

# StarPRNT iOS SDK ユーザーズマニュアル

2019 年 09 月 04 日

## 要約

---

このマニュアルは StarPRNT iOS 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 04 月 01 日	発行
2016 年 08 月 11 日	誤記訂正
2016 年 09 月 12 日	SMCSAllReceipts クラスに uploadData メソッドを追加
2017 年 04 月 17 日	ブラックマーク、ページモードに対応
2017 年 08 月 16 日	StarIO フレームワークに含まれるクラスの API リファレンスを追加 SMBluetoothManagerFactory 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	ISCBBuilder インターフェイスクラスに 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 にリトライオプションを追加 StarPrinterStatus 構造体に connectedInterface メンバを追加 ISCBBuilder インターフェイスクラスの appendSound メソッドに driveTime パラメータと delayTime パラメータを追加 ISCBBuilder インターフェイスクラスに appendTopMargin メソッドを追加 ISCBBuilder インターフェイスクラスに appendPrintableArea メソッドを追加 USB プリンタのデバッグ方法を Xcode 9 のものに変更 getPort メソッドと searchPrinter メソッドについて、Bluetooth, USB(Lightning)両対応モデル特有の注意事項を追加。 readPort の互換性に関する注意を追加 Swift 対応 API の情報を追加
2018 年 06 月 29 日	APS シリーズのサポート終了
2018 年 10 月 19 日	StarloExtManager クラスに connectAsync メソッドを追加 StarloExtManagerDelegate プロトコルに didConnectPort メソッドを追加 StarloExtManagerDelegate プロトコルに didFailToConnectPort メソッドを追加
2018 年 11 月 20 日	Zebra Technologies (Symbol) 社 DS9208 対応追加 mC-Sound 対応追加 StarloExt クラスに createMelodySpeakerCommandBuilder メソッドを追加 StarloExt クラスに createMelodySpeakerConnectParser メソッドを追加 ISMCBBuilder インターフェイスを追加 StarloExtMelodySpeakerModel コンスタントを追加 SMCBMelodySpeakerSoundStorageArea コンスタントを追加 SMSoundSetting クラスを追加
2019 年 03 月 07 日	Objective-C Bridging Header の情報を追加 ISCBBuilder インターフェイスに appendCjkUnifiedIdeographFont メソッドを追加 SCBBarcodeWidth コンスタントに ExtMode を追加
2019 年 09 月 04 日	Bluetooth Always Usage Description の設定方法を追加 Objective-C Bridging Header の情報を追加 以下メソッドのエラーハンドリングに関する情報の追加 getPort beginCheckedBlock endCheckedBlock writePort readPort getParsedStatus getFirmwareInformation searchPrinter disconnect MCP30 に関する注意事項を追加

## 本書に関して

本マニュアルは、StarIO と Star POS プリンタが通信を行う、iOS アプリケーションの作成方法を解説しています。

また、このマニュアルは、アプリケーション・システム開発者を対象に作成しており、利用者は Swift もしくは Objective-C 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:

- iPad, iPhone, iPod, iPod touch, Retina は、米国および他の国々で登録された Apple Inc. の商標です。iPad Air, iPad mini は、Apple Inc. の商標です。“iPhone” の商標は、アイホン株式会社のライセンスにもとづき使用されています。IOS は、米国およびその他の国における Cisco 社の商標または登録商標であり、ライセンスに基づき使用されています。
- 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	10
1.1	対応プリンタ	10
1.2	対応外部機器	12
1.3	プリンタを iOS デバイスに接続する	14
1.4	プリンタ設定用ソフトウェア	18
2	iOS StarPRNT プロジェクト構成方法	21
2.1	iOS StarPRNT SDK をプロジェクトに追加	21
2.2	必要なフレームワークをプロジェクトに追加	21
2.3	Information Property List へ項目を設定	21
2.3.1	(Bluetooth プリンタを使用する場合)Supported external accessory protocols の設定	21
2.3.2	(iOS13 以上で Bluetooth Low Energy プリンタを使用する場合)Bluetooth Always Usage Description の設定	21
2.4	(Swift で開発する場合)Objective-C Bridging Header の追加	23
3	SMPort クラス (StarIO.framework)	24
3.1	SMPort クラスを使った印刷処理フロー	25
3.2	モデル: SMPort クラス	28
3.3	getPort メソッド	29
3.3.1	Objective-C 用	29
3.3.2	Swift 用	30
3.4	releasePort メソッド	35
3.5	searchPrinter メソッド	36
3.5.1	Objective-C 用	36
3.5.2	Swift 用	38
3.6	writePort メソッド	40
3.6.1	Objective-C 用	40
3.6.2	Swift 用	41
3.7	readPort メソッド	42
3.7.1	Objective-C 用	42
3.7.2	Swift 用	43
3.8	getParsedStatus メソッド	44
3.8.1	Objective-C 用	44
3.8.2	Swift 用	46
3.9	beginCheckedBlock メソッド	47
3.9.1	Objective-C 用	47
3.9.2	Swift 用	48
3.10	endCheckedBlock メソッド	49
3.10.1	Objective-C 用	49
3.10.2	Swift 用	50
3.11	disconnect メソッド	51
3.11.1	Objective-C 用	51
3.11.2	Swift 用	52
3.12	getFirmwareInformation メソッド	53
3.12.1	Objective-C 用	53
3.12.2	Swift 用	54
3.13	StarIOVersion メソッド	55
3.14	portName プロパティ	55
3.15	portSettings プロパティ	55
3.16	timeoutMillis プロパティ	55
3.17	endCheckedBlockTimeoutMillis プロパティ	56
3.18	connected プロパティ	56
4	StarPrinterStatus 構造体 (StarIO.framework)	57

4.1	モデル: StarPrinterStatus 構造体 .....	59
5	PortInfo クラス (StarIO.framework) .....	60
5.1	portName プロパティ .....	60
5.2	macAddress プロパティ .....	60
5.3	modelName プロパティ .....	60
6	SMBluetoothManager クラス (StarIO.framework) .....	61
6.1	SMBluetoothManager を使った Bluetooth 設定変更フロー .....	62
6.2	モデル: SMBluetoothManager クラス .....	63
6.3	initWithName メソッド .....	65
6.4	open メソッド .....	65
6.5	loadSetting メソッド .....	66
6.6	close メソッド .....	66
6.7	apply メソッド .....	67
6.8	portName プロパティ .....	67
6.9	deviceType プロパティ .....	67
6.10	opened プロパティ .....	67
6.11	deviceName プロパティ .....	68
6.12	iOSPortName プロパティ .....	68
6.13	autoConnect プロパティ .....	68
6.14	security プロパティ .....	69
6.15	pinCode プロパティ .....	69
6.16	deviceNameCapability プロパティ .....	69
6.17	iOSPortNameCapability プロパティ .....	69
6.18	autoConnectCapability プロパティ .....	69
6.19	securityTypeCapability プロパティ .....	70
6.20	pinCodeCapability プロパティ .....	70
6.21	SMDeviceType コンスタント .....	71
6.22	SMBluetoothSecurity コンスタント .....	72
6.23	SMBluetoothSettingCapability コンスタント .....	73
7	StarloExt クラス (StarIO_Extension.framework) .....	74
7.1	createCommandBuilder メソッド .....	75
7.2	createDisplayCommandBuilder メソッド .....	76
7.3	createMelodySpeakerCommandBuilder メソッド .....	77
7.4	createBcrConnectParser メソッド .....	78
7.5	createDisplayConnectParser メソッド .....	79
7.6	createMelodySpeakerConnectParser メソッド .....	80
7.7	StarloExtEmulation コンスタント .....	81
7.8	StarloExtCharacterCode コンスタント .....	82
7.9	StarloExtBcrModel コンスタント .....	82
7.10	StarloExtDisplayModel コンスタント .....	82
7.11	StarloExtMelodySpeakerModel コンスタント .....	83
8	ISCBBuilder クラス (StarIO_Extension.framework) .....	84
8.1	モデル: ISCBBuilder interface メソッド .....	87
8.2	beginDocument メソッド .....	92
8.3	endDocument メソッド .....	93
8.4	appendInitialization メソッド .....	94
8.5	appendData メソッド .....	95
8.6	appendRawData メソッド .....	96
8.7	appendFontStyle メソッド .....	97
8.8	appendCodePage メソッド .....	98
8.9	appendInternational メソッド .....	100
8.10	appendLineFeed メソッド .....	102
8.11	appendUnitFeed メソッド .....	103

8.12	appendCharacterSpace メソッド	104
8.13	appendLineSpace メソッド	105
8.14	appendTopMargin メソッド	106
8.15	appendEmphasis メソッド	107
8.16	appendInvert メソッド	108
8.17	appendMultiple メソッド	109
8.18	appendUnderLine メソッド	111
8.19	appendLogo メソッド	112
8.20	appendAbsolutePosition メソッド	113
8.21	appendAlignment メソッド	114
8.22	appendHorizontalTabPosition メソッド	116
8.23	appendCutPaper メソッド	117
8.24	appendPeripheral メソッド	118
8.25	appendSound メソッド	119
8.26	appendBarcodeData メソッド	120
8.27	appendBarcodeDataWithAbsolutePosition メソッド	122
8.28	appendBarcodeDataWithAlignment メソッド	124
8.29	appendPdf417Data メソッド	127
8.30	appendPdf417DataWithAbsolutePosition メソッド	129
8.31	appendPdf417DataWithAlignment メソッド	131
8.32	appendQrCodeData メソッド	133
8.33	appendQrCodeDataWithAbsolutePosition メソッド	134
8.34	appendQrCodeDataWithAlignment メソッド	135
8.35	appendBitmap メソッド	137
8.36	appendBitmapWithAbsolutePosition メソッド	139
8.37	appendBitmapWithAlignment メソッド	141
8.38	appendBlackMark メソッド	143
8.39	beginPageMode メソッド	144
8.40	endPageMode メソッド	145
8.41	appendPageModeVerticalAbsolutePosition メソッド	146
8.42	appendPageModeRotation メソッド	147
8.43	appendPrintableArea メソッド	149
8.44	appendCjkUnifiedIdeographFont メソッド	152
8.45	commands プロパティ	154
8.46	SCBInitializationType コンスタント	155
8.47	SCBFontStyleType コンスタント	155
8.48	SCBCodePageType コンスタント	156
8.49	SCBInternationalType コンスタント	158
8.50	SCBLogoSize コンスタント	159
8.51	SCBAlignmentPosition コンスタント	159
8.52	SCBCutPaperAction コンスタント	160
8.53	SCBPeripheralChannel コンスタント	160
8.54	SCBSoundChannel コンスタント	161
8.55	SCBBarcodeSymbology コンスタント	161
8.56	SCBBarcodeWidth コンスタント	162
8.57	SCBPdf417Level コンスタント	163
8.58	SCBQrCodeModel コンスタント	163
8.59	SCBQrCodeLevel コンスタント	164
8.60	SCBBitmapConverterRotation コンスタント	164
8.61	SCBBlackMarkType コンスタント	165
8.62	SCBPrintableAreaType コンスタント	165
8.63	SCBCjkUnifiedIdeographFont コンスタント	166
9	ISDCBBuilder インターフェイス (StarIO_Extension.framework)	167

9.1	モデル : ISDCBBuilder インターフェイス .....	168
9.2	appendData メソッド .....	169
9.3	appendBackSpace メソッド .....	170
9.4	appendHorizontalTab メソッド .....	170
9.5	appendLineFeed メソッド .....	170
9.6	appendCarriageReturn メソッド .....	171
9.7	appendBitmap メソッド .....	172
9.8	appendInternational メソッド .....	173
9.9	appendCodePage メソッド .....	174
9.10	appendDeleteToEndOfLine メソッド .....	175
9.11	appendClearScreen メソッド .....	175
9.12	appendHomePosition メソッド .....	175
9.13	appendTurnOn メソッド .....	176
9.14	appendSpecifiedPosition メソッド .....	176
9.15	appendCursorMode メソッド .....	177
9.16	appendContrastMode メソッド .....	177
9.17	appendUserDefinedCharacter メソッド .....	178
9.18	appendUserDefinedDbcsCharacter メソッド .....	179
9.19	commands プロパティ .....	181
9.20	passThroughCommands プロパティ .....	181
9.21	SDCBInternationalType コンスタント .....	182
9.22	SDCBCodePageType コンスタント .....	183
9.23	SDCBCursorMode コンスタント .....	184
9.24	SDCBContrastMode コンスタント .....	184
10	ISMCBBuilder インターフェイス (StarIO_Extension.framework) .....	185
10.1	モデル : ISMCBBuilder インターフェイス .....	186
10.2	appendSoundWithSetting メソッド .....	187
10.3	appendSoundWithSound メソッド .....	190
10.4	commands プロパティ .....	193
10.5	SMCBMelodySpeakerSoundStorageArea コンスタント .....	194
11	SMSoundSetting クラス (StarIO_Extension.framework) .....	195
11.1	soundStorageArea プロパティ .....	196
11.2	soundNumber プロパティ .....	196
11.3	volume プロパティ .....	196
11.4	count プロパティ .....	196
11.5	delay プロパティ .....	197
11.6	interval プロパティ .....	197
11.7	SMSoundSettingVolume コンスタント .....	197
12	ISCPParser インターフェイス (StarIO_Extension.framework) .....	198
12.1	モデル : ISCPParser インターフェイス .....	198
12.2	createSendCommands メソッド .....	199
12.3	completionHandler プロパティ .....	201
12.4	StarIoExtParserCompletionResult コンスタント .....	201
13	ISCPConnectParser インターフェイス (StarIO_Extension.framework) .....	202
13.1	モデル : ISCPConnectParser インターフェイス .....	202
13.2	connect メソッド .....	203
14	StarPRNT iOS SDK Sample .....	204
14.1	Communication .....	204
15	StarIoExtManager クラス (StarIO_Extension.framework) .....	207
15.1	initWithType メソッド .....	208
15.2	connectAsync メソッド .....	209
15.3	connect メソッド .....	210
15.4	disconnect メソッド .....	211



15.5	port プロパティ .....	211
15.6	lock プロパティ .....	212
15.7	delegate プロパティ .....	213
15.8	printerOnlineStatus プロパティ .....	214
15.9	printerPaperStatus プロパティ .....	214
15.10	printerCoverStatus プロパティ .....	215
15.11	cashDrawerStatus プロパティ .....	215
15.12	barcodeReaderStatus プロパティ .....	216
15.13	cashDrawerOpenActiveHigh プロパティ .....	216
15.14	StarIoExtManagerType コンスタント .....	217
15.15	StarIoExtManagerPrinterStatus コンスタント .....	217
15.16	StarIoExtManagerPrinterPaperStatus コンスタント .....	218
15.17	StarIoExtManagerPrinterCoverStatus コンスタント .....	218
15.18	StarIoExtManagerCashDrawerStatus コンスタント .....	219
15.19	StarIoExtManagerBarcodeReaderStatus コンスタント .....	219
16	StarIoExtManagerDelegate プロトコル (StarIO_Extension.framework) .....	220
16.1	didConnectPort メソッド .....	221
16.2	didFailToConnectPort メソッド .....	222
16.3	didPrinterImpossible メソッド .....	223
16.4	didPrinterOnline メソッド .....	224
16.5	didPrinterOffline メソッド .....	224
16.6	didPrinterPaperReady メソッド .....	225
16.7	didPrinterPaperNearEmpty メソッド .....	225
16.8	didPrinterPaperEmpty メソッド .....	226
16.9	didPrinterCoverOpen メソッド .....	226
16.10	didPrinterCoverClose メソッド .....	227
16.11	didCashDrawerOpen メソッド .....	227
16.12	didCashDrawerClose メソッド .....	228
16.13	didBarcodeReaderImpossible メソッド .....	228
16.14	didBarcodeReaderConnect メソッド .....	229
16.15	didBarcodeReaderDisconnect メソッド .....	229
16.16	didBarcodeDataReceive メソッド .....	230
16.17	didAccessoryConnectSuccess メソッド .....	231
16.18	didAccessoryConnectFailure メソッド .....	231
16.19	didAccessoryDisconnect メソッド .....	232
16.20	didStatusUpdate メソッド .....	233
17	SMBluetoothManagerFactory クラス (StarIO_Extension.framework) .....	234
17.1	getManager メソッド .....	234
18	SMCloudServices クラス (SMCloudServices.framework) .....	235
18.1	showRegistrationView メソッド .....	235
18.2	isRegistered メソッド .....	236
19	SMCSAllReceipts クラス (SMCloudServices.framework) .....	237
19.1	モデル : SMCSAllReceipts class メソッド .....	238
19.2	uploadBitmap メソッド .....	239
19.3	uploadData メソッド .....	240
19.4	updateStatus メソッド .....	242
19.5	generateAllReceipts メソッド .....	243
20	非推奨 API .....	245
20.1	SMPort クラス .....	245
20.1.1	getPort メソッド .....	245
20.1.2	searchPrinter メソッド .....	246
20.1.3	writePort メソッド .....	247
20.1.4	readPort メソッド .....	248



---

20.1.5	getParsedStatus メソッド .....	249
20.1.6	beginCheckedBlock メソッド .....	250
20.1.7	endCheckedBlock メソッド .....	251
20.1.8	disconnect メソッド .....	252
20.1.9	getFirmwareInformation メソッド .....	253
Appendix A.	AllReceipt™ 利用方法案内 (配布用) .....	254

## 1 はじめに

### 1.1 対応プリンタ

	対応プリンタ	ファームウェアバージョン	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	StarPRNT
	FVP10	1.2 以降	Bluetooth Ethernet	StarLine
	TSP100IIIW	1.0 以降	Wireless LAN	StarGraphic
	TSP100IIILAN	1.0 以降	Ethernet	StarGraphic
	TSP100IIIBI	1.0 以降	Bluetooth	StarGraphic
	TSP100IIIU	1.0 以降	USB	StarGraphic
	TSP100ECO	1.0 以降	USB *1	StarGraphic
	TSP100U	1.3 以降	USB *1	StarGraphic
	TSP100GT	1.0 以降	USB *1	StarGraphic
	TSP100LAN	2.0 以降	Ethernet	StarGraphic
	TSP650II	1.0 以降	Bluetooth Ethernet	StarLine
	TSP700II	4.0 以降	Bluetooth	StarLine
		3.0 以降	Ethernet	
	TSP800II	2.0 以降	Bluetooth Ethernet	StarLine
		1.2 以降		
	BSC10 LAN	1.0 以降	Ethernet	ESC/POS
	SP700	4.0 以降	Bluetooth(欧米のみ)	StarDotImpact
		3.0 以降	Ethernet	
モバイル プリンタ	SM-S210i	2.4 以降	Bluetooth	StarPRNT *2 / EscPosMobile
	SM-S220i	2.0 以降	Bluetooth	StarPRNT *2 / EscPosMobile
	SM-S230i	1.0 以降	Bluetooth	StarPRNT / EscPosMobile
	SM-T300i	2.4 以降	Bluetooth	StarPRNT *2 / EscPosMobile

対応プリンタ	ファームウェアバージョン	対応インターフェイス	対応エミュレーション
SM-T300	1.1 以降	Wireless LAN	EscPosMobile
SM-T400i	2.4 以降	Bluetooth	StarPRNT *2 / EscPosMobile
SM-L200	1.0 以降	Bluetooth Low Energy	StarPRNT / StarPRNTL *3
SM-L300	1.1 以降	Bluetooth Low Energy	StarPRNT / StarPRNTL

\*1 : Apple AirMac Express が必要です。

\*2 : StarPRNT 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 3.0 以降

\*3 : 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

## 1.2 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
DS9208	Zebra Technologies (Symbol) 社 バーコードリーダー
SCD222U	カスタマーディスプレイ
mC-Sound	メロディスピーカー

外部機器	mC-Print2	mC-Print3	mPOP	FVP40	TSP100	TSP650H	TSP700H	TSP800H	SM-S240i	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
BCR-POP1	✓	✓	✓	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
DS9208	✓	✓	✓	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
SCD222U	✓	✓	✓	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
mC-Sound	-	*1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

- : 対象外

\*1: ファームウェアバージョン 1.3 以降が必要です。



## 1.3 プリンタを iOS 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。

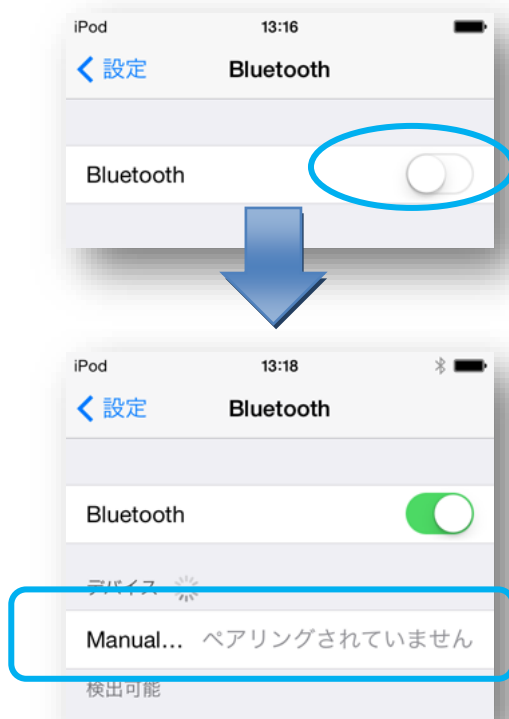
Bluetooth デバイス名の変更等、Star POS デバイスの LAN/Bluetooth 設定値は専用のソフトウェアを使用して変更することができます。各プリンタで使用するソフトウェアは[プリンタ設定用ソフトウェア](#)を参照ください。

### ◆ペアリング

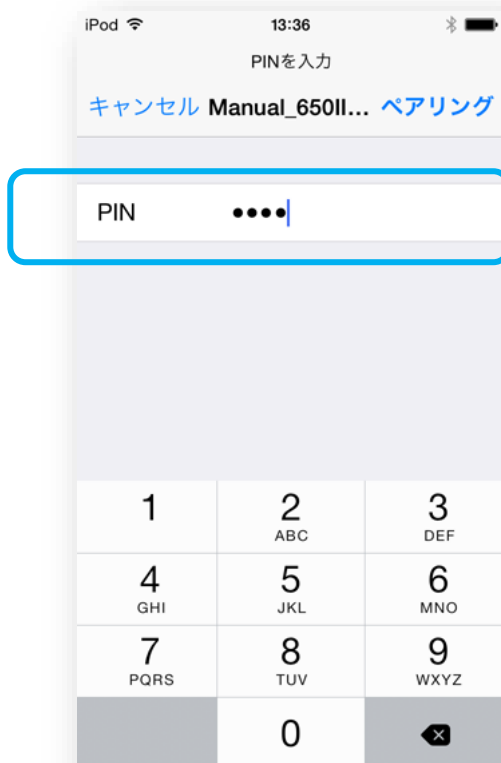
1. Star POS プリンタを、ペアリングを行う iOS デバイスと接続が可能な範囲に設置して電源を投入します。
2. iOS の[設定]より、[Bluetooth]をタップします。



3. [Bluetooth]を ” オン” に設定すると、iOS デバイスとペアリングが可能な Bluetooth デバイスの検索を行い、表示します。ペアリングを行う Star POS プリンタをタップします。



4. PIN を入力します。（Star POS プリンタの Bluetooth セキュリティが PIN Code の場合）





5. 以下の表示が確認できればペアリング完了です。



#### ◆Bluetooth 名称の変更

専用のソフトウェアを使って iOS ポート名を変更することができます。必要に応じてご使用ください。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。

iOS ポート名はペアリング実行後、以下の手順で確認できます。

「設定」 - 「一般」 - 「情報」 Bluetooth アドレスの下に表示

#### Ethernet / Wireless LAN 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 を利用できる環境の場合、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。

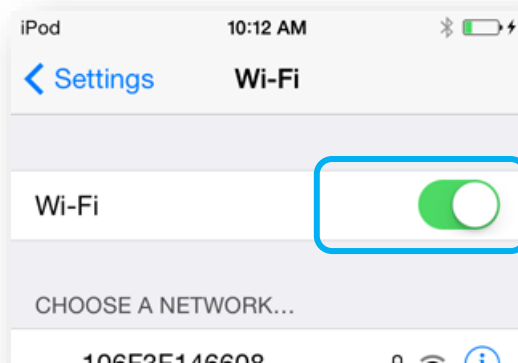
また、プリンタ本体の #9100 Multi Session を無効に設定して使用してください。設定は、FEED ボタンを押しながら電源スイッチを ON することで実行される自己印字で確認できます。

#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設定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。TSP100IIIUtility、Star Setting Utility では 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。

本ソフトウェアは TCP 通信ポート 9100、TCP 通信ポート 9101、UDP 通信ポート 22222 を使用します。

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。

### USB インターフェイス

プリンタと iOS デバイスを Lightning ケーブルで接続してください。

※使用可能なモデルは[対応プリンタ](#)を参照ください。

### ◆USB プリンタのデバッグ方法

※ 下記の表記、手順等は iOS および Xcode のバージョンアップにより、変更される可能性があります。  
本内容は 2018/4/13 時点で、macOS 10.13.4, Xcode 9.3, iOS 11.3 を使用して確認しています。

最新の情報は、Apple 社のサイトをご参照ください。

USB プリンタを使用する場合、iOS デバイスの Lightning ポートはプリンタとの接続に使用されるため、Lightning ケーブル経由でのアプリのインストールやデバッグができません。しかし、Xcode 9 と iOS 11 ではワイヤレスデバッグを使用する事が可能です。

### <セットアップ>

- \* iOS デバイスと mac が同一ネットワークに接続されている必要があります。
- 1. iOS デバイスの“設定”アプリより、iOS デバイスのパスコードを設定します。
- 2. iOS デバイスと mac を Lightning ケーブルで接続します。
- 3. Xcode を起動し、“Window”-“Devices and Simulators”を選択します。
- 4. 左ペインで使用する iOS デバイスを選択し、“connect via network”にチェックを入れます。

これ以降、Lightning ケーブルで接続していない状態でも Xcode 上にネットワーク上の iOS デバイスが表示され、アプリのインストールとデバッグが可能になります。

## 1.4 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。

### 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。

## 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。

## 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 iOS StarPRNT プロジェクト構成方法

### 2.1 iOS StarPRNT SDK をプロジェクトに追加

- “Build Phases”タブの”Link Binary With Libraries”に StarIO.framework を追加する
- “Build Phases”タブの”Link Binary With Libraries”に StarIO\_Extension.framework を追加する
- “Build Phases”タブの”Link Binary With Libraries”に SMCloudServices.framework を追加する
- “Build Phases”タブの”Copy Bundle Resources”に SMCloudServices.framework フォルダ配下にある SMCloudServicesResources.bundle を追加する

