCutWithFeed);

    builder.endDocument();

    return builder.getCommands();
}
```

ApiFunctions.java /各コマンド仕様書を参照ください。

8.31 appendPdf417WithAlignment メソッド

位置揃え PDF417 印刷コマンドを生成し、コマンドバッファに追加します。

宣言

```
void appendPdf417WithAlignment(byte[] data, int line, int column, Pdf417Level level, int module, int aspect, AlignmentPosition position);
```

引数

名称	説明	型
position	位置揃え指定 <ul style="list-style-type: none"> Left ... 左揃え Center ... 中心揃え Right ... 右揃え 	com.starmicronics.starioextension.ICommandBuilder.AlignmentPosition
data	PDF417 データ	byte[]
line	ライン数	int
column	カラム数	int
level	PDF417 ECC (セキュリティレベル) <ul style="list-style-type: none"> ECC0 ... セキュリティレベル 0 ECC1 ... セキュリティレベル 1 ECC2 ... セキュリティレベル 2 ECC3 ... セキュリティレベル 3 ECC4 ... セキュリティレベル 4 ECC5 ... セキュリティレベル 5 ECC6 ... セキュリティレベル 6 ECC7 ... セキュリティレベル 7 ECC8 ... セキュリティレベル 8 	com.starmicronics.starioextension.ICommandBuilder.Pdf417Level
module	モジュール X 方向サイズ	int
aspect	モジュールアスペクト比	int

戻り値

説明	型
-	-

Example

```
public static byte[] createData(Emulation emulation) {  
    byte[] data = "Hello World.".getBytes();  
  
    ICommandBuilder builder = StarIoExt.createCommandBuilder(emulation);  
  
    builder.beginDocument();  
  
    builder.appendPdf417WithAlignment(data, 0, 1, Pdf417Level.ECC0, 2, 2, AlignmentPosition.Center);  
    builder.appendUnitFeed(32);  
  
    builder.appendPdf417WithAlignment(data, 0, 1, Pdf417Level.ECC0, 2, 2, AlignmentPosition.Right);  
    builder.appendUnitFeed(32);  
  
    builder.appendCutPaper(CutPaperAction.PartialCutWithFeed);  
  
    builder.endDocument();  
  
    return builder.getCommands();  
}
```

ApiFunctions.java /各コマンド仕様書を参照ください。

8.32 appendQrCode メソッド

QR コード印刷コマンドを生成し、コマンドバッファに追加します。

宣言

```
void appendQrCode(byte[] data, QrCodeModel model, QrCodeLevel level, int cell);
```

引数

名称	説明	型
data	QR コードデータ	byte[]
model	QR コードモデル <ul style="list-style-type: none"> No1 ... モデル 1 No2 ... モデル 2 	com.starmicronics.starioextension.ICommandBuilder.QrCodeModel
level	QR コード誤り訂正レベル <ul style="list-style-type: none"> L ... 誤り訂正レベル L M ... 誤り訂正レベル M Q ... 誤り訂正レベル Q H ... 誤り訂正レベル H 	com.starmicronics.starioextension.ICommandBuilder.QrCodeLevel
cell	QR コードセルサイズ	int

戻り値

説明	型
-	-

Example

```
public static byte[] createData(Emulation emulation) {
    byte[] data = "Hello World.".getBytes();

    ICommandBuilder builder = StarIoExt.createCommandBuilder(emulation);

    builder.beginDocument();

    builder.appendQrCode(data, QrCodeModel.No2, QrCodeLevel.L, 4);
    builder.appendUnitFeed(32);

    builder.appendCutPaper(CutPaperAction.PartialCutWithFeed);

    builder.endDocument();

    return builder.getCommands();
}
```

ApiFunctions.java /各コマンド仕様書を参照ください。

8.33 appendQrCodeWithAbsolutePosition メソッド

絶対位置指定 QR コード印刷コマンドを生成し、コマンドバッファに追加します。

宣言

```
void appendQrCodeWithAbsolutePosition(byte[] data, QrCodeModel model, QrCodeLevel level, int cell, int position);
```

引数

名称	説明	型
position	絶対位置（ドット数単位）	int
data	QR コードデータ	byte[]
model	QR コードモデル <ul style="list-style-type: none"> No1 ... モデル 1 No2 ... モデル 2 	com.starmicronics.starioextension.ICommandBuilder.QrCodeModel
level	QR コード誤り訂正レベル <ul style="list-style-type: none"> L ... 誤り訂正レベル L M ... 誤り訂正レベル M Q ... 誤り訂正レベル Q H ... 誤り訂正レベル H 	com.starmicronics.starioextension.ICommandBuilder.QrCodeLevel
cell	QR コードセルサイズ	int

戻り値

説明	型
-	-

Example

```
public static byte[] createData(Emulation emulation) {
    byte[] data = "Hello World.".getBytes();

    ICommandBuilder builder = StarIoExt.createCommandBuilder(emulation);

    builder.beginDocument();

    builder.appendQrCodeWithAbsolutePosition(data, QrCodeModel.No2, QrCodeLevel.L, 4, 40);
    builder.appendUnitFeed(32);

    builder.appendCutPaper(CutPaperAction.PartialCutWithFeed);

    builder.endDocument();

    return builder.getCommands();
}
```

ApiFunctions.java /各コマンド仕様書を参照ください。

8.34 appendQrCodeWithAlignment メソッド

位置揃え QR コード印刷コマンドを生成し、コマンドバッファに追加します。

宣言

```
void appendQrCodeWithAlignment(byte[] data, QrCodeModel model, QrCodeLevel level, int cell, AlignmentPosition position);
```

引数

名称	説明	型
position	位置揃え指定 <ul style="list-style-type: none"> Left ... 左揃え Center ... 中心揃え Right ... 右揃え 	com.starmicronics.starioextension.ICommandBuilder.AlignmentPosition
data	QR コードデータ	byte[]
model	QR コードモデル <ul style="list-style-type: none"> No1 ... モデル 1 No2 ... モデル 2 	com.starmicronics.starioextension.ICommandBuilder.QrCodeModel
level	QR コード誤り訂正レベル <ul style="list-style-type: none"> L ... 誤り訂正レベル L M ... 誤り訂正レベル M Q ... 誤り訂正レベル Q H ... 誤り訂正レベル H 	com.starmicronics.starioextension.ICommandBuilder.QrCodeLevel
cell	QR コードセルサイズ	int

戻り値

説明	型
-	-

Example

```
public static byte[] createData(Emulation emulation) {
    byte[] data = "Hello World.".getBytes();

    ICommandBuilder builder = StarIoExt.createCommandBuilder(emulation);

    builder.beginDocument();

    builder.appendQrCodeWithAlignment(data, QrCodeModel.No2, QrCodeLevel.L, 4, AlignmentPosition.Center);
    builder.appendUnitFeed(32);

    builder.appendQrCodeWithAlignment(data, QrCodeModel.No2, QrCodeLevel.L, 4, AlignmentPosition.Right);
    builder.appendUnitFeed(32);

    builder.appendCutPaper(CutPaperAction.PartialCutWithFeed);

    builder.endDocument();

    return builder.getCommands();
}
```

ApiFunctions.java /各コマンド仕様書を参照ください。

8.35 appendBitmap メソッド

ビットマップ印刷コマンドを生成し、コマンドバッファに追加します。

宣言

```
void appendBitmap(Bitmap bitmap, boolean diffusion, int width, boolean bothScale,
BitmapConverterRotation rotation);
```

```
void appendBitmap(Bitmap bitmap, boolean diffusion, BitmapConverterRotation rotation);
```

```
void appendBitmap(Bitmap bitmap, boolean diffusion, int width, boolean bothScale);
```

```
void appendBitmap(Bitmap bitmap, boolean diffusion);
```

引数

名称	説明	型
bitmap	元となるビットマップオブジェクト	android.graphics.Bitmap
diffusion	誤差拡散 <ul style="list-style-type: none"> • true ... 誤差拡散を行う • false ... 誤差拡散を行わない 	boolean
width	変換後のビットマップ幅（ドット数単位）	int
bothScale	width プロパティによる変換率に合わせた高さ変換 <ul style="list-style-type: none"> • true ... 高さ変換を行う • false ... 高さ変換を行わない 	boolean
rotation	変換後の回転 <ul style="list-style-type: none"> • Normal ... 回転なし • Right90 ... 右 90 度回転 • Left90 ... 左 90 度回転 • Rotate180 ... 180 度回転 	com.starmicronics.starioextension.ICommandBuilder.BitmapConverterRotation

戻り値

説明	型
-	-

Example

```
public static byte[] createData(Emulation emulation, int width, Context context) {
    Bitmap starLogoBitmap = BitmapFactory.decodeResource(context.getResources(), R.drawable.star_logo_image);

    ICommandBuilder builder = StarIoExt.createCommandBuilder(emulation);

    builder.beginDocument();

    builder.append("\n*Normal*\n".getBytes());
    builder.appendBitmap(starLogoBitmap, true);

    builder.append("\n*width:Full, bothScale:true*\n".getBytes());
    builder.appendBitmap(starLogoBitmap, true, width, true);
    builder.append("\n*width:Full, bothScale:false*\n".getBytes());
    builder.appendBitmap(starLogoBitmap, true, width, false);

    builder.append("\n*Rotate180*\n".getBytes());
    builder.appendBitmap(starLogoBitmap, true, BitmapConverterRotation.Rotate180);

    builder.appendCutPaper(CutPaperAction.PartialCutWithFeed);

    builder.endDocument();

    return builder.getCommands();
}
```

ApiFunctions.java を参照ください。

8.36 appendBitmapWithAbsolutePosition メソッド

絶対位置指定ビットマップ印刷コマンドを生成し、コマンドバッファに追加します。

宣言

```
void appendBitmapWithAbsolutePosition(Bitmap bitmap, boolean diffusion, int width, boolean bothScale, BitmapConverterRotation rotation, int position);
```

```
void appendBitmapWithAbsolutePosition(Bitmap bitmap, boolean diffusion, BitmapConverterRotation rotation, int position);
```

```
void appendBitmapWithAbsolutePosition(Bitmap bitmap, boolean diffusion, int width, boolean bothScale, int position);
```

```
void appendBitmapWithAbsolutePosition(Bitmap bitmap, boolean diffusion, int position);
```

引数

名称	説明	型
position	絶対位置（ドット数単位）	int
bitmap	元となるビットマップオブジェクト	android.graphics.Bitmap
diffusion	誤差拡散 <ul style="list-style-type: none"> true ... 誤差拡散を行う false ... 誤差拡散を行わない 	boolean
width	変換後のビットマップ幅（ドット数単位）	int
bothScale	width プロパティによる変換率に合わせた高さ変換 <ul style="list-style-type: none"> true ... 高さ変換を行う false ... 高さ変換を行わない 	boolean
rotation	変換後の回転 <ul style="list-style-type: none"> Normal ... 回転なし Right90 ... 右 90 度回転 Left90 ... 左 90 度回転 Rotate180 ... 180 度回転 	com.starmicronics.starioextension.ICommandBuilder.BitmapConverterRotation

戻り値

説明	型
-	-

Example

```
public static byte[] createData(Emulation emulation, int width, Context context) {
    Bitmap starLogoBitmap = BitmapFactory.decodeResource(context.getResources(), R.drawable.star_logo_image);

    ICommandBuilder builder = StarIoExt.createCommandBuilder(emulation);

    builder.beginDocument();

    builder.append("\n*Normal, AbsolutePosition:40*\n".getBytes());
    builder.appendBitmapWithAbsolutePosition(starLogoBitmap, true, 40);

    builder.append("\n*width:Full, bothScale:true, AbsolutePosition:40*\n".getBytes());
    builder.appendBitmap(starLogoBitmap, true, width, true, 40);
    builder.append("\n*width:Full, bothScale:false, AbsolutePosition:40*\n".getBytes());
    builder.appendBitmap(starLogoBitmap, true, width, false, 40);

    builder.append("\n*Rotate180, AbsolutePosition:40*\n".getBytes());
    builder.appendBitmapWithAbsolutePosition(starLogoBitmap, true, BitmapConverterRotation.Rotate180, 40);

    builder.appendCutPaper(CutPaperAction.PartialCutWithFeed);

    builder.endDocument();

    return builder.getCommands();
}
```

ApiFunctions.java を参照ください。

8.37 appendBitmapWithAlignment メソッド

位置揃えビットマップ印刷コマンドを生成し、コマンドバッファに追加します。

宣言

```
void appendBitmapWithAlignment(Bitmap bitmap, boolean diffusion, int width, boolean bothScale,
BitmapConverterRotation rotation, AlignmentPosition position);
```

```
void appendBitmapWithAlignment(Bitmap bitmap, boolean diffusion, BitmapConverterRotation
rotation, AlignmentPosition position);
```

```
void appendBitmapWithAlignment(Bitmap bitmap, boolean diffusion, int width, boolean bothScale,
AlignmentPosition position);
```

```
void appendBitmapWithAlignment(Bitmap bitmap, boolean diffusion, AlignmentPosition position);
```

引数

名称	説明	型
position	位置揃え指定 <ul style="list-style-type: none"> Left ... 左揃え Center ... 中心揃え Right ... 右揃え 	com.starmicronics.starioextension.ICommandBuilder.AlignmentPosition
bitmap	元となるビットマップオブジェクト	android.graphics.Bitmap
diffusion	誤差拡散 <ul style="list-style-type: none"> true ... 誤差拡散を行う false ... 誤差拡散を行わない 	boolean
width	変換後のビットマップ幅（ドット数単位）	int
bothScale	width プロパティによる変換率に合わせた高さ変換 <ul style="list-style-type: none"> true ... 高さ変換を行う false ... 高さ変換を行わない 	boolean
rotation	変換後の回転 <ul style="list-style-type: none"> Normal ... 回転なし Right90 ... 右 90 度回転 Left90 ... 左 90 度回転 Rotate180 ... 180 度回転 	com.starmicronics.starioextension.ICommandBuilder.BitmapConverterRotation

戻り値

説明	型
-	-

Example