-SMCloudServices.framework 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。

### 2.2 必要なフレームワークをプロジェクトに追加

- CoreBluetooth.framework
- ExternalAccessory.framework

### 2.3 Information Property List へ項目を設定

#### 2.3.1 (Bluetooth プリンタを使用する場合)Supported external accessory protocols の設定

1. Information Property List (デフォルトでは"Info.plist") を選択します。
2. Key "Supported external accessory protocols"を追加します。
3. 項目名左側の▽をクリックして表示される"Item 0"の[Value]に "jp.star-m.starpro" を設定します。

**※Bluetooth プリンタを使用しない場合は、この設定を行わないでください。**

#### 2.3.2 (iOS13 以上で Bluetooth Low Energy プリンタを使用する場合)Bluetooth Always Usage Description の設定

1. Information Property List (デフォルトでは"Info.plist") を選択します。
2. Key "Privacy – Bluetooth Always Usage Description"を追加します。
3. Value に Bluetooth を利用する理由を (例 "Use Bluetooth for communication with the printer.") を設定します。
4. iOS13 以上で Bluetooth Low Energy にてプリンタと通信するとき、Bluetooth へのアクセス許可を求めるダイアログが表示されます。その際 Value に設定した文字列が Bluetooth を利用する理由として表示されます。

Swift SDK

SMCloudServicesResources.bundle

SMCloudServices.framework

StarIO\_Extension.framework

StarIO.framework

CoreBluetooth.framework

Swift SDK

AllReceiptsExtViewController.swift

AllReceiptsViewController.swift

ApiController.swift

AppDelegate.swift

Assets.xcassets

BarcodeReaderE...iewController.swift

BlackMarkPasteViewController.swift

BlackMarkViewController.swift

CashDrawerExtViewController.swift

CashDrawerViewController.swift

CombinationExtViewController.swift

CombinationViewController.swift

CommonViewController.swift

Communications

DeviceStatusViewController.swift

DisplayExtViewController.swift

DisplayViewController.swift

Functions

Info.plist

LaunchScreen.storyboard

Swift SDK > Swift SDK > Info.plist > No Selection

Key	Type	Value
Information Property List	Dictionary	(24 items)
App Category	String	
Localization native development region	String	en
Bundle display name	String	StarPRNT SDK
Executable file	String	\$(EXECUTABLE_NAME)
Bundle identifier	String	\$(PRODUCT_BUNDLE_IDENTIFIER)
InfoDictionary version	String	6.0
Bundle name	String	\$(PRODUCT_NAME)
Bundle OS Type code	String	APPL
Bundle versions string, short	String	5.11.0
Bundle creator OS Type code	String	????
Bundle version	String	1
Application requires iPhone environment	Boolean	YES
App Transport Security Settings	Dictionary	(1 item)
Privacy - Bluetooth Always Usage Description	String	Use Bluetooth for communication with the printer.
Privacy - Bluetooth Peripheral Usage Description	String	Use Bluetooth for communication with the printer.
Privacy - Camera Usage Description	String	Use a Photo file as print data.
Privacy - Photo Library Usage Description	String	Use a Photo file as print data.
Application supports iTunes file sharing	Boolean	YES
Launch screen interface file base name	String	LaunchScreen
Main storyboard file base name	String	Main
Required device capabilities	Array	(1 item)
Supported external accessory protocols	Array	(1 item)
Item 0	String	jp.star-m.starpro
Supported interface orientations	Array	(4 items)
Supported interface orientations (iPad)	Array	(4 items)



## 2.4 (Swift で開発する場合)Objective-C Bridging Header の追加

iOS StarPRNT SDK で提供するフレームワークは Objective-C で作成されています。

Swift からアクセスする場合は、プロジェクトに Objective-C Bridging-Header を追加する必要があります。

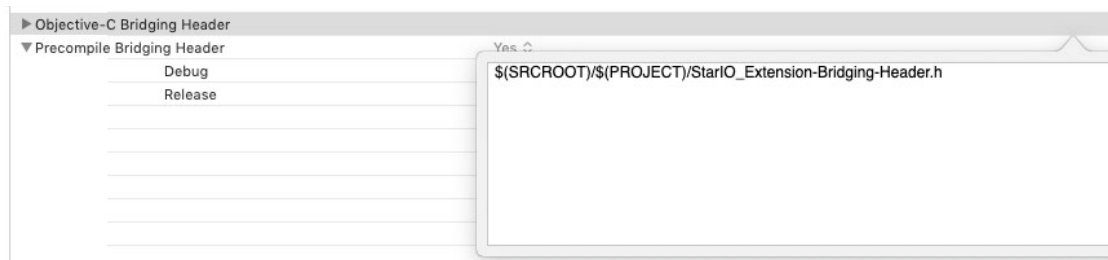
1. Xcode メニューの “File” – “New” – “File...” を選択します。
2. “Header File” を選択し、“Next” をクリックします。
3. 任意の名前を入力し、“Create” をクリックして空のヘッダファイルを作成します。  
以下の例では、ヘッダファイルの名前を “StarIO\_Extension-Bridging-Header.h” とします。
4. 3 で作成したヘッダファイルに以下の import 文を記載します。

```
#import <StarIO/SMPortSwift.h>

#import <StarIO_Extension/StarIoExt.h>
#import <StarIO_Extension/StarIoExtManager.h>
#import <StarIO_Extension/SMBluetoothManagerFactory.h>
#import <StarIO_Extension/SMSoundSetting.h>

// SMCloudServices.framework を使用する場合は以下も追加する
#import <SMCloudServices/SMCloudServices.h>
#import <SMCloudServices/SMCSAllReceipts.h>
```

5. Build Settings の Objective-C Bridging Header 項目に 3 で作成したヘッダファイルへのパスを指定します。



6. 上記手順により、Swift から各フレームワークへのアクセスが可能になります。

### 3 SMPort クラス (StarIO.framework)

SMPort 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

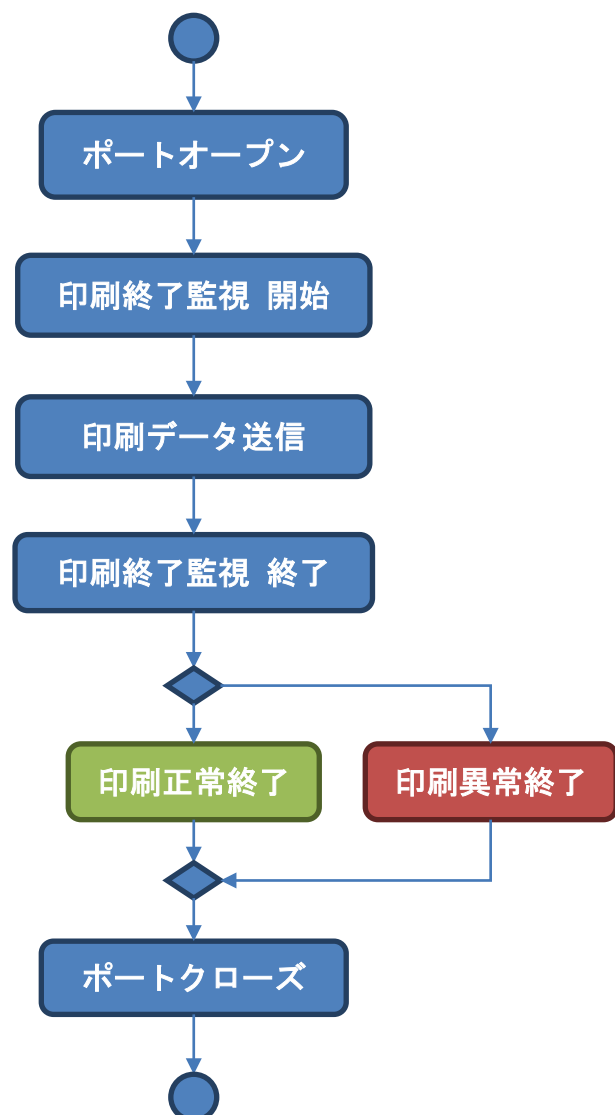
名称	説明
getPort	SMPort オブジェクトを生成し、指定されたポートへの接続をオープンします。
releasePort	指定されたポートへの接続をクローズし、SMPort オブジェクトを破棄します。
searchPrinter	iOS デバイスと接続できるプリンタを検索します。
writePort	プリンタにデータを書き込みます。
readPort	プリンタからデータを読み込みます。
getParsedStatus	プリンタのステータスを取得します。
beginCheckedBlock	印刷終了監視を開始します。
endCheckedBlock	印刷終了監視を終了します。
disconnect	指定された Bluetooth プリンタとのコネクションを切断します。
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。
StarIOVersion	StarIO ライブラリのバージョン番号を取得します。

#### プロパティ

名称	説明
portName	プリンタと通信するためのポートを特定する文字列
portSettings	プリンタとの接続設定情報
timeoutMillis	内部制御および API のタイムアウト値（ミリ秒指定）
endCheckedBlockTimeoutMillis	endCheckedBlock メソッドのタイムアウト値（ミリ秒指定）
Connected	指定された Bluetooth/Bluetooth Low Energy プリンタの iOS デバイスとの接続状態

### 3.1 SMPort クラスを使った印刷処理フロー

SMPort クラスを利用し、以下の手順に沿って印刷を行います。



## Objective-C

```
unsigned char command[] = {0x41, 0x42, 0x43, 0x44, 0x1B, 0x7A, 0x00, 0x1B, 0x64, 0x02};

StarPrinterStatus_2 printerStatus;
SMPort *port = nil;

while (YES) {
    NSError *error = nil;

    // ポートオープン
    port = [SMPort getPort: "BT:Star Micronics" :@"":10000 :&error];

    if (port == nil) { break; /* エラー発生 */ }

    // 印刷終了監視を開始
    [port beginCheckedBlock:&printerStatus :2 :&error];

    if (error != nil) { break; /* エラー発生 */ }

    if (printerStatus.offline == SM_TRUE) {
        break; // プリンタオフライン
    }

    uint32_t total = 0;

    while (total < (uint32_t) commands.length) {
        // 印刷データ送信
        uint32_t written = [port writePort:command :total : sizeof(command) - total :&error];

        if (error != nil) { break; /* エラー発生 */ }

        total += written;
    }

    // 印刷終了監視を終了
    [port endCheckedBlock:&printerStatus :2 :&error];

    if (error != nil) { break; /* エラー発生 */ }

    if (printerStatus.offline == SM_TRUE) {
        break; // プリンタオフライン
    }

    // 印刷正常終了
    break;
}

if (port != nil) {
    // ポートクローズ
    [SMPort releasePort:port];
    port = nil;
}
```

Communication.m を参照ください

## Swift

```
var command: [UInt8] = [0x41, 0x42, 0x43, 0x44, 0x1b, 0x7a, 0x00, 0x1b, 0x64, 0x02]

while true {
    var port : SMPort

    do {
        // ポートオープン
        port = try SMPort.getPort(portName: "BT:Star Micronics", portSettings: "", ioTimeoutMillis: 10000)

        defer {
            SMPort.release(port)
        }

        var printerStatus: StarPrinterStatus_2 = StarPrinterStatus_2()

        // 印刷終了監視を開始
        try port.beginCheckedBlock(starPrinterStatus: &printerStatus, level: 2)

        if printerStatus.offline == sm_true {
            break // プリンタオフライン
        }

        var total: UInt32 = 0

        while total < UInt32(command.count) {
            var written: UInt32 = 0

            // 印刷データ送信
            try port.write(writeBuffer: command, offset: total, size: UInt32(command.count) - total,
numberOfBytesWritten: &written)

            total += written
        }

        // 印刷終了監視を終了
        try port.endCheckedBlock(starPrinterStatus: &printerStatus, level: 2)

        if printerStatus.offline == sm_true {
            break // プリンタオフライン
        }

        // 印刷正常終了
        break
    }
    catch let error as NSError {
        break // エラー発生
    }
}
```

Communication.swift を参照ください

## 3.2 モデル: SMPort 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 StarPRNT	SM-T400i StarPRNT	SM-L200	SM-L300	SP700
getPort	*1	*1	*1	*1	*1	*1	*1	*1	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
releasePort	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
searchPrinter	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
writePort	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
readPort	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
getParsedStatus	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
beginCheckedBlock	✓	✓	✓	✓	✓	✓	*2	✓	✓	*4	✓	*4	✓	✓	✓	✓	✓	✓	✓	✓	✓	*2
endCheckedBlock	✓	✓	✓	✓	✓	✓	*2	✓	✓	*4	✓	*4	✓	✓	✓	✓	✓	✓	✓	✓	✓	*2
disconnect *6	✓	✓	✓	✓	-	✓	✓	✓	-	-	-	-	-	-	-	-	-	-	-	-	-	✓
getFirmwareInformation	✓	✓	✓	*5	*3*5	*5	*5	*5	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	*5
StarIOVersion	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓

- : 無視

\*1: Bluetooth でのデータタイムアウト機能をサポート。FVP10, 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。

\*2: ファームウェアバージョン 3.0 以降が必要です。

\*3: 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。

\*4: 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。

\*5: 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。

\*6: Bluetooth インターフェイスでのみ動作します。

### 3.3 getPort メソッド

SMPort オブジェクトを生成し、指定されたポートへの接続をオープンします。  
印刷の手順については [SMPort クラスを使った印刷処理フロー](#) を参照ください。

#### 3.3.1 Objective-C 用

##### 宣言

```
+ (SMPort *)getPort:(NSString *)portName
                :(NSString *)portSettings
                :(u_int32_t)ioTimeoutMillis
                :(NSError **)error;
```

##### 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	NSString *
portSettings	プリンタとの接続設定情報	NSString *
timeoutMillis	内部制御および API のタイムアウト値（ミリ秒指定）	u_int32_t
error	エラー発生時にはエラー情報が格納されます。 code プロパティにエラーコードがセットされています。	NSError **

##### 戻り値

説明	型
SMPort オブジェクト ポートオープンに失敗した場合、既にオープン済みであった場合は、 nil が返されます。	SMPort *

##### エラーコード

値	説明
SMStarIOResultCodeFailedError	何らかのエラーが発生した
SMStarIOResultCodeInUseError	プリンタから接続拒否された（他ホストが接続中など）



### 3.3.2 Swift 用

#### 宣言

```
open class func getPort(portName: String!, portSettings: String!, ioTimeoutMillis: UInt32) throws -> SMPort
```

```
open class func getPort(_ portName: String!, _ portSettings: String!, _ ioTimeoutMillis: UInt32, _ error: NSErrorPointer) -> SMPort!
```

#### 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	String
timeoutMillis	内部制御および API のタイムアウト値（ミリ秒指定）	UInt32
error	エラー発生時にはエラー情報が格納されます。 code プロパティにエラーコードがセットされています。	NSErrorPointer

#### 戻り値

説明	型
SMPort オブジェクト ポートオープンに失敗した場合、既にオープン済みであった場合は、 nil が返されます。	SMPort

#### エラーコード

値	説明
SMStarIOResultCodeFailedError	何らかのエラーが発生した
SMStarIOResultCodeInUseError	プリンタから接続拒否された（他ホストが接続中など）

## 1. portName 引数

インターフェイス	portName	説明	例
Bluetooth*1	BT:デバイス名	iOS ポート名で指定する場合	@"BT:Star Micronics"
	BT:シリアルナンバー	プリンタのシリアルナンバーで指定する場合	@"BT:2580217090400032"
Bluetooth Low Energy	BLE:デバイス名	Bluetooth デバイス名で指定する場合	@"BLE:STAR L200-00001"
	BLE:MAC アドレス	MAC アドレスで指定する場合	@"BLE:00:11:62:1b:4d:f4"
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*1	USB:デバイス名	iOS ポート名で指定してください。	@"USB:TSP100"
	USB: シリアルナンバー	プリンタのシリアルナンバーで指定する場合	@"USB:2580217090400032"

\*1 Bluetooth, USB(Lightning)両対応モデルでは、Bluetooth と USB は区別されず、同じ挙動となります。

Bluetooth 接続中に Lightning ケーブル接続を行うと、自動的に Bluetooth は切断され Lightning 経由の通信となりますが、この時 portName を"BT:"から"USB:"に変更する必要はありません。

## 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	659
mC-Print3	659
mPOP	659
FVP10	659
TSP100	659
TSP650II	659
TSP700II	659
TSP800II	659
SM-S210i	"Portable;escpos"もしくは"mini"
SM-S220i	"Portable;escpos"もしくは"mini"
SM-S230i	"Portable;escpos"もしくは"mini"
SM-T300i/T300	"Portable;escpos"もしくは"mini"
SM-T400i	"Portable;escpos"もしくは"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	659

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	“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。</p> <p>このオプションをサポートするモデルは<a href="#">モデル: StarIOPort クラス</a>を参照ください。</p> <p>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。</p> <p>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"> <li>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）</li> <li>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</li> <li>③ 上記以外の場合 : 3 秒</li> </ul> <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。</p> <p>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"> <li>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</li> <li>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）</li> <li>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</li> <li>④ 上記以外の場合 : リトライしない設定となります</li> </ul>
	“[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	“;l10000”
SM-L200 をデフォルト設定で使用する	“Portable”
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. timeoutMillis 引数

timeoutMillis 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。

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。

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、timeoutMillis 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。

### 4. 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。

### 5. SM-L シリーズの注意事項

iOS 端末で Bluetooth Low Energy にて通信する場合、プリンタとの接続に時間がかかることがあります。接続に失敗した場合は、接続が成功するまでリトライを行ってください。

また、接続時間を短縮したい場合は、プリンタとの接続が常に維持されるようアプリケーションを設計してください。

\* ただし、接続中のプリンタは他のアプリケーション、および他の端末から検出することができなくなります。

### 3.4 releasePort メソッド

指定されたポートへの接続をクローズし、SMPort オブジェクトを破棄します。

#### 宣言

+ (void) releasePort: (SMPort \*) 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 SMPort オブジェクト	SMPort *

#### 戻り値

説明	型
-	-

印刷の手順については [SMPort 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

getPort を実行した後は、必ず releasePort してから次の getPort を行ってください。releasePort をせずに次の getPort を行くと、nil が返されます。

## 3.5 searchPrinter メソッド

iOS デバイスと接続できるプリンタを検索します。

### 3.5.1 Objective-C 用

#### 宣言

```
+ (NSArray *)searchPrinter:(NSString *)target
                        :(NSError **)error;
```

#### 引数

名称	説明	型
target	すべてのインターフェイスタイプ	"ALL:"
	Bluetooth の場合	"BT:"
	Bluetooth Low Energy の場合	"BLE:"
	Ethernet / Wireless LAN の場合	"TCP:"
	USB の場合	"USB:"
error	エラー発生時にはエラー情報が格納されます。 code プロパティにエラーコードがセットされています。	NSError **

※ Bluetooth, USB(Lightning)両対応モデルでは、Bluetooth と USB は区別されません。

実際の接続方法に関わらず、target を指定しないか"BT:"を指定した場合は Bluetooth プリンタ、"USB:"を指定した場合は USB プリンタとして検出されます。

#### 戻り値

説明	型
Star プリンタの検索結果 戻り値には、PortInfo クラスのインスタンスが含まれます。 取得できる情報については <a href="#">PortInfo クラス</a> を参照ください。	NSArray *

#### エラーコード

値	説明
SMStarIOResultCodeFailedError	何らかのエラーが発生した



### Example

```
NSArray *portInfoArray;

NSError *error = nil;

portInfoArray = [SMPort searchPrinter:@"ALL:" :&error];

if (error != nil) {
    // Some error occurred.
}

if (portInfoArray == nil) {
    return;
}

for (PortInfo *portInfo in portInfoArray) {
    NSLog(@"Port Name: %@", portInfo.portName);
    NSLog(@"MAC Address: %@", portInfo.macAddress);
    NSLog(@"Model Name: %@", portInfo.modelName);
}
```

*SearchPortViewController.m* を参照ください。

### 3.5.2 Swift 用

#### 宣言

```
open class func searchPrinter(target: String!) throws -> [Any]
```

```
open class func searchPrinter(_ target: String!, _ error: NSErrorPointer) -> [Any]!
```

#### 引数

名称	説明	型
target	すべてのインターフェイスタイプ	"ALL:"
	Bluetooth の場合	"BT:"
	Bluetooth Low Energy の場合	"BLE:"
	Ethernet / Wireless LAN の場合	"TCP:"
	USB の場合	"USB:"
error	エラー発生時にはエラー情報が格納されます。 code プロパティにエラーコードがセットされています。	NSErrorPointer

※ Bluetooth, USB(Lightning)両対応モデルでは、Bluetooth と USB は区別されません。

実際の接続方法に関わらず、target を指定しないか"BT:"を指定した場合は Bluetooth プリンタ、"USB:"を指定した場合は USB プリンタとして検出されます。

#### 戻り値

説明	型
Star プリンタの検索結果 戻り値には、PortInfo クラスのインスタンスが含まれます。 取得できる情報については <a href="#">PortInfo クラス</a> を参照ください。	[Any]

#### エラーコード

値	説明
SMStarIOResultCodeFailedError	何らかのエラーが発生した

#### Example

```
var searchPrinterResult: [PortInfo]? = nil

do {
    searchPrinterResult = try SMPort.searchPrinter(target: "ALL:") as? [PortInfo]
} catch {
    // Some error occurred.
}

guard let portInfoArray: [PortInfo] = searchPrinterResult else {
    return
}

for portInfo: PortInfo in portInfoArray {
    print("Port Name: \(portInfo.portName ?? "")")
    print("MAC Address: \(portInfo.macAddress ?? "")")
    print("Model Name: \(portInfo.modelName ?? "")")
}
```

[SearchPortViewController.swift](#) を参照ください。

## 1. 検索実行時間

検索したいインターフェイスによって、searchPrinter メソッドの実行時間が異なります。

インターフェイス	実行時間
すべてのインターフェイス	7[s]
Bluetooth	すぐに制御を返します。
Bluetooth Low Energy	7[s]
Ethernet / Wireless LAN	6[s]
USB	すぐに制御を返します。

## 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. Bluetooth Low Energy インターフェイスでの注意事項

初めて searchPrinter メソッドでプリンタ情報を取得した際、portName が@"BLE:"となる場合があります。  
この場合、一旦 getPort メソッドでプリンタに接続を行ってください。これ以降、正しいデバイス名を取得、使用可能になります。

## 3.6 writePort メソッド

プリンタにデータを書き込みます。コマンドや印刷データの送信に使用します。  
 印字終了の確認を行うため、このメソッドの前後で beginCheckedBlock/endCheckedBlock を使用してください。

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

### 3.6.1 Objective-C 用

#### 宣言

```
-(u_int32_t)writePort:(u_int8_t const *)writeBuffer
                  :(u_int32_t)offset
                  :(u_int32_t)size
                  :(NSError **)error;
```

#### 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	u_int8_t const *
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	u_int32_t
size	書き込む byte 数	u_int32_t
error	エラー発生時にはエラー情報が格納されます。 code プロパティにエラーコードがセットされています。	NSError **

#### 戻り値

説明	型
<b>Bluetooth/Ethernet/Wireless LAN/USB インターフェイス</b> 実際書き込まれたバイト数。 データが全て書き込めなかった時でも、writePort 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、writePort 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。	u_int32_t
<b>Bluetooth Low Energy インターフェイス</b> 成功時には、送信データのサイズが返ります。失敗時は 0 が返ります。	

#### エラーコード

値	説明
SMStarIOResultCodeFailedError	何らかのエラーが発生した

## 3.6.2 Swift 用

### 宣言

```
open func write(writeBuffer: UnsafePointer<UInt8>!, offset: UInt32, size: UInt32, numberOfBytesWritten:
UnsafeMutablePointer<UInt32>!) throws
```

```
open func write(_ writeBuffer: UnsafePointer<UInt8>!, _ offSet: UInt32, _ size: UInt32, _ error:
NSErrorPointer) -> UInt32
```

### 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	UnsafePointer<UInt8>
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	UInt32
size	書き込む byte 数	UInt32
numberOfBytesWritten	<b>Bluetooth/Ethernet/Wireless LAN/USB インターフェイス</b> 実際書き込まれたバイト数。 データが全て書き込めなかった時でも、writePort 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、writePort 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。 <b>Bluetooth Low Energy インターフェイス</b> 成功時には、送信データのサイズが返ります。失敗時は 0 が返ります。	UnsafeMutablePointer<UInt32>
error	エラー発生時にはエラー情報が格納されます。 code プロパティにエラーコードがセットされています。	NSErrorPointer

### 戻り値

説明	型
<b>Bluetooth/Ethernet/Wireless LAN/USB インターフェイス</b> 実際書き込まれたバイト数。 データが全て書き込めなかった時でも、writePort 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、writePort 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。 <b>Bluetooth Low Energy インターフェイス</b> 成功時には、送信データのサイズが返ります。失敗時は 0 が返ります。	UInt32

### エラーコード

値	説明
SMStarIOResultCodeFailedError	何らかのエラーが発生した

## 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。

### 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 getParsedStatus メソッドを使用してください。

### 2. 互換性に関する注意

StarIO.framework v2.4.0(StarPRNT SDK v5.6.0)より、LAN プリンタ使用時の readPort の挙動が以下のように変更されています。

readPort 実行時に受信すべきデータが存在しなかった場合

- v2.3.3 以前: PortException をスローします。
- v2.4.0 以降: 0 を返します。

### 3.7.1 Objective-C 用

#### 宣言

```
-(u_int32_t)readPort:(u_int8_t *)readBuffer
                  :(u_int32_t)offset
                  :(u_int32_t)size
                  :(NSError **)error;
```

#### 引数

名称	説明	型
readBuffer	データが読み込まれる byte 配列	u_int8_t *
offset	readBuffer にデータを書き込み始めるインデックス	u_int32_t
size	読み込む byte 数	u_int32_t
error	エラー発生時にはエラー情報が格納されます。 code プロパティにエラーコードがセットされています。	NSError **

#### 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。	u_int32_t

#### エラーコード

値	説明
SMStarIOResultCodeFailedError	何らかのエラーが発生した

### 3.7.2 Swift 用

#### 宣言

```
open func read(readBuffer: UnsafeMutablePointer<UInt8>!, offset: UInt32, size: UInt32, numberOfBytesRead:
UnsafeMutablePointer<UInt32>!) throws
```

```
open func read(_ readBuffer: UnsafeMutablePointer<UInt8>!, _ offSet: UInt32, _ size: UInt32, _ error:
NSErrorPointer) -> UInt32
```

#### 引数

名称	説明	型
readBuffer	データが読み込まれる byte 配列	UnsafeMutablePointer<UInt8>
offset	readBuffer にデータを書き込み始めるインデックス	UInt32
size	読み込む byte 数	UInt32
numberOfBytesRead	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。	UnsafeMutablePointer<UInt32>
error	エラー発生時にはエラー情報が格納されます。 code プロパティにエラーコードがセットされています。	NSErrorPointer

#### 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。	UInt32

#### エラーコード

値	説明
SMStarIOResultCodeFailedError	何らかのエラーが発生した

## 3.8 getParsedStatus メソッド

プリンタのステータスを取得します。

### 3.8.1 Objective-C 用

#### 宣言

```
-(NSUInteger)getParsedStatus:(void *)starPrinterStatus
                        :(u_int32_t)level
                        :(NSError **)error;
```

#### 引数

名称	説明	型
starPrinterStatus	現在のプリンタのステータスが格納されます。 取得できるステータスの種類は <a href="#">StarPrinterStatus 構造体</a> を参照ください。	void *
level	StarPrinterStatus 構造体のレベル (0,1,2 の指定が可能だが、通常は 2 を指定)	u_int32_t
error	エラー発生時にはエラー情報が格納されます。 code プロパティにエラーコードがセットされています。	NSError **

#### 戻り値

説明	型
成功時は 1, 失敗時は 0 を返します。	NSUInteger

#### エラーコード

値	説明
SMStarIOResultCodeFailedError	何らかのエラーが発生した



### Example

```
StarPrinterStatus_2 printerStatus;

NSError *error = nil;

[port getParsedStatus: &printerStatus : 2 : &error];

if (error != nil) {
    // Some error occurred.
}

if (printerStatus.offline == SM_TRUE)
{
    if (printerStatus.coverOpen == SM_TRUE) {
        // Cover is open.
    }
    else if (printerStatus.receiptPaperEmpty == SM_TRUE) {
        // Receipt paper is empty.
    }
    else {
        // The printer is offline.
    }
}
else {
    // The printer is online.
}
```

*Refer to DeviceStatusViewController.m.*

## 3.8.2 Swift 用

### 宣言

open func getParsedStatus(starPrinterStatus: UnsafeMutableRawPointer!, level: UInt32) throws

open func getParsedStatus(\_ starPrinterStatus: UnsafeMutableRawPointer!, \_ level: UInt32, \_ error: NSErrorPointer) -> UInt32

### 引数

名称	説明	型
starPrinterStatus	現在のプリンタのステータスが格納されます。 取得できるステータスの種類は <a href="#">StarPrinterStatus 構造体</a> を参照ください。	UnsafeMutableRawPointer
level	StarPrinterStatus 構造体のレベル（0,1,2 の指定が可能だが、通常は 2 を指定）	UInt32
error	エラー発生時にはエラー情報が格納されます。 code プロパティにエラーコードがセットされています。	NSErrorPointer

### 戻り値

説明	型
成功時は 1, 失敗時は 0 を返します。	UInt32

### エラーコード

値	説明
SMStarIOResultCodeFailedError	何らかのエラーが発生した

### Example

```
var printerStatus: StarPrinterStatus_2 = StarPrinterStatus_2()

do {
    try port.getParsedStatus(starPrinterStatus: &printerStatus, level: 2)
}
catch {
    // Some error occurred.
}

if printerStatus.offline == sm_true {
    if printerStatus.coverOpen == sm_true {
        // Cover is open.
    }
    else if printerStatus.receiptPaperEmpty == sm_true {
        // Receipt paper is empty.
    }
    else {
        // The printer is offline.
    }
}
else {
    // The printer is online.
}
```

Refer to DeviceStatusViewController.swift.

## 3.9 beginCheckedBlock メソッド

endCheckedBlock メソッドとセットで使用する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。

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

### 3.9.1 Objective-C 用

#### 宣言

```
-(NSInteger)beginCheckedBlock:(void *)starPrinterStatus
                        :(u_int32_t)level
                        :(NSError **)error;
```

#### 引数

名称	説明	型
starPrinterStatus	StarPrinterStatus 構造体へのポインタ（StarPrinterStatus, StarPrinterStatus_1, StarPrinterStatus_2 の指定が可能だが、通常は 2 を指定） このメソッドが成功したとき、現在のプリンタのステータスが格納されます。	void *
level	StarPrinterStatus 構造体のレベル（0,1,2 の指定が可能だが、通常は 2 を指定）	u_int32_t
error	エラー発生時にはエラー情報が格納されます。 code プロパティにエラーコードがセットされています。	NSError **

#### 戻り値

説明	型
成功時は 1, 失敗時は 0 を返します。	NSInteger

#### エラーコード

値	説明
SMStarIOResultCodeFailedError	何らかのエラーが発生した

### 3.9.2 Swift 用

#### 宣言

open func beginCheckedBlock(starPrinterStatus: UnsafeMutableRawPointer!, level: UInt32) throws

open func beginCheckedBlock(\_ starPrinterStatus: UnsafeMutableRawPointer!, \_ level: UInt32, \_ error: NSErrorPointer) -> UInt32

#### 引数

名称	説明	型
starPrinterStatus	StarPrinterStatus 構造体へのポインタ（StarPrinterStatus, StarPrinterStatus_1, StarPrinterStatus_2 の指定が可能だが、通常は 2 を指定） このメソッドが成功したとき、現在のプリンタのステータスが格納されます。	UnsafeMutableRawPointer
level	StarPrinterStatus 構造体のレベル（0,1,2 の指定が可能だが、通常は 2 を指定）	UInt32
error	エラー発生時にはエラー情報が格納されます。 code プロパティにエラーコードがセットされています。	NSErrorPointer

#### 戻り値

説明	型
成功時は 1, 失敗時は 0 を返します。	UInt32

#### エラーコード

値	説明
SMStarIOResultCodeFailedError	何らかのエラーが発生した

## 3.10 endCheckedBlock メソッド

beginCheckedBlock 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。  
 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。

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

### 3.10.1 Objective-C 用

#### 宣言

```
-(NSInteger)endCheckedBlock:(void *)starPrinterStatus
                        :(u_int32_t)level
                        :(NSError **)error;
```

#### 引数

名称	説明	型
starPrinterStatus	StarPrinterStatus 構造体へのポインタ (StarPrinterStatus, StarPrinterStatus_1, StarPrinterStatus_2 の指定が可能だが、通常は 2 を指定) このメソッドが成功したとき、現在のプリンタのステータスが格納されます。	void *
level	StarPrinterStatus 構造体のレベル (0,1,2 の指定が可能だが、通常は 2 を指定)	u_int32_t
error	エラー発生時にはエラー情報が格納されます。 code プロパティにエラーコードがセットされています。	NSError **

#### 戻り値

説明	型
成功時は 1, 失敗時は 0 を返します。	NSInteger

#### エラーコード

値	説明
SMStarIOResultCodeFailedError	何らかのエラーが発生した

#### 1. タイムアウト時間

タイムアウト時間は、endCheckedBlockTimeoutMillis プロパティの値が使用されます。  
 初期値は getPort で指定したタイムアウト時間となります。  
 endCheckedBlockTimeoutMillis の値は、印刷時間より長くなるよう調整してください。  
 また、10 秒未満の値が設定された場合にはタイムアウトは 10 秒になります。

### 3.10.2 Swift 用

#### 宣言

open func endCheckedBlock(starPrinterStatus: UnsafeMutableRawPointer!, level: UInt32) throws

open func endCheckedBlock(\_ starPrinterStatus: UnsafeMutableRawPointer!, \_ level: UInt32, \_ error: NSErrorPointer) -> UInt32

#### 引数

名称	説明	型
starPrinterStatus	StarPrinterStatus 構造体へのポインタ（StarPrinterStatus, StarPrinterStatus_1, StarPrinterStatus_2 の指定が可能だが、通常は 2 を指定） このメソッドが成功したとき、現在のプリンタのステータスが格納されます。	UnsafeMutableRawPointer
level	StarPrinterStatus 構造体のレベル（0,1,2 の指定が可能だが、通常は 2 を指定）	UInt32
error	エラー発生時にはエラー情報が格納されます。 code プロパティにエラーコードがセットされています。	NSErrorPointer

#### 戻り値

説明	型
成功時は 1, 失敗時は 0 を返します。	UInt32

#### エラーコード

値	説明
SMStarIOResultCodeFailedError	何らかのエラーが発生した

#### 1. タイムアウト時間

タイムアウト時間は、endCheckedBlockTimeoutMillis プロパティの値が使用されます。

初期値は getPort で指定したタイムアウト時間となります。

endCheckedBlockTimeoutMillis の値は、印刷時間より長くなるよう調整してください。

また、10 秒未満の値が設定された場合にはタイムアウトは 10 秒になります。

## 3.11 disconnect メソッド

指定された Bluetooth デバイスへのコネクションを切断します。

コネクションの切断後、Bluetooth デバイスは再び他の iOS 端末から接続することができるようになります。

### 3.11.1 Objective-C 用

#### 宣言

```
-(BOOL)disconnect:(NSError **)error;
```

#### 引数

名称	説明	型
error	エラー発生時にはエラー情報が格納されます。 code プロパティにエラーコードがセットされています。	NSError **

#### 戻り値

説明	型
切断成功・失敗 このメソッドは、以下の場合に失敗となります。 <ul style="list-style-type: none"> <li>● getPort で指定したタイムアウト時間内に切断が完了しなかった場合</li> <li>● プリンタが切断機能に対応していない場合（モバイルプリンタ等）</li> </ul> Ethernet デバイスに対しては何も行いません。 Ethernet デバイスに対して実行した場合は常に YES を返します。	BOOL

#### エラーコード

値	説明
SMStarIOResultCodeFailedError	何らかのエラーが発生した

### 3.11.2 Swift 用

#### 宣言

open func disconnectAccessory() throws

open func disconnect(\_ error: NSErrorPointer) -> Bool

#### 引数

名称	説明	型
error	エラー発生時にはエラー情報が格納されます。 code プロパティにエラーコードがセットされています。	NSErrorPointer

#### 戻り値

説明	型
切断成功・失敗 このメソッドは、以下の場合に失敗となります。 <ul style="list-style-type: none"> <li>● getPort で指定したタイムアウト時間内に切断が完了しなかった場合</li> <li>● プリンタが切断機能に対応していない場合（モバイルプリンタ等）</li> </ul> Ethernet デバイスに対しては何も行いません。 Ethernet デバイスに対して実行した場合は常に YES を返します。	Bool

#### エラーコード

値	説明
SMStarIOResultCodeFailedError	何らかのエラーが発生した



## 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。

### 3.12.1 Objective-C 用

#### 宣言

```
-(NSDictionary *)getFirmwareInformation:(NSError **)error;
```

#### 引数

名称	説明	型
error	エラー発生時にはエラー情報が格納されます。 code プロパティにエラーコードがセットされています。	NSError **

#### 戻り値

説明	型
ファームウェア情報取得結果の NSDictionary 戻り値の NSDictionary 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。	NSDictionary *

#### エラーコード

値	説明
SMStarIOResultCodeFailedError	何らかのエラーが発生した

### 3.12.2 Swift 用

#### 宣言

```
open func getFirmwareInformation() throws -> [AnyHashable : Any]
```

```
open func getFirmwareInformation(_ error: NSErrorPointer) -> [AnyHashable : Any]!
```

#### 引数

名称	説明	型
error	エラー発生時にはエラー情報が格納されます。 code プロパティにエラーコードがセットされています。	NSErrorPointer

#### 戻り値

説明	型
ファームウェア情報取得結果の Dictionary 戻り値の Dictionary 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。	[AnyHashable : Any]

#### エラーコード

値	説明
SMStarIOResultCodeFailedError	何らかのエラーが発生した

### 3.13 StarIOVersion メソッド

StarIO ライブラリのバージョンを取得します。

#### 宣言

```
+ (NSString *) StarIOVersion;
```

#### 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	NSString *

### 3.14 portName プロパティ

プリンタと通信するためのポートを特定する文字列

#### 宣言

```
- (NSString *) portName;
```

### 3.15 portSettings プロパティ

プリンタとの接続設定情報

#### 宣言

```
- (NSString *) portSettings;
```

### 3.16 timeoutMillis プロパティ

内部制御および API のタイムアウト値（ミリ秒指定）  
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。

#### 宣言

```
- (u_int32_t) timeoutMillis;
```

### 3.17 endCheckedBlockTimeoutMillis プロパティ

endCheckedBlock メソッドのタイムアウト値（ミリ秒指定）

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。

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 秒以上長くす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。

#### 宣言

```
@property (assign, readwrite, nonatomic) u_int32_t endCheckedBlockTimeoutMillis;
```

### 3.18 connected プロパティ

プリンタが iOS デバイスと接続されていれば "YES" を、そうでなければ "NO" を返します。

Bluetooth/Bluetooth Low Energy インターフェイスのみ対応しています。非対応のインターフェイスでは、常に "Yes" を返します。

Bluetooth/Bluetooth Low Energy 通信が切断されてからこのプロパティに反映されるまで、iOS デバイスの制限により 5 秒程度かかります。

#### 宣言

```
-(BOOL) connected;
```

## 4 StarPrinterStatus 構造体 (StarIO.framework)

プリンタのステータスを保持する構造体です。

boolean 型とバイナリ(raw バイト配列)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メンバー一覧をご覧ください。

### メンバ

名称	説明	型	詳細
blackMarkError	ブラックマークエラー	SM_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 SM_TRUE となる。通常時は SM_FALSE。
compulsionSwitch	コンパルジョンスイッチ	SM_BOOLEAN	ドロワのコンパルジョン SW が押されていると SM_TRUE となる。通常は SM_FALSE。
coverOpen	カバーオープン	SM_BOOLEAN	カバーが開いている場合に SM_TRUE となる。閉じている場合は SM_FALSE。
cutterError	オートカッターエラー	SM_BOOLEAN	カッターエラー発生時に SM_TRUE となる。
etbAvailable	ETB 使用可否	SM_BOOLEAN	ETB が使用可能な場合に SM_TRUE となる。使用できない場合は SM_FALSE。
etbCounter	ETB カウンタ	UCHAR	現在の ETB カウンタの値。
headThermistorError	ヘッドサーミスタエラー	SM_BOOLEAN	ヘッドサーミスタ異常値検出時に SM_TRUE となる。通常は SM_FALSE。
offline	オフライン	SM_BOOLEAN	オフラインの場合に SM_TRUE となる。オンライン時は SM_FALSE。
overTemp	印字ヘッド高温による停止中	SM_BOOLEAN	ヘッドが高温になり印刷停止している状態で SM_TRUE となる。通常は SM_FALSE。
raw	ステータスバイト列	UCHAR[63]	ステータスのバイト列 (例: HEX 23 86 00 00 00 00 00 00 00)

名称	説明	型	詳細
rawLength	raw の長さ	CHAR	raw の長さ
receiptPaperEmpty	用紙エンド	SM_BOOLEAN	用紙切れの場合は SM_TRUE となる。通常は SM_FALSE。
receiptPaperNearEmptyInner	用紙ニアエンド(内側)	SM_BOOLEAN	用紙ニアエンド状態の時に SM_TRUE となる。通常は SM_FALSE。
receiveBufferOverflow	受信バッファオーバーフロー	SM_BOOLEAN	受信バッファフルの時に SM_TRUE となる。通常は SM_FALSE。
unrecoverableError	復帰不可能エラー	SM_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 SM_TRUE となる。通常は SM_FALSE。
voltageError	電源電圧エラー	SM_BOOLEAN	電源電圧異常値検出時に SM_TRUE となる。通常は SM_FALSE。
connectedInterface	現在通信しているインターフェイスの種類	UCHAR	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 構造体

各モデルにおいて機能するメンバ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	SM-T400i	BSC10	SM-S210i StarPRNT	SM-S220i StarPRNT	SM-S230i StarPRNT	SM-T300i StarPRNT	SM-T400i StarPRNT	SM-L200	SM-L300	SP700
blackMarkError	-	-	-	✓	-	-	✓	✓	-	-	-	-	-	-	✓	✓	✓	✓	✓	✓	✓	✓
compulsionSwitch	✓	✓	✓	✓	✓	✓	✓	✓	-	-	-	-	-	✓	-	-	-	-	-	-	-	✓
coverOpen	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
cutterError	✓	✓	✓	✓	✓	✓	✓	✓	-	-	-	-	-	-	-	-	-	-	-	-	-	✓
etbAvailable	✓	✓	✓	✓	✓	✓	✓	✓	-	-	-	-	-	-	✓	✓	✓	✓	✓	✓	✓	✓
etbCounter	✓	✓	✓	✓	✓	✓	✓	✓	-	-	-	-	-	-	✓	✓	✓	✓	✓	✓	✓	✓
headThermistorError	✓	✓	✓	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
offline	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
overTemp	✓	✓	✓	✓	✓	✓	✓	✓	-	-	-	-	-	-	✓	✓	✓	✓	✓	✓	✓	✓
raw	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	-	✓	✓	✓	✓	✓	✓	✓	✓
rawLength	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	-	✓	✓	✓	✓	✓	✓	✓	✓
receiptPaperEmpty	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
receiptPaperNearEmptyInner	-	*1	-	✓	-	✓	✓	✓	-	-	-	-	-	-	-	-	-	-	-	-	-	✓
receiveBufferOverflow	-	-	-	✓	-	✓	✓	✓	-	-	-	-	-	-	-	-	-	-	-	-	-	✓
unrecoverableError	✓	✓	✓	✓	✓	✓	✓	✓	-	-	-	-	-	-	-	-	-	-	-	-	-	✓
voltageError	✓	✓	✓	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
connectedInterface	✓	✓	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

- : 機能しない

\*1 : MCP30 は非対応

## 5 PortInfo クラス (StarIO.framework)

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。

### プロパティ

名称	説明
portName	プリンタと通信するためのポートを特定する文字列。
macAddress	プリンタの MAC アドレスを表す文字列。
modelName	プリンタのモデル名を表す文字列。

### 5.1 portName プロパティ

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。

#### 宣言

```
@property (retain, readonly) NSString *portName;
```

### 5.2 macAddress プロパティ

プリンタの MAC アドレスを表す文字列。

#### 宣言

```
@property (retain, readonly) NSString *macAddress;
```

### 5.3 modelName プロパティ

プリンタのモデル名を表す文字列。

#### 宣言

```
@property (retain, readonly) NSString *modelName;
```



## 6 SMBluetoothManager クラス (StarIO.framework)

SMBluetoothManager 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。SMPort クラスと同時に使用しないでください。

### メソッド

名称	説明
initWithName	SMBluetoothManager のインスタンスを作成します。
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。

### プロパティ

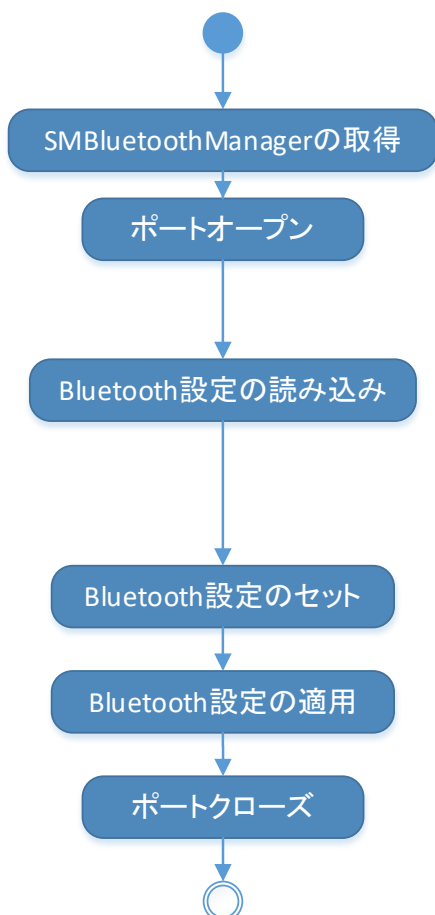
名称	説明
portName	プリンタと通信するためのポートを特定する文字列
deviceType	接続先プリンタの種類
opened	ポートがオープンされているかどうか
deviceName	Bluetooth デバイス名
iOSPortName	iOSPort 名
autoConnect	AutoConnection（自動接続機能）のオン/オフ
security	Bluetooth セキュリティタイプ
pinCode	Bluetooth ペアリング時に使用する PIN コード
deviceNameCapability	Bluetooth デバイス名の設定可否情報
iOSPortNameCapability	iOSPort 名の設定可否情報
autoConnectCapability	AutoConnection の設定可否情報
securityTypeCapability	Bluetooth セキュリティタイプの設定可否情報
pinCodeCapability	PIN コード設定可否情報

### コンスタント

名称	説明
SMDeviceType	プリンタタイプ定数。
SMBluetoothSecurity	Bluetooth セキュリティタイプ定数。
SMBluetoothSettingCapability	Bluetooth 設定可否定数。

## 6.1 SMBluetoothManager を使った Bluetooth 設定変更フロー

SMBluetoothManager 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 の取得
SMBluetoothManager *bluetoothManager = [SMBluetoothManagerFactory
getManager:portName emulation:emulation];

// ポートオープン
if ([bluetoothManager open] == NO) {
    // Fail to open port.
    return;
}

// Bluetooth 設定の読み込み
if ([bluetoothManager loadSetting] == NO) {
    // Fail to load settings.
    [bluetoothManager close];
    return;
}

// Bluetooth 設定のセット
// 例) デバイス名を"Star Micronics"に設定したい場合
bluetoothManager.deviceName = @"Star Micronics";

// Bluetooth 設定の適用
if ([bluetoothManager apply] == NO) {
    // Fail to apply settings.
    [bluetoothManager close];
    return;
}

// ポートクローズ
[bluetoothManager close];
  
```

*BluetoothSettingViewController.m* を参照ください。

## 6.2 モデル: SMBluetoothManager 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 *1	FVP10	TSP100 (TSP100IIIのみ)	TSP650II	TSP700II	TSP800II	SM-S210i *2	SM-S220i *2	SM-S230i *2	SM-T300i *2	SM-T400i *2	BSC10	SM-S210i StarPRNT	SM-S220i StarPRNT	SM-S230i StarPRNT	SM-T300i StarPRNT	SM-T400i StarPRNT	SM-L200	SM-L300	SP700
initWithName	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	-	✓	✓	✓	✓	✓	✓	✓	✓
open	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	-	✓	✓	✓	✓	✓	✓	✓	✓
loadSetting	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	-	✓	✓	✓	✓	✓	✓	✓	✓
close	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	-	✓	✓	✓	✓	✓	✓	✓	✓
apply	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	-	✓	✓	✓	✓	✓	✓	✓	✓

-: 機能しない

\*1: ファームウェアバージョン 1.1 以降が必要です

\*2: ファームウェアバージョン 3.0 以降が必要です

各モデルにおいて設定できる（apply メソッド実行時に Bluetooth デバイスに反映される）プロパティは以下のとおりです。

Property	mC-Print2	mC-Print3	mPOP *1	FVP10	TSP100 (TSP100IIBIのみ)	TSP650II	TSP700II	TSP800II	SM-S210i *2	SM-S220i *2	SM-S230i *2	SM-T300i *2	SM-T400i *2	BSC10	SM-S210i StarPRNT	SM-S220i StarPRNT	SM-S230i StarPRNT	SM-T300i StarPRNT	SM-T400i StarPRNT	SM-L200	SM-L300	SP700
deviceName	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	-	✓	✓	✓	✓	✓	✓	✓	✓
iOSPortName	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	-	✓	✓	✓	✓	✓	-	-	✓
pinCode	-	-	-	✓	-	✓	✓	✓	✓	✓	✓	✓	✓	-	✓	✓	✓	✓	✓	✓	✓	✓
autoConnect	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	-	✓	✓	✓	✓	✓	-	-	✓
security	-	-	-	*3	-	*3	*3	*3	*4	*4	*4	*4	*4	-	*4	*4	*4	*4	*4	*4	*4	*3

- : 機能しない

\*1 : ファームウェアバージョン 1.1 以降が必要です

\*2 : ファームウェアバージョン 3.0 以降が必要です

\*3 : 設定・取得できる security は PIN コードもしくは SSP

\*4 : 設定・取得できる security は PIN コードもしくは Disable

## 6.3 initWithName メソッド

SMBluetoothManager のインスタンスを作成します。

### 宣言

```
-(id) initWithPortName: (NSString *) portName deviceType: (SMDeviceType) deviceType;
```

### 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 <a href="#">SMPort クラスの getPort メソッド</a> の portName と同じです。	NSString *
deviceType	プリンタタイプ定数	SMDeviceType

### 戻り値

説明	型
成功時は SMBluetoothManager のインスタンスを返します。 失敗時は nil を返します。	id

Bluetooth 設定変更の手順は [SMBluetoothManager 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 SMDeviceType コンスタント定数は [SMDeviceType コンスタント](#)を参照ください。

#### 1. SMBluetoothManagerFactory クラス

[SMBluetoothManagerFactory クラス](#)は、プリンタのエミュレーションに対して適切な SMBluetoothManager 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 メソッドで現在の設定を取得してください。

### 宣言

```
-(BOOL) open;
```

### 引数

名称	説明	型
-	-	-

### 戻り値

説明	型
成功・失敗	BOOL

Bluetooth 設定変更の手順は [SMBluetoothManager 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

- (BOOL) loadSetting;

### 引数

名称	説明	型
-	-	-

### 戻り値

説明	型
-	-

Bluetooth 設定変更の手順は [SMBluetoothManager 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

- (void) close;

### 引数

名称	説明	型
-	-	-

### 戻り値

説明	型
-	-

Bluetooth 設定変更の手順は [SMBluetoothManager 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

deviceName、iOSSPortName、autoConnect、security、pinCode プロパティの値をデバイスに適用します。

### 宣言

- (BOOL) apply;

### 引数

名称	説明	型
-	-	-

### 戻り値

説明	型
成功・失敗	BOOL

1. apply メソッドで適用した値は、デバイスの電源再投入・再ペアリングを行った後に有効になります。

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

## 6.8 portName プロパティ

プリンタと通信するためのポートを特定する文字列

### 宣言

@property (nonatomic, readonly) NSString \*portName;

## 6.9 deviceType プロパティ

接続先プリンタの種類

### 宣言

@property(nonatomic, readonly) SMDeviceType deviceType;

## 6.10 opened プロパティ

ポートがオープンされているかどうか

open メソッドが成功すると YES になります。その後 close メソッドを呼び出すと NO になります。

### 宣言

@property(nonatomic, readonly) BOOL opened;

## 6.11 deviceName プロパティ

Bluetooth デバイス名

この名前は、Bluetooth ではペアリング時に表示されます。

Bluetooth Low Energy では、通信時の接続ポート名として使用されます。

loadSetting メソッドを呼び出した際に現在の設定値が読み込まれます。

設定するには、本プロパティを変更後 apply メソッドを実行します。

### 宣言