```
public static byte[] createData(Emulation emulation, int width, Context context) {
    Bitmap starLogoBitmap = BitmapFactory.decodeResource(context.getResources(), R.drawable.star_logo_image);

    ICommandBuilder builder = StarIoExt.createCommandBuilder(emulation);

    builder.beginDocument();

    builder.append("\n*Normal, Alignment:Center*\n".getBytes());
    builder.appendBitmapWithAlignment(starLogoBitmap, true, AlignmentPosition.Center);

    builder.append("\n*width:Full, bothScale:true, Alignment:Center*\n".getBytes());
    builder.appendBitmapWithAlignment(starLogoBitmap, true, width, true, AlignmentPosition.Center);
    builder.append("\n*width:Full, bothScale:false, Alignment:Center*\n".getBytes());
    builder.appendBitmapWithAlignment(starLogoBitmap, true, width, false, AlignmentPosition.Center);

    builder.append("\n*Rotate180, Alignment:Center*\n".getBytes());
    builder.appendBitmapWithAlignment(starLogoBitmap, true, BitmapConverterRotation.Rotate180, AlignmentPosition.Center);

    builder.appendCutPaper(CutPaperAction.PartialCutWithFeed);

    builder.endDocument();

    return builder.getCommands();
}
```

ApiFunctions.java を参照ください。

8.38 appendBlackMark メソッド

ブラックマークコマンドを生成し、コマンドバッファに追加します。

宣言

```
void appendBlackMark(BlackMarkType type);
```

引数

名称	説明	型
type	ブラックマーク <ul style="list-style-type: none"> Invalid ... ブラックマーク無効 Valid ... ブラックマーク有効 ValidWithDetection ... ブラックマーク有効 + ブラックマーク検出動作 	com.starmicronics.starioextension.ICommandBuilder.BlackMarkType

戻り値

説明	型
-	-

Example

```
public static byte[] createData(Emulation emulation) {
    byte[] data = "Hello World.\n".getBytes();

    ICommandBuilder builder = StarIoExt.createCommandBuilder(emulation);

    builder.beginDocument();

    builder.appendBlackMark(BlackMarkType.Valid);

    builder.append(data);

    builder.appendCutPaper(CutPaperAction.PartialCutWithFeed);

    // builder.appendBlackMark(BlackMarkType.Invalid);

    builder.endDocument();

    return builder.getCommands();
}
```

ApiFunctions.java を参照ください。

8.39 beginPageMode メソッド

ページモード開始コマンドを生成し、コマンドバッファに追加します。

宣言

```
void beginPageMode(Rect rect, BitmapConverterRotation rotation);
```

引数

名称	説明	型
rect	位置とサイズ（ドット数単位）	android.graphics.Rect
rotation	印刷方向 <ul style="list-style-type: none"> Normal ... 回転なし Right90 ... 右 90 度回転 Left90 ... 左 90 度回転 Rotate180 ... 180 度回転 	com.starmicronics.starioextension.ICommandBuilder.BitmapConverterRotation

戻り値

説明	型
-	-

Example

```
public static byte[] createData(Emulation emulation, int width) {
    byte[] data = "Hello World.\n".getBytes();

    int left = 0;
    int top = 0;
    int height = 30 * 8;    // 30mm!!!

    ICommandBuilder builder = StarIoExt.createCommandBuilder(emulation);

    builder.beginDocument();

    builder.append("\n*Rotate180 Text*\n".getBytes());

    Rect rect = new Rect(left, top, left + width, top + height);

    builder.beginPageMode(rect, BitmapConverterRotation.Rotate180);

    builder.appendPageModeVerticalAbsolutePosition(height / 2);

    builder.appendAbsolutePosition(data, width / 2);

    builder.endPageMode();

    builder.appendCutPaper(CutPaperAction.PartialCutWithFeed);

    builder.endDocument();

    return builder.getCommands();
}
```

ApiFunctions.java を参照ください。

8.40 endPageMode メソッド

ページモード終了コマンドを生成し、コマンドバッファに追加します。

宣言

```
void endPageMode();
```

引数

名称	説明	型
-	-	-

戻り値

説明	型
-	-

Example

```
public static byte[] createData(Emulation emulation, int width) {
    byte[] data = "Hello World.\n".getBytes();

    int left = 0;
    int top = 0;
    int height = 30 * 8;    // 30mm!!!

    ICommandBuilder builder = StarIoExt.createCommandBuilder(emulation);

    builder.beginDocument();

    builder.append("\n*Rotate180 Text*\n".getBytes());

    Rect rect = new Rect(left, top, left + width, top + height);

    builder.beginPageMode(rect, BitmapConverterRotation.Rotate180);

    builder.appendPageModeVerticalAbsolutePosition(height / 2);

    builder.appendAbsolutePosition(data, width / 2);

    builder.endPageMode();

    builder.appendCutPaper(CutPaperAction.PartialCutWithFeed);

    builder.endDocument();

    return builder.getCommands();
}
```

ApiFunctions.java を参照ください。

8.41 appendPageModeVerticalAbsolutePosition メソッド

ページモードの縦方向絶対位置指定コマンドを生成し、コマンドバッファに追加します。

宣言

```
void appendPageModeVerticalAbsolutePosition(int position);
```

引数

名称	説明	型
position	縦方向の絶対位置（ドット数単位）	int

戻り値

説明	型
-	-

Example

```
public static byte[] createData(Emulation emulation, int width) {
    byte[] data = "Hello World.\n".getBytes();

    int left = 0;
    int top = 0;
    int height = 30 * 8;    // 30mm!!!

    ICommandBuilder builder = StarIoExt.createCommandBuilder(emulation);

    builder.beginDocument();

    builder.append("\n*Rotate180 Text*\n".getBytes());

    Rect rect = new Rect(left, top, left + width, top + height);

    builder.beginPageMode(rect, BitmapConverterRotation.Normal);

    builder.appendPageModeVerticalAbsolutePosition(height / 2);

    builder.appendAbsolutePosition(data, width / 2);

    builder.endPageMode();

    builder.appendCutPaper(CutPaperAction.PartialCutWithFeed);

    builder.endDocument();

    return builder.getCommands();
}
```

ApiFunctions.java を参照ください。

8.42 appendPageModeRotation メソッド

ページモード印刷方向コマンドを生成し、コマンドバッファに追加します。

宣言

```
void appendPageModeRotation(BitmapConverterRotation rotation);
```

引数

名称	説明	型
rotation	印刷方向 <ul style="list-style-type: none"> • Normal ... 回転なし • Right90 ... 右 90 度回転 • Left90 ... 左 90 度回転 • Rotate180 ... 180 度回転 	com.starmicronics.starioextension.ICommandBuilder.BitmapConverterRotation

戻り値

説明	型
-	-

Example

```
public static byte[] createData(Emulation emulation, int width) {
    byte[] data = "Hello World.\n".getBytes();

    int left = 0;
    int top = 0;
    int height = 30 * 8;    // 30mm!!!

    ICommandBuilder builder = StarIoExt.createCommandBuilder(emulation);

    builder.beginDocument();

    builder.append("\n*Mixed Text*\n".getBytes());

    Rect rect = new Rect(left, top, left + width, top + height);

    builder.beginPageMode(rect, BitmapConverterRotation.Normal);

    builder.appendPageModeVerticalAbsolutePosition(height / 2);

    builder.appendAbsolutePosition(data, width / 2);

    builder.appendPageModeRotation(BitmapConverterRotation.Rotate180);

    builder.appendPageModeVerticalAbsolutePosition(height / 2);

    builder.appendAbsolutePosition(data, width / 2);

    builder.endPageMode();

    builder.appendCutPaper(CutPaperAction.PartialCutWithFeed);

    builder.endDocument();

    return builder.getCommands();
}
```

ApiFunctions.java を参照ください。

8.43 appendPrintableArea メソッド

印字領域設定コマンドを生成し、コマンドバッファに追加します。

宣言

```
void appendPrintableArea(PrintableAreaType type);
```

引数

名称	説明	型
type	印字領域タイプ <ul style="list-style-type: none"> • Standard ... 標準タイプ • Type1 ... タイプ 1 • Type2 ... タイプ 2 • Type3 ... タイプ 3 • Type4 ... タイプ 4 	com.starmicronics.starioextension.ICommandBuilder.PrintableAreaType

戻り値

説明	型
-	-

指定する PrintableAreaType と設定される印字領域の対応は以下のとおりです。

PrintableAreaType	mC-Print2	mC-Print3	mPOP	FVP10	TSP100	TSP650II	TSP700II	TSP800II	SM-S210i	SM-S220i	SM-S230i	SM-T300i/T300	SM-T400i	BSC10	SM-S210i-StarPRNT	SM-S220i-StarPRNT	SM-S230i-StarPRNT	SM-T300i/T300-StarPRNT	SM-T400i-StarPRNT	SM-L200	SM-L300	SP700
Standard	48	72 *1	48	72 *1	72 *1	72 *1	72 *1	104 *2	-	-	-	-	-	64 *1*3	-	-	-	-	-	-	72 *1	63
Type1	54	-	54	52.5	51	-	52.5	-	-	-	-	-	-	-	-	-	-	-	-	-	-	48
Type2	-	48	-	-	-	-	80 *1	72	-	-	-	-	-	-	-	-	-	-	-	-	-	60
Type3	50.8	50.8	50.8	50.8	-	50.8	50.8	-	-	-	-	-	-	45 *3	-	-	-	-	-	-	50.8	45
Type4	-	-	-	52	-	-	52	-	-	-	-	-	-	-	-	-	-	-	-	-	-	57

※単位はミリメートル

-: 無視

*1: 2 インチ紙を使用時、3 インチ幅の印字領域設定を行わないでください

*2: 3 インチ紙を使用時、4 インチ幅の印字領域設定を行わないでください

*3: メモリスイッチの印字桁数を"最大桁数"に設定することで、Standard の場合 72mm、Type3 の場合 50.8mm に設定されます。

(メモリスイッチ 4 のビット 4 を 1 に設定)

Example

```
public static byte[] createPrintableAreaData(Emulation emulation, PrintableAreaType type, Context context) {
    ICommandBuilder builder = StarIoExt.createCommandBuilder(emulation);

    Bitmap printableAreaImage = BitmapFactory.decodeResource(context.getResources(),
R.drawable.printable_area_image);

    byte[] data1 = "123456789".getBytes();
    byte[] data2 = "0".getBytes();

    builder.beginDocument();

    builder.appendPrintableArea(type);

    switch (type) {
//    case Standard:
        default:
            builder.append("**Standard*\n".getBytes());
            break;
        case Type1:
            builder.append("**Type1*\n".getBytes());
            break;
        case Type2:
            builder.append("**Type2*\n".getBytes());
            break;
        case Type3:
            builder.append("**Type3*\n".getBytes());
            break;
        case Type4:
            builder.append("**Type4*\n".getBytes());
            break;
    }

    builder.appendBitmap(printableAreaImage, true);

    builder.append(data1);
    builder.appendInvert(data2);
    builder.append(data1);
    builder.appendInvert(data2);
    builder.append(data1);
    builder.appendInvert(data2);
    builder.append(data1);
    builder.appendInvert(data2);
    builder.append(data1);
    builder.appendInvert(data2);
    builder.append(data1);
    builder.appendInvert(data2);
    builder.append(data1);
    builder.appendInvert(data2);
    builder.append(data1);
    builder.appendInvert(data2);
    builder.append(data1);
    builder.appendInvert(data2);

    builder.appendCutPaper(CutPaperAction.PartialCutWithFeed);

    builder.endDocument();

    return builder.getCommands();
}
```

ApiFunctions.java を参照ください。

8.44 getCommands メソッド

コマンドバッファを取得します。

宣言

```
byte[] getCommands();
```

引数

名称	説明	型
-	-	-

戻り値

説明	型
コマンドバッファ	byte[]

Example

```
public static byte[] createData(Emulation emulation) {
    byte[] data = "Hello World.\n".getBytes();

    ICommandBuilder builder = StarIoExt.createCommandBuilder(emulation);

    builder.beginDocument();

    builder.append(data);

    builder.appendCutPaper(CutPaperAction.PartialCutWithFeed);

    builder.endDocument();

    return builder.getCommands();
}
```

ApiFunctions.java を参照ください。

8.45 InitializationType コンスタント

初期化指定定数。

宣言

```
enum InitializationType {
    Command
// Reset,
// ResetWithPrint
}
```

定数

名称	説明
Command	コマンド初期化
Reset	プリンタリセット
ResetWithPrint	プリンタリセット（自己印字実行）

ApiFunctions.java を参照ください。

8.46 FontStyleType コンスタント

フォントスタイル指定定数。

宣言

```
enum FontStyleType {
    A,
    B
}
```

定数

名称	説明
A	フォント A（12 x 24 ドット） / 7 x 9 フォント（ハーフドット）
B	フォント B（9 x 24 ドット） / 5 x 9 フォント（2P-1）

ApiFunctions.java / 各コマンド仕様書を参照ください。

8.47 CodePageType コンスタント

コードページ指定定数。

宣言