```
@property(nonatomic, retain) NSString *deviceName;
```

#### 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

#### 3. 注意事項

Bluetooth Low Energy の場合、変更した Bluetooth デバイス名は、デバイスの電源再投入・再接続を行った後に有効になります。

## 6.12 iOSPortName プロパティ

iOSPort 名

StarIO で Bluetooth 通信時に使用するポート名を取得、設定します。

Bluetooth Low Energy では使用されません。

loadSetting メソッドを呼び出した際に現在の設定値が読み込まれます。

設定するには、本プロパティを変更後 apply メソッドを実行します。

### 宣言

```
@property(nonatomic, retain) NSString *iOSPortName;
```

#### 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

## 6.13 autoConnect プロパティ

AutoConnection（自動接続機能）のオン／オフ

本機能は Bluetooth インターフェイスのみ対応しています。

loadSetting メソッドを呼び出した際に現在の設定値が読み込まれます。

設定するには、本プロパティを変更後 apply メソッドを実行します。

### 宣言

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

#### 1. 注意事項

security プロパティが PIN コード設定の場合は、この値に NO を設定してください。



## 6.14 security プロパティ

Bluetooth セキュリティタイプ

本機能は Bluetooth のみ対応しています。

loadSetting メソッドを呼び出した際に現在の設定値が読み込まれます。

設定するには、本プロパティを変更後 apply メソッドを実行します。

### 宣言

```
@property(nonatomic, assign) SMBluetoothSecurity security;
```

## 6.15 pinCode プロパティ

Bluetooth ペアリング時に使用する PIN コード

本機能は Bluetooth のみ対応しています。

現在の設定値を取得することはできません。

現在の PIN コードから値を変更しない場合は nil を指定します。

### 宣言

```
@property(nonatomic, retain) NSString *pinCode;
```

#### 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)

## 6.16 deviceNameCapability プロパティ

Bluetooth デバイス名の設定可否情報

### 宣言

```
@property (assign, readonly) SMBluetoothSettingCapability deviceNameCapability;
```

## 6.17 iOSPortNameCapability プロパティ

iOSPort 名の設定可否情報

### 宣言

```
@property (assign, readonly) SMBluetoothSettingCapability iOSPortNameCapability;
```

## 6.18 autoConnectCapability プロパティ

AutoConnection の設定可否情報

### 宣言

```
@property (assign, readonly) SMBluetoothSettingCapability autoConnectCapability;
```

## 6.19 securityTypeCapability プロパティ

Bluetooth セキュリティタイプの設定可否情報

### 宣言

@property (assign, readonly) SMBluetoothSettingCapability securityTypeCapability;

## 6.20 pinCodeCapability プロパティ

PIN コード設定可否情報

### 宣言

@property (assign, readonly) SMBluetoothSettingCapability pinCodeCapability;

## 6.21 SMDeviceType コンスタント

プリンタタイプ定数。

### 宣言

```
typedef enum _SMDeviceType {
    SMDeviceTypeUnknown = 0,
    SMDeviceTypeDesktopPrinter,
    SMDeviceTypePortablePrinter,
    SMDeviceTypeDKAirCash,
} SMDeviceType;
```

### 定数

名称	説明
SMDeviceTypeDesktopPrinter	デスクトッププリンタ
SMDeviceTypePortablePrinter	モバイルプリンタ

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

Model	SMDeviceType
TSP100	SMDeviceTypeDesktopPrinter
FVP10	
TSP650II	
TSP700II	
TSP800II	
SP700	
mC-Print2	SMDeviceTypePortablePrinter
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	

## 6.22 SMBluetoothSecurity コンスタント

Bluetooth セキュリティタイプ定数。

### 宣言

```
typedef enum _SMBluetoothSecurity {
    SMBluetoothSecurityDisable,
    SMBluetoothSecuritySSP,
    SMBluetoothSecurityPINcode
} SMBluetoothSecurity;
```

### 定数

名称	説明
SMBluetoothSecurityPINcode	PIN コード入力によるセキュリティ
SMBluetoothSecuritySSP	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
SMBluetoothSecurityDisable	セキュリティなし

## 6.23 SMBluetoothSettingCapability コンスタント

Bluetooth 設定可否定数。

### 宣言

```
typedef enum _SMBluetoothSettingCapability {
    SMBluetoothSettingCapabilitySupport,
    SMBluetoothSettingCapabilityNoSupport
} SMBluetoothSettingCapability;
```

### 定数

名称	説明
SMBluetoothSettingCapabilitySupport	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。
SMBluetoothSettingCapabilityNoSupport	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 tarloExt クラス (StarIO\_Extension.framework)

### 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。
createMelodySpeakerCommandBuilder	メロディスピーカー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。
createMelodySpeakerConnectParser	メロディスピーカ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。

### コンスタント

名称	説明
StarloExtEmulation	エミュレーション指定定数。
StarloExtCharacterCode	文字コード指定定数。
StarloExtBcrModel	バーコードリーダーモデル指定定数。
StarloExtDisplayModel	カスタマーディスプレイモデル指定定数。
StarloExtMelodySpeakerModel	メロディスピーカーモデル指定定数。

## 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

```
+ (ISCBBuilder *)createCommandBuilder:(StarloExtEmulation)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"> <li>StarloExtEmulationStarPRNT ... StarPRNT エミュレーション</li> <li>StarloExtEmulationStarLine ... STAR Line Mode エミュレーション</li> <li>StarloExtEmulationStarGraphic ... STAR Graphic Mode エミュレーション</li> <li>StarloExtEmulationEscPos ... ESC/POS エミュレーション</li> <li>StarloExtEmulationEscPosMobile ... ESC/POS Mobile エミュレーション</li> <li>StarloExtEmulationStarDotImpact ... STAR ドットインパクトエミュレーション</li> <li>StarloExtEmulationStarPRNTL... StarPRNTL エミュレーション</li> </ul>	StarloExtEmulation

### 戻り値

説明	型
ISCBBuilder オブジェクト	ISCBBuilder *

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

### Example

```
+ (NSData *)createCommandImage:(StarloExtEmulation)emulation image:(UIImage *)image {
    ISCBBuilder *builder = [StarloExt createCommandBuilder:emulation];

    [builder beginDocument];

    [builder appendBitmap:image diffusion:NO];

    [builder appendCutPaper:SCBCutPaperActionPartialCutWithFeed];

    [builder endDocument];

    return [builder.commands copy];
}
```

PrinterFunctions.m 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

```
+ (ISDCBBuilder *)createDisplayCommandBuilder:(StarloExtDisplayModel)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	StarloExtDisplayModel

### 戻り値

説明	型
ISDCBBuilder オブジェクト	ISDCBBuilder *

### Example

```
- (void)alertView:(UIAlertView *)alertView didDismissWithButtonIndex:(NSInteger)buttonIndex {
    if (buttonIndex != alertView.cancelButtonIndex) {
        ISDCBBuilder *builder = [StarloExt createDisplayCommandBuilder:StarloExtDisplayModelSCD222];
        ...
    }
}
```

DisplayViewController.m / DisplayExtViewController.m を参照ください。



## 7.3 createMelodySpeakerCommandBuilder メソッド

メロディスピーカーコマンド生成オブジェクトを生成します。

### 宣言

```
+ (ISMCBBuilder *)createMelodySpeakerCommandBuilder:(StarloExtMelodySpeakerModel) model;
```

### 引数

名称	説明	型
model	メロディスピーカーモデル指定定数 <ul style="list-style-type: none"> <li>StarloExtMelodySpeakerModelMCS10</li> <li>StarloExtMelodySpeakerModelFVP10</li> </ul>	StarloExtMelodySpeakerModel

### 戻り値

説明	型
ISMCBBuilder オブジェクト	ISMCBBuilder *

### Example

```
+ (NSData *)createPlayingRegisteredSound:(StarloExtMelodySpeakerModel)model
    specifySound:(BOOL)specifySound
    soundStorageArea:(SMCBSoundStorageArea)soundStorageArea
    soundNumber:(NSInteger)soundNumber
    specifyVolume:(BOOL)specifyVolume
    volume:(NSInteger)volume
    error:(NSError * _Nullable * _Nullable) error {
    ISMCBBuilder *builder = [StarloExt createMelodySpeakerCommandBuilder:model];

    SMSoundSetting *setting = [SMSoundSetting new];

    if (specifySound == YES) {
        [setting setSoundStorageArea:soundStorageArea];
        [setting setSoundNumber:soundNumber];
    }

    if (specifyVolume == YES) {
        [setting setVolume:volume];
    }

    [builder appendSoundWithSetting:setting
                        error:error];

    if (*error != nil) {
        return nil;
    }

    return [builder.commands copy];
}
```

*MelodySpeakerFunctions.m* を参照ください。

## 7.4 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

```
+ (ISCPConnectParser *)createBcrConnectParser:(StarloExtBcrModel)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 • StarloExtBcrModelPOP1	StarloExtBcrModel

### 戻り値

説明	型
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	ISCPConnectParser *

## 7.5 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

```
+ (ISCPConnectParser *)createDisplayConnectParser:(StarloExtDisplayModel)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	StarloExtDisplayModel

### 戻り値

説明	型
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	ISCPConnectParser *

### Example

```
- (void)tableView:(UITableView *)tableView didSelectRowAtIndexPath:(NSIndexPath *)indexPath {
    ...

    ISCPConnectParser *parser = [StarloExt createDisplayConnectParser:StarloExtDisplayModelISCD222];

    [Communication parseDoNotCheckCondition:parser
     port:port
     completionHandler:^(CommunicationResult *communicationResult) {
         if (communicationResult.result == CommResultSuccess) {
             if (parser.connect == YES) {
                 [self showSimpleAlertWithTitle:@"Check Status"
                  message:@"Display Connect."
                  buttonTitle:@"OK"
                  buttonStyle:UIAlertActionStyleCancel
                  completion:nil];
             }
             else {
                 [self showSimpleAlertWithTitle:@"Check Status"
                  message:@"Display Disconnect."
                  buttonTitle:@"OK"
                  buttonStyle:UIAlertActionStyleCancel
                  completion:nil];
             }
         }
     }];
    ...
}
```

DisplayViewController.m / DisplayExtViewController.m を参照ください。

## 7.6 createMelodySpeakerConnectParser メソッド

メロディスピーカ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

```
+ (ISCPConnectParser *)createMelodySpeakerConnectParser:(StarloExtMelodySpeakerModel)model
error:(NSError **)error;
```

### 引数

名称	説明	型
model	メロディスピーカーモデル指定定数 <ul style="list-style-type: none"> <li>StarloExtMelodySpeakerModelMCS10</li> <li>StarloExtMelodySpeakerModelFVP10 (非対応)</li> </ul>	StarloExtMelodySpeakerModel
error	オブジェクト生成に失敗した際のエラー情報	NSError **

### 戻り値

説明	型
メロディスピーカー接続確認コマンド応答解析オブジェクト	ISCPConnectParser *

### エラーコード

値	説明
StarloExtErrorInvalidOperation	StarloExtMelodySpeakerModelFVP10 を指定した場合

### Example

```
if (_melodySpeakerModel == StarloExtMelodySpeakerModelMCS10) {
    NSError *parserError = nil;

    ISCPConnectParser *parser = [StarloExt createMelodySpeakerConnectParser:_melodySpeakerModel
                                error:&parserError];

    [Communication parseDoNotCheckCondition:parser
     port:port
     completionHandler:^(CommunicationResult *communicationResult) {
        if (communicationResult.result == CommResultSuccess) {
            if (parser.connect == YES) {
                ...
            }
            ...
        }
        ...
    }];
}
```

MelodySpeakerViewController.m を参照ください。

## 7.7 StarloExtEmulation コンスタント

エミュレーション指定定数。

### 宣言

```
typedef NS_ENUM(NSInteger, StarloExtEmulation) {
    StarloExtEmulationNone = 0,
    StarloExtEmulationStarPRNT,
    StarloExtEmulationStarLine,
    StarloExtEmulationStarGraphic,
    StarloExtEmulationEscPos,
    StarloExtEmulationEscPosMobile,
    StarloExtEmulationStarDotImpact,
    StarloExtEmulationStarPRNTL
};
```

### 定数

名称	説明
StarloExtEmulationStarPRNT	StarPRNT エミュレーション
StarloExtEmulationStarLine	STAR Line Mode エミュレーション
StarloExtEmulationStarGraphic	STAR Graphic Mode エミュレーション
StarloExtEmulationEscPos	ESC/POS エミュレーション
StarloExtEmulationEscPosMobile	ESC/POS Mobile エミュレーション
StarloExtEmulationStarDotImpact	STAR ドットインパクトエミュレーション
StarloExtEmulationStarPRNTL	StarPRNTL 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 StarloExtEmulation コンスタント定数は以下のとおりです。

Model	StarloExtEmulation
mC-Print2	StarloExtEmulationStarPRNT
mC-Print3	StarloExtEmulationStarPRNT
mPOP	StarloExtEmulationStarPRNT
FVP10	StarloExtEmulationStarLine
TSP100	StarloExtEmulationStarGraphic
TSP650II	StarloExtEmulationStarLine
TSP700II	StarloExtEmulationStarLine
TSP800II	StarloExtEmulationStarLine
SM-S210i	StarloExtEmulationEscPosMobile
SM-S220i	StarloExtEmulationEscPosMobile
SM-S230i	StarloExtEmulationEscPosMobile
SM-T300i/T300	StarloExtEmulationEscPosMobile
SM-T400i	StarloExtEmulationEscPosMobile
BSC10	StarloExtEmulationEscPos
SM-S210i StarPRNT	StarloExtEmulationStarPRNT
SM-S220i StarPRNT	StarloExtEmulationStarPRNT
SM-S230i StarPRNT	StarloExtEmulationStarPRNT
SM-T300i/T300 StarPRNT	StarloExtEmulationStarPRNT
SM-T400i StarPRNT	StarloExtEmulationStarPRNT

## 7.8 StarloExtCharacterCode コンスタント

文字コード指定定数。

### 宣言

```
typedef NS_ENUM(NSInteger, StarloExtCharacterCode) {
    StarloExtCharacterCodeNone = 0,
    StarloExtCharacterCodeStandard,
    StarloExtCharacterCodeJapanese,
    StarloExtCharacterCodeSimplifiedChinese,
    StarloExtCharacterCodeTraditionalChinese
};
```

### 定数

名称	説明
StarloExtCharacterCodeStandard	スタンダード
StarloExtCharacterCodeJapanese	日本語
StarloExtCharacterCodeSimplifiedChinese	簡体字中国語
StarloExtCharacterCodeTraditionalChinese	繁体字中国語

## 7.9 StarloExtBcrModel コンスタント

バーコードリーダーモデル指定定数。

### 宣言

```
typedef NS_ENUM(NSInteger, StarloExtBcrModel) {
    StarloExtBcrModelNone = 0,
    StarloExtBcrModelPOP1,
    StarloExtBcrModelDS9208
};
```

### 定数

名称	説明
StarloExtBcrModelPOP1	バーコードリーダー BCR-POP1
StarloExtBcrModelDS9208	バーコードリーダー DS9208

## 7.10 StarloExtDisplayModel 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

```
typedef NS_ENUM(NSInteger, StarloExtDisplayModel) {
    StarloExtDisplayModelNone = 0,
    StarloExtDisplayModelSCD222
};
```

### 定数

名称	説明
StarloExtDisplayModelSCD222	カスタマーディスプレイ SCD222U

## 7.11 StarloExtMelodySpeakerModel コ ン ス タ ント

メロディスピーカーモデル指定定数。

### 宣言

```
typedef NS_ENUM(NSUInteger, StarloExtMelodySpeakerModel) {
    StarloExtMelodySpeakerModelMCS10,
    StarloExtMelodySpeakerModelFVP10
};
```

### 定数

名称	説明
StarloExtMelodySpeakerModelMCS10	メロディスピーカー MCS10
StarloExtMelodySpeakerModelFVP10	FVP10 内蔵スピーカー

## 8 ISCBBuilder クラス (StarIO\_Extension.framework)

### 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、commands プロパティに追加します。
endDocument	ドキュメント終了コマンドを生成し、commands プロパティに追加します。
appendInitialization	初期化コマンドを生成し、commands プロパティに追加します。
appendByte	データ（テキストやコマンド）を commands プロパティに追加します。
appendData	
appendBytes	
appendRawByte	生のデータ（テキストやコマンド）を commands プロパティに追加します。
appendRawData	
appendRawBytes	
appendFontStyle	フォントスタイル指定コマンドを生成し、commands プロパティに追加します。
appendCodePage	コードページ指定コマンドを生成し、commands プロパティに追加します。
appendInternational	国際文字指定コマンドを生成し、commands プロパティに追加します。
appendLineFeed	行単位の用紙送りコマンドを生成し、commands プロパティに追加します。
appendDataWithLineFeed	
appendBytesWithLineFeed	
appendUnitFeed	1 ドットライン単位の用紙送りコマンドを生成し、commands プロパティに追加します。
appendDataWithUnitFeed	
appendBytesWithUnitFeed	
appendCharacterSpace	文字間スペース設定コマンドを生成し、commands プロパティに追加します。
appendLineSpace	行間スペース設定コマンドを生成し、commands プロパティに追加します。
appendTopMargin	トップマージン設定コマンドを生成し、commands プロパティに追加します。
appendEmphasis	強調指定コマンドを生成し、commands プロパティに追加します。
appendDataWithEmphasis	
appendBytesWithEmphasis	
appendInvert	白黒反転指定コマンドを生成し、commands プロパティに追加します。
appendDataWithInvert	
appendBytesWithInvert	
appendMultiple	拡大指定コマンドを生成し、commands プロパティに追加します。
appendDataWithMultiple	
appendBytesWithMultiple	
appendMultipleHeight	
appendDataWithMultipleHeight	
appendBytesWithMultipleHeight	
appendMultipleWidth	



名称	説明
appendDataWithMultipleWidth	アンダーライン指定コマンドを生成し、commands プロパティに追加します。
appendBytesWithMultipleWidth	
appendUnderLine	
appendDataWithUnderLine	
appendBytesWithUnderLine	
appendLogo	ロゴ印刷コマンドを生成し、commands プロパティに追加します。
appendAbsolutePosition	絶対位置指定コマンドを生成し、commands プロパティに追加します。
appendDataWithAbsolutePosition	
appendBytesWithAbsolutePosition	
appendAlignment	位置揃え指定コマンドを生成し、commands プロパティに追加します。
appendDataWithAlignment	
appendBytesWithAlignment	
appendHorizontalTabPosition	水平タブ設定／クリアコマンドを生成し、commands プロパティに追加します。
appendCutPaper	用紙カットコマンドを生成し、commands プロパティに追加します。
appendPeripheral	キャッシュドローア駆動コマンドを生成し、commands プロパティに追加します。
appendSound	サウンド/ブザー駆動コマンドを生成し、commands プロパティに追加します。
appendBarcodeData	バーコード印刷コマンドを生成し、commands プロパティに追加します。
appendBarcodeBytes	
appendBarcodeDataWithAbsolutePosition	絶対位置指定バーコード印刷コマンドを生成し、commands プロパティに追加します。
appendBarcodeBytesWithAbsolutePosition	
appendBarcodeDataWithAlignment	位置揃えバーコード印刷コマンドを生成し、commands プロパティに追加します。
appendBarcodeBytesWithAlignment	
appendPdf417Data	PDF417 印刷コマンドを生成し、commands プロパティに追加します。
appendPdf417Bytes	
appendPdf417DataWithAbsolutePosition	絶対位置指定 PDF417 印刷コマンドを生成し、commands プロパティに追加します。
appendPdf417BytesWithAbsolutePosition	
appendPdf417DataWithAlignment	位置揃え PDF417 印刷コマンドを生成し、commands プロパティに追加します。
appendPdf417BytesWithAlignment	
appendQrCodeData	QR コード印刷コマンドを生成し、commands プロパティに追加します。
appendQrCodeBytes	
appendQrCodeDataWithAbsolutePosition	絶対位置指定 QR コード印刷コマンドを生成し、commands プロパティに追加します。
appendQrCodeBytesWithAbsolutePosition	
appendQrCodeDataWithAlignment	位置揃え QR コード印刷コマンドを生成し、commands プロパティに追加します。
appendQrCodeBytesWithAlignment	
appendBitmap	ビットマップ印刷コマンドを生成し、commands プロパティに追加します。

名称	説明
appendBitmapWithAbsolutePosition	絶対位置指定ビットマップ印刷コマンドを生成し、commands プロパティに追加します。
appendBitmapWithAlignment	位置揃えビットマップ印刷コマンドを生成し、commands プロパティに追加します。
appendBlackMark	ブラックマークコマンドを生成し、commands プロパティに追加します。
beginPageMode	ページモード開始コマンドを生成し、commands プロパティに追加します。
endPageMode	ページモード終了コマンドを生成し、commands プロパティに追加します。
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、commands プロパティに追加します。
appendPageModeRotation	ページモード印刷方向コマンドを生成し、commands プロパティに追加します。
appendPrintableArea	印字領域設定コマンドを生成し、commands プロパティに追加します。
appendCjkUnifiedIdeographFont	UTF-8 における CJK 統合漢字のフォント設定コマンドを生成し、commands プロパティに追加します。

## プロパティ

名称	説明
commands	生成、追加されたコマンド列です。

## コンスタント

名称	説明
SCBInitializationType	初期化指定定数。
SCBFontStyleType	フォントスタイル指定定数。
SCBCodePageType	コードページ指定定数。
SCBInternationalType	国際文字指定定数。
SCBLogoSize	ロゴサイズ指定定数。
SCBAlignmentPosition	位置揃え指定定数。
SCBCutPaperAction	用紙カット指定定数。
SCBPeripheralChannel	キャッシュドローアーチャネル指定定数。
SCBSoundChannel	サウンド/ブザーチャネル指定定数。
SCBBarcodeSymbology	バーコードシンボル指定定数。
SCBBarcodeWidth	バーコード幅指定定数。
SCBPdf417Level	PDF417 ECC（セキュリティレベル）指定定数。
SCBQRCodeModel	QR コードモデル指定定数
SCBQRCodeLevel	QR コード誤り訂正レベル指定定数。
SCBBitmapConverterRotation	ビットマップ回転指定定数。
SCBBlackMarkType	ブラックマーク指定定数。
SCBPrintableAreaType	印字領域タイプ指定定数。

## 8.1 モデル : ISCBBuilder 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	SM-T400i	BSC10	SM-S210i StarPRNT	SM-S220i StarPRNT	SM-S230i StarPRNT	SM-T300i StarPRNT	SM-T400i StarPRNT	SM-L200	SM-L300	SP700
Document control	beginDocument	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	endDocument	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Initialization	appendInitialization	✓	✓	✓	✓	-	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Data (Text and Command)	appendByte	✓	✓	✓	✓	-	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	appendData	✓	✓	✓	✓	-	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	appendBytes	✓	✓	✓	✓	-	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Raw data (Text and Command)	appendRawByte	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	appendRawData	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	appendRawBytes	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Font style	appendFontStyle	✓	✓	✓	✓	-	✓	✓	✓	-	-	-	-	-	✓	✓	✓	✓	✓	✓	✓	✓	✓
Code page	appendCodePage	✓	✓	✓	✓	-	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
International	appendInternational	✓	✓	✓	✓	-	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Line feed	appendLineFeed	✓	✓	✓	✓	-	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	appendDataWithLineFeed	✓	✓	✓	✓	-	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	appendBytesWithLineFeed	✓	✓	✓	✓	-	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Unit feed	appendUnitFeed	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	appendDataWithUnitFeed	✓	✓	✓	✓	*1	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	appendBytesWithUnitFeed	✓	✓	✓	✓	*1	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Character space	appendCharacterSpace	✓	✓	✓	✓	-	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Line space	appendLineSpace	✓	✓	✓	✓	-	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Top margin	appendTopMargin	-	*2 *3	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Emphasis	appendEmphasis	✓	✓	✓	✓	-	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	appendDataWithEmphasis	✓	✓	✓	✓	-	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓

Function	Method	mC-Print2	mC-Print3	mPOP	FVP10	TSP100	TSP650II	TSP700II	TSP800II	SM-S210i	SM-S220i	SM-S230i	SM-T300i	SM-T400i	BSC10	SM-S210i StarPRNT	SM-S220i StarPRNT	SM-S230i StarPRNT	SM-T300i StarPRNT	SM-T400i StarPRNT	SM-L200	SM-L300	SP700
	appendBytesWithEmphasis	✓	✓	✓	✓	-	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Invert	appendInvert	✓	✓	✓	✓	-	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	appendDataWithInvert	✓	✓	✓	✓	-	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	appendBytesWithInvert	✓	✓	✓	✓	-	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	appendMultiple	✓	✓	✓	✓	-	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Multiple	appendDataWithMultiple	✓	✓	✓	✓	-	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	appendBytesWithMultiple	✓	✓	✓	✓	-	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	appendMultipleHeight	✓	✓	✓	✓	-	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	appendDataWithMultipleHeight	✓	✓	✓	✓	-	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	appendBytesWithMultipleHeight	✓	✓	✓	✓	-	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	appendMultipleWidth	✓	✓	✓	✓	-	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	appendDataWithMultipleWidth	✓	✓	✓	✓	-	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	appendBytesWithMultipleWidth	✓	✓	✓	✓	-	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Under line	appendUnderLine	✓	✓	✓	✓	-	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	appendDataWithUnderLine	✓	✓	✓	✓	-	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	appendBytesWithUnderLine	✓	✓	✓	✓	-	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Logo	appendLogo	✓	✓	✓	✓	-	✓	✓	✓	*4	*4	*4	*4	*4	✓	✓	✓	✓	✓	✓	✓	✓	✓
Absolute position	appendAbsolutePosition	✓	✓	✓	✓	-	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	appendDataWithAbsolutePosition	✓	✓	✓	✓	-	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	appendBytesWithAbsolutePosition	✓	✓	✓	✓	-	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Alignment	appendAlignment	✓	✓	✓	✓	-	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	appendDataWithAlignment	✓	✓	✓	✓	-	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	appendBytesWithAlignment	✓	✓	✓	✓	-	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓

Function	Method	mC-Print2	mC-Print3	mPOP	FVP10	TSP100	TSP650II	TSP700II	TSP800II	SM-S210i	SM-S220i	SM-S230i	SM-T300i	SM-T400i	BSC10	SM-S210i StarPRNT	SM-S220i StarPRNT	SM-S230i StarPRNT	SM-T300i StarPRNT	SM-T400i StarPRNT	SM-L200	SM-L300	SP700
Horizontal Tab position	appendHorizontalTabPosition	✓	✓	✓	✓	-	✓	✓	✓	-	-	-	-	-	✓	✓	✓	✓	✓	✓	✓	✓	✓
Cut paper	appendCutPaper	✓	✓	✓	✓	✓	✓	✓	✓	*5	*5	*5	*5	*5	✓	*5	*5	*5	*5	*5	*5	*5	✓
Peripheral	appendPeripheral	✓	✓	✓	✓	✓	✓	✓	✓	-	-	-	-	-	✓	-	-	-	-	-	-	-	✓
Sound	appendSound	✓	✓	✓	✓	✓	✓	✓	✓	-	-	-	-	-	✓	-	-	-	-	-	-	-	✓
Barcode *6	appendBarcodeData	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	-
	appendBarcodeBytes	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	-
	appendBarcodeDataWithAbsolutePosition	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	-
	appendBarcodeBytesWithAbsolutePosition	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	-
	appendBarcodeDataWithAlignment	✓	✓	✓	✓	*7	✓	✓	✓	*7	*7	*7	*7	*7	✓	✓	✓	✓	✓	✓	✓	✓	-
	appendBarcodeBytesWithAlignment	✓	✓	✓	✓	*7	✓	✓	✓	*7	*7	*7	*7	*7	✓	✓	✓	✓	✓	✓	✓	✓	-
PDF417	appendPdf417Data	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	-
	appendPdf417Bytes	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	-
	appendPdf417DataWithAbsolutePosition	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	-
	appendPdf417BytesWithAbsolutePosition	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	-
	appendPdf417DataWithAlignment	✓	✓	✓	✓	*7	✓	✓	✓	*7	*7	*7	*7	*7	✓	✓	✓	✓	✓	✓	✓	✓	-
	appendPdf417BytesWithAlignment	✓	✓	✓	✓	*7	✓	✓	✓	*7	*7	*7	*7	*7	✓	✓	✓	✓	✓	✓	✓	✓	-
QR code	appendQrCodeData	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	-
	appendQrCodeBytes	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	-
	appendQrCodeDataWithAbsolutePosition	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	-
	appendQrCodeBytesWithAbsolutePosition	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	-

Function	Method	mC-Print2	mC-Print3	mPOP	FVP10	TSP100	TSP650II	TSP700II	TSP800II	SM-S210i	SM-S220i	SM-S230i	SM-T300i	SM-T400i	BSC10	SM-S210i StarPRNT	SM-S220i StarPRNT	SM-S230i StarPRNT	SM-T300i StarPRNT	SM-T400i StarPRNT	SM-L200	SM-L300	SP700
	appendQrCodeDataWithAlignment	✓	✓	✓	✓	*7	✓	✓	✓	*7	*7	*7	*7	*7	✓	✓	✓	✓	✓	✓	✓	✓	-
	appendQrCodeBytesWithAlignment	✓	✓	✓	✓	*7	✓	✓	✓	*7	*7	*7	*7	*7	✓	✓	✓	✓	✓	✓	✓	✓	-
Bitmap	appendBitmap	✓	✓	✓	✓	✓	✓	✓	✓	*8	*8	*8	*8	*8	✓	✓	✓	✓	✓	✓	✓	✓	✓
	appendBitmapWithAbsolutePosition	✓	✓	✓	✓	✓	✓	✓	✓	*8	*8	*8	*8	*8	✓	✓	✓	✓	✓	✓	✓	✓	✓
	appendBitmapWithAlignment	✓	✓	✓	✓	*7	✓	✓	✓	*7 *8	*7 *8	*7 *8	*7 *8	*7 *8	✓	✓	✓	✓	✓	✓	✓	✓	✓
Black mark	appendBlackMark	-	-	-	✓	-	-	✓	✓	-	-	-	*9	*9	-	-	-	-	*	*	*	*9	✓
Page mode	beginPageMode	✓	✓	✓	✓	-	✓	-	-	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	-
	endPageMode	✓	✓	✓	✓	-	✓	-	-	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	-
	appendPageModeVerticalAbsolutePosition	✓	✓	✓	✓	-	✓	-	-	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	-
	appendPageModeRotation	✓	✓	✓	✓	-	✓	-	-	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	-
Printable area	appendPrintableArea	✓	✓	✓	✓	✓	✓	✓	✓	-	-	-	-	-	✓	-	-	-	-	-	-	✓	✓
CJK Unified Ideograph Font	appendCjkUnifiedIdeographFont	✓	✓	-	-	-	*	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

- : 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 : 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）

\*3 : MCP30 は非対応

\*4 : 常にノーマルサイズ

\*5 : 用紙カットを行わない

\*6 : 印字されるバーコードが正常に読み取れることを実環境で確認すること。

\*7 : 常に左揃え

\*8 : 縦方向の最大印刷可能領域は 30cm (2400dot)まで

\*9 : プリンタ側でブラックマーク設定を有効にする必要有り。

設定方法はハードウェアマニュアル参照のこと。

\*10 : 用紙表面のブラックマークのみ有効無効設定可能。

用紙裏面のブラックマーク設定はプリンタ側で有効にする必要有り (SM-T300i/T300 は用紙裏面ブラックマーク非対応)。

設定方法はハードウェアマニュアル参照のこと。

\*11 : JP2/TW モデルのファームウェアバージョン 4.0 以降が必要。

## 8.2 beginDocument メソッド

ドキュメント開始コマンドを生成し、commands プロパティに追加します。

### 宣言

- (void)beginDocument;

### 引数

名称	説明	型
-	-	-

### 戻り値

説明	型
-	-

### Example