```
enum CodePageType {  
    CP437,  
    CP737,  
    CP772,  
    CP774,  
    CP851,  
    CP852,  
    CP855,  
    CP857,  
    CP858,  
    CP860,  
    CP861,  
    CP862,  
    CP863,  
    CP864,  
    CP865,  
    CP866,  
    CP869,  
    CP874,  
    CP928,  
    CP932,  
    CP998,  
    CP999,  
    CP1001,  
    CP1250,  
    CP1251,  
    CP1252,  
    CP2001,  
    CP3001,  
    CP3002,  
    CP3011,  
    CP3012,  
    CP3021,  
    CP3041,  
    CP3840,  
    CP3841,  
    CP3843,  
    CP3844,  
    CP3845,  
    CP3846,  
    CP3847,  
    CP3848,  
    UTF8,  
    Blank  
}
```

定数

名称	説明
CP437	CodePage437 (USA, Std. Europe).
CP737	Codepage 737 (Greek).
CP772	Codepage 772 (Lithuanian).
CP774	Codepage 774 (Lithuanian).
CP851	Codepage 851 (Greek).
CP852	Codepage 852 (Latin-2).
CP855	Codepage 855 (Cyrillic Bulgarian).
CP857	Codepage 857 (Turkey).
CP858	Codepage 858 (Multilingual).
CP860	Codepage 860 (Portuguese).
CP861	Codepage 861 (Icelandic).
CP862	Codepage 862 (Israel (Hebrew)).
CP863	Codepage 863 (Canadian French).
CP864	Codepage 864 (Arabic).
CP865	Codepage 865 (Nordic).
CP866	Codepage 866 (Cyrillic Russian).
CP869	Codepage 869 (Greek).
CP874	Codepage 874 (Thai).
CP928	Codepage 928 (Greek).
CP932	Katakana.
CP998	Normal.
CP999	Codepage 1252 (Windows Latin-1).
CP1001	Codepage 1001 (Arabic).
CP1250	Codepage 1250 (Windows Latin-2).
CP1251	Codepage 1251 (Windows Cyrillic).
CP1252	Codepage 1252 (Windows Latin-1).
CP2001	Codepage 2001 (Lithuanian-KBL).
CP3001	Codepage 3001 (Estonian-1).
CP3002	Codepage 3002 (Estonian-2).
CP3011	Codepage 3011 (Latvian-1).
CP3012	Codepage 3012 (Latvian-2).
CP3021	Codepage 3021 (Bulgarian).
CP3041	Codepage 3041 (Maltese).
CP3840	Codepage 3840 (IBM-Russian).
CP3841	Codepage 3841 (Gost).
CP3843	Codepage 3843 (Polish).
CP3844	Codepage 3844 (CS2).
CP3845	Codepage 3845 (Hungarian).
CP3846	Codepage 3846 (Turkish).
CP3847	Codepage 3847 (Brazil-ABNT).
CP3848	Codepage 3848 (Brazil-ABICOMP).
UTF8	UTF-8.
Blank	User Setting Blank Code Page.

ApiFunctions.java / 各コマンド仕様書を参照ください。

8.48 InternationalType コンスタント

国際文字指定定数。

宣言

```
enum InternationalType {
    USA,
    France,
    Germany,
    UK,
    Denmark,
    Sweden,
    Italy,
    Spain,
    Japan,
    Norway,
    Denmark2,
    Spain2,
    LatinAmerica,
    Korea,
    Ireland,
    Legal
}
```

定数

名称	説明
USA	USA
France	フランス
Germany	ドイツ
UK	イギリス
Denmark	デンマーク
Sweden	スウェーデン
Italy	イタリア
Spain	スペイン
Japan	日本
Norway	ノルウェー
Denmark2	デンマーク II
Spain2	スペイン II
LatinAmerica	ラテンアメリカ
Korea	韓国
Ireland	アイルランド
Legal	Legal

ApiFunctions.java / 各コマンド仕様書を参照ください。

8.49 LogoSize コンスタント

ロゴサイズ指定定数。

宣言

```
enum LogoSize {
    Normal,
    DoubleWidth,
    DoubleHeight,
    DoubleWidthDoubleHeight
}
```

定数

名称	説明
Normal	ノーマルモード
DoubleWidth	横 2 倍モード
DoubleHeight	縦 2 倍モード
DoubleWidthDoubleHeight	縦 2 倍、横 2 倍モード

ApiFunctions.java / 各コマンド仕様書を参照ください。

8.50 AlignmentPosition コンスタント

位置揃え指定定数。

宣言

```
enum AlignmentPosition {
    Left,
    Center,
    Right
}
```

定数

名称	説明
Left	左揃え
Center	中心揃え
Right	右揃え

ApiFunctions.java / 各コマンド仕様書を参照ください。

8.51 CutPaperAction コンスタント

用紙カット指定定数。

宣言

```
enum CutPaperAction {
    FullCut,
    PartialCut,
    FullCutWithFeed,
    PartialCutWithFeed
}
```

定数

名称	説明
FullCut	フルカット
PartialCut	パーシャルカット
FullCutWithFeed	用紙送り付きフルカット
PartialCutWithFeed	用紙送り付きパーシャルカット

ApiFunctions.java / 各コマンド仕様書を参照ください。

8.52 PeripheralChannel コンスタント

キャッシュドロアーチャネル指定定数。

宣言

```
enum PeripheralChannel {
    No1,
    No2
}
```

定数

名称	説明
No1	チャネル 1
No2	チャネル 2

ApiFunctions.java を参照ください。

8.53 SoundChannel コンスタント

サウンド/ブザーチャネル指定定数。

宣言

```
enum SoundChannel {
    No1,
    No2
}
```

定数

名称	説明
No1	チャネル 1
No2	チャネル 2

ApiFunctions.java を参照ください。

8.54 BarcodeSymbology コンスタント

バーコードシンボル指定定数。

宣言

```
enum BarcodeSymbology {
    UPCE,
    UPCA,
    JAN8,
    JAN13,
    Code39,
    ITF,
    Code128,
    Code93,
    // Codabar,
    NW7
}
```

定数

名称	説明
UPCE	UPC-E.
UPCA	UPC-A.
JAN8	JAN/EAN8.
JAN13	JAN/EAN13.
Code39	Code39.
ITF	ITF.
Code128	Code128.
Code93	Code93.
Codabar	Codabar.
NW7	NW7.

ApiFunctions.java / 各コマンド仕様書を参照ください。

8.55 BarcodeWidth コンスタント

バーコード幅指定定数。

宣言

```
enum BarcodeWidth {
    Mode1,
    Mode2,
    Mode3,
    Mode4,
    Mode5,
    Mode6,
    Mode7,
    Mode8,
    Mode9
}
```

定数

名称	説明
Mode1	モード 1
Mode2	モード 2
Mode3	モード 3
Mode4	モード 4
Mode5	モード 5
Mode6	モード 6
Mode7	モード 7
Mode8	モード 8
Mode9	モード 9

ApiFunctions.java / 各コマンド仕様書を参照ください。

8.56 Pdf417Level コンスタント

PDF417 ECC（セキュリティレベル）指定定数。

宣言

```
enum Pdf417Level {
    ECC0,
    ECC1,
    ECC2,
    ECC3,
    ECC4,
    ECC5,
    ECC6,
    ECC7,
    ECC8
}
```

定数

名称	説明
ECC0	セキュリティレベル 0
ECC1	セキュリティレベル 1
ECC2	セキュリティレベル 2
ECC3	セキュリティレベル 3
ECC4	セキュリティレベル 4
ECC5	セキュリティレベル 5
ECC6	セキュリティレベル 6
ECC7	セキュリティレベル 7
ECC8	セキュリティレベル 8

ApiFunctions.java / 各コマンド仕様書を参照ください。

8.57 QrCodeModel コンスタント

QR コードモデル指定定数

宣言

```
enum QrCodeModel {
    No1,
    No2
}
```

定数

名称	説明
No1	モデル 1
No2	モデル 2

ApiFunctions.java / 各コマンド仕様書を参照ください。

8.58 QrCodeLevel コンスタント

QR コード誤り訂正レベル指定定数。

宣言

```
enum QrCodeLevel {
    L,
    M,
    Q,
    H
}
```

定数

名称	説明
L	誤り訂正レベル L
M	誤り訂正レベル M
Q	誤り訂正レベル Q
H	誤り訂正レベル H

ApiFunctions.java / 各コマンド仕様書を参照ください。

8.59 BitmapConverterRotation コンスタント

ビットマップ回転指定定数。

宣言

```
enum BitmapConverterRotation {
    Normal,
    Right90,
    Left90,
    Rotate180
}
```

定数

名称	説明
Normal	回転なし
Right90	右 90 度回転
Left90	左 90 度回転
Rotate180	180 度回転

ApiFunctions.java を参照ください。

8.60 BlackMarkType コンスタント

ブラックマーク指定定数。

宣言

```
enum BlackMarkType {
    Invalid,
    Valid,
    ValidWithDetection
}
```

定数

名称	説明
Invalid	ブラックマーク無効
Valid	ブラックマーク有効
ValidWithDetection	ブラックマーク有効 + ブラックマーク検出動作

ApiFunctions.java / 各コマンド仕様書を参照ください。

8.61 PrintableAreaType コンスタント

印字領域タイプ指定定数。

宣言

```
enum PrintableAreaType {
    Standard,
    Type1,
    Type2,
    Type3,
    Type4
}
```

定数

名称	説明
Standard	標準タイプ
Type1	タイプ 1
Type2	タイプ 2
Type3	タイプ 3
Type4	タイプ 4

ApiFunctions.java を参照ください。

9 IDisplayCommandBuilder インターフェイス (starioextension.jar)

カスタマーディスプレイ制御用コマンドを生成する機能を提供するインターフェイスです。

メソッド

名称	説明
append	データ（テキストやコマンド）をコマンドバッファに追加します。
appendBackSpace	バックスペースコマンドを生成し、コマンドバッファに追加します。
appendHorizontalTab	水平タブコマンドを生成し、コマンドバッファに追加します。
appendLineFeed	ラインフィードコマンドを生成し、コマンドバッファに追加します。
appendCarriageReturn	キャリッジリターンコマンドを生成し、コマンドバッファに追加します。
appendBitmap	グラフィック表示コマンドを生成し、コマンドバッファに追加します。
appendInternational	国際文字指定コマンドを生成し、コマンドバッファに追加します。
appendCodePage	コードページ指定コマンドを生成し、コマンドバッファに追加します。
appendDeleteToEndOfLine	画面を行末までクリアするコマンドを生成し、コマンドバッファに追加します。
appendClearScreen	画面をクリアするコマンドを生成し、コマンドバッファに追加します。
appendHomePosition	カーソルをホームポジションに移動するコマンドを生成し、コマンドバッファに追加します。
appendTurnOn	バックライトをオン・オフするコマンドを生成し、コマンドバッファに追加します。
appendSpecifiedPosition	カーソルを指定された位置に移動するコマンドを生成し、コマンドバッファに追加します。
appendCursorMode	カーソルモード変更コマンドを生成し、コマンドバッファに追加します。
appendContrastMode	コントラストモード変更コマンドを生成し、コマンドバッファに追加します。
appendUserDefinedCharacter	ユーザー定義文字(SBCS)登録コマンドを生成し、コマンドバッファに追加します。
appendUserDefinedDbcsCharacter	ユーザー定義文字(DBCS)登録コマンドを生成し、コマンドバッファに追加します。
getCommands	コマンドバッファ（生成したコマンド）を取得します。
getPassThroughCommands	コマンドバッファ（生成したコマンド）の先頭にプリンタへのパススルーコマンドが追加されたコマンド列を取得します。

コンスタント

名称	説明
InternationalType	国際文字指定定数。
CodePageType	コードページ指定定数。
CursorMode	カーソルモード指定定数。
ContrastMode	コントラストモード指定定数。

9.1 モデル : IDisplayCommandBuilder インターフェイス

IDisplayCommandBuilder インターフェイスはカスタマーディスプレイに対応しているモデル専用です。対応しているモデルは[対応外部機器](#)を参照ください。

9.2 append メソッド

データ（テキストやコマンド）をコマンドバッファに追加します。

宣言

```
void append(byte data);
```

```
void append(byte[] data);
```

引数

名称	説明	型
data	データ（テキストやコマンド）	byte
		byte[]

戻り値

説明	型
-	-

Example

```
public static byte[] createTextPattern(int number) {
    IDisplayCommandBuilder builder = StarIoExt.createDisplayCommandBuilder(DisplayModel.SCD222);

    builder.appendClearScreen();
    builder.appendCursorMode(CursorMode.Off);
    builder.appendHomePosition();

    byte[] pattern;

    ...

    builder.append(pattern);

    return builder.getPassThroughCommands();
}
```

DisplayFunctions.java を参照ください。

9.3 appendBackSpace メソッド

バックスペースコマンドを生成し、コマンドバッファに追加します。

宣言

```
void appendBackSpace();
```

引数

名称	説明	型
-	-	-

戻り値

説明	型
-	-

9.4 appendHorizontalTab メソッド

水平タブコマンドを生成し、コマンドバッファに追加します。

宣言

```
void appendHorizontalTab();
```

引数

名称	説明	型
-	-	-

戻り値

説明	型
-	-

9.5 appendLineFeed メソッド

ラインフィードコマンドを生成し、コマンドバッファに追加します。

宣言

```
void appendLineFeed();
```

引数

名称	説明	型
-	-	-

戻り値

説明	型
-	-

9.6 appendCarriageReturn メソッド

キャリッジリターンコマンドを生成し、コマンドバッファに追加します。

宣言

```
void appendCarriageReturn();
```

引数

名称	説明	型
-	-	-

戻り値

説明	型
-	-

9.7 appendBitmap メソッド

グラフィック表示コマンドを生成し、コマンドバッファに追加します。

宣言

```
void appendBitmap(Bitmap bitmap, boolean diffusion);
```

引数

名称	説明	型
bitmap	表示したいイメージ 横 160dot、縦 40dot のモノクロ Bitmap クラスのインスタンスを指定してください。 異なるサイズの Bitmap が指定された場合、横 160dot、縦 40dot にリサイズされます。 Bitmap の各ピクセルの色は自動でモノクロ 2 値に変換されます。	Bitmap
diffusion	誤差拡散 <ul style="list-style-type: none"> true ... 誤差拡散を行う false ... 誤差拡散を行わない 	boolean

戻り値

説明	型
-	-

Example

```
public static byte[] createGraphicPattern(int number, Resources resources) {
    IDisplayCommandBuilder builder = StarIoExt.createDisplayCommandBuilder(DisplayModel.SCD222);

    builder.appendClearScreen();
    builder.appendCursorMode(CursorMode.Off);
    // builder.appendHomePosition();

    Bitmap bitmap;

    switch (number) {
        default:
        case 0:
            bitmap = BitmapFactory.decodeResource(resources, R.drawable.display_image_1);
            break;
        case 1:
            bitmap = BitmapFactory.decodeResource(resources, R.drawable.display_image_2);
            break;
        case 2:
            bitmap = BitmapFactory.decodeResource(resources, R.drawable.display_image_3);
            break;
        case 3:
            bitmap = BitmapFactory.decodeResource(resources, R.drawable.display_image_4);
            break;
    }

    builder.appendBitmap(bitmap, false);

    return builder.getPassThroughCommands();
}
```

DisplayFunctions.java を参照ください。

9.8 appendInternational メソッド

国際文字指定コマンドを生成し、コマンドバッファに追加します。

宣言

```
void appendInternational(InternationalType internationalType);
```

引数

名称	説明	型
internationalType	国際文字指定定数	com.starmicronics.starioextension.IDisplayCommandBuilder.InternationalType

戻り値

説明	型
-	-

Example

```
public static byte[] createCharacterSet(InternationalType internationalType, CodePageType codePageType) {
    IDisplayCommandBuilder builder = StarIoExt.createDisplayCommandBuilder(DisplayModel.SCD222);

    builder.appendClearScreen();
    builder.appendCursorMode(CursorMode.Off);
    builder.appendHomePosition();

    builder.appendInternational(internationalType);
    builder.appendCodePage(codePageType);

    ...

    switch (codePageType) {
        default: builder.append(pattern1); break; //
CP437,Katakana,CP850,CP860,CP863,CP865,CP1252,CP866,CP852,CP858
        case Japanese: builder.append(pattern2); break;
        case SimplifiedChinese: builder.append(pattern3); break;
        case TraditionalChinese: builder.append(pattern4); break;
        case Hangul: builder.append(pattern5); break;
    }

    return builder.getPassThroughCommands();
}
```

DisplayFunctions.java を参照ください。

9.9 appendCodePage メソッド

コードページ指定コマンドを生成し、コマンドバッファに追加します。

宣言

```
void appendCodePage(CodePageType codePageType);
```

引数

名称	説明	型
codePageType	コードページ指定定数	com.starmicronics.starioextension.IDisplayCommandBuilder.CodePageType

戻り値

説明	型
-	-

Example

```
public static byte[] createCharacterSet(InternationalType internationalType, CodePageType codePageType) {
    IDisplayCommandBuilder builder = StarIoExt.createDisplayCommandBuilder(DisplayModel.SCD222);

    builder.appendClearScreen();
    builder.appendCursorMode(CursorMode.Off);
    builder.appendHomePosition();

    builder.appendInternational(internationalType);
    builder.appendCodePage(codePageType);

    ...

    switch (codePageType) {
        default: builder.append(pattern1); break; //
CP437,Katakana,CP850,CP860,CP863,CP865,CP1252,CP866,CP852,CP858
        case Japanese: builder.append(pattern2); break;
        case SimplifiedChinese: builder.append(pattern3); break;
        case TraditionalChinese: builder.append(pattern4); break;
        case Hangul: builder.append(pattern5); break;
    }

    return builder.getPassThroughCommands();
}
```

DisplayFunctions.java を参照ください。

9.10 appendDeleteToEndOfLine メソッド

画面を行末までクリアするコマンドを生成し、コマンドバッファに追加します。

宣言

```
void appendDeleteToEndOfLine();
```

引数

名称	説明	型
-	-	-

戻り値

説明	型
-	-

9.11 appendClearScreen メソッド

画面をクリアするコマンドを生成し、コマンドバッファに追加します。

宣言

```
void appendClearScreen();
```

引数

名称	説明	型
-	-	-

戻り値

説明	型
-	-

Example

```
public static byte[] createTextPattern(int number) {
    IDisplayCommandBuilder builder = StarIoExt.createDisplayCommandBuilder(DisplayModel.SCD222);

    builder.appendClearScreen();
    builder.appendCursorMode(CursorMode.Off);
    builder.appendHomePosition();

    byte[] pattern;

    ...

    builder.append(pattern);

    return builder.getPassThroughCommands();
}
```

DisplayFunctions.java を参照ください。

9.12 appendHomePosition メソッド

カーソルをホームポジションに移動するコマンドを生成し、コマンドバッファに追加します。

宣言

```
void appendHomePosition();
```

引数

名称	説明	型
-	-	-

戻り値

説明	型
-	-

Example

```
public static byte[] createTextPattern(int number) {
    IDisplayCommandBuilder builder = StarIoExt.createDisplayCommandBuilder(DisplayModel.SCD222);

    builder.appendClearScreen();
    builder.appendCursorMode(CursorMode.Off);
    builder.appendHomePosition();

    byte[] pattern;

    ...

    builder.append(pattern);

    return builder.getPassThroughCommands();
}
```

DisplayFunctions.java を参照ください。

9.13 appendTurnOn メソッド

バックライトをオン・オフするコマンドを生成し、コマンドバッファに追加します。

宣言

```
void appendTurnOn(boolean turnOn);
```

引数

名称	説明	型
turnOn	バックライトのオン・オフ	boolean

戻り値

説明	型
-	-

Example

```
public static byte[] createTurnOn(boolean isTurnOn) {
    IDisplayCommandBuilder builder = StarIoExt.createDisplayCommandBuilder(DisplayModel.SCD222);

    ...

    builder.appendTurnOn(isTurnOn);

    return builder.getPassThroughCommands();
}
```

DisplayFunctions.java を参照ください。

9.14 appendSpecifiedPosition メソッド

カーソルを指定された位置に移動するコマンドを生成し、コマンドバッファに追加します。

宣言

```
void appendSpecifiedPosition(int x, int y);
```

引数

名称	説明	型
x	桁位置（一番左の桁が 1）	int
y	行位置（一番上の行が 1）	int

戻り値

説明	型
-	-

Example

```
public static byte[] createCursorMode(CursorMode mode) {  
    IDisplayCommandBuilder builder = StarloExt.createDisplayCommandBuilder(DisplayModel.SCD222);  
  
    builder.appendClearScreen();  
    builder.appendCursorMode(CursorMode.Off);  
    builder.appendHomePosition();  
  
    byte[] pattern = {'S', 't', 'a', 'r', ' ', 'M', 'i', 'c', 'r', 'o', 'n', 'i', 'c', 's', ' ', ' ', ' ', ' ', ' ', ' ', '  
        'T', 'o', 't', 'a', 'l', ' ', ':', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', '1', '2', '3', '4', '5'};  
  
    builder.append(pattern);  
  
    builder.appendSpecifiedPosition(20, 2);  
  
    builder.appendCursorMode(mode);  
  
    return builder.getPassThroughCommands();  
}
```

DisplayFunctions.java を参照ください。

9.15 appendCursorMode メソッド

カーソルモード変更コマンドを生成し、コマンドバッファに追加します。

宣言

```
void appendCursorMode(CursorMode cursorMode);
```

引数

名称	説明	型
cursorMode	カーソルモード指定定数	com.starmicronics.starioextension.IDisplay CommandBuilder.CursorMode

戻り値

説明	型
-	-

Example

```
public static byte[] createCursorMode(CursorMode mode) {  
    IDisplayCommandBuilder builder = StarloExt.createDisplayCommandBuilder(DisplayModel.SCD222);  
  
    builder.appendClearScreen();  
    builder.appendCursorMode(CursorMode.Off);  
    builder.appendHomePosition();  
  
    byte[] pattern = {'S', 't', 'a', 'r', 'l', 'o', 'M', 'e', 'n', 't', 'c', 'o', 'u', 'n', 't', 's', ' ', '1', '2', '3', '4', '5'},  
                    'T', 'o', 't', 'a', 'l', ' ', ':', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', '1', '2', '3', '4', '5'};  
  
    builder.append(pattern);  
  
    builder.appendSpecifiedPosition(20, 2);  
  
    builder.appendCursorMode(mode);  
  
    return builder.getPassThroughCommands();  
}
```

DisplayFunctions.java を参照ください。

9.16 appendContrastMode メソッド

コントラストモード変更コマンドを生成し、コマンドバッファに追加します。

宣言

```
void appendContrastMode(ContrastMode contrastMode);
```

引数

名称	説明	型
contrastMode	コントラストモード指定定数	com.starmicronics.starioextension.IDisplayCommandBuilder.ContrastMode

戻り値

説明	型
-	-

Example

[illegible]

DisplayFunctions.java を参照ください。

9.17 appendUserDefinedCharacter メソッド

ユーザー定義文字(SBCS)登録コマンドを生成し、コマンドバッファに追加します。

宣言

```
void appendUserDefinedCharacter(int index, int code, byte[] font);
```

引数

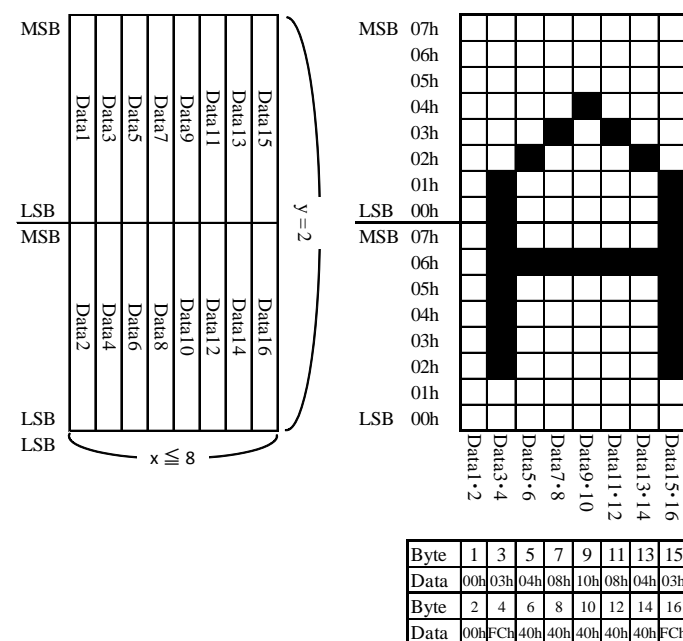
名称	説明	型
index	フォント番号 $00h \leq \text{index} \leq 1Fh$	int
code	ユーザー定義文字の文字コード $20h \leq \text{code} \leq 7Fh$ appendCharacterSet メソッド、appendCharacterSetCodePage メソッドで CodePageType. Japanese を指定した場合、以下の範囲も使用可能です。 $A0h \leq \text{code} \leq DFh$	int
font	16byte のフォントデータ ユーザー定義文字のフォントデータのフォーマットを参照してください。 null を指定すると、指定したフォント番号のユーザー定義文字を解除します。	byte[]

戻り値

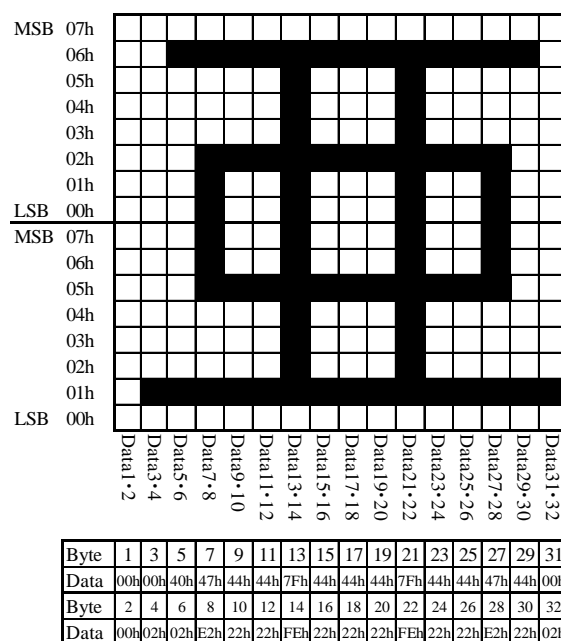
説明	型
-	-

ユーザー定義文字のフォントデータのフォーマット

ユーザー定義文字 (SBCD)



ユーザー定義文字 (DBCS)



Example

```
public static byte[] createUserDefinedCharacter(boolean set) {
    IDisplayCommandBuilder builder = StarIoExt.createDisplayCommandBuilder(DisplayModel.SCD222);

    builder.appendClearScreen();
    builder.appendCursorMode(CursorMode.Off);
    builder.appendHomePosition();

    builder.appendInternational(InternationalType.USA);
    builder.appendCodePage(CodePageType.Japanese);

    if (set) {
        byte[] font = {
            (byte) 0x00, (byte) 0x00, (byte) 0x32, (byte) 0x00, (byte) 0x49, (byte) 0x00, (byte) 0x49, (byte) 0x7f, (byte) 0x26,
            (byte) 0x48, (byte) 0x00, (byte) 0x48, (byte) 0x00, (byte) 0x30, (byte) 0x00, (byte) 0x00
        };
        byte[] dbcsFont = {
            (byte) 0x00, (byte) 0x00, (byte) 0x00, (byte) 0x00, (byte) 0x00, (byte) 0x00, (byte) 0x00, (byte) 0x00, (byte) 0x03,
            (byte) 0x20, (byte) 0x04, (byte) 0x90, (byte) 0x04, (byte) 0x90, (byte) 0x02, (byte) 0x60,
            (byte) 0x00, (byte) 0x00, (byte) 0x07, (byte) 0xf0, (byte) 0x04, (byte) 0x80, (byte) 0x04, (byte) 0x80, (byte) 0x03,
            (byte) 0x00, (byte) 0x00, (byte) 0x00, (byte) 0x00, (byte) 0x00, (byte) 0x00, (byte) 0x00
        };
        builder.appendUserDefinedCharacter(0, 0x20, font);
        builder.appendUserDefinedDbcsCharacter(0, 0x8140, dbcsFont);
    }
    else {
        builder.appendUserDefinedCharacter(0, 0x20, null);
        builder.appendUserDefinedDbcsCharacter(0, 0x8140, null);
    }

    byte[] pattern = {
        (byte) 0x5b, (byte) 0x20, (byte) 0x20, (byte) 0x53, (byte) 0x74, (byte) 0x61, (byte) 0x72, (byte) 0x20, (byte) 0x4d, (byte)
        0x69, (byte) 0x63, (byte) 0x72, (byte) 0x6f, (byte) 0x6e, (byte) 0x69, (byte) 0x63, (byte) 0x73, (byte) 0x20, (byte) 0x20, (byte)
        0x5d,
        (byte) 0x5b, (byte) 0x81, (byte) 0x40, (byte) 0x81, (byte) 0x40, (byte) 0x83, (byte) 0x58, (byte) 0x83, (byte) 0x5e, (byte)
        0x81, (byte) 0x5b, (byte) 0x90, (byte) 0xb8, (byte) 0x96, (byte) 0xa7, (byte) 0x81, (byte) 0x40, (byte) 0x81, (byte) 0x40, (byte)
        0x5d,
    };
    builder.append(pattern);

    return builder.getPassThroughCommands();
}
```

DisplayFunctions.java を参照ください。

9.18 appendUserDefinedDbcsCharacter メソッド

ユーザー定義文字(DBCS)登録コマンドを生成し、コマンドバッファに追加します。

宣言