```
+ (NSData *)createData:(StarIoExtEmulation)emulation {
    NSData *otherData = [@"Hello World.\n" dataUsingEncoding:NSUTF8StringEncoding];

    ISCBBuilder *builder = [StarIoExt createCommandBuilder:emulation];

    [builder beginDocument];

    [builder appendData:otherData];

    [builder appendCutPaper:SCBCutPaperActionPartialCutWithFeed];

    [builder endDocument];

    return [builder.commands copy];
}
```

*ApiFunctions.m* 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、commands プロパティに追加します。

#### 宣言

- (void)endDocument;

#### 引数

名称	説明	型
-	-	-

#### 戻り値

説明	型
-	-

#### Example

```
+ (NSData *)createData:(StarIoExtEmulation)emulation {
    NSData *otherData = [@"Hello World.\n" dataUsingEncoding:NSUTF8StringEncoding];

    ISCBBuilder *builder = [StarIoExt createCommandBuilder:emulation];

    [builder beginDocument];

    [builder appendData:otherData];

    [builder appendCutPaper:SCBCutPaperActionPartialCutWithFeed];

    [builder endDocument];

    return [builder.commands copy];
}
```

ApiFunctions.m 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、commands プロパティに追加します。

### 宣言

- (void)appendInitialization:(SCBInitializationType)type;

### 引数

名称	説明	型
type	初期化タイプ <ul style="list-style-type: none"> <li>SCBInitializationTypeCommand ... コマンド初期化</li> <li>SCBInitializationTypeReset ... プリンタリセット</li> <li>SCBInitializationTypeResetWithPrint ... プリンタリセット (自己印字実行)</li> </ul>	SCBInitializationType

### 戻り値

説明	型
-	-

### Example

```
+ (NSData *)createData:(StarloExtEmulation)emulation {
    NSData *otherData = [@"Hello World.\n" dataUsingEncoding:NSUTF8StringEncoding];

    ISCBBuilder *builder = [StarloExt createCommandBuilder:emulation];

    [builder beginDocument];

    [builder appendUnderLine:YES];
    [builder appendMultiple:2 height:2];
    [builder appendData:otherData];

    [builder appendInitialization:SCBInitializationTypeCommand];

    [builder appendData:otherData];

    [builder appendCutPaper:SCBCutPaperActionPartialCutWithFeed];

    [builder endDocument];

    return [builder.commands copy];
}
```

ApiFunctions.m を参照ください。

## 8.5 appendData メソッド

データ（テキストやコマンド）を commands プロパティに追加します。

### 宣言

- (void)appendByte:(unsigned char)data;
- (void)appendData:(NSData \*)otherData;
- (void)appendBytes:(const void \*)bytes length:(NSUInteger)length;

### 引数

名称	説明	型
data	データ（テキストやコマンド）	unsigned char
otherData		NSData
bytes		const void *
length		NSUInteger

### 戻り値

説明	型
-	-

### Example

```
+ (NSData *)createData:(StarIoExtEmulation)emulation {
    NSData *otherData = [@"Hello World." dataUsingEncoding:NSUTF8StringEncoding];

    unsigned char bytes[] = {0x48, 0x65, 0x6c, 0x6c, 0x6f, 0x20, 0x57, 0x6f, 0x72, 0x6c, 0x64, 0x2e};

    NSUInteger length = sizeof(bytes);

    ISCBBuilder *builder = [StarIoExt createCommandBuilder:emulation];

    [builder beginDocument];

    [builder appendData:otherData];
    [builder appendByte:0x0d];

    [builder appendBytes:bytes length:length];
    [builder appendByte:0x0d];

    [builder appendCutPaper:SCBCutPaperActionPartialCutWithFeed];

    [builder endDocument];

    return [builder.commands copy];
}
```

ApiFunctions.m /各コマンド仕様書を参照ください。

## 8.6 appendRawData メソッド

生のデータ（テキストやコマンド）を commands プロパティに追加します。

### 宣言

- (void)appendRawByte:(unsigned char)data;
- (void)appendRawData:(NSData \*)otherData;
- (void)appendRawBytes:(const void \*)bytes length:(NSUInteger)length;

### 引数

名称	説明	型
data	生のデータ（テキストやコマンド）	unsigned char
otherData		NSData
bytes		const void *
length		NSUInteger

### 戻り値

説明	型
-	-

### Example

```
+ (NSData *)createData:(StarIoExtEmulation)emulation image:(UIImage *)image {
    NSString *urlString = [SMCSAllReceipts uploadBitmap:image completion:nil];

    ISCBBuilder *builder = [StarIoExt createCommandBuilder:emulation];

    [builder beginDocument];

    [builder appendBitmap:image diffusion:NO];

    NSData *data = [SMCSAllReceipts generateAllReceipts:urlString emulation:emulation info:YES qrCode:YES];
    [builder appendRawData:data];

    [builder appendCutPaper:SCBCutPaperActionPartialCutWithFeed];

    [builder endDocument];

    return [builder.commands copy];
}
```

AllReceiptsFunctions.m /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、commands プロパティに追加します。

### 宣言

```
- (void)appendFontStyle:(SCBFontStyleType)type;
```

### 引数

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

### 戻り値

説明	型
-	-

### Example

```
+ (NSData *)createData:(StarIoExtEmulation)emulation {
    NSData *otherData = [@"Hello World.\n" dataUsingEncoding:NSUTF8StringEncoding];

    ISCBBuilder *builder = [StarIoExt createCommandBuilder:emulation];

    [builder beginDocument];

    [builder appendFontStyle:SCBFontStyleTypeA];
    [builder appendData:otherData];
    [builder appendFontStyle:SCBFontStyleTypeB];
    [builder appendData:otherData];

    [builder appendCutPaper:SCBCutPaperActionPartialCutWithFeed];

    [builder endDocument];

    return [builder.commands copy];
}
```

ApiFunctions.m /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、commands プロパティに追加します。

### 宣言

```
(void)appendCodePage:(SCBCodePageType)type;
```

### 引数

名称	説明	型
type	<p>コードページ</p> <ul style="list-style-type: none"> <li>• SCBCodePageTypeCP437 ... CodePage437 (USA, Std. Europe).</li> <li>• SCBCodePageTypeCP737 ... Codepage 737 (Greek).</li> <li>• SCBCodePageTypeCP772 ... Codepage 772 (Lithuanian).</li> <li>• SCBCodePageTypeCP774 ... Codepage 774 (Lithuanian).</li> <li>• SCBCodePageTypeCP851 ... Codepage 851 (Greek).</li> <li>• SCBCodePageTypeCP852 ... Codepage 852 (Latin-2).</li> <li>• SCBCodePageTypeCP855 ... Codepage 855 (Cyrillic Bulgarian).</li> <li>• SCBCodePageTypeCP857 ... Codepage 857 (Turkey).</li> <li>• SCBCodePageTypeCP858 ... Codepage 858 (Multilingual).</li> <li>• SCBCodePageTypeCP860 ... Codepage 860 (Portuguese).</li> <li>• SCBCodePageTypeCP861 ... Codepage 861 (Icelandic).</li> <li>• SCBCodePageTypeCP862 ... Codepage 862 (Israel (Hebrew)).</li> <li>• SCBCodePageTypeCP863 ... Codepage 863 (Canadian French).</li> <li>• SCBCodePageTypeCP864 ... Codepage 864 (Arabic).</li> <li>• SCBCodePageTypeCP865 ... Codepage 865 (Nordic).</li> <li>• SCBCodePageTypeCP866 ... Codepage 866 (Cyrillic Russian).</li> <li>• SCBCodePageTypeCP869 ... Codepage 869 (Greek).</li> <li>• SCBCodePageTypeCP874 ... Codepage 874 (Thai).</li> <li>• SCBCodePageTypeCP928 ... Codepage 928 (Greek).</li> <li>• SCBCodePageTypeCP932 ... Katakana.</li> <li>• SCBCodePageTypeCP998 ... Normal.</li> <li>• SCBCodePageTypeCP999 ... Codepage 1252 (Windows Latin-1).</li> <li>• SCBCodePageTypeCP1001 ... Codepage 1001 (Arabic).</li> <li>• SCBCodePageTypeCP1250 ... Codepage 1250 (Windows Latin-2).</li> <li>• SCBCodePageTypeCP1251 ... Codepage 1251 (Windows Cyrillic).</li> <li>• SCBCodePageTypeCP1252 ... Codepage 1252 (Windows Latin-1).</li> <li>• SCBCodePageTypeCP2001 ... Codepage 2001 (Lithuanian-KBL).</li> <li>• SCBCodePageTypeCP3001 ... Codepage 3001 (Estonian-1).</li> <li>• SCBCodePageTypeCP3002 ... Codepage 3002 (Estonian-2).</li> <li>• SCBCodePageTypeCP3011 ... Codepage 3011 (Latvian-1).</li> <li>• SCBCodePageTypeCP3012 ... Codepage 3012 (Latvian-2).</li> <li>• SCBCodePageTypeCP3021 ... Codepage 3021 (Bulgarian).</li> <li>• SCBCodePageTypeCP3041 ... Codepage 3041 (Maltese).</li> <li>• SCBCodePageTypeCP3840 ... Codepage 3840 (IBM-Russian).</li> <li>• SCBCodePageTypeCP3841 ... Codepage 3841 (Gost).</li> </ul>	SCBCodePageType

名称	説明	型
	<ul style="list-style-type: none"> <li>SCBCodePageTypeCP3843 ... Codepage 3843 (Polish).</li> <li>SCBCodePageTypeCP3844 ... Codepage 3844 (CS2).</li> <li>SCBCodePageTypeCP3845 ... Codepage 3845 (Hungarian).</li> <li>SCBCodePageTypeCP3846 ... Codepage 3846 (Turkish).</li> <li>SCBCodePageTypeCP3847 ... Codepage 3847 (Brazil-ABNT).</li> <li>SCBCodePageTypeCP3848 ... Codepage 3848 (Brazil-ABICOMP).</li> <li>SCBCodePageTypeUTF8 ... UTF-8.</li> <li>SCBCodePageTypeBlank ... User Setting Blank Code Page.</li> </ul>	

## 戻り値

説明	型
-	-

## Example

```
+ (NSData *)createData:(StarloExtEmulation)emulation {
    unsigned char bytes8[] = {0x80, 0x81, 0x82, 0x83, 0x84, 0x85, 0x86, 0x87, 0x88, 0x89, 0x8a, 0x8b, 0x8c, 0x8d, 0x8e, 0x8f, 0x0a};
    unsigned char bytes9[] = {0x90, 0x91, 0x92, 0x93, 0x94, 0x95, 0x96, 0x97, 0x98, 0x99, 0x9a, 0x9b, 0x9c, 0x9d, 0x9e, 0x9f, 0x0a};
    unsigned char bytesA[] = {0xa0, 0xa1, 0xa2, 0xa3, 0xa4, 0xa5, 0xa6, 0xa7, 0xa8, 0xa9, 0xaa, 0xab, 0xac, 0xad, 0xae, 0xaf, 0x0a};
    unsigned char bytesB[] = {0xb0, 0xb1, 0xb2, 0xb3, 0xb4, 0xb5, 0xb6, 0xb7, 0xb8, 0xb9, 0xba, 0xbb, 0xbc, 0xbd, 0xbe, 0xbf, 0x0a};
    unsigned char bytesC[] = {0xc0, 0xc1, 0xc2, 0xc3, 0xc4, 0xc5, 0xc6, 0xc7, 0xc8, 0xc9, 0xca, 0xcb, 0xcc, 0xcd, 0xce, 0xcf, 0x0a};
    unsigned char bytesD[] = {0xd0, 0xd1, 0xd2, 0xd3, 0xd4, 0xd5, 0xd6, 0xd7, 0xd8, 0xd9, 0xda, 0xdb, 0xdc, 0xdd, 0xde, 0xdf, 0x0a};
    unsigned char bytesE[] = {0xe0, 0xe1, 0xe2, 0xe3, 0xe4, 0xe5, 0xe6, 0xe7, 0xe8, 0xe9, 0xea, 0xeb, 0xec, 0xed, 0xee, 0xef, 0x0a};
    unsigned char bytesF[] = {0xf0, 0xf1, 0xf2, 0xf3, 0xf4, 0xf5, 0xf6, 0xf7, 0xf8, 0xf9, 0xfa, 0xfb, 0xfc, 0xfd, 0xfe, 0xff, 0x0a};

    NSUInteger length = sizeof(bytes8);

    ISCBBuilder *builder = [StarloExt createCommandBuilder:emulation];

    [builder beginDocument];

    [builder appendCodePage:SCBCodePageTypeCP932];

    [builder appendBytes:bytes8 length:length];
    [builder appendBytes:bytes9 length:length];
    [builder appendBytes:bytesA length:length];
    [builder appendBytes:bytesB length:length];
    [builder appendBytes:bytesC length:length];
    [builder appendBytes:bytesD length:length];
    [builder appendBytes:bytesE length:length];
    [builder appendBytes:bytesF length:length];

    [builder appendCutPaper:SCBCutPaperActionPartialCutWithFeed];

    [builder endDocument];

    return [builder.commands copy];
}
```

ApiFunctions.m /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、commands プロパティに追加します。

### 宣言

```
- (void)appendInternational:(SCBInternationalType)type;
```

### 引数

名称	説明	型
type	国際文字 <ul style="list-style-type: none"> <li>• SCBInternationalTypeUSA ... USA</li> <li>• SCBInternationalTypeFrance ... フランス</li> <li>• SCBInternationalTypeGermany ... ドイツ</li> <li>• SCBInternationalTypeUK ... イギリス</li> <li>• SCBInternationalTypeDenmark ... デンマーク</li> <li>• SCBInternationalTypeSweden ... スウェーデン</li> <li>• SCBInternationalTypeItaly ... イタリア</li> <li>• SCBInternationalTypeSpain ... スペイン</li> <li>• SCBInternationalTypeJapan ... 日本</li> <li>• SCBInternationalTypeNorway ... ノルウェー</li> <li>• SCBInternationalTypeDenmark2 ... デンマーク II</li> <li>• SCBInternationalTypeSpain2 ... スペイン II</li> <li>• SCBInternationalTypeLatinAmerica ... ラテンアメリカ</li> <li>• SCBInternationalTypeKorea ... 韓国</li> <li>• SCBInternationalTypeIreland ... アイルランド</li> <li>• SCBInternationalTypeLegal ... Legal</li> </ul>	SCBInternationalType

### 戻り値

説明	型
-	-



### Example

```
+ (NSData *)createData:(StarIoExtEmulation)emulation {
    unsigned char bytes[] = {0x23, 0x24, 0x40, 0x58, 0x5a, 0x5b, 0x5c, 0x5d, 0x5e, 0x60, 0x7b, 0x7c, 0x7d, 0x7e, 0x0a};

    NSUInteger length = sizeof(bytes);

    ISCBBuilder *builder = [StarIoExt createCommandBuilder:emulation];

    [builder beginDocument];

    [builder appendInternational:SCBInternationalTypeUSA];
    [builder appendBytes:bytes length:length];

    [builder appendInternational:SCBInternationalTypeJapan];
    [builder appendBytes:bytes length:length];

    [builder appendCutPaper:SCBCutPaperActionPartialCutWithFeed];

    [builder endDocument];

    return [builder.commands copy];
}
```

ApiFunctions.m /各コマンド仕様書を参照ください。

## 8.10 appendLineFeed メソッド

行単位の用紙送りコマンドを生成し、commands プロパティに追加します。

### 宣言

- (void)appendLineFeed;
- (void)appendDataWithLineFeed:(NSData \*)otherData;
- (void)appendBytesWithLineFeed:(const void \*)bytes length:(NSUInteger)length;
- (void)appendLineFeed:(NSInteger)line;
- (void)appendDataWithLineFeed:(NSData \*)otherData line:(NSInteger)line;
- (void)appendBytesWithLineFeed:(const void \*)bytes length:(NSUInteger)length line:(NSInteger)line;

### 引数

名称	説明	型
line	紙送り量（行数単位）	NSInteger
otherData	データ（テキストやコマンド）	NSData
bytes		const void *
length		NSUInteger

### 戻り値

説明	型
-	-

### Example

```
+ (NSData *)createData:(StarloExtEmulation)emulation {
    NSData *otherData = [@"Hello World." dataUsingEncoding:NSUTF8StringEncoding];

    ISCBBuilder *builder = [StarloExt createCommandBuilder:emulation];

    [builder beginDocument];

    [builder appendData:otherData];
    [builder appendLineFeed];

    [builder appendDataWithLineFeed:otherData];

    [builder appendData:otherData];
    [builder appendLineFeed:2];

    [builder appendDataWithLineFeed:otherData line:2];

    [builder appendCutPaper:SCBCutPaperActionPartialCutWithFeed];

    [builder endDocument];

    return [builder.commands copy];
}
```

ApiFunctions.m 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、commands プロパティに追加します。

### 宣言

- (void)appendUnitFeed:(NSInteger)unit;
- (void)appendDataWithUnitFeed:(NSData \*)otherData unit:(NSInteger)unit;
- (void)appendBytesWithUnitFeed:(const void \*)bytes length:(NSUInteger)length unit:(NSInteger)unit;

### 引数

名称	説明	型
unit	紙送り量（ドット数単位）	NSInteger
otherData	データ（テキストやコマンド）	NSData
bytes		const void *
length		NSUInteger

### 戻り値

説明	型
-	-

### Example

```
+ (NSData *)createData:(StarIoExtEmulation)emulation {
    NSData *otherData = [@"Hello World." dataUsingEncoding:NSUTF8StringEncoding];

    ISCBBuilder *builder = [StarIoExt createCommandBuilder:emulation];

    [builder beginDocument];

    [builder appendData:otherData];
    [builder appendUnitFeed:64];

    [builder appendDataWithUnitFeed:otherData unit:64];

    [builder appendCutPaper:SCBCutPaperActionPartialCutWithFeed];

    [builder endDocument];

    return [builder.commands copy];
}
```

ApiFunctions.m 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、commands プロパティに追加します。

### 宣言

- (void)appendCharacterSpace:(NSInteger)space;

### 引数

名称	説明	型
space	文字間スペース（ドット数単位）	NSInteger

### 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

```
+ (NSData *)createData:(StarloExtEmulation)emulation {
    NSData *otherData = [@"Hello World.\n" dataUsingEncoding:NSUTF8StringEncoding];

    ISCBBuilder *builder = [StarloExt createCommandBuilder:emulation];

    [builder beginDocument];

    [builder appendCharacterSpace:0];
    [builder appendData:otherData];
    [builder appendCharacterSpace:4];
    [builder appendData:otherData];

    [builder appendCutPaper:SCBCutPaperActionPartialCutWithFeed];

    [builder endDocument];

    return [builder.commands copy];
}
```

ApiFunctions.m 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、commands プロパティに追加します。

### 宣言

- (void)appendLineSpace:(NSInteger)lineSpace;

### 引数

名称	説明	型
lineSpace	行間スペース（ドット数単位）	NSInteger

### 戻り値

説明	型
-	-

### Example

```
+ (NSData *)createData:(StarIoExtEmulation)emulation {
    NSData *otherData = [@"Hello World.\n" dataUsingEncoding:NSUTF8StringEncoding];

    ISCBBuilder *builder = [StarIoExt createCommandBuilder:emulation];

    [builder beginDocument];

    [builder appendLineSpace:32];
    [builder appendData:otherData];
    [builder appendData:otherData];
    [builder appendData:otherData];
    [builder appendLineSpace:24];
    [builder appendData:otherData];
    [builder appendData:otherData];
    [builder appendData:otherData];

    [builder appendCutPaper:SCBCutPaperActionPartialCutWithFeed];

    [builder endDocument];

    return [builder.commands copy];
}
```

ApiFunctions.m /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、commands プロパティに追加します。

### 宣言

- (void)appendTopMargin:(NSInteger)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)	NSInteger

### 戻り値

説明	型
-	-

### Example

```
+ (NSData *)createTopMarginData:(StarloExtEmulation)emulation {
    NSData *data = [@"Hello, World.\n" dataUsingEncoding:NSUTF8StringEncoding];

    ISCBBuilder *builder = [StarloExt createCommandBuilder:StarloExtEmulationStarPRNT];

    [builder beginDocument];

    [builder appendTopMargin:2];
    [builder appendData:[@"**Top margin:2mm*\n" dataUsingEncoding:NSUTF8StringEncoding]];
    [builder appendData:data];
    [builder appendData:data];
    [builder appendData:data];

    [builder appendCutPaper:SCBCutPaperActionPartialCutWithFeed];

    [builder appendTopMargin:6];
    [builder appendData:[@"**Top margin:6mm*\n" dataUsingEncoding:NSUTF8StringEncoding]];
    [builder appendData:data];
    [builder appendData:data];
    [builder appendData:data];