```
void appendUserDefinedDbcsCharacter(int index, int code, byte[] font);
```

引数

名称	説明	型
index	フォント番号 $00h \leq \text{index} \leq 0Fh$	int
code	ユーザー定義文字の文字コード $20h \leq \text{code} \leq 7Fh$ appendCharacterSet メソッド、appendCharacterSetCodePage メソッドで CodePageType. Japanese を指定した場合、以下の範囲も使用可能です。 $80\ 00h \leq \text{code} \leq FF\ FFh$	int
font	32byte のフォントデータ ユーザー定義文字のフォントデータのフォーマットを参照してください。 null を指定すると、指定したフォント番号のユーザー定義文字を解除します。	byte[]

戻り値

説明	型
-	-

Example

```
public static byte[] createUserDefinedCharacter(boolean set) {
    IDisplayCommandBuilder builder = StarIoExt.createDisplayCommandBuilder(DisplayModel.SCD222);

    builder.appendClearScreen();
    builder.appendCursorMode(CursorMode.Off);
    builder.appendHomePosition();

    builder.appendInternational(InternationalType.USA);
    builder.appendCodePage(CodePageType.Japanese);

    if (set) {
        byte[] font = {
            (byte) 0x00, (byte) 0x00, (byte) 0x32, (byte) 0x00, (byte) 0x49, (byte) 0x00, (byte) 0x49, (byte) 0x7f, (byte) 0x26,
            (byte) 0x48, (byte) 0x00, (byte) 0x48, (byte) 0x00, (byte) 0x30, (byte) 0x00, (byte) 0x00
        };
        byte[] dbcsFont = {
            (byte) 0x00, (byte) 0x00, (byte) 0x00, (byte) 0x00, (byte) 0x00, (byte) 0x00, (byte) 0x00, (byte) 0x00, (byte) 0x03,
            (byte) 0x20, (byte) 0x04, (byte) 0x90, (byte) 0x04, (byte) 0x90, (byte) 0x02, (byte) 0x60,
            (byte) 0x00, (byte) 0x00, (byte) 0x07, (byte) 0xf0, (byte) 0x04, (byte) 0x80, (byte) 0x04, (byte) 0x80, (byte) 0x03,
            (byte) 0x00, (byte) 0x00, (byte) 0x00, (byte) 0x00, (byte) 0x00, (byte) 0x00, (byte) 0x00
        };
        builder.appendUserDefinedCharacter(0, 0x20, font);
        builder.appendUserDefinedDbcsCharacter(0, 0x8140, dbcsFont);
    }
    else {
        builder.appendUserDefinedCharacter(0, 0x20, null);
        builder.appendUserDefinedDbcsCharacter(0, 0x8140, null);
    }

    byte[] pattern = {
        (byte) 0x5b, (byte) 0x20, (byte) 0x20, (byte) 0x53, (byte) 0x74, (byte) 0x61, (byte) 0x72, (byte) 0x20, (byte) 0x4d, (byte)
        0x69, (byte) 0x63, (byte) 0x72, (byte) 0x6f, (byte) 0x6e, (byte) 0x69, (byte) 0x63, (byte) 0x73, (byte) 0x20, (byte) 0x20, (byte)
        0x5d,
        (byte) 0x5b, (byte) 0x81, (byte) 0x40, (byte) 0x81, (byte) 0x40, (byte) 0x83, (byte) 0x58, (byte) 0x83, (byte) 0x5e, (byte)
        0x81, (byte) 0x5b, (byte) 0x90, (byte) 0xb8, (byte) 0x96, (byte) 0xa7, (byte) 0x81, (byte) 0x40, (byte) 0x81, (byte) 0x40, (byte)
        0x5d,
    };
    builder.append(pattern);

    return builder.getPassThroughCommands();
}
```

DisplayFunctions.java を参照ください。

9.19 getCommands メソッド

コマンドバッファ（生成したコマンド）を取得します。

宣言

```
byte[] getCommands();
```

引数

名称	説明	型
-	-	-

戻り値

説明	型
コマンドバッファ（生成したコマンド）	byte[]

9.20 getPassThroughCommands メソッド

コマンドバッファ（生成したコマンド）の先頭にプリンタへのパススルーコマンドが追加されたコマンド列を取得します。

宣言

```
byte[] appendPassThroughCommands();
```

引数

名称	説明	型
-	-	-

戻り値

説明	型
コマンドバッファ（生成したコマンド）	byte[]

Example

```
public static byte[] createTextPattern(int number) {
    IDisplayCommandBuilder builder = StarIoExt.createDisplayCommandBuilder(DisplayModel.SCD222);

    builder.appendClearScreen();
    builder.appendCursorMode(CursorMode.Off);
    builder.appendHomePosition();

    byte[] pattern;

    ...

    builder.append(pattern);

    return builder.getPassThroughCommands();
}
```

DisplayFunctions.java を参照ください。

9.21 InternationalType コンスタント

国際文字指定定数。

宣言

```
enum InternationalType {
    USA,
    France,
    Germany,
    UK,
    Denmark,
    Sweden,
    Italy,
    Spain,
    Japan,
    Norway,
    Denmark2,
    Spain2,
    LatinAmerica,
    Korea
}
```

定数

名称	説明
USA	USA
France	フランス
Germany	ドイツ
UK	イギリス
Denmark	デンマーク
Sweden	スウェーデン
Italy	イタリア
Spain	スペイン
Japan	日本
Norway	ノルウェー
Denmark2	デンマーク II
Spain2	スペイン II
LatinAmerica	ラテンアメリカ
Korea	韓国

9.22 CodePageType コンスタント

コードページ指定定数。

宣言

```
enum CodePageType {
    CP437,
    Katakana,
    CP850,
    CP860,
    CP863,
    CP865,
    CP1252,
    CP866,
    CP852,
    CP858,
    Japanese,
    SimplifiedChinese,
    TraditionalChinese,
    Hangul
}
```

定数

名称	説明
CP437	CodePage437 (USA, Std. Europe).
Katakana	Katakana.
CP850	PC850 (Multilingual)
CP860	PC860 (Portuguese)
CP863	PC863 (Canadian-French)
CP865	PC865 (Norwegian)
CP1252	WPC1252
CP866	PC866 [Cyrillic #2]
CP852	PC852 [Latin 2]
CP858	Page 19 [PC858]
Japanese	Japanese font (shift JIS)
SimplifiedChinese	Simplified Chinese (GB2312)
TraditionalChinese	Traditional Chinese (Big5)
Hangul	Hangul (KSC5601)

9.23 CursorMode コンスタント

カーソルモード指定定数。

宣言

```
enum CursorMode {
    Off,
    Blink,
    On,
}
```

定数

名称	説明
Off	カーソル消灯
Blink	カーソル点滅
On	カーソル点灯

9.24 ContrastMode コンスタント

コントラストモード指定定数。

宣言

```
enum ContrastMode {
    Minus3,
    Minus2,
    Minus1,
    Default,
    Plus1,
    Plus2,
    Plus3,
}
```

定数

名称	説明
Minus3	コントラストマイナス 3
Minus2	コントラストマイナス 2
Minus1	コントラストマイナス 1
Default	デフォルト
Plus1	コントラストプラス 1
Plus2	コントラストプラス 2
Plus3	コントラストプラス 3

10 IPeripheralCommandParser インターフェイス (starioextension.jar)

外部機器（バーコードリーダー・カスタマディスプレイ）制御用コマンドの応答を解析する機能を提供するインターフェイスです。

メソッド

名称	説明
createSendCommands	外部機器(カスタマディスプレイ・バーコードリーダ)からの応答を受け取るコマンドを生成します。
parse	createSendCommands で生成したコマンドを送信後の、外部機器(カスタマディスプレイ・バーコードリーダ)からのコマンド応答を解析します。

コンスタント

名称	説明
ParseResult	解析結果定数。

10.1 モデル : IPeripheralCommandParser インターフェイス

IPeripheralCommandParser インターフェイスは外部機器(カスタマディスプレイ・バーコードリーダ)に対応しているモデル専用です。対応しているモデルは[対応外部機器](#)を参照ください。

10.2 createSendCommands メソッド

外部機器(カスタマディスプレイ・バーコードリーダ)からの応答を受け取るコマンドを生成します。

宣言

```
byte[] createSendCommands();
```

引数

名称	説明	型
-	-	-

戻り値

説明	型
生成されたコマンド列	byte[]

Example

```
public void run() {
    Communication.Result communicateResult = Communication.Result.ErrorOpenPort;
    boolean result = false;

    synchronized (mLock) {
        try {
            if (mPort == null) {
                mPort = StarIOPort.getPort(mPortName, mPortSettings, mTimeout, mContext);
            }

            communicateResult = Communication.Result.ErrorWritePort;
            StarPrinterStatus status = mPort.retrieveStatus();
            if (status.rawLength == 0) {
                throw new StarIOPortException("Unable to communicate with printer.");
            }

            communicateResult = Communication.Result.ErrorWritePort;
            byte[] sendData = mParser.createSendCommands();
            mPort.writePort(sendData, 0, sendData.length);

            List<Byte> receiveDataList = new ArrayList<>();
            byte[] readBuffer = new byte[1024];

            long start = System.currentTimeMillis();
            while (true) {
                if (1000 < (System.currentTimeMillis() - start)) {
                    communicateResult = Communication.Result.ErrorReadPort;
                    break;
                }

                try {
                    Thread.sleep(10);
                } catch (InterruptedException ignore) {}

                int receiveSize = mPort.readPort(readBuffer, 0, readBuffer.length);

                if (0 < receiveSize) {
                    for (int i = 0; i < receiveSize; i++) {
                        receiveDataList.add(readBuffer[i]);
                    }
                } else {
                    continue;
                }

                byte[] receiveData = new byte[receiveDataList.size()];
                int receiveDataLength = receiveDataList.size();

                for (int i = 0; i < receiveDataLength; i++) {
                    receiveData[i] = receiveDataList.get(i);
                }

                if (mParser.parse(receiveData, receiveDataLength) == ParseResult.Success) {
                    result = true;
                    communicateResult = Communication.Result.Success;
                    break;
                }
            }
        }
    }
}
```

Communication.java を参照ください。

10.3 parse メソッド

createSendCommands メソッドで生成したコマンドを送信後の、外部機器(カスタマディスプレイ・バーコードリーダ)からのコマンド応答を解析します。

宣言

```
ParseResult parse(byte[] response, int length);
```

引数

名称	説明	型
response	コマンド応答	byte[]
length	コマンド応答の長さ	int

戻り値

説明	型
解析結果	com.starmicronics.starioextension.IPeripheralCommandParser.ParseResult

Example

createSendCommands メソッドの Example を参照ください。

10.4 ParseResult コンスタント

解析結果定数。

宣言

```
enum ParseResult {
    Invalid,
    Success,
    Failure,
}
```

定数

名称	説明
Invalid	不定（コマンド応答未完了）
Success	成功
Failure	失敗

11 IPeripheralConnectParser インターフェイス (starioextension.jar)

外部機器(カスタマーディスプレイ・バーコードリーダー)の接続・切断ステータスを取得する機能を提供するインターフェイスです。

宣言

```
public interface IPeripheralConnectParser extends IPeripheralCommandParser
```

メソッド

名称	説明
isConnected	外部機器(カスタマーディスプレイ・バーコードリーダー)の接続・切断ステータスを取得します。

11.1 モデル : IPeripheralConnectParser インターフェイス

IPeripheralConnectParser インターフェイスは外部機器(カスタマディスプレイ・バーコードリーダー)に対応しているモデル専用です。対応しているモデルは[対応外部機器](#)を参照ください。

11.2 isConnected メソッド

外部機器(カスタマーディスプレイ・バーコードリーダー)の接続・切断ステータスを取得します。
parse メソッドが ParseResult.Success を返した後に実行してください。

宣言

```
boolean isConnected();
```

引数

名称	説明	型
-	-	-

戻り値

説明	型
接続・切断ステータス	boolean

Example

```
Communication.parseDoNotCheckCondition(DisplayFragment.class, parser, setting.getPortName(),
setting.getPortSettings(), 10000, getActivity(), new Communication.SendCallback() {
    @Override
    public void onStatus(boolean result, Communication.Result communicateResult) {
        if (!isForeground) {
            return;
        }

        String msg;
        String title;

        if (result) {
            title = "Check Status";

            if (parser.isConnected()) {
                msg = "Display Connect";
            }
            else {
                msg = "Display Disconnect";
            }
        }
        else {
            title = "Communication Result";
            msg = "Printer Impossible";
        }

        ...
    }
});
```

DisplayFragment.java/ DisplayExtFragment.java を参照ください。

12 StarPRNT Android SDK Sample

12.1 Communication

StarIO Android SDK 相当の印刷データ送信の一例と StarIoExtManager オブジェクトを使用した印刷データ送信の一例が用意されています。

-StarIoExtManager について-

StarIoExtManager を用いた場合、デバイス間が常時接続となります。

そのため、他のアプリケーションや他の端末とデバイスを共有する場合、また他のアプリケーションへの遷移やスリープ状態への遷移が想定される場合、デバイスとの接続状態に関する実装上の配慮が必要となります。

Example (StarIoExtManager オブジェクトを使用)

```
public static Result sendCommands(byte[] commands, StarIoPort port, Context context) {
    Result result = Result.ErrorUnknown;

    try {
        if (port == null) {
            result = Result.ErrorOpenPort;
            return result;
        }

        StarPrinterStatus status;

        result = Result.ErrorBeginCheckedBlock;

        status = port.beginCheckedBlock();

        if (status.offline) {
            throw new StarIoPortException("A printer is offline");
        }

        result = Result.ErrorWritePort;

        port.writePort(commands, 0, commands.length);

        result = Result.ErrorEndCheckedBlock;

        port.setEndCheckedBlockTimeoutMillis(30000); // 30000mS!!!

        status = port.endCheckedBlock();

        if (status.coverOpen) {
            throw new StarIoPortException("Printer cover is open");
        }
        else if (status.receiptPaperEmpty) {
            throw new StarIoPortException("Receipt paper is empty");
        }
        else if (status.offline) {
            throw new StarIoPortException("Printer is offline");
        }

        result = Result.Success;
    }
    catch (StarIoPortException e) {

        ...