    [builder appendCutPaper:SCBCutPaperActionPartialCutWithFeed];

    [builder appendTopMargin:11];
    [builder appendData:[@"**Top margin:11mm*\n" dataUsingEncoding:NSUTF8StringEncoding]];
    [builder appendData:data];
    [builder appendData:data];
    [builder appendData:data];

    [builder appendCutPaper:SCBCutPaperActionPartialCutWithFeed];

    [builder endDocument];

    return builder.commands;
}
```

ApiFunctions.m 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、commands プロパティに追加します。

### 宣言

- (void)appendEmphasis:(BOOL)emphasis;
- (void)appendDataWithEmphasis:(NSData \*)otherData;
- (void)appendBytesWithEmphasis:(const void \*)bytes length:(NSUInteger)length;

### 引数

名称	説明	型
emphasis	強調 <ul style="list-style-type: none"> <li>YES ... 強調設定</li> <li>NO ... 強調解除</li> </ul>	BOOL
otherData	データ (テキストやコマンド)	NSData
bytes		const void *
length		NSUInteger

### 戻り値

説明	型
-	-

### Example

```
+ (NSData *)createData:(StarloExtEmulation)emulation {
    NSData *otherData = [@"Hello World.\n" dataUsingEncoding:NSUTF8StringEncoding];
    NSData *otherDataHalf0 = [@"Hello " dataUsingEncoding:NSUTF8StringEncoding];
    NSData *otherDataHalf1 = [@"World.\n" dataUsingEncoding:NSUTF8StringEncoding];

    ISCBBuilder *builder = [StarloExt createCommandBuilder:emulation];

    [builder beginDocument];

    [builder appendData:otherData];
    [builder appendEmphasis:YES];
    [builder appendData:otherData];
    [builder appendData:otherData];
    [builder appendEmphasis:NO];
    [builder appendData:otherData];

    [builder appendDataWithEmphasis:otherData];
    [builder appendData:otherData];

    [builder appendDataWithEmphasis:otherDataHalf0];
    [builder appendData:otherDataHalf1];

    [builder appendCutPaper:SCBCutPaperActionPartialCutWithFeed];

    [builder endDocument];

    return [builder.commands copy];
}
```

ApiFunctions.m 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、commands プロパティに追加します。

### 宣言

- (void)appendInvert:(BOOL)invert;
- (void)appendDataWithInvert:(NSData \*)otherData;
- (void)appendBytesWithInvert:(const void \*)bytes length:(NSUInteger)length;

### 引数

名称	説明	型
invert	白黒反転 <ul style="list-style-type: none"> <li>YES ... 白黒反転設定</li> <li>NO ... 白黒反転解除</li> </ul>	BOOL
otherData		NSData
bytes	データ (テキストやコマンド)	const void *
length		NSUInteger

### 戻り値

説明	型
-	-

### Example

```
+ (NSData *)createData:(StarloExtEmulation)emulation {
    NSData *otherData = [@"Hello World.\n" dataUsingEncoding:NSUTF8StringEncoding];
    NSData *otherDataHalf0 = [@"Hello " dataUsingEncoding:NSUTF8StringEncoding];
    NSData *otherDataHalf1 = [@"World.\n" dataUsingEncoding:NSUTF8StringEncoding];

    ISCBBuilder *builder = [StarloExt createCommandBuilder:emulation];

    [builder beginDocument];

    [builder appendData:otherData];
    [builder appendInvert:YES];
    [builder appendData:otherData];
    [builder appendData:otherData];
    [builder appendInvert:NO];
    [builder appendData:otherData];

    [builder appendDataWithInvert:otherData];
    [builder appendData:otherData];

    [builder appendDataWithInvert:otherDataHalf0];
    [builder appendData:otherDataHalf1];

    [builder appendCutPaper:SCBCutPaperActionPartialCutWithFeed];

    [builder endDocument];

    return [builder.commands copy];
}
```

ApiFunctions.m 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、commands プロパティに追加します。

### 宣言

- (void)appendMultiple:(NSInteger)width height:(NSInteger)height;
- (void)appendDataWithMultiple:(NSData \*)otherData width:(NSInteger)width height:(NSInteger)height;
- (void)appendBytesWithMultiple:(const void \*)bytes length:(NSUInteger)length width:(NSInteger)width height:(NSInteger)height;
- (void)appendMultipleHeight:(NSInteger)height;
- (void)appendDataWithMultipleHeight:(NSData \*)otherData height:(NSInteger)height;
- (void)appendBytesWithMultipleHeight:(const void \*)bytes length:(NSUInteger)length height:(NSInteger)height;
- (void)appendMultipleWidth:(NSInteger)width;
- (void)appendDataWithMultipleWidth:(NSData \*)otherData width:(NSInteger)width;
- (void)appendBytesWithMultipleWidth:(const void \*)bytes length:(NSUInteger)length width:(NSInteger)width;

### 引数

名称	説明	型
width	横倍率	NSInteger
height	縦倍率	NSInteger
otherData	データ（テキストやコマンド）	NSData
bytes		const void *
length		NSUInteger

### 戻り値

説明	型
-	-

## Example

```
+ (NSData *)createData:(StarIoExtEmulation)emulation {
    NSData *otherData = [@"Hello World.\n" dataUsingEncoding:NSUTF8StringEncoding];
    NSData *otherDataHalf0 = [@"Hello " dataUsingEncoding:NSUTF8StringEncoding];
    NSData *otherDataHalf1 = [@"World.\n" dataUsingEncoding:NSUTF8StringEncoding];

    ISCBBuilder *builder = [StarIoExt createCommandBuilder:emulation];

    [builder beginDocument];

    [builder appendData:otherData];

    [builder appendMultiple:2 height:2];
    [builder appendData:otherData];
    [builder appendData:otherData];
    [builder appendMultiple:1 height:1];
    [builder appendData:otherData];

    [builder appendDataWithMultiple:otherData width:2 height:2];
    [builder appendData:        otherData];

    [builder appendDataWithMultiple:otherDataHalf0 width:2 height:2];
    [builder appendData:        otherDataHalf1];

    [builder appendData:otherDataHalf0];
    [builder appendDataWithMultiple:otherDataHalf1 width:2 height:2];

    [builder appendMultipleHeight:2];
    [builder appendData:otherData];
    [builder appendData:otherData];
    [builder appendMultipleHeight:1];
    [builder appendData:otherData];

    [builder appendDataWithMultipleHeight:otherDataHalf0 height:2];
    [builder appendData:otherDataHalf1];

    [builder appendData:otherDataHalf0];
    [builder appendDataWithMultipleHeight:otherDataHalf1 height:2];

    [builder appendMultipleWidth:2];
    [builder appendData:otherData];
    [builder appendData:otherData];
    [builder appendMultipleWidth:1];
    [builder appendData:otherData];

    [builder appendDataWithMultipleWidth:otherDataHalf0 width:2];
    [builder appendData:otherDataHalf1];

    [builder appendData:otherDataHalf0];
    [builder appendDataWithMultipleWidth:otherDataHalf1 width:2];

    [builder appendCutPaper:SCBCutPaperActionPartialCutWithFeed];

    [builder endDocument];

    return [builder.commands copy];
}
```

ApiFunctions.m /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、commands プロパティに追加します。

### 宣言

- (void)appendUnderLine:(BOOL)underLine;
- (void)appendDataWithUnderLine:(NSData \*)otherData;
- (void)appendBytesWithUnderLine:(const void \*)bytes length:(NSUInteger)length;

### 引数

名称	説明	型
underLine	アンダーライン <ul style="list-style-type: none"> <li>YES ... アンダーライン設定</li> <li>NO ... アンダーライン解除</li> </ul>	BOOL
otherData		NSData
bytes	データ (テキストやコマンド)	const void *
length		NSUInteger

### 戻り値

説明	型
-	-

### Example

```
+ (NSData *)createData:(StarloExtEmulation)emulation {
    NSData *otherData = [@"Hello World.\n" dataUsingEncoding:NSUTF8StringEncoding];
    NSData *otherDataHalf0 = [@"Hello " dataUsingEncoding:NSUTF8StringEncoding];
    NSData *otherDataHalf1 = [@"World.\n" dataUsingEncoding:NSUTF8StringEncoding];

    ISCBBuilder *builder = [StarloExt createCommandBuilder:emulation];

    [builder beginDocument];

    [builder appendData:otherData];

    [builder appendUnderLine:YES];
    [builder appendData:otherData];
    [builder appendData:otherData];
    [builder appendUnderLine:NO];
    [builder appendData:otherData];

    [builder appendDataWithUnderLine:otherData];
    [builder appendData:otherData];

    [builder appendDataWithUnderLine:otherDataHalf0];
    [builder appendData:otherDataHalf1];

    [builder appendCutPaper:SCBCutPaperActionPartialCutWithFeed];

    [builder endDocument];

    return [builder.commands copy];
}
```

ApiFunctions.m 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、commands プロパティに追加します。

### 宣言

- (void)appendLogo:(SCBLogoSize)size number:(NSInteger)number;

### 引数

名称	説明	型
size	ロゴサイズ <ul style="list-style-type: none"> <li>SCBLogoSizeNormal ... ノーマルモード</li> <li>SCBLogoSizeDoubleWidth ... 横 2 倍モード</li> <li>SCBLogoSizeDoubleHeight ... 縦 2 倍モード</li> <li>SCBLogoSizeDoubleWidthDoubleHeight ... 縦 2 倍、横 2 倍モード</li> </ul>	SCBLogoSize
number	ロゴ番号	NSInteger

### 戻り値

説明	型
-	-

### Example

```
+ (NSData *)createData:(StarloExtEmulation)emulation {
    ISCBBuilder *builder = [StarloExt createCommandBuilder:emulation];

    [builder beginDocument];

    [builder appendLogo:SCBLogoSizeNormal number:1];
    [builder appendLogo:SCBLogoSizeDoubleWidth number:1];
    [builder appendLogo:SCBLogoSizeDoubleHeight number:1];
    [builder appendLogo:SCBLogoSizeDoubleWidthDoubleHeight number:1];

    [builder appendCutPaper:SCBCutPaperActionPartialCutWithFeed];

    [builder endDocument];

    return [builder.commands copy];
}
```

ApiFunctions.m /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、commands プロパティに追加します。

### 宣言

- (void)appendAbsolutePosition:(NSInteger)position;
- (void)appendDataWithAbsolutePosition:(NSData \*)otherData position:(NSInteger)position;
- (void)appendBytesWithAbsolutePosition:(const void \*)bytes length:(NSUInteger)length position:(NSInteger)position;

### 引数

名称	説明	型
position	絶対位置（ドット数単位）	NSInteger
otherData	データ（テキストやコマンド）	NSData
bytes		const void *
length		NSUInteger

### 戻り値

説明	型
-	-

### Example

```
+ (NSData *)createData:(StarIoExtEmulation)emulation {
    NSData *otherData = [@"Hello World.\n" dataUsingEncoding:NSUTF8StringEncoding];

    ISCBBuilder *builder = [StarIoExt createCommandBuilder:emulation];

    [builder beginDocument];

    [builder appendData:otherData];

    [builder appendAbsolutePosition:40];
    [builder appendData:otherData];
    [builder appendData:otherData];

    [builder appendDataWithAbsolutePosition:otherData position:40];
    [builder appendData:otherData];

    [builder appendCutPaper:SCBCutPaperActionPartialCutWithFeed];

    [builder endDocument];

    return [builder.commands copy];
}
```

ApiFunctions.m 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、commands プロパティに追加します。

### 宣言

```
- (void)appendAlignment:(SCBAlignmentPosition)position;

- (void)appendDataWithAlignment:(NSData *)otherData position:(SCBAlignmentPosition)position;

- (void)appendBytesWithAlignment:(const void *)bytes length:(NSUInteger)length
position:(SCBAlignmentPosition)position;
```

### 引数

名称	説明	型
position	位置揃え <ul style="list-style-type: none"> <li>SCBAlignmentPositionLeft ... 左揃え</li> <li>SCBAlignmentPositionCenter ... 中心揃え</li> <li>SCBAlignmentPositionRight ... 右揃え</li> </ul>	SCBAlignmentPosition
otherData	データ（テキストやコマンド）	NSData
bytes		const void *
length		NSUInteger

### 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

```
+ (NSData *)createData:(StarIoExtEmulation)emulation {
    NSData *otherData = [@"Hello World.\n" dataUsingEncoding:NSUTF8StringEncoding];

    ISCBBuilder *builder = [StarIoExt createCommandBuilder:emulation];

    [builder beginDocument];

    [builder appendData:otherData];

    [builder appendAlignment:SCBAlignmentPositionRight];
    [builder appendData:otherData];
    [builder appendData:otherData];
    [builder appendAlignment:SCBAlignmentPositionLeft];
    [builder appendData:otherData];
    [builder appendData:otherData];

    [builder appendDataWithAlignment:otherData position:SCBAlignmentPositionRight];
    [builder appendData:otherData];

    [builder appendCutPaper:SCBCutPaperActionPartialCutWithFeed];

    [builder endDocument];

    return [builder.commands copy];
}
```

ApiFunctions.m /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、commands プロパティに追加します。

### 宣言

```
-(void)appendHorizontalTabPosition:(NSArray<NSNumber *> *)positions;
```

### 引数

名称	説明	型
positions	水平タブ位置（ANK 文字ピッチ単位）のリスト  nil もしくは空のリストを指定した場合、現在設定されている水平タブ位置をすべてクリアします。	NSArray<NSNumber *> *

### 戻り値

説明	型
-	-

### Example

```
+ (NSData *)createHorizontalTabPositionData:(StarIoExtEmulation)emulation {
    NSData *otherData1 = [@"QTY\tITEM\tTOTAL\n" dataUsingEncoding:NSUTF8StringEncoding];
    NSData *otherData2 = [@"1\tApple\t1.50\n" dataUsingEncoding:NSUTF8StringEncoding];
    NSData *otherData3 = [@"2\tOrange\t2.00\n" dataUsingEncoding:NSUTF8StringEncoding];
    NSData *otherData4 = [@"5\tBanana\t3.00\n" dataUsingEncoding:NSUTF8StringEncoding];

    NSArray<NSNumber *> *positions = @[5, 27];

    ISCBBuilder *builder = [StarIoExt createCommandBuilder:emulation];

    [builder beginDocument];

    [builder appendHorizontalTabPosition:positions];

    [builder appendData:[@"*Tab Position:5, 27\n" dataUsingEncoding:NSUTF8StringEncoding]];

    [builder appendData:otherData1];
    [builder appendData:otherData2];
    [builder appendData:otherData3];
    [builder appendData:otherData4];

    [builder appendCutPaper:SCBCutPaperActionPartialCutWithFeed];

    [builder endDocument];

    return [builder.commands copy];
}
```

ApiFunctions.m 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、commands プロパティに追加します。

### 宣言

```
- (void)appendCutPaper:(SCBCutPaperAction)action;
```

### 引数

名称	説明	型
action	用紙カット <ul style="list-style-type: none"> <li>SCBCutPaperActionFullCut ... フルカット</li> <li>SCBCutPaperActionPartialCut ... パーシャルカット</li> <li>SCBCutPaperActionFullCutWithFeed ... 用紙送り付きフルカット</li> <li>SCBCutPaperActionPartialCutWithFeed ... 用紙送り付きパーシャルカット</li> </ul>	SCBCutPaperAction

### 戻り値

説明	型
-	-

### Example

```
+ (NSData *)createData:(StarIoExtEmulation)emulation {
    NSData *otherData = [@"Hello World.\n" dataUsingEncoding:NSUTF8StringEncoding];

    ISCBBuilder *builder = [StarIoExt createCommandBuilder:emulation];

    [builder beginDocument];

    [builder appendData:otherData];
    [builder appendData:otherData];
    [builder appendData:otherData];
    [builder appendData:otherData];
    [builder appendData:otherData];

    [builder appendCutPaper:SCBCutPaperActionPartialCutWithFeed];

    [builder endDocument];

    return [builder.commands copy];
}
```

ApiFunctions.m 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、commands プロパティに追加します。

### 宣言

```
- (void)appendPeripheral:(SCBPeripheralChannel)channel;
- (void)appendPeripheral:(SCBPeripheralChannel)channel time:(NSInteger)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"> <li>SCBPeripheralChannelNo1 ... チャンネル 1</li> <li>SCBPeripheralChannelNo2 ... チャンネル 2</li> </ul>	SCBPeripheralChannel
time	駆動時間（1 ミリ秒単位） ※チャンネル 1 のみ有効	NSInteger

### 戻り値

説明	型
-	-

### Example

```
+ (NSData *)createData:(StarIoExtEmulation)emulation {
    ISCBBuilder *builder = [StarIoExt createCommandBuilder:emulation];

    [builder beginDocument];

    [builder appendPeripheral:SCBPeripheralChannelNo1];
    [builder appendPeripheral:SCBPeripheralChannelNo2];
    [builder appendPeripheral:SCBPeripheralChannelNo1 time:2000];
    [builder appendPeripheral:SCBPeripheralChannelNo2 time:2000];

    [builder endDocument];

    return [builder.commands copy];
}
```

ApiFunctions.m /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、commands プロパティに追加します。

### 宣言

```
- (void)appendSound:(SCBSoundChannel)channel;
- (void)appendSound:(SCBSoundChannel)channel repeat:(NSInteger)repeat;
- (void)appendSound:(SCBSoundChannel)channel repeat:(NSInteger)repeat driveTime:(NSInteger) driveTime
delayTime:(NSInteger)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"> <li>SCBSoundChannelNo1 ... チャンネル 1</li> <li>SCBSoundChannelNo2 ... チャンネル 2</li> </ul>	SCBSoundChannel
repeat	繰り返し回数	NSInteger
driveTime	駆動時間 (1 ミリ秒単位)	NSInteger
delayTime	ディレイ時間 (1 ミリ秒単位)	NSInteger

### 戻り値

説明	型
-	-

### Example

```
+ (NSData *)createData:(StarIoExtEmulation)emulation {
    ISCBBuilder *builder = [StarIoExt createCommandBuilder:emulation];

    [builder beginDocument];

    [builder appendSound:SCBSoundChannelNo1];
    [builder appendSound:SCBSoundChannelNo2];
    [builder appendSound:SCBSoundChannelNo1 repeat:3];
    [builder appendSound:SCBSoundChannelNo2 repeat:3];
    [builder appendSound:SCBSoundChannelNo1 repeat:1 driveTime:1000 delayTime:1000];
    [builder appendSound:SCBSoundChannelNo2 repeat:1 driveTime:1000 delayTime:1000];

    [builder endDocument];

    return [builder.commands copy];
}
```

ApiFunctions.m /各コマンド仕様書を参照ください。

## 8.26 appendBarcodeData メソッド

バーコード印刷コマンドを生成し、commands プロパティに追加します。

### 宣言

```
(void)appendBarcodeData:(NSData *)otherData symbology:(SCBBarcodeSymbology)symbology
width:(SCBBarcodeWidth)width height:(NSInteger)height hri:(BOOL)hri;
```

```
(void)appendBarcodeBytes:(const void *)bytes length:(NSUInteger)length
symbology:(SCBBarcodeSymbology)symbology width:(SCBBarcodeWidth)width height:(NSInteger)height
hri:(BOOL)hri;
```

### 引数

名称	説明	型
otherData	バーコードデータ	NSData
bytes		const void *
length		NSUInteger
symbology	バーコードシンボル <ul style="list-style-type: none"> <li>• SCBBarcodeSymbologyUPCE ... UPC-E.</li> <li>• SCBBarcodeSymbologyUPCA ... UPC-A.</li> <li>• SCBBarcodeSymbologyJAN8 ... JAN/EAN8.</li> <li>• SCBBarcodeSymbologyJAN13 ... JAN/EAN13.</li> <li>• SCBBarcodeSymbologyCode39 ... Code39.</li> <li>• SCBBarcodeSymbologyITF ... ITF.</li> <li>• SCBBarcodeSymbologyCode128 ... Code128.</li> <li>• SCBBarcodeSymbologyCode93 ... Code93.</li> <li>• SCBBarcodeSymbologyCodabar ... Codabar.</li> <li>• SCBBarcodeSymbologyNW7 ... NW7.</li> </ul>	SCBBarcodeSymbology

名称	説明	型
width	バーコード幅 <ul style="list-style-type: none"> <li>SCBBarcodeWidthMode1 ... モード 1</li> <li>SCBBarcodeWidthMode2 ... モード 2</li> <li>SCBBarcodeWidthMode3 ... モード 3</li> <li>SCBBarcodeWidthMode4 ... モード 4</li> <li>SCBBarcodeWidthMode5 ... モード 5</li> <li>SCBBarcodeWidthMode6 ... モード 6</li> <li>SCBBarcodeWidthMode7 ... モード 7</li> <li>SCBBarcodeWidthMode8 ... モード 8</li> <li>SCBBarcodeWidthMode9 ... モード 9</li> <li>SCBBarcodeWidthExtMode1 ... Ext モード 1</li> <li>SCBBarcodeWidthExtMode2 ... Ext モード 2</li> <li>SCBBarcodeWidthExtMode3 ... Ext モード 3</li> <li>SCBBarcodeWidthExtMode4 ... Ext モード 4</li> <li>SCBBarcodeWidthExtMode5 ... Ext モード 5</li> <li>SCBBarcodeWidthExtMode6 ... Ext モード 6</li> <li>SCBBarcodeWidthExtMode7 ... Ext モード 7</li> <li>SCBBarcodeWidthExtMode8 ... Ext モード 8</li> <li>SCBBarcodeWidthExtMode9 ... Ext モード 9</li> </ul>	SCBBarcodeWidth
height	バーコードの高さ（ドット数単位）	NSInteger
hri	バー下文字列 <ul style="list-style-type: none"> <li>YES ... 印刷あり</li> <li>NO ... 印刷なし</li> </ul>	BOOL

## 戻り値

説明	型
-	-

## Example

```

+ (NSData *)createData:(StarIoExtEmulation)emulation {
    NSData *otherDataCode128 = [@"{B0123456789" dataUsingEncoding:NSUTF8StringEncoding];

    ISCBBuilder *builder = [StarIoExt createCommandBuilder:emulation];

    [builder beginDocument];

    [builder appendBarcodeData:otherDataCode128 symbology:SCBBarcodeSymbologyCode128
    width:SCBBarcodeWidthMode1 height:40 hri:YES];
    [builder appendUnitFeed:32];

    [builder appendCutPaper:SCBCutPaperActionPartialCutWithFeed];

    [builder endDocument];

    return [builder.commands copy];
}

```

ApiFunctions.m /各コマンド仕様書を参照ください。

## 8.27 appendBarcodeDataWithAbsolutePosition 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、commands プロパティに追加します。

### 宣言

```
- (void)appendBarcodeDataWithAbsolutePosition:(NSData *)otherData
symbology:(SCBBarcodeSymbology)symbology width:(SCBBarcodeWidth)width height:(NSInteger)height
hri:(BOOL)hri position:(NSInteger)position;
```

```
- (void)appendBarcodeBytesWithAbsolutePosition:(const void *)bytes length:(NSUInteger)length
symbology:(SCBBarcodeSymbology)symbology width:(SCBBarcodeWidth)width height:(NSInteger)height
hri:(BOOL)hri position:(NSInteger)position;
```

### 引数

名称	説明	型
position	絶対位置（ドット数単位）	NSInteger
otherData	バーコードデータ	NSData
bytes		const void *
length		NSUInteger
symbology	バーコードシンボル <ul style="list-style-type: none"> <li>• SCBBarcodeSymbologyUPCE ... UPC-E.</li> <li>• SCBBarcodeSymbologyUPCA ... UPC-A.</li> <li>• SCBBarcodeSymbologyJAN8 ... JAN/EAN8.</li> <li>• SCBBarcodeSymbologyJAN13 ... JAN/EAN13.</li> <li>• SCBBarcodeSymbologyCode39 ... Code39.</li> <li>• SCBBarcodeSymbologyITF ... ITF.</li> <li>• SCBBarcodeSymbologyCode128 ... Code128.</li> <li>• SCBBarcodeSymbologyCode93 ... Code93.</li> <li>• <del>SCBBarcodeSymbologyCodabar ... Codabar.</del></li> <li>• SCBBarcodeSymbologyNW7 ... NW7.</li> </ul>	SCBBarcodeSymbology

名称	説明	型
width	バーコード幅 <ul style="list-style-type: none"> <li>SCBBarcodeWidthMode1 ... モード 1</li> <li>SCBBarcodeWidthMode2 ... モード 2</li> <li>SCBBarcodeWidthMode3 ... モード 3</li> <li>SCBBarcodeWidthMode4 ... モード 4</li> <li>SCBBarcodeWidthMode5 ... モード 5</li> <li>SCBBarcodeWidthMode6 ... モード 6</li> <li>SCBBarcodeWidthMode7 ... モード 7</li> <li>SCBBarcodeWidthMode8 ... モード 8</li> <li>SCBBarcodeWidthMode9 ... モード 9</li> <li>SCBBarcodeWidthExtMode1 ... Ext モード 1</li> <li>SCBBarcodeWidthExtMode2 ... Ext モード 2</li> <li>SCBBarcodeWidthExtMode3 ... Ext モード 3</li> <li>SCBBarcodeWidthExtMode4 ... Ext モード 4</li> <li>SCBBarcodeWidthExtMode5 ... Ext モード 5</li> <li>SCBBarcodeWidthExtMode6 ... Ext モード 6</li> <li>SCBBarcodeWidthExtMode7 ... Ext モード 7</li> <li>SCBBarcodeWidthExtMode8 ... Ext モード 8</li> <li>SCBBarcodeWidthExtMode9 ... Ext モード 9</li> </ul>	SCBBarcodeWidth
height	バーコードの高さ (ドット数単位)	NSInteger
hri	バー下文字列 <ul style="list-style-type: none"> <li>YES ... 印刷あり</li> <li>NO ... 印刷なし</li> </ul>	BOOL

## 戻り値

説明	型
-	-

## Example

```

+ (NSData *)createData:(StarIoExtEmulation)emulation {
    NSData *otherDataCode128 = [@"{B0123456789" dataUsingEncoding:NSUTF8StringEncoding];

    ISCBBuilder *builder = [StarIoExt createCommandBuilder:emulation];

    [builder beginDocument];

    [builder appendBarcodeDataWithAbsolutePosition:otherDataCode128
     symbology:SCBBarcodeSymbologyCode128 width:SCBBarcodeWidthMode1 height:40 hri:YES position:40];
    [builder appendUnitFeed:32];