    }

    return result;
}
```


13 StarIoExtManager クラス (starioextension.jar)

コンストラクタ

名称	説明
StarIoExtManager	StarIoExtManager を初期化します。

メソッド

名称	説明
connect	マネージメントを開始します。
disconnect	マネージメントを停止します。
getPort	StarIoPort オブジェクトを取得します。
getPrinterStatus	プリンタステータスを取得します。
getPrinterPaperStatus	プリンタ用紙ステータスを取得します。
getPrinterCoverStatus	プリンタカバーステータスを取得します。
getCashDrawerStatus	キャッシュドロアーステータスを取得します。
getBarcodeReaderStatus	バーコードリーダーステータスを取得します。
getCashDrawerOpenActiveHigh	キャッシュドロア開閉センサのモードを取得します。
setCashDrawerOpenActiveHigh	キャッシュドロア開閉センサのモードを設定します。
setListener	StarIoExtManager のリスナーオブジェクトを設定します。

コンスタント

名称	説明
Type	マネージャタイプ定数。
PrinterStatus	プリンタステータス定数。
PrinterPaperStatus	プリンタ用紙ステータス定数。
PrinterCoverStatus	プリンタカバーステータス定数。
CashDrawerStatus	キャッシュドロアーステータス定数。
BarcodeReaderStatus	バーコードリーダーステータス定数。

- BSC10 の USB 接続時のプリンタビジー条件について -

BSC10 の USB 接続においてはプリンタビジー条件を“受信バッファフル”（工場出荷時設定は“受信バッファフル、オフライン”）に設定してください。（メモリスイッチ 7 のビット 4 を 1 に設定）

-USB 接続時の自動パワーダウンモードについて-

SM-S シリーズ、SM-T シリーズにおいては自動パワーダウンモードを“NO USE”（工場出荷時設定）に設定してください。

13.1 StarloExtManager コンストラクタ

StarloExtManager を初期化します。

宣言

```
StarloExtManager(Type type, String portName, String portSetting, int ioTimeoutMillis, Context context);
```

引数

名称	説明	型
type	マネージャタイプ <ul style="list-style-type: none"> Standard ... プリンタ、キャッシュドロワのマネージメント WithBarcodeReader ... プリンタ、キャッシュドロワ、バーコードリーダーのマネージメント OnlyBarcodeReader ... バーコードリーダーのマネージメント 	com.starmicronics.starioextension.StarloExtManager.Type
portName	プリンタポート名 StarloPort クラスの getPort メソッド の portName と同じです。	String
portSettings	ポートセッティング StarloPort クラスの getPort メソッド の portSetting と同じです。	String
ioTimeoutMillis	内部制御および API のタイムアウト値	int
context	コンテキスト	android.content.Context

戻り値

説明	型
StarloExtManager オブジェクト	com.starmicronics.starioextension.StarloExtManager

Example

```
public void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);

    PrinterSetting setting = new PrinterSetting(getContext());

    mStarloExtManager = new StarloExtManager(StarloExtManager.Type.Standard, setting.getPortName(),
    setting.getPrinterType(), 10000, getContext()); // 10000mS!!!

    mStarloExtManager.setListener(mStarloExtManagerListener);
}
```

PrinterExtFragment.java / CashDrawerExtFragment.java / BarcodeReaderExtFragment.java / CombinationExtFragment.java を参照ください。

13.2 connect メソッド

マネージメントを開始します。

宣言

```
public void connect(final IConnectionCallback callback);
```

引数

名称	説明	型
callback	接続が完了した際、接続結果を onConnected メソッドの引数によって捉えることができます。 -result- • 接続結果	com.starmicronics.starioextension.IConnectionCallback

戻り値

説明	型
-	-

Example

```
public class PrinterExtFragment extends Fragment implements IConnectionCallback {
    ...

    @Override
    public void onStart() {
        super.onStart();

        ...

        mStarloExtManager.connect(this);
    }

    ...

    @Override
    public void onConnected() {
        Log.d("PrinterExtFragment", "onConnected " + result);

        ...
    }

    ...
}
```

PrinterExtFragment.java / CashDrawerExtFragment.java / BarcodeReaderExtFragment.java / CombinationExtFragment.java を参照ください。

13.3 disconnect メソッド

マネージメントを停止します。

宣言

```
public void disconnect(IConnectionCallback callback);
```

引数

名称	説明	型
callback	接続が完了した際、切断完了を onDisconnected メソッドによって捉えることができます。	com.starmicronics.starioextension.IConnectionCallback

戻り値

説明	型
-	-

Example

```
public class PrinterExtFragment extends Fragment implements IConnectionCallback {

    ...

    @Override
    public void onStop() {
        super.onStop();

        ...

        mStarloExtManager.disconnect(this);
    }

    ...

    @Override
    public void onDisconnected() {
        Log.d("PrinterExtFragment", "onDisconnected");

        ...

    }

    ...

}
```

PrinterExtFragment.java / CashDrawerExtFragment.java / BarcodeReaderExtFragment.java / CombinationExtFragment.java を参照ください。

13.4 getPort メソッド

StarIOPort オブジェクトを取得します。

宣言

```
StarIOPort getPort();
```

引数

名称	説明	型
-	-	-

戻り値

説明	型
StarIOPort オブジェクト	com.starmicronics.stario.StarIOPort

13.5 getPrinterOnlineStatus メソッド

プリンタオンラインステータスを取得します。

宣言

```
PrinterStatus getPrinterOnlineStatus();
```

引数

名称	説明	型
-	-	-

値

説明	型
プリンタステータス <ul style="list-style-type: none"> Invalid ... 不定 Impossible ... プリンタ使用不能 Online ... プリンタオンライン検出 Offline ... プリンタオフライン検出 	com.starmicronics.starioextension.StarIoExtManager.PrinterStatus

13.6 getPrinterPaperStatus メソッド

プリンタ用紙ステータスを取得します。

宣言

```
PrinterPaperStatus getPrinterPaperStatus();
```

引数

名称	説明	型
-	-	-

値

説明	型
プリンタ用紙ステータス <ul style="list-style-type: none"> Invalid ... 不定 Impossible ... 機器使用不能 Ready ... プリンタ用紙レディ検出 NearEmpty ... プリンタ用紙ニアエンド検出 Empty ... プリンタ用紙エンプティ検出 	com.starmicronics.starioextension.StarIoExtManager.PrinterPaperStatus

13.7 getPrinterCoverStatus メソッド

プリンタカバーステータスを取得します。

宣言

```
PrinterCoverStatus getPrinterCoverStatus();
```

引数

名称	説明	型
-	-	-

値

説明	型
プリンタカバーステータス <ul style="list-style-type: none"> Invalid ... 不定 Impossible ... 機器使用不能 Open ... プリンタカバーオープン検出 Close ... プリンタカバークローズ検出 	com.starmicronics.starioextension.StarIoExtManager.PrinterCoverStatus

13.8 getCashDrawerStatus メソッド

キャッシュドロアステータスを取得します。

宣言

```
CashDrawerStatus getCashDrawerStatus();
```

引数

名称	説明	型
-	-	-

値

説明	型
キャッシュドロアステータス <ul style="list-style-type: none"> Invalid ... 不定 Impossible ... 機器使用不能 Open ... キャッシュドロアオープン検出 Close ... キャッシュドロアクローズ検出 	com.starmicronics.starioextension.StarloExtManager.CashDrawerStatus

13.9 getBarcodeReaderStatus メソッド

バーコードリーダーステータスを取得します。

宣言

```
BarcodeReaderStatus getBarcodeReaderStatus();
```

引数

名称	説明	型
-	-	-

値

説明	型
バーコードリーダーステータス <ul style="list-style-type: none"> Invalid ... 不定 Impossible ... 機器使用不能 Connect ... バーコードリーダー接続検出 Disconnect ... バーコードリーダー切断検出 	com.starmicronics.starioextension.StarloExtManager.BarcodeReaderStatus

13.10 getCashDrawerOpenActiveHigh メソッド

キャッシュドロアー開閉センサのモードを取得します。

宣言

```
boolean getCashDrawerOpenActiveHigh();
```

引数

名称	説明	型
-	-	-

値

説明	型
キャッシュドロアー開閉センサモード <ul style="list-style-type: none"> • true ... High アクティブ • false ... Low アクティブ 	boolean

13.11 setCashDrawerOpenActiveHigh メソッド

キャッシュドロアー開閉センサのモードを設定します。

宣言

```
void setCashDrawerOpenActiveHigh(boolean isActiveHigh);
```

引数

名称	説明	型
isActiveHigh	キャッシュドロアー開閉センサモード <ul style="list-style-type: none"> • YES ... High アクティブ • NO ... Low アクティブ ※ mPOP メカニズム構造においてはデフォルト値 (YES) のこと。	boolean

戻り値

説明	型
-	-

13.12 setListener メソッド

StarloExtManager のリスナーオブジェクトを設定します。

宣言

```
void setListener(StarloExtManagerListener listener);
```

引数

名称	説明	型
listener	StarloExtManager のリスナーオブジェクト	com.starmicronics.starioextension.StarloExtManagerListener

戻り値

説明	型
-	-

Example

```
public class PrinterExtFragment extends Fragment implements IConnectionCallback {
    ...

    public void onStart() {
        super.onStart();
        ...

        mStarloExtManager.setListener(mStarloExtManagerListener);

        ...
    }

    private final StarloExtManagerListener mStarloExtManagerListener = new StarloExtManagerListener() {
        ...
    };
}
```

PrinterExtFragment.java / CashDrawerExtFragment.java / BarcodeReaderExtFragment.java / CombinationExtFragment.java を参照ください。

13.13 Type コンスタント

マネージャタイプ定数。

宣言

```
public enum Type {
    Standard,
    WithBarcodeReader,
    OnlyBarcodeReader,
}
```

定数

名称	説明
Standard	プリンタ、キャッシュドロワのマネージメント
WithBarcodeReader	プリンタ、キャッシュドロワ、バーコードリーダーのマネージメント
OnlyBarcodeReader	バーコードリーダーのマネージメント

13.14 PrinterStatus コンスタント

プリンタステータス定数。

宣言

```
public enum PrinterStatus {
    Invalid,
    Impossible,
    Online,
    Offline
}
```

定数

名称	説明
Invalid	不定
Impossible	プリンタ使用不能
Online	プリンタオンライン検出
Offline	プリンタオフライン検出

13.15 PrinterPaperStatus コンスタント

プリンタ用紙ステータス定数。

宣言

```
public enum PrinterPaperStatus {
    Invalid,
    Impossible,
    Ready,
    NearEmpty,
    Empty
}
```

定数

名称	説明
Invalid	不定
Impossible	プリンタ使用不能
Ready	プリンタ用紙レディ検出
NearEmpty	プリンタ用紙ニアエンド検出
Empty	プリンタ用紙エンプティ検出

13.16 PrinterCoverStatus コンスタント

プリンタカバーステータス定数。

宣言

```
public enum PrinterCoverStatus {
    Invalid,
    Impossible,
    Open,
    Close
}
```

定数

名称	説明
Invalid	不定
Impossible	プリンタ使用不能
Open	プリンタカバーオープン検出
Close	プリンタカバークローズ検出

13.17 CashDrawerStatus コンスタント

キャッシュドロアステータス定数。

宣言

```
public enum CashDrawerStatus {
    Invalid,
    Impossible,
    Open,
    Close
}
```

定数

名称	説明
Invalid	不定
Impossible	キャッシュドロア使用不能
Open	キャッシュドロアオープン検出
Close	キャッシュドロアクローズ検出

13.18 BarcodeReaderStatus コンスタント

バーコードリーダーステータス定数。

宣言

```
public enum BarcodeReaderStatus {
    Invalid,
    Impossible,
    Connect,
    Disconnect
}
```

定数

名称	説明
Invalid	不定
Impossible	バーコードリーダー使用不能
Connect	バーコードリーダー接続検出
Disconnect	バーコードリーダー切断検出

14 IConnectionCallback インターフェイス (starioextension.jar)

メソッド

名称	説明
onConnected	接続結果を通知します。* <i>Override</i>
onDisconnected	切断完了を通知します。* <i>Override</i>

コンスタント

名称	説明
ConnectResult	接続結果定数。

14.1 onConnected メソッド

接続結果を通知します。* *Override*

宣言

```
void onConnected(ConnectResult result);
```

引数

名称	説明	型
result	接続結果	com.starmicronics.starioextension.IConnectionCallback.ConnectResult

戻り値

説明	型
-	-

Example

```
@Override
public void onConnected(ConnectResult result) {
    Log.d("PrinterExtFragment", "onConnected " + result);
    ...
}
```

PrinterExtFragment.java / *CashDrawerExtFragment.java* / *BarcodeReaderExtFragment.java* / *CombinationExtFragment.java* を参照ください。

14.2 onDisconnected メソッド

切断完了を通知します。* *Override*

宣言

```
void onDisconnected();
```

引数

名称	説明	型
-	-	-

戻り値

説明	型
-	-

Example

```
@Override
public void onDisconnected() {
    Log.d("PrinterExtFragment", "onDisconnected");

    ...
}
```

PrinterExtFragment.java / CashDrawerExtFragment.java / BarcodeReaderExtFragment.java / CombinationExtFragment.java を参照ください。

14.3 ConnectResult コンスタント

接続結果定数。

宣言

```
enum ConnectResult {
    Success,
    Failure,
    AlreadyConnected
}
```

定数

名称	説明
Success	接続成功
Failure	接続失敗
AlreadyConnected	接続済み

15 StarloExtManagerListener (starioextension.jar)

メソッド

名称	説明
onPrinterImpossible	プリンタ使用不能を通知します。 * <i>Override</i>
onPrinterOnline	プリンタオンラインを通知します。 * <i>Override</i>
onPrinterOffline	プリンタオフラインを通知します。 * <i>Override</i>
onPrinterPaperReady	プリンタ用紙レディを通知します。 * <i>Override</i>
onPrinterPaperNearEmpty	プリンタ用紙ニアエンプティを通知します。 * <i>Override</i>
onPrinterPaperEmpty	プリンタ用紙エンプティを通知します。 * <i>Override</i>
onPrinterCoverOpen	プリンタカバーオープンを通知します。 * <i>Override</i>
onPrinterCoverClose	プリンタカバークローズを通知します。 * <i>Override</i>
onCashDrawerOpen	キャッシュドロアーオープンを通知します。 * <i>Override</i>
onCashDrawerClose	キャッシュドロアークローズを通知します。 * <i>Override</i>
onBarcodeReaderImpossible	バーコードリーダー使用不能を通知します。 * <i>Override</i>
onBarcodeReaderConnect	バーコードリーダー接続を通知します。 * <i>Override</i>
onBarcodeReaderDisconnect	バーコードリーダー切断を通知します。 * <i>Override</i>
onBarcodeDataReceive	バーコードデータ受信を通知します。 * <i>Override</i>
onAccessoryConnectSuccess	アクセサリ（Bluetooth / USB）切断状態でのアクセサリ再接続成功を通知します。 * <i>Override</i>
onAccessoryConnectFailure	アクセサリ（Bluetooth / USB）切断状態でのアクセサリ再接続失敗を通知します。 * <i>Override</i>
onAccessoryDisconnect	アクセサリ（Bluetooth / USB）切断を通知します。 * <i>Override</i>
onStatusUpdate	ステータスアップデートを通知します。 * <i>Override</i>

15.1 onPrinterImpossible メソッド

プリンタ使用不能を通知します。 * *Override*

宣言

```
void onPrinterImpossible();
```

引数

名称	説明	型
-	-	-

戻り値

説明	型
-	-

Example

```
@Override
public void onPrinterImpossible() {
    mComment.setText("Printer Impossible.");
    mComment.setTextColor(Color.RED);
}
```

PrinterExtFragment.java / CashDrawerExtFragment.java / CombinationExtFragment.java を参照ください。

15.2 onPrinterOnline メソッド

プリンタオンラインを通知します。* *Override*

宣言

```
void onPrinterOnline();
```

引数

名称	説明	型
-	-	-

戻り値

説明	型
-	-

Example

```
@Override
void onPrinterOnline() {
    mComment.setText("Printer Online.");
    mComment.setTextColor(Color.BLUE);
}
```

PrinterExtFragment.java / CombinationExtFragment.java を参照ください。

15.3 onPrinterOffline メソッド

プリンタオフラインを通知します。* *Override*

宣言

```
void onPrinterOffline();
```

引数

名称	説明	型
-	-	-

戻り値

説明	型
-	-

Example

```
@Override
void onPrinterOffline() {
    mComment.setText("Printer Offline.");
    mComment.setTextColor(Color.RED);
}
```

PrinterExtFragment.java / CombinationExtFragment.java を参照ください。

15.4 onPrinterPaperReady メソッド

プリンタ用紙レディを通知します。* *Override*

宣言

```
void onPrinterPaperReady();
```

引数

名称	説明	型
-	-	-

戻り値

説明	型
-	-

Example

```
@Override
void onPrinterPaperReady() {
    mComment.setText("Printer Paper Ready.");

    mComment.setTextColor(Color.BLUE);
}
```

PrinterExtFragment.java / CombinationExtFragment.java を参照ください。

15.5 onPrinterPaperNearEmpty メソッド

プリンタ用紙ニアエンプティを通知します。* *Override*

宣言

```
void onPrinterPaperNearEmpty();
```

引数

名称	説明	型
-	-	-

戻り値

説明	型
-	-

Example

```
@Override
void onPrinterPaperNearEmpty() {
    mComment.setText("Printer Paper Near Empty.");

    mComment.setTextColor(0xfffffa500); // Orange
}
```

PrinterExtFragment.java / CombinationExtFragment.java を参照ください。

15.6 onPrinterPaperEmpty メソッド

プリンタ用紙エンプティを通知します。* *Override*

宣言

```
void onPrinterPaperEmpty();
```

引数

名称	説明	型
-	-	-

戻り値

説明	型
-	-

Example

```
@Override
void onPrinterPaperEmpty() {
    mComment.setText("Printer Paper Empty.");

    mComment.setTextColor(Color.RED);
}
```

PrinterExtFragment.java / CombinationExtFragment.java を参照ください。

15.7 onPrinterCoverOpen メソッド

プリンタカバーオープンを通知します。* *Override*

宣言

```
void onPrinterCoverOpen();
```

引数

名称	説明	型
-	-	-

戻り値

説明	型
-	-

Example

```
@Override
void onPrinterCoverOpen() {
    mComment.setText("Printer Cover Open.");

    mComment.setTextColor(Color.RED);
}
```

PrinterExtFragment.java / CombinationExtFragment.java を参照ください。

15.8 onPrinterCoverClose メソッド

プリンタカバークローズを通知します。* *Override*

宣言

```
void onPrinterCoverClose();
```

引数

名称	説明	型
-	-	-

戻り値

説明	型
-	-

Example

```
@Override
void onPrinterCoverClose() {
    mComment.setText("Printer Cover Close.");

    mComment.setTextColor(Color.BLUE);
}
```

PrinterExtFragment.java / CombinationExtFragment.java を参照ください。

15.9 onCashDrawerOpen メソッド

キャッシュドロアーオープンを通知します。* *Override*

宣言

```
void onCashDrawerOpen();
```

引数

名称	説明	型
-	-	-

戻り値

説明	型
-	-

Example

```
@Override
void onCashDrawerOpen() {
    mComment.setText("Cash Drawer Open.");

    mComment.setTextColor(Color.MAGENTA);
}
```

CashDrawerExtFragment.java / CombinationExtFragment.java を参照ください。

15.10 onCashDrawerClose メソッド

キャッシュドロアークローズを通知します。* *Override*

宣言

```
void onCashDrawerClose();
```

引数

名称	説明	型
-	-	-

戻り値

説明	型
-	-

Example

```
@Override
void onCashDrawerClose() {
    mComment.setText("Cash Drawer Close.");

    mComment.setTextColor(Color.BLUE);
}
```

CashDrawerExtFragment.java / CombinationExtFragment.java を参照ください。

15.11 onBarcodeReaderImpossible メソッド

バーコードリーダー使用不能を通知します。* *Override*

宣言

```
void onBarcodeReaderImpossible();
```

引数

名称	説明	型
-	-	-

戻り値

説明	型
-	-

Example

```
@Override
void onBarcodeReaderImpossible() {
    mComment.setText("Barcode Reader Impossible.");

    mComment.setTextColor(Color.RED);
}
```

BarcodeReaderExtFragment.java / CombinationExtFragment.java を参照ください。

15.12 onBarcodeReaderConnect メソッド

バーコードリーダー接続を通知します。* *Override*

宣言

```
void onBarcodeReaderConnect();
```

引数

名称	説明	型
-	-	-

戻り値

説明	型
-	-

Example

```
@Override
void onBarcodeReaderConnect() {
    mComment.setText("Barcode Reader Connect.");

    mComment.setTextColor(Color.BLUE);
}
```

BarcodeReaderExtFragment.java / CombinationExtFragment.java を参照ください。

15.13 onBarcodeReaderDisconnect メソッド

バーコードリーダー切断を通知します。* *Override*

宣言

```
void onBarcodeReaderDisconnect();
```

引数

名称	説明	型
-	-	-

戻り値

説明	型
-	-

Example

```
@Override
void onBarcodeReaderDisconnect() {
    mComment.setText("Barcode Reader Disconnect.");

    mComment.setTextColor(Color.RED);
}
```

BarcodeReaderExtFragment.java / CombinationExtFragment.java を参照ください。

15.14 onBarcodeDataReceive メソッド

バーコードデータ受信を通知します。* *Override*

宣言

```
void onBarcodeDataReceive(byte[] data);
```

引数

名称	説明	型
data	受信したバーコードデータ	byte[]

戻り値

説明	型
-	-

Example

```
@Override
public void onBarcodeDataReceive( byte[] data) {
    String[] barcodeDataArray = new String(data).split("\n");

    for(String barcodeData:barcodeDataArray) {
        ...
    }
}
```

BarcodeReaderExtFragment.java / CombinationExtFragment.java を参照ください。

15.15 onAccessoryConnectSuccess メソッド

アクセサリ（Bluetooth / USB）切断状態でのアクセサリ再接続成功を通知します。* *Override*

宣言

```
void onAccessoryConnectSuccess();
```

引数

名称	説明	型
-	-	-

戻り値

説明	型
-	-

Example

```
@Override
void onAccessoryConnectSuccess() {
    mComment.setText("Accessory Connect Success.");

    mComment.setTextColor(Color.BLUE);
}
```

PrinterExtFragment.java / CashDrawerExtFragment.java / BarcodeReaderExtFragment.java / CombinationExtFragment.java を参照ください。

15.16 onAccessoryConnectFailure メソッド

アクセサリ（Bluetooth / USB）切断状態でのアクセサリ再接続失敗を通知します。* *Override*

宣言

```
void onAccessoryConnectFailure();
```

引数

名称	説明	型
-	-	-

戻り値

説明	型
-	-

Example

```
@Override
void onAccessoryConnectFailure() {
    mComment.setText("Accessory Connect Failure.");

    mComment.setTextColor(Color.RED);
}
```

PrinterExtFragment.java / CashDrawerExtFragment.java / BarcodeReaderExtFragment.java / CombinationExtFragment.java を参照ください。

15.17 onAccessoryDisconnect メソッド

アクセサリ（Bluetooth / USB）切断を通知します。* *Override*

宣言

```
void onAccessoryDisconnect();
```

引数

名称	説明	型
-	-	-

戻り値

説明	型
-	-

Example

```
@Override
void onAccessoryDisconnect() {
    mComment.setText("Accessory Disconnect.");

    mComment.setTextColor(Color.RED);
}
```

PrinterExtFragment.java / CashDrawerExtFragment.java / BarcodeReaderExtFragment.java / CombinationExtFragment.java を参照ください。

15.18 onStatusUpdate メソッド

ステータスアップデートを通知します。* *Override*

宣言

```
void onStatusUpdate(String status);
```

引数

名称	説明	型
status	<p>アップデートステータス</p> <p>自動ステータス（※）の3バイト目～6バイト目を16進数文字列化したもの（例："28000000"）</p> <p>-ステータス通知タイミング-</p> <ul style="list-style-type: none"> ステータス変化時 ステータス未変化時、5分毎 <p>※STAR Line Mode コマンド仕様書もしくは StarPRNT コマンド仕様書参照ください</p>	String

戻り値

説明	型
-	-

Example

```
@Override
void onStatusUpdate(String status) {
    mComment.setText(status);

    mComment.setTextColor(Color.GREEN);
}
```

AllReceiptsExtFragment.java を参照ください。

16 StarBluetoothManagerFactory クラス (starioextension.jar)

エミュレーションを指定することで StarBluetoothManager オブジェクトを取得するためのクラスです。

メソッド

名称	説明
getManager	StarBluetoothManager クラスのインスタンスを取得します。

16.1 getManager メソッド

引数で渡すエミュレーション用の StarBluetoothManager クラスのオブジェクトを取得します。

宣言

```
public static StarBluetoothManager getManager(String portName, String portSettings, int
ioTimeoutMillis, Emulation emulation) throws StarIOPortException;
```

引数

名称	説明	型
portName	プリンタポート名 StarIOPort クラスの getPort メソッド の portName と同じです。	String
portSettings	ポートセッティング StarIOPort クラスの getPort メソッド の portSetting と同じです。	String
ioTimeoutMillis	内部制御および API のタイムアウト値	int
emulation	エミュレーション	com.starmicronics.starioextension. StarIoExt.Emulation

戻り値

説明	型
StarIoExtManager オブジェクト	com.starmicronics.stario.StarBluetoothManager

Bluetooth 設定変更の手順は [StarBluetoothManager を使った Bluetooth 設定変更フロー](#)を参照ください。

各モデルに対して適用すべき Emulation コンスタント定数は [Emulation コンスタント](#)を参照ください。

17 CloudServices クラス (smcloudservices.aar)

メソッド

名称	説明
showRegistrationView	スター精密クラウドサービス登録のビューを表示します。
isRegistered	アプリケーションがスター精密クラウドサービスに登録されているか確認します。

17.1 showRegistrationView メソッド

スター精密クラウドサービス登録のビューを表示します。



宣言

```
public synchronized static void showRegistrationView(FragmentManager fragmentManager,
DialogInterface.OnDismissListener listener);
```

引数

名称	説明	型
fragmentManager	フラグメントマネージャ	android.support.v4.app.FragmentManager
listener	ビューが閉じられた際の Listener オブジェクト	android.content.DialogInterface.OnDismissListener

戻り値

説明	型
-	-

Example

```
@Override
public void onItemClick(AdapterView<?> parent, View view, int position, long id) {
    super.onItemClick(parent, view, position, id);

    ...