    [builder appendCutPaper:SCBCutPaperActionPartialCutWithFeed];

    [builder endDocument];

    return [builder.commands copy];
}

```

ApiFunctions.m /各コマンド仕様書を参照ください。

## 8.28 appendBarcodeDataWithAlignment メソッド

位置揃えバーコード印刷コマンドを生成し、commands プロパティに追加します。

### 宣言

```
- (void)appendBarcodeDataWithAlignment:(NSData *)otherData
symbology:(SCBBarcodeSymbology)symbology width:(SCBBarcodeWidth)width height:(NSInteger)height
hri:(BOOL)hri position:(SCBAlignmentPosition)position;
```

```
- (void)appendBarcodeBytesWithAlignment:(const void *)bytes length:(NSUInteger)length
symbology:(SCBBarcodeSymbology)symbology width:(SCBBarcodeWidth)width height:(NSInteger)height
hri:(BOOL)hri position:(SCBAlignmentPosition)position;
```

### 引数

名称	説明	型
position	位置揃え指定 <ul style="list-style-type: none"> <li>SCBAlignmentPositionLeft ... 左揃え</li> <li>SCBAlignmentPositionCenter ... 中心揃え</li> <li>SCBAlignmentPositionRight ... 右揃え</li> </ul>	SCBAlignmentPosition
otherData		NSData
bytes	バーコードデータ	const void *
length		NSUInteger
symbology	バーコードシンボル <ul style="list-style-type: none"> <li>SCBBarcodeSymbologyUPCE ... UPC-E.</li> <li>SCBBarcodeSymbologyUPCA ... UPC-A.</li> <li>SCBBarcodeSymbologyJAN8 ... JAN/EAN8.</li> <li>SCBBarcodeSymbologyJAN13 ... JAN/EAN13.</li> <li>SCBBarcodeSymbologyCode39 ... Code39.</li> <li>SCBBarcodeSymbologyITF ... ITF.</li> <li>SCBBarcodeSymbologyCode128 ... Code128.</li> <li>SCBBarcodeSymbologyCode93 ... Code93.</li> <li><del>SCBBarcodeSymbologyCodabar ... Codabar.</del></li> <li>SCBBarcodeSymbologyNW7 ... NW7.</li> </ul>	SCBBarcodeSymbology



名称	説明	型
width	バーコード幅 <ul style="list-style-type: none"> <li>• SCBBarcodeWidthMode1 ... モード 1</li> <li>• SCBBarcodeWidthMode2 ... モード 2</li> <li>• SCBBarcodeWidthMode3 ... モード 3</li> <li>• SCBBarcodeWidthMode4 ... モード 4</li> <li>• SCBBarcodeWidthMode5 ... モード 5</li> <li>• SCBBarcodeWidthMode6 ... モード 6</li> <li>• SCBBarcodeWidthMode7 ... モード 7</li> <li>• SCBBarcodeWidthMode8 ... モード 8</li> <li>• SCBBarcodeWidthMode9 ... モード 9</li> <li>• SCBBarcodeWidthExtMode1 ... Ext モード 1</li> <li>• SCBBarcodeWidthExtMode2 ... Ext モード 2</li> <li>• SCBBarcodeWidthExtMode3 ... Ext モード 3</li> <li>• SCBBarcodeWidthExtMode4 ... Ext モード 4</li> <li>• SCBBarcodeWidthExtMode5 ... Ext モード 5</li> <li>• SCBBarcodeWidthExtMode6 ... Ext モード 6</li> <li>• SCBBarcodeWidthExtMode7 ... Ext モード 7</li> <li>• SCBBarcodeWidthExtMode8 ... Ext モード 8</li> <li>• SCBBarcodeWidthExtMode9 ... Ext モード 9</li> </ul>	SCBBarcodeWidth
height	バーコードの高さ（ドット数単位）	NSInteger
hri	バー下文字列 <ul style="list-style-type: none"> <li>• YES ... 印刷あり</li> <li>• NO ... 印刷なし</li> </ul>	BOOL

## 戻り値

説明	型
-	-

### Example

```
+ (NSData *)createData:(StarIoExtEmulation)emulation {
    NSData *otherDataCode128 = [@"{B0123456789" dataUsingEncoding:NSUTF8StringEncoding];

    ISCBBuilder *builder = [StarIoExt createCommandBuilder:emulation];

    [builder beginDocument];

    [builder appendBarcodeDataWithAlignment:otherDataCode128 symbology:SCBBarcodeSymbologyCode128
    width:SCBBarcodeWidthMode1 height:40 hri:YES position:SCBAlignmentPositionCenter];
    [builder appendUnitFeed:32];

    [builder appendBarcodeDataWithAlignment:otherDataCode128 symbology:SCBBarcodeSymbologyCode128
    width:SCBBarcodeWidthMode1 height:40 hri:YES position:SCBAlignmentPositionRight];
    [builder appendUnitFeed:32];

    [builder appendCutPaper:SCBCutPaperActionPartialCutWithFeed];

    [builder endDocument];

    return [builder.commands copy];
}
```

ApiFunctions.m /各コマンド仕様書を参照ください。

## 8.29 appendPdf417Data メソッド

PDF417 印刷コマンドを生成し、commands プロパティに追加します。

### 宣言

```
- (void)appendPdf417Data:(NSData *)otherData line:(NSInteger)line column:(NSInteger)column
level:(SCBPdf417Level)level module:(NSInteger)module aspect:(NSInteger)aspect;
```

```
- (void)appendPdf417Bytes:(const void *)bytes length:(NSUInteger)length line:(NSInteger)line
column:(NSInteger)column level:(SCBPdf417Level)level module:(NSInteger)module
aspect:(NSInteger)aspect;
```

### 引数

名称	説明	型
otherData	PDF417 データ	NSData
bytes		const void *
length		NSUInteger
line	ライン数	NSInteger
column	カラム数	NSInteger
level	PDF417 ECC (セキュリティレベル) <ul style="list-style-type: none"> <li>SCBPdf417LevelECC0 ... セキュリティレベル 0</li> <li>SCBPdf417LevelECC1 ... セキュリティレベル 1</li> <li>SCBPdf417LevelECC2 ... セキュリティレベル 2</li> <li>SCBPdf417LevelECC3 ... セキュリティレベル 3</li> <li>SCBPdf417LevelECC4 ... セキュリティレベル 4</li> <li>SCBPdf417LevelECC5 ... セキュリティレベル 5</li> <li>SCBPdf417LevelECC6 ... セキュリティレベル 6</li> <li>SCBPdf417LevelECC7 ... セキュリティレベル 7</li> <li>SCBPdf417LevelECC8 ... セキュリティレベル 8</li> </ul>	SCBPdf417Level
module	モジュール X 方向サイズ	NSInteger
aspect	モジュールアスペクト比	NSInteger

### 戻り値

説明	型
-	-

### Example

```
+ (NSData *)createData:(StarloExtEmulation)emulation {
    NSData *otherData = [@"Hello World." dataUsingEncoding:NSUTF8StringEncoding];

    ISCBBuilder *builder = [StarloExt createCommandBuilder:emulation];

    [builder beginDocument];

    [builder appendPdf417Data:otherData line:0 column:1 level:SCBPdf417LevelECC0 module:2 aspect:2];
    [builder appendUnitFeed:32];

    [builder appendCutPaper:SCBCutPaperActionPartialCutWithFeed];

    [builder endDocument];

    return [builder.commands copy];
}
```

*ApiFunctions.m* /各コマンド仕様書を参照ください。

## 8.30 appendPdf417DataWithAbsolutePosition メソッド

絶対位置指定 PDF417 印刷コマンドを生成し、commands プロパティに追加します。

### 宣言

```
- (void)appendPdf417DataWithAbsolutePosition:(NSData *)otherData line:(NSInteger)line
column:(NSInteger)column level:(SCBPdf417Level)level module:(NSInteger)module aspect:(NSInteger)aspect
position:(NSInteger)position;
```

```
- (void)appendPdf417BytesWithAbsolutePosition:(const void *)bytes length:(NSUInteger)length
line:(NSInteger)line column:(NSInteger)column level:(SCBPdf417Level)level module:(NSInteger)module
aspect:(NSInteger)aspect position:(NSInteger)position;
```

### 引数

名称	説明	型
position	絶対位置（ドット数単位）	NSInteger
otherData	PDF417 データ	NSData
bytes		const void *
length		NSUInteger
line	ライン数	NSInteger
column	カラム数	NSInteger
level	PDF417 ECC（セキュリティレベル） <ul style="list-style-type: none"> <li>SCBPdf417LevelECC0 ... セキュリティレベル 0</li> <li>SCBPdf417LevelECC1 ... セキュリティレベル 1</li> <li>SCBPdf417LevelECC2 ... セキュリティレベル 2</li> <li>SCBPdf417LevelECC3 ... セキュリティレベル 3</li> <li>SCBPdf417LevelECC4 ... セキュリティレベル 4</li> <li>SCBPdf417LevelECC5 ... セキュリティレベル 5</li> <li>SCBPdf417LevelECC6 ... セキュリティレベル 6</li> <li>SCBPdf417LevelECC7 ... セキュリティレベル 7</li> <li>SCBPdf417LevelECC8 ... セキュリティレベル 8</li> </ul>	SCBPdf417Level
module	モジュール X 方向サイズ	NSInteger
aspect	モジュールアスペクト比	NSInteger

### 戻り値

説明	型
-	-

### Example

```
+ (NSData *)createData:(StarIoExtEmulation)emulation {
    NSData *otherData = [@"Hello World." dataUsingEncoding:NSUTF8StringEncoding];

    ISCBBuilder *builder = [StarIoExt createCommandBuilder:emulation];

    [builder beginDocument];

    [builder appendPdf417DataWithAbsolutePosition:otherData line:0 column:1 level:SCBPdf417LevelECC0
     module:2 aspect:2 position:40];
    [builder appendUnitFeed:32];

    [builder appendCutPaper:SCBCutPaperActionPartialCutWithFeed];

    [builder endDocument];

    return [builder.commands copy];
}
```

ApiFunctions.m /各コマンド仕様書を参照ください。

## 8.31 appendPdf417DataWithAlignment メソッド

位置揃え PDF417 印刷コマンドを生成し、commands プロパティに追加します。

### 宣言

```
- (void)appendPdf417DataWithAlignment:(NSData *)otherData line:(NSInteger)line column:(NSInteger)column
level:(SCBPdf417Level)level module:(NSInteger)module aspect:(NSInteger)aspect
position:(SCBAlignmentPosition)position;
```

```
- (void)appendPdf417BytesWithAlignment:(const void *)bytes length:(NSUInteger)length line:(NSInteger)line
column:(NSInteger)column level:(SCBPdf417Level)level module:(NSInteger)module aspect:(NSInteger)aspect
position:(SCBAlignmentPosition)position;
```

### 引数

名称	説明	型
position	位置揃え指定 <ul style="list-style-type: none"> <li>SCBAlignmentPositionLeft ... 左揃え</li> <li>SCBAlignmentPositionCenter ... 中心揃え</li> <li>SCBAlignmentPositionRight ... 右揃え</li> </ul>	SCBAlignmentPosition
otherData	PDF417 データ	NSData
bytes		const void *
length		NSUInteger
line	ライン数	NSInteger
column	カラム数	NSInteger
level	PDF417 ECC (セキュリティレベル) <ul style="list-style-type: none"> <li>SCBPdf417LevelECC0 ... セキュリティレベル 0</li> <li>SCBPdf417LevelECC1 ... セキュリティレベル 1</li> <li>SCBPdf417LevelECC2 ... セキュリティレベル 2</li> <li>SCBPdf417LevelECC3 ... セキュリティレベル 3</li> <li>SCBPdf417LevelECC4 ... セキュリティレベル 4</li> <li>SCBPdf417LevelECC5 ... セキュリティレベル 5</li> <li>SCBPdf417LevelECC6 ... セキュリティレベル 6</li> <li>SCBPdf417LevelECC7 ... セキュリティレベル 7</li> <li>SCBPdf417LevelECC8 ... セキュリティレベル 8</li> </ul>	SCBPdf417Level
module	モジュール X 方向サイズ	NSInteger
aspect	モジュールアスペクト比	NSInteger

### 戻り値

説明	型
-	-

### Example

```
+ (NSData *)createData:(StarIoExtEmulation)emulation {
    NSData *otherData = [@"Hello World." dataUsingEncoding:NSUTF8StringEncoding];

    ISCBBuilder *builder = [StarIoExt createCommandBuilder:emulation];

    [builder beginDocument];

    [builder appendPdf417DataWithAlignment:otherData line:0 column:1 level:SCBPdf417LevelECC0 module:2
    aspect:2 position:SCBAlignmentPositionCenter];
    [builder appendUnitFeed:32];

    [builder appendPdf417DataWithAlignment:otherData line:0 column:1 level:SCBPdf417LevelECC0 module:2
    aspect:2 position:SCBAlignmentPositionRight];
    [builder appendUnitFeed:32];

    [builder appendCutPaper:SCBCutPaperActionPartialCutWithFeed];

    [builder endDocument];

    return [builder.commands copy];
}
```

ApiFunctions.m /各コマンド仕様書を参照ください。



## 8.32 appendQrCodeData メソッド

QR コード印刷コマンドを生成し、commands プロパティに追加します。

### 宣言

```
- (void)appendQrCodeData:(NSData *)otherData model:(SCBQrCodeModel)model
level:(SCBQrCodeLevel)level cell:(NSInteger)cell;
```

```
- (void)appendQrCodeBytes:(const void *)bytes length:(NSUInteger)length model:(SCBQrCodeModel)model
level:(SCBQrCodeLevel)level cell:(NSInteger)cell;
```

### 引数

名称	説明	型
otherData	QR コードデータ	NSData
bytes		const void *
length		NSUInteger
model	QR コードモデル <ul style="list-style-type: none"> <li>SCBQrCodeModelNo1 ... モデル 1</li> <li>SCBQrCodeModelNo2 ... モデル 2</li> </ul>	SCBQrCodeModel
level	QR コード誤り訂正レベル <ul style="list-style-type: none"> <li>SCBQrCodeLevelL ... 誤り訂正レベル L</li> <li>SCBQrCodeLevelM ... 誤り訂正レベル M</li> <li>SCBQrCodeLevelQ ... 誤り訂正レベル Q</li> <li>SCBQrCodeLevelH ... 誤り訂正レベル H</li> </ul>	SCBQrCodeLevel
cell	QR コードセルサイズ	NSInteger

### 戻り値

説明	型
-	-

### Example

```
+ (NSData *)createData:(StarloExtEmulation)emulation {
    NSData *otherData = [@"Hello World." dataUsingEncoding:NSUTF8StringEncoding];

    ISCBBuilder *builder = [StarloExt createCommandBuilder:emulation];

    [builder beginDocument];

    [builder appendQrCodeData:otherData model:SCBQrCodeModelNo2 level:SCBQrCodeLevelL cell:4];
    [builder appendUnitFeed:32];

    [builder appendCutPaper:SCBCutPaperActionPartialCutWithFeed];

    [builder endDocument];

    return [builder.commands copy];
}
```

ApiFunctions.m /各コマンド仕様書を参照ください。

## 8.33 appendQrCodeDataWithAbsolutePosition メソッド

絶対位置指定 QR コード印刷コマンドを生成し、commands プロパティに追加します。

### 宣言

```
- (void)appendQrCodeDataWithAbsolutePosition:(NSData *)otherData model:(SCBQrCodeModel)model
level:(SCBQrCodeLevel)level cell:(NSInteger)cell position:(NSInteger)position;
```

```
- (void)appendQrCodeBytesWithAbsolutePosition:(const void *)bytes length:(NSUInteger)length
model:(SCBQrCodeModel)model level:(SCBQrCodeLevel)level cell:(NSInteger)cell
position:(NSInteger)position;
```

### 引数

名称	説明	型
position	絶対位置（ドット数単位）	NSInteger
otherData	QR コードデータ	NSData
bytes		const void *
length		NSUInteger
model	QR コードモデル <ul style="list-style-type: none"> <li>SCBQrCodeModelNo1 ... モデル 1</li> <li>SCBQrCodeModelNo2 ... モデル 2</li> </ul>	SCBQrCodeModel
level	QR コード誤り訂正レベル <ul style="list-style-type: none"> <li>SCBQrCodeLevelL ... 誤り訂正レベル L</li> <li>SCBQrCodeLevelM ... 誤り訂正レベル M</li> <li>SCBQrCodeLevelQ ... 誤り訂正レベル Q</li> <li>SCBQrCodeLevelH ... 誤り訂正レベル H</li> </ul>	SCBQrCodeLevel
cell	QR コードセルサイズ	NSInteger

### 戻り値

説明	型
-	-

### Example

```
+ (NSData *)createData:(StarIoExtEmulation)emulation {
    NSData *otherData = [@"Hello World." dataUsingEncoding:NSUTF8StringEncoding];

    ISCBBuilder *builder = [StarIoExt createCommandBuilder:emulation];

    [builder beginDocument];

    [builder appendQrCodeDataWithAbsolutePosition:otherData model:SCBQrCodeModelNo2
level:SCBQrCodeLevelL cell:4 position:40];
    [builder appendUnitFeed:32];

    [builder appendCutPaper:SCBCutPaperActionPartialCutWithFeed];

    [builder endDocument];

    return [builder.commands copy];
}
```

ApiFunctions.m /各コマンド仕様書を参照ください。

## 8.34 appendQrCodeDataWithAlignment メソッド

位置揃え QR コード印刷コマンドを生成し、commands プロパティに追加します。

### 宣言

```
- (void)appendQrCodeDataWithAlignment:(NSData *)otherData model:(SCBQrCodeModel)model
level:(SCBQrCodeLevel)level cell:(NSInteger)cell position:(SCBAlignmentPosition)position;
```

```
- (void)appendQrCodeBytesWithAlignment:(const void *)bytes length:(NSUInteger)length
model:(SCBQrCodeModel)model level:(SCBQrCodeLevel)level cell:(NSInteger)cell
position:(SCBAlignmentPosition)position;
```

### 引数

名称	説明	型
position	位置揃え指定 <ul style="list-style-type: none"> <li>SCBAlignmentPositionLeft ... 左揃え</li> <li>SCBAlignmentPositionCenter ... 中心揃え</li> <li>SCBAlignmentPositionRight ... 右揃え</li> </ul>	SCBAlignmentPosition
otherData	QR コードデータ	NSData
bytes		const void *
length		NSUInteger
model	QR コードモデル <ul style="list-style-type: none"> <li>SCBQrCodeModelNo1 ... モデル 1</li> <li>SCBQrCodeModelNo2 ... モデル 2</li> </ul>	SCBQrCodeModel
level	QR コード誤り訂正レベル <ul style="list-style-type: none"> <li>SCBQrCodeLevelL ... 誤り訂正レベル L</li> <li>SCBQrCodeLevelM ... 誤り訂正レベル M</li> <li>SCBQrCodeLevelQ ... 誤り訂正レベル Q</li> <li>SCBQrCodeLevelH ... 誤り訂正レベル H</li> </ul>	SCBQrCodeLevel
cell	QR コードセルサイズ	NSInteger

### 戻り値

説明	型
-	-

### Example

```
+ (NSData *)createData:(StarIoExtEmulation)emulation {
    NSData *otherData = [@"Hello World." dataUsingEncoding:NSUTF8StringEncoding];

    ISCBBuilder *builder = [StarIoExt createCommandBuilder:emulation];

    [builder beginDocument];

    [builder appendQrCodeDataWithAlignment:otherData model:SCBQrCodeModelNo2 level:SCBQrCodeLevelL
    cell:4 position:SCBAlignmentPositionCenter];
    [builder appendUnitFeed:32];

    [builder appendQrCodeDataWithAlignment:otherData model:SCBQrCodeModelNo2 level:SCBQrCodeLevelL
    cell:4 position:SCBAlignmentPositionRight];
    [builder appendUnitFeed:32];

    [builder appendCutPaper:SCBCutPaperActionPartialCutWithFeed];

    [builder endDocument];

    return [builder.commands copy];
}
```

ApiFunctions.m /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、commands プロパティに追加します。

### 宣言

```
- (void)appendBitmap:(UIImage *)image diffusion:(BOOL)diffusion width:(NSInteger)width  
bothScale:(BOOL)bothScale rotation:(SCBBitmapConverterRotation)rotation;
```

```
- (void)appendBitmap:(UIImage *)image diffusion:(BOOL)diffusion  
rotation:(SCBBitmapConverterRotation)rotation;
```

```
- (void)appendBitmap:(UIImage *)image diffusion:(BOOL)diffusion width:(NSInteger)width  
bothScale:(BOOL)bothScale;
```

```
- (void)appendBitmap:(UIImage *)image diffusion:(BOOL)diffusion;
```

### 引数

名称	説明	型
image	元となるビットマップオブジェクト	UIImage
diffusion	誤差拡散 <ul style="list-style-type: none"> <li>YES ... 誤差拡散を行う</li> <li>NO ... 誤差拡散を行わない</li> </ul>	BOOL
width	変換後のビットマップ幅（ドット数単位）	NSInteger
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"> <li>YES ... 高さ変換を行う</li> <li>NO ... 高さ変換を行わない</li> </ul>	BOOL
rotation	変換後の回転 <ul style="list-style-type: none"> <li>SCBBitmapConverterRotationNormal ... 回転なし</li> <li>SCBBitmapConverterRotationRight90 ... 右 90 度回転</li> <li>SCBBitmapConverterRotationLeft90 ... 左 90 度回転</li> <li>SCBBitmapConverterRotationRotate180 ... 180 度回転</li> </ul>	SCBBitmapConverterRotation

### 戻り値

説明	型
-	-

### Example

```
+ (NSData *)createData:(StarloExtEmulation)emulation width:(NSInteger)width {
    UIImage *starLogoImage = [UIImage imageNamed:@"StarLogoImage"];

    ISCBBuilder *builder = [StarloExt createCommandBuilder:emulation];

    [builder beginDocument];

    [builder appendData:@"\nNormal\n" dataUsingEncoding:NSUTF8StringEncoding];
    [builder appendBitmap:starLogoImage diffusion:YES];

    [builder appendData:@"\nwidth:Full, bothScale:YES\n" dataUsingEncoding:NSUTF8StringEncoding];
    [builder appendBitmap:starLogoImage diffusion:YES width:width bothScale:YES];
    [builder appendData:@"\nwidth:Full, bothScale:NO\n" dataUsingEncoding:NSUTF8StringEncoding];
    [builder appendBitmap:starLogoImage diffusion:YES width:width bothScale:NO];

    [builder appendData:@"\nRotate180\n" dataUsingEncoding:NSUTF8StringEncoding];
    [builder appendBitmap:starLogoImage diffusion:YES rotation:SCBBitmapConverterRotationRotate180];

    [builder appendCutPaper:SCBCutPaperActionPartialCutWithFeed];

    [builder endDocument];

    return [builder.commands copy];
}
```

*ApiFunctions.m* 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、commands プロパティに追加します。

### 宣言

```
- (void)appendBitmapWithAbsolutePosition:(UIImage *)image diffusion:(BOOL)diffusion
width:(NSInteger)width bothScale:(BOOL)bothScale rotation:(SCBBitmapConverterRotation)rotation
position:(NSInteger)position;
```

```
- (void)appendBitmapWithAbsolutePosition:(UIImage *)image diffusion:(BOOL)diffusion
rotation:(SCBBitmapConverterRotation)rotation position:(NSInteger)position;
```

```
- (void)appendBitmapWithAbsolutePosition:(UIImage *)image diffusion:(BOOL)diffusion
width:(NSInteger)width bothScale:(BOOL)bothScale position:(NSInteger)position;
```

```
- (void)appendBitmapWithAbsolutePosition:(UIImage *)image diffusion:(BOOL)diffusion
position:(NSInteger)position;
```

### 引数

名称	説明	型
position	絶対位置（ドット数単位）	NSInteger
image	元となるビットマップオブジェクト	UIImage
diffusion	誤差拡散 <ul style="list-style-type: none"> <li>YES ... 誤差拡散を行う</li> <li>NO ... 誤差拡散を行わない</li> </ul>	BOOL
width	変換後のビットマップ幅（ドット数単位）	NSInteger
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"> <li>YES ... 高さ変換を行う</li> <li>NO ... 高さ変換を行わない</li> </ul>	BOOL
rotation	変換後の回転 <ul style="list-style-type: none"> <li>SCBBitmapConverterRotationNormal ... 回転なし</li> <li>SCBBitmapConverterRotationRight90 ... 右 90 度回転</li> <li>SCBBitmapConverterRotationLeft90 ... 左 90 度回転</li> <li>SCBBitmapConverterRotationRotate180 ... 180 度回転</li> </ul>	SCBBitmapConverterRotation

### 戻り値

説明	型
-	-

## Example

```
+ (NSData *)createData:(StarloExtEmulation)emulation width:(NSInteger)width {
    UIImage *starLogolmage = [UIImage imageNamed:@"StarLogolmage"];

    ISCBBuilder *builder = [StarloExt createCommandBuilder:emulation];

    [builder beginDocument];

    [builder appendData:@"\nNormal, AbsolutePosition:40*\n" dataUsingEncoding:NSUTF8StringEncoding];
    [builder appendBitmapWithAbsolutePosition:starLogolmage diffusion:YES position:40];

    [builder appendData:@"\nwidth:Full, bothScale:YES, AbsolutePosition:40*\n"
dataUsingEncoding:NSUTF8StringEncoding];
    [builder appendBitmap:starLogolmage diffusion:YES width:width bothScale:YES position:40];
    [builder appendData:@"\nwidth:Full, bothScale:NO, AbsolutePosition:40*\n"
dataUsingEncoding:NSUTF8StringEncoding];
    [builder appendBitmap:starLogolmage diffusion:YES width:width bothScale:NO position:40];

    [builder appendData:@"\nRotate180, AbsolutePosition:40*\n" dataUsingEncoding:NSUTF8StringEncoding];
    [builder appendBitmapWithAbsolutePosition:starLogolmage diffusion:YES
rotation:SCBBitmapConverterRotationRotate180 position:40];

    [builder appendCutPaper:SCBCutPaperActionPartialCutWithFeed];

    [builder endDocument];

    return [builder.commands copy];
}
```

*ApiFunctions.m* 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、commands プロパティに追加します。

### 宣言

```
- (void)appendBitmapWithAlignment:(UIImage *)image diffusion:(BOOL)diffusion width:(NSInteger)width  
bothScale:(BOOL)bothScale rotation:(SCBBitmapConverterRotation)rotation  
position:(SCBAlignmentPosition)position;
```