    CloudServices.showRegistrationView(getChildFragmentManager(), new DialogInterface.OnDismissListener() {
        @Override
        public void onDismiss(DialogInterface dialog) {

            ...

        }
    });

    ...
}
```

AllReceiptsFragment.java を参照ください。

17.2 isRegistered メソッド

アプリケーションがスター精密クラウドサービスに登録されているか確認します。

宣言

```
public synchronized static boolean isRegistered(Context context);
```

引数

名称	説明	型
context	コンテキスト	android.content.Context

戻り値

説明	型
リザルト <ul style="list-style-type: none"> true ... 登録されている false ... 登録されていない 	boolean

Example

```
private void updateList() {
    ...

    boolean isRegistered = CloudServices.isRegistered(getContext());

    ...

    if (isRegistered) {
        ...
    }
    else {
        ...
    }

    ...
}
```

AllReceiptsFragment.java を参照ください。

18 AllReceipts クラス (smcloudservices.aar)

メソッド

名称	説明
uploadBitmap	ビットマップをスター精密クラウドサービスへアップロードします。
uploadData	データをスター精密クラウドサービスへアップロードします。
updateStatus	スター精密クラウドサービス上のデバイスステータス情報をアップデートします。
generateAllReceipts	AllReceipts™利用のための印刷データを生成します。

18.1 モデル : AllReceipts class メソッド

各モデルにおいて機能するメソッドは以下のとおりです。

Function	Method	mC-Print2	mC-Print3	mPOP	FVP10	TSP100	TSP650II	TSP700II	TSP800II	SM-S210i	SM-S220i	SM-S230i	SM-T300i/T300	SM-T400i	BSC10	SM-S210i StarPRNT	SM-S220i StarPRNT	SM-S230i StarPRNT	SM-T300i/T300 StarPRNT	SM-T400i StarPRNT	SM-L200	SM-L300	SP700
Upload	uploadBitmap	✓	✓	✓	✓	✓	✓	✓	✓	-	-	-	-	-	✓	✓	✓	✓	✓	✓	✓	✓	-
	uploadData	✓	✓	✓	✓	-	✓	✓	✓	-	-	-	-	-	✓	✓	✓	✓	✓	✓	✓	✓	-
Update	updateStatus	✓	✓	✓	✓	✓	✓	✓	✓	-	-	-	-	-	✓	✓	✓	✓	✓	✓	✓	✓	-
Generate	generateAllReceipts	✓	✓	✓	✓	✓	✓	✓	✓	-	-	-	-	-	✓	✓	✓	✓	✓	✓	✓	✓	-

- : 対象外

18.2 uploadBitmap メソッド

ビットマップ(Bitmap オブジェクト)をスター精密クラウドサービスへアップロードします。

宣言

```
public synchronized static String uploadBitmap(Context context, Bitmap bitmap, RequestCallback callback);
```

引数

名称	説明	型
context	コンテキスト	android.content.Context
bitmap	アップロードするビットマップ	android.graphics.Bitmap
callback	アップロードが完了した際、アップロード結果を以下の引数によって捉えることができます。 -statusCode- • HTTP ステータスコード -error- • アップロードに失敗した際のエラー情報	com.starmicronics.cloudservices. .RequestCallback

戻り値

説明	型
アップロード URL	String

Example

```
public static byte[] createData(Context context, Emulation emulation, Bitmap bitmap, RequestCallback callback) {
    String urlString = AllReceipts.uploadBitmap(context, bitmap, callback);

    ICommandBuilder builder = StarIoExt.createCommandBuilder(emulation);

    builder.beginDocument();

    builder.appendBitmap(bitmap, false);

    byte[] data = AllReceipts.generateAllReceipts(context, urlString, emulation, true, true);

    builder.appendRaw(data);

    builder.appendCutPaper(CutPaperAction.PartialCutWithFeed);

    builder.endDocument();

    return builder.getCommands();
}
```

AllReceiptsFunctions.java を参照ください。

18.3 uploadData メソッド

データ (byte[] オブジェクト) をスター精密クラウドサービスへアップロードします。

宣言

```
public synchronized static String uploadData(Context context, byte[] data, Emulation emulation,
CharacterCode characterCode, int width, RequestCallback callback);
```

引数

名称	説明	型
context	コンテキスト	android.content.Context
data	アップロードするデータ	byte[]
emulation	エミュレーション指定 <ul style="list-style-type: none"> StarPRNT ... StarPRNT エミュレーション StarLine ... STAR Line Mode エミュレーション StarGraphic ... STAR Graphic Mode エミュレーション EscPos ... ESC/POS エミュレーション EscPosMobile ... ESC/POS Mobile エミュレーション StarDotImpact ... STAR ドットインパクトエミュレーション 	com.starmicronics.starioextension.StarloExt.Emulation
characterCode	文字コード指定 <ul style="list-style-type: none"> Standard ... スタンダード Japanese ... 日本語 SimplifiedChinese ... 簡体字中国語 TraditionalChinese ... 繁体字中国語 	com.starmicronics.starioextension.StarloExt.CharacterCode
width	印刷領域幅 (ドット数単位)	int
callback	アップロードが完了した際、アップロード結果を以下の引数によって捉えることができます。 -statusCode- <ul style="list-style-type: none"> HTTP ステータスコード -error- <ul style="list-style-type: none"> アップロードに失敗した際のエラー情報 	com.starmicronics.cloudservices.RequestCallback

戻り値

説明	型
アップロード URL	String

各モデルに対して適用すべき Emulation コンスタント定数は [Emulation コンスタント](#) を参照ください。

Example

```
public static byte[] createData(Context context, Emulation emulation, CharacterCode characterCode, byte[] data,
RequestCallback callback) {
    ICommandBuilder uploadDataBuilder = StarIoExt.createCommandBuilder(emulation);

    uploadDataBuilder.beginDocument();

    uploadDataBuilder.append(data);

    uploadDataBuilder.appendCutPaper(CutPaperAction.PartialCutWithFeed);

    uploadDataBuilder.endDocument();

    byte[] receiptData = uploadDataBuilder.getCommands();

    String urlString = AllReceipts.uploadData(context, receiptData, emulation, characterCode, width, callback);

    ICommandBuilder printDataBuilder = StarIoExt.createCommandBuilder(emulation);

    printDataBuilder.beginDocument();

    printDataBuilder.append(data);

    byte[] allReceiptsData = AllReceipts.generateAllReceipts(context, urlString, emulation, true, true);

    printDataBuilder.appendRaw(allReceiptsData);

    printDataBuilder.appendCutPaper(CutPaperAction.PartialCutWithFeed);

    printDataBuilder.endDocument();

    return printDataBuilder.getCommands();
}
```

AllReceiptsFunctions.java を参照ください。

18.4 updateStatus メソッド

スター精密クラウドサービス上のデバイスステータス情報をアップデートします。

宣言

```
public synchronized static void updateStatus(Context context, String status, RequestCallback callback);
```

引数

名称	説明	型
context	コンテキスト	android.content.Context
status	アップデートステータス 自動ステータス（※）の3バイト目～6バイト目を16進数文字列化したもの（例：“28000000”） ※STAR Line Mode コマンド仕様書もしくはStarPRNT コマンド仕様書参照ください	String
callback	アップデートが完了した際、アップデート結果を以下の引数によって捉えることができます。 -statusCode- • HTTP ステータスコード -error- • アップデートに失敗した際のエラー情報	com.starmicronics.cloudservices.RequestCallback

戻り値

説明	型
-	-

Example

```
@Override
public void onStatusUpdate(StarloExtManager manager, String status) {
    AllReceipts.updateStatus(getContext(), status, new RequestCallback() {
        @Override
        public void onRequestResult(int statusCode, RequestError requestError) {
            String message;

            if (requestError != null) {
                message = requestError.getMessage();
            }
            else {
                message = "Status Code : " + statusCode;
            }

            Toast.makeText(getContext(), message, Toast.LENGTH_SHORT).show();
        }
    });
}
```

AllReceiptsExtFragment.java を参照ください。

18.5 generateAllReceipts メソッド

AllReceipts™利用のための印刷データを生成します。

宣言

```
public synchronized static byte[] generateAllReceipts(Context context, String urlString, Emulation emulation, boolean info, boolean qrCode);
public synchronized static byte[] generateAllReceipts(Context context, String urlString, Emulation emulation, boolean info, boolean qrCode, int width);
```

引数

名称	説明	型
context	コンテキスト	android.content.Context
urlString	アップロード URL	String
emulation	エミュレーション <ul style="list-style-type: none"> StarPRNT ... StarPRNT エミュレーション StarLine ... STAR Line Mode エミュレーション StarGraphic ... STAR Graphic Mode エミュレーション EscPos ... ESC/POS エミュレーション EscPosMobile ... ESC/POS Mobile エミュレーション StarDotImpact ... STAR ドットインパクトエミュレーション 	com.starmicronics.starioextension.StarIoExt.Emulation
info	インフォメーションロゴ <ul style="list-style-type: none"> true ... 印刷データに含める false ... 印刷データに含めない 	boolean
qrCode	QR コード <ul style="list-style-type: none"> true ... 印刷データに含める false ... 印刷データに含めない 	boolean
width	印刷領域幅 (ドット数単位)	int

戻り値

説明	型
生成された印刷データ	byte[]

各モデルに対して適用すべき Emulation コンスタント定数は [Emulation コンスタント](#) を参照ください。

Example

```
public static byte[] createData(Context context, Emulation emulation, Bitmap bitmap, int width, RequestCallback callback) {
    String urlString = AllReceipts.uploadBitmap(context, bitmap, callback);

    ICommandBuilder builder = StarIoExt.createCommandBuilder(emulation);

    builder.beginDocument();

    builder.appendBitmap(bitmap, false);

    byte[] data;

    if (emulation == Emulation.StarGraphic) {
        data = AllReceipts.generateAllReceipts(context, urlString, emulation, true, true, width); // Support to
        centering in Star Graphic.
    }
    else {
        data = AllReceipts.generateAllReceipts(context, urlString, emulation, true, true); // Non support to
        centering in Star Graphic.
    }

    builder.appendRaw(data);

    builder.appendCutPaper(CutPaperAction.PartialCutWithFeed);

    builder.endDocument();

    return builder.getCommands();
}
```

AllReceiptsFunctions.java を参照ください。

19 RequestCallback クラス (smcloudservices.aar)

メソッド

名称	説明
onRequestResult	非同期実行の完了を通知します。 * <i>Override</i>

19.1 onRequestResult メソッド

非同期実行の完了を通知します。 * *Override*

宣言

```
public abstract void onRequestResult(int statusCode, RequestError error);
```

引数

名称	説明	型
statusCode	HTTP ステータスコード	int
Error	データアップロードやステータスアップデートに失敗した際のエラー情報	com.starmicronics.cloudservices. .RequestError

戻り値

説明	型
-	-

Example

```
private final RequestCallback mAllReceiptsCallback = new RequestCallback() {
    @Override
    public void onRequestResult(int statusCode, RequestError error) {
        String message;

        if (error != null) {
            message = error.getMessage();
        }
        else {
            message = "Status Code : " + statusCode;
        }

        Toast.makeText(getContext(), message, Toast.LENGTH_SHORT).show();
    }
};
```

AllReceiptsFragment.java / *AllReceiptsExtFragment.java* を参照ください。

20 RequestError クラス (smcloudservices.aar)

メソッド

名称	説明
getMessage	非同期実行中に発生したエラーに関するメッセージを取得します。
getInternalException	非同期実行中に発生した例外を取得します。(デバッグ用)

20.1 getMessage メソッド

非同期実行中に発生したエラーに関するメッセージを取得します。

宣言

```
public String getMessage();
```

引数

名称	説明	型
-	-	-

戻り値

説明	型
非同期実行中に発生したエラーに関するメッセージ	String

Example

```
private final RequestCallback mAllReceiptsCallback = new RequestCallback() {
    @Override
    public void onRequestResult(int statusCode, RequestError error) {
        String message;

        if (error != null) {
            message = error.getMessage();
        }
        else {
            message = "Status Code : " + statusCode;
        }

        Toast.makeText(getContext(), message, Toast.LENGTH_SHORT).show();
    }
};
```

AllReceiptsFragment.java / *AllReceiptsExtFragment.java* を参照ください。

20.2 getInternalException メソッド

非同期実行中に発生した例外を取得します。(デバッグ用)

宣言

```
public Exception getInternalException();
```

引数

名称	説明	型
-	-	-

戻り値

説明	型
非同期実行中に発生した例外	Exception

Example

```
private final RequestCallback mAllReceiptsCallback = new RequestCallback() {
    @Override
    public void onRequestResult(int statusCode, RequestError error) {
        String message;

        if (error != null) {
            message = error.getMessage();

            Exception exception = error.getInternalException();

            if (exception != null) {
                message = exception.getMessage();
            }
        }
        else {
            message = "Status Code : " + statusCode;
        }

        Toast.makeText(getContext(), message, Toast.LENGTH_SHORT).show();
    }
};
```

AllReceiptsFragment.java / AllReceiptsExtFragment.java を参照ください。

Appendix A. AllReceipt™ 利用方法案内 (配布用)

無料の電子レシートサービス『AllReceipts™』について、詳しくは下記 URL をご参照ください。

www.allreceipts.com

下記サイトにて店舗アカウントを登録いただくだけで、その日からサービスの利用が可能です。

<<スター精密クラウドサービス 店舗アカウント登録サイト>>

www.starmicronicscloud.com

1) デバイスの登録

ご登録いただいたスター精密クラウドサービスのアカウントでデバイスを登録します。

デバイスの登録をすると、スター精密クラウドサービスのダッシュボードから登録したデバイスの管理が可能となります。

2) あらかじめご登録いただいた、スター精密クラウドサービスアカウントのユーザー名 (Eメール) とパスワードを入力して [登録] をタップします。



3) デバイスの登録が成功すると、接続したスター精密クラウドサービスのアカウント情報が表示されます。