```
- (void)appendBitmapWithAlignment:(UIImage *)image diffusion:(BOOL)diffusion  
rotation:(SCBBitmapConverterRotation)rotation position:(SCBAlignmentPosition)position;
```

```
- (void)appendBitmapWithAlignment:(UIImage *)image diffusion:(BOOL)diffusion width:(NSInteger)width  
bothScale:(BOOL)bothScale position:(SCBAlignmentPosition)position;
```

```
- (void)appendBitmapWithAlignment:(UIImage *)image diffusion:(BOOL)diffusion  
position:(SCBAlignmentPosition)position;
```

### 引数

名称	説明	型
position	位置揃え指定 <ul style="list-style-type: none"> <li>SCBAlignmentPositionLeft ... 左揃え</li> <li>SCBAlignmentPositionCenter ... 中心揃え</li> <li>SCBAlignmentPositionRight ... 右揃え</li> </ul>	SCBAlignmentPosition
image	元となるビットマップオブジェクト	UIImage
diffusion	誤差拡散 <ul style="list-style-type: none"> <li>YES ... 誤差拡散を行う</li> <li>NO ... 誤差拡散を行わない</li> </ul>	BOOL
width	変換後のビットマップ幅（ドット数単位）	NSInteger
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"> <li>YES ... 高さ変換を行う</li> <li>NO ... 高さ変換を行わない</li> </ul>	BOOL
rotation	変換後の回転 <ul style="list-style-type: none"> <li>SCBBitmapConverterRotationNormal ... 回転なし</li> <li>SCBBitmapConverterRotationRight90 ... 右 90 度回転</li> <li>SCBBitmapConverterRotationLeft90 ... 左 90 度回転</li> <li>SCBBitmapConverterRotationRotate180 ... 180 度回転</li> </ul>	SCBBitmapConverterRotation

### 戻り値

説明	型
-	-

### Example

```
+ (NSData *)createData:(StarloExtEmulation)emulation width:(NSInteger)width {
    UIImage *starLogoImage = [UIImage imageNamed:@"StarLogoImage"];

    ISCBBuilder *builder = [StarloExt createCommandBuilder:emulation];

    [builder beginDocument];

    [builder appendData:@"\nNormal, Alignment:Center*\n" dataUsingEncoding:NSUTF8StringEncoding];
    [builder appendBitmapWithAlignment:starLogoImage diffusion:YES position:SCBAlignmentPositionCenter];

    [builder appendData:@"\nwidth:Full, bothScale:YES, Alignment:Center*\n"
    dataUsingEncoding:NSUTF8StringEncoding];
    [builder appendBitmapWithAlignment:starLogoImage diffusion:YES width:width bothScale:YES
    position:SCBAlignmentPositionCenter];

    [builder appendData:@"\nwidth:Full, bothScale:NO, Alignment:Center*\n"
    dataUsingEncoding:NSUTF8StringEncoding];
    [builder appendBitmapWithAlignment:starLogoImage diffusion:YES width:width bothScale:NO
    position:SCBAlignmentPositionCenter];

    [builder appendData:@"\nRotate180, Alignment:Center*\n" dataUsingEncoding:NSUTF8StringEncoding];
    [builder appendBitmapWithAlignment:starLogoImage diffusion:YES
    rotation:SCBBitmapConverterRotationRotate180 position:SCBAlignmentPositionCenter];

    [builder appendCutPaper:SCBCutPaperActionPartialCutWithFeed];

    [builder endDocument];

    return [builder.commands copy];
}
```

*ApiFunctions.m* 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、commands プロパティに追加します。

### 宣言

- (void)appendBlackMark:(SCBBlackMarkType)type;

### 引数

名称	説明	型
Type	ブラックマーク <ul style="list-style-type: none"> <li>SCBBlackMarkTypeInvalid ... ブラックマーク無効</li> <li>SCBBlackMarkTypeValid ... ブラックマーク有効</li> <li>SCBBlackMarkTypeValidWithDetection ... ブラックマーク有効 + ブラックマーク検出動作</li> </ul>	SCBBlackMarkType

### 戻り値

説明	型
-	-

### Example

```
+ (NSData *)createData:(StarloExtEmulation)emulation {
    NSData *otherData = [@"Hello World.\n" dataUsingEncoding:NSUTF8StringEncoding];

    ISCBBuilder *builder = [StarloExt createCommandBuilder:emulation];

    [builder beginDocument];

    [builder appendBlackMark:SCBBlackMarkTypeValid];

    [builder appendData:otherData];

    [builder appendCutPaper:SCBCutPaperActionPartialCutWithFeed];

    // [builder appendBlackMark:SCBBlackMarkTypeInvalid];

    [builder endDocument];

    return [builder.commands copy];
}
```

ApiFunctions.m 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、commands プロパティに追加します。

### 宣言

- (void)beginPageMode:(CGRect)rect rotation:(SCBBitmapConverterRotation)rotation;

### 引数

名称	説明	型
rect	位置とサイズ（ドット数単位）	CGRect
rotation	印刷方向 <ul style="list-style-type: none"> <li>SCBBitmapConverterRotationNormal ... 回転なし</li> <li>SCBBitmapConverterRotationRight90 ... 右 90 度回転</li> <li>SCBBitmapConverterRotationLeft90 ... 左 90 度回転</li> <li>SCBBitmapConverterRotationRotate180 ... 180 度回転</li> </ul>	SCBBitmapConverterRotation

### 戻り値

説明	型
-	-

### Example

```
+ (NSData *)createData:(StarloExtEmulation)emulation width:(NSInteger)width {
    NSData *otherData = [@"Hello World.\n" dataUsingEncoding:NSUTF8StringEncoding];

    int height = 30 * 8;    // 30mm!!!

    CGRect rect;

    ISCBBuilder *builder = [StarloExt createCommandBuilder:emulation];

    [builder beginDocument];

    [builder appendData:[@"\n*Rotate180 Text*\n" dataUsingEncoding:NSUTF8StringEncoding]];

    rect = CGRectMake(0, 0, width, height);

    [builder beginPageMode:rect rotation:SCBBitmapConverterRotationRotate180];

    [builder appendPageModeVerticalAbsolutePosition:height / 2];

    [builder appendDataWithAbsolutePosition:otherData position:width / 2];

    [builder endPageMode];

    [builder appendCutPaper:SCBCutPaperActionPartialCutWithFeed];

    [builder endDocument];

    return [builder.commands copy];
}
```

ApiFunctions.m 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、commands プロパティに追加します。

### 宣言

- (void)endPageMode;

### 引数

名称	説明	型
-	-	-

### 戻り値

説明	型
-	-

### Example

```
+ (NSData *)createData:(StarIoExtEmulation)emulation width:(NSInteger)width {
    NSData *otherData = [@"Hello World.\n" dataUsingEncoding:NSUTF8StringEncoding];

    int height = 30 * 8;    // 30mm!!!

    CGRect rect;

    ISCBBuilder *builder = [StarIoExt createCommandBuilder:emulation];

    [builder beginDocument];

    [builder appendData:[@"\n*Rotate180 Text*\n" dataUsingEncoding:NSUTF8StringEncoding]];

    rect = CGRectMake(0, 0, width, height);

    [builder beginPageMode:rect rotation:SCBBitmapConverterRotationRotate180];

    [builder appendPageModeVerticalAbsolutePosition:height / 2];

    [builder appendDataWithAbsolutePosition:otherData position:width / 2];

    [builder endPageMode];

    [builder appendCutPaper:SCBCutPaperActionPartialCutWithFeed];

    [builder endDocument];

    return [builder.commands copy];
}
```

ApiFunctions.m 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、commands プロパティに追加します。

### 宣言

- (void)appendPageModeVerticalAbsolutePosition:(NSInteger)position;

### 引数

名称	説明	型
position	縦方向の絶対位置（ドット数単位）	NSInteger

### 戻り値

説明	型
-	-

### Example

```
+ (NSData *)createData:(StarIoExtEmulation)emulation width:(NSInteger)width {
    NSData *otherData = [@"Hello World.\n" dataUsingEncoding:NSUTF8StringEncoding];

    int height = 30 * 8;    // 30mm!!!

    CGRect rect;

    ISCBBuilder *builder = [StarIoExt createCommandBuilder:emulation];

    [builder beginDocument];

    [builder appendData:[@"\n*Rotate180 Text*\n" dataUsingEncoding:NSUTF8StringEncoding]];

    rect = CGRectMake(0, 0, width, height);

    [builder beginPageMode:rect rotation:SCBBitmapConverterRotationRotate180];

    [builder appendPageModeVerticalAbsolutePosition:height / 2];

    [builder appendDataWithAbsolutePosition:otherData position:width / 2];

    [builder endPageMode];

    [builder appendCutPaper:SCBCutPaperActionPartialCutWithFeed];

    [builder endDocument];

    return [builder.commands copy];
}
```

ApiFunctions.m 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、commands プロパティに追加します。

### 宣言

```
- (void)appendPageModeRotation:(SCBBitmapConverterRotation)rotation;
```

### 引数

名称	説明	型
rotation	印刷方向 <ul style="list-style-type: none"> <li>• SCBBitmapConverterRotationNormal ... 回転なし</li> <li>• SCBBitmapConverterRotationRight90 ... 右 90 度回転</li> <li>• SCBBitmapConverterRotationLeft90 ... 左 90 度回転</li> <li>• SCBBitmapConverterRotationRotate180 ... 180 度回転</li> </ul>	SCBBitmapConverterRotation

### 戻り値

説明	型
-	-

### Example

```
+ (NSData *)createData:(StarIoExtEmulation)emulation width:(NSInteger)width {
    NSData *otherData = [@"Hello World.\n" dataUsingEncoding:NSUTF8StringEncoding];

    int height = 30 * 8;    // 30mm!!!

    CGRect rect;

    ISCBBuilder *builder = [StarIoExt createCommandBuilder:emulation];

    [builder beginDocument];

    [builder appendData:[@"\n*Mixed Text*\n" dataUsingEncoding:NSUTF8StringEncoding]];

    rect = CGRectMake(0, 0, width, height);

    [builder beginPageMode:rect rotation:SCBBitmapConverterRotationNormal];

    [builder appendPageModeVerticalAbsolutePosition:height / 2];

    [builder appendDataWithAbsolutePosition:otherData position:width / 2];

    [builder appendPageModeRotation:SCBBitmapConverterRotationRotate180];

    [builder appendPageModeVerticalAbsolutePosition:height / 2];

    [builder appendDataWithAbsolutePosition:otherData position:width / 2];

    [builder endPageMode];

    [builder appendCutPaper:SCBCutPaperActionPartialCutWithFeed];

    [builder endDocument];

    return [builder.commands copy];
}
```

*ApiFunctions.m* 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、commands プロパティに追加します。

### 宣言

```
- (void)appendPrintableArea:(SCBPrintableAreaType)type;
```

### 引数

名称	説明	型
type	印字領域タイプ	SCBPrintableAreaType
	• SCBPrintableAreaTypeStandard ... 標準タイプ	
	• SCBPrintableAreaTypeType1 ... タイプ 1	
	• SCBPrintableAreaTypeType2 ... タイプ 2	
	• SCBPrintableAreaTypeType3 ... タイプ 3	
	• SCBPrintableAreaTypeType4 ... タイプ 4	

### 戻り値

説明	型
-	-

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

SCBPrintableAreaType	mC-Print2	mC-Print3	mPOP	FVP10	TSP100	TSP650II	TSP700II	TSP800II	SM-S240i	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
SCBPrintableAreaTypeStandard	48	72 *1	48	72 *1	72 *1	72 *1	72 *1	104 *2	-	-	-	-	-	64 *1* 3	-	-	-	-	-	-	72 *1	63
SCBPrintableAreaTypeType1	54	-	54	52. 5	51	-	52. 5	-	-	-	-	-	-	-	-	-	-	-	-	-	-	48
SCBPrintableAreaTypeType2	-	48	-	-	-	-	80 *1	72	-	-	-	-	-	-	-	-	-	-	-	-	-	60
SCBPrintableAreaTypeType3	50. 8	50. 8	50.8	50. 8	-	50. 8	50. 8	-	-	-	-	-	-	45 *3	-	-	-	-	-	-	50. 8	45
SCBPrintableAreaTypeType4	-	-	-	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

```
+ (NSData *)createPrintableAreaDataWithEmulation:(StarIoExtEmulation)emulation
                                type:(SCBPrintableAreaType)type {
    ISCBBuilder *builder = [StarIoExt createCommandBuilder:emulation];

    [builder beginDocument];

    [builder appendPrintableArea:type];

    switch (type) {
        case SCBPrintableAreaTypeStandard:
            [builder appendData:@"Standard\n" dataUsingEncoding:NSUTF8StringEncoding];
            break;
        case SCBPrintableAreaTypeType1:
            [builder appendData:@"Type1\n" dataUsingEncoding:NSUTF8StringEncoding];
            break;
        case SCBPrintableAreaTypeType2:
            [builder appendData:@"Type2\n" dataUsingEncoding:NSUTF8StringEncoding];
            break;
        case SCBPrintableAreaTypeType3:
            [builder appendData:@"Type3\n" dataUsingEncoding:NSUTF8StringEncoding];
            break;
        case SCBPrintableAreaTypeType4:
            [builder appendData:@"Type4\n" dataUsingEncoding:NSUTF8StringEncoding];
            break;
    }

    UIImage *image = [UIImage imageNamed:@"PrintableAreaImage.png"];
    [builder appendBitmap:image diffused:true];

    NSData *data1 = [@"123456789" dataUsingEncoding:NSUTF8StringEncoding];
    NSData *data2 = [@"0" dataUsingEncoding:NSUTF8StringEncoding];

    for (int i = 0; i < 8; i++) {
        [builder appendData:data1];
        [builder appendDataWithInvert:data2];
    }

    [builder appendCutPaper:SCBCutPaperActionPartialCutWithFeed];

    [builder endDocument];

    return builder.commands;
}
```

*ApiFunctions.m* を参照ください。

## 8.44 appendCjkUnifiedIdeographFont メソッド

UTF-8 における CJK 統合漢字のフォント設定コマンドを生成し、commands プロパティに追加します。

### 宣言

```
(void)appendCjkUnifiedIdeographFont:(NSArray<NSNumber *> *)fonts;
```

### 引数

名称	説明	型
fonts	<p>CJK 統合漢字のフォントのリスト</p> <ul style="list-style-type: none"> <li>SCBCjkUnifiedIdeographFontJapanese ... 日本語フォント</li> <li>SCBCjkUnifiedIdeographFontSimplifiedChinese ... 簡体字中国語フォント</li> <li>SCBCjkUnifiedIdeographFontTraditionalChinese ... 繁体字中国語フォント</li> <li>SCBCjkUnifiedIdeographFontHangul ... 韓国語フォント</li> </ul> <p>2つ以上のフォントを指定した場合、先に指定したフォントから優先して印字されます。 nil もしくは空のリストを指定した場合、CJK 統合漢字のフォント設定をメモリスイッチ設定に戻します。</p>	NSArray<NSNumber *> *

### 戻り値

説明	型
-	-

## Example

```
- (void)append3inchTextReceiptData:(ISCBBuilder *)builder utf8:(BOOL)utf8 {
    NSStringEncoding encoding = NSUTF8StringEncoding;

    // This function is supported by TSP650II(JP2/TW models only) with F/W version 4.0 or later and mC-Print 2/3.
    // Switch Kanji/Hangul font by specifying the font for Unicode CJK Unified Ideographs before each word.

    [builder appendCodePage:SCBCodePageTypeUTF8];

    [builder appendCharacterSpace:0];

    [builder appendAlignment:SCBAlignmentPositionCenter];

    [builder appendData:@"2017 / 5 / 15 AM 10:00\n" dataUsingEncoding:encoding];

    [builder appendMultiple:2 height:2];

    [builder appendCjkUnifiedIdeographFont:@[@(SCBCjkUnifiedIdeographFontJapanese)]];
    [builder appendData:@"受付票 " dataUsingEncoding:encoding];

    [builder appendCjkUnifiedIdeographFont:@[@(SCBCjkUnifiedIdeographFontTraditionalChinese)]];
    [builder appendData:@"排號單\n" dataUsingEncoding:encoding];

    [builder appendCjkUnifiedIdeographFont:@[@(SCBCjkUnifiedIdeographFontSimplifiedChinese)]];
    [builder appendData:@"排号单 " dataUsingEncoding:encoding];

    [builder appendCjkUnifiedIdeographFont:@[@(SCBCjkUnifiedIdeographFontHangul)]];
    [builder appendData:@"접수표\n\n" dataUsingEncoding:encoding];

    [builder appendMultiple:1 height:1];

    [builder appendCjkUnifiedIdeographFont:@[]];
    [builder appendDataWithMultiple:@"1\n" dataUsingEncoding:encoding] width:6 height:6;
    [builder appendData:@"-----\n" dataUsingEncoding:encoding];

    [builder appendCjkUnifiedIdeographFont:@[@(SCBCjkUnifiedIdeographFontJapanese)]];
    [builder appendData:@"ご本人がお持ちください。 \n" dataUsingEncoding:encoding];
    [builder appendData:@"※紛失しないようにご注意ください。 \n" dataUsingEncoding:encoding];
}
```

*Utf8MultiLanguageReceiptsImpl.m* を参照ください。

## 8.45 commands プロパティ

生成、追加されたコマンド列です。

### 宣言

@property (nonatomic, readonly) NSMutableData \*commands;

### 値

説明	型
生成、追加されたコマンド列	NSMutableData

### Example

```
+ (NSData *)createData:(StarloExtEmulation)emulation {
    NSData *otherData = [@"Hello World.\n" dataUsingEncoding:NSUTF8StringEncoding];

    ISCBBuilder *builder = [StarloExt createCommandBuilder:emulation];

    [builder beginDocument];

    [builder appendData:otherData];

    [builder appendCutPaper:SCBCutPaperActionPartialCutWithFeed];

    [builder endDocument];

    return [builder.commands copy];
}
```

ApiFunctions.m を参照ください。

## 8.46 SCBInitializationType コンスタント

初期化指定定数。

### 宣言

```
typedef NS_ENUM(NSUInteger, SCBInitializationType) {
    SCBInitializationTypeCommand
    // SCBInitializationTypeReset,
    // SCBInitializationTypeResetWithPrint
};
```

### 定数

名称	説明
SCBInitializationTypeCommand	コマンド初期化
SCBInitializationTypeReset	プリンタリセット
SCBInitializationTypeResetWithPrint	プリンタリセット (自己印字実行)

*ApiFunctions.m* を参照ください。

## 8.47 SCBFontStyleType コンスタント

フォントスタイル指定定数。

### 宣言

```
typedef NS_ENUM(NSUInteger, SCBFontStyleType) {
    SCBFontStyleTypeA,
    SCBFontStyleTypeB
};
```

### 定数

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

*ApiFunctions.m* / 各コマンド仕様書を参照ください。

## 8.48 SCBCodePageType コンスタント

コードページ指定定数。

### 宣言

```
typedef NS_ENUM(NSUInteger, SCBCodePageType) {
    SCBCodePageTypeCP437,
    SCBCodePageTypeCP737,
    SCBCodePageTypeCP772,
    SCBCodePageTypeCP774,
    SCBCodePageTypeCP851,
    SCBCodePageTypeCP852,
    SCBCodePageTypeCP855,
    SCBCodePageTypeCP857,
    SCBCodePageTypeCP858,
    SCBCodePageTypeCP860,
    SCBCodePageTypeCP861,
    SCBCodePageTypeCP862,
    SCBCodePageTypeCP863,
    SCBCodePageTypeCP864,
    SCBCodePageTypeCP865,
    SCBCodePageTypeCP866,
    SCBCodePageTypeCP869,
    SCBCodePageTypeCP874,
    SCBCodePageTypeCP928,
    SCBCodePageTypeCP932,
    SCBCodePageTypeCP998,
    SCBCodePageTypeCP999,
    SCBCodePageTypeCP1001,
    SCBCodePageTypeCP1250,
    SCBCodePageTypeCP1251,
    SCBCodePageTypeCP1252,
    SCBCodePageTypeCP2001,
    SCBCodePageTypeCP3001,
    SCBCodePageTypeCP3002,
    SCBCodePageTypeCP3011,
    SCBCodePageTypeCP3012,
    SCBCodePageTypeCP3021,
    SCBCodePageTypeCP3041,
    SCBCodePageTypeCP3840,
    SCBCodePageTypeCP3841,
    SCBCodePageTypeCP3843,
    SCBCodePageTypeCP3844,
    SCBCodePageTypeCP3845,
    SCBCodePageTypeCP3846,
    SCBCodePageTypeCP3847,
    SCBCodePageTypeCP3848,
    SCBCodePageTypeUTF8,
    SCBCodePageTypeBlank
};
```



## 定数

名称	説明
SCBCodePageTypeCP437	CodePage437 (USA, Std. Europe).
SCBCodePageTypeCP737	Codepage 737 (Greek).
SCBCodePageTypeCP772	Codepage 772 (Lithuanian).
SCBCodePageTypeCP774	Codepage 774 (Lithuanian).
SCBCodePageTypeCP851	Codepage 851 (Greek).
SCBCodePageTypeCP852	Codepage 852 (Latin-2).
SCBCodePageTypeCP855	Codepage 855 (Cyrillic Bulgarian).
SCBCodePageTypeCP857	Codepage 857 (Turkey).
SCBCodePageTypeCP858	Codepage 858 (Multilingual).
SCBCodePageTypeCP860	Codepage 860 (Portuguese).
SCBCodePageTypeCP861	Codepage 861 (Icelandic).
SCBCodePageTypeCP862	Codepage 862 (Israel (Hebrew)).
SCBCodePageTypeCP863	Codepage 863 (Canadian French).
SCBCodePageTypeCP864	Codepage 864 (Arabic).
SCBCodePageTypeCP865	Codepage 865 (Nordic).
SCBCodePageTypeCP866	Codepage 866 (Cyrillic Russian).
SCBCodePageTypeCP869	Codepage 869 (Greek).
SCBCodePageTypeCP874	Codepage 874 (Thai).
SCBCodePageTypeCP928	Codepage 928 (Greek).
SCBCodePageTypeCP932	Katakana.
SCBCodePageTypeCP998	Normal.
SCBCodePageTypeCP999	Codepage 1252 (Windows Latin-1).
SCBCodePageTypeCP1001	Codepage 1001 (Arabic).
SCBCodePageTypeCP1250	Codepage 1250 (Windows Latin-2).
SCBCodePageTypeCP1251	Codepage 1251 (Windows Cyrillic).
SCBCodePageTypeCP1252	Codepage 1252 (Windows Latin-1).
SCBCodePageTypeCP2001	Codepage 2001 (Lithuanian-KBL).
SCBCodePageTypeCP3001	Codepage 3001 (Estonian-1).
SCBCodePageTypeCP3002	Codepage 3002 (Estonian-2).
SCBCodePageTypeCP3011	Codepage 3011 (Latvian-1).
SCBCodePageTypeCP3012	Codepage 3012 (Latvian-2).
SCBCodePageTypeCP3021	Codepage 3021 (Bulgarian).
SCBCodePageTypeCP3041	Codepage 3041 (Maltese).
SCBCodePageTypeCP3840	Codepage 3840 (IBM-Russian).
SCBCodePageTypeCP3841	Codepage 3841 (Gost).
SCBCodePageTypeCP3843	Codepage 3843 (Polish).
SCBCodePageTypeCP3844	Codepage 3844 (CS2).
SCBCodePageTypeCP3845	Codepage 3845 (Hungarian).
SCBCodePageTypeCP3846	Codepage 3846 (Turkish).
SCBCodePageTypeCP3847	Codepage 3847 (Brazil-ABNT).
SCBCodePageTypeCP3848	Codepage 3848 (Brazil-ABICOMP).
SCBCodePageTypeUTF8	UTF-8.
SCBCodePageTypeBlank	User Setting Blank Code Page.

ApiFunctions.m / 各コマンド仕様書を参照ください。

## 8.49 SCBInternationalType コンスタント

国際文字指定定数。

### 宣言

```
typedef NS_ENUM(NSUInteger, SCBInternationalType) {
    SCBInternationalTypeUSA,
    SCBInternationalTypeFrance,
    SCBInternationalTypeGermany,
    SCBInternationalTypeUK,
    SCBInternationalTypeDenmark,
    SCBInternationalTypeSweden,
    SCBInternationalTypeItaly,
    SCBInternationalTypeSpain,
    SCBInternationalTypeJapan,
    SCBInternationalTypeNorway,
    SCBInternationalTypeDenmark2,
    SCBInternationalTypeSpain2,
    SCBInternationalTypeLatinAmerica,
    SCBInternationalTypeKorea,
    SCBInternationalTypeIreland,
    SCBInternationalTypeLegal
};
```

### 定数

名称	説明
SCBInternationalTypeUSA	USA
SCBInternationalTypeFrance	フランス
SCBInternationalTypeGermany	ドイツ
SCBInternationalTypeUK	イギリス
SCBInternationalTypeDenmark	デンマーク
SCBInternationalTypeSweden	スウェーデン
SCBInternationalTypeItaly	イタリア
SCBInternationalTypeSpain	スペイン
SCBInternationalTypeJapan	日本
SCBInternationalTypeNorway	ノルウェー
SCBInternationalTypeDenmark2	デンマーク II
SCBInternationalTypeSpain2	スペイン II
SCBInternationalTypeLatinAmerica	ラテンアメリカ
SCBInternationalTypeKorea	韓国
SCBInternationalTypeIreland	アイルランド
SCBInternationalTypeLegal	Legal

*ApiFunctions.m* / 各コマンド仕様書を参照ください。

## 8.50 SCBLogoSize コンスタント

ロゴサイズ指定定数。

### 宣言

```
typedef NS_ENUM(NSUInteger, SCBLogoSize) {
    SCBLogoSizeNormal,
    SCBLogoSizeDoubleWidth,
    SCBLogoSizeDoubleHeight,
    SCBLogoSizeDoubleWidthDoubleHeight
};
```

### 定数

名称	説明
SCBLogoSizeNormal	ノーマルモード
SCBLogoSizeDoubleWidth	横 2 倍モード
SCBLogoSizeDoubleHeight	縦 2 倍モード
SCBLogoSizeDoubleWidthDoubleHeight	縦 2 倍、横 2 倍モード

*ApiFunctions.m* / 各コマンド仕様書を参照ください。

## 8.51 SCBAlignmentPosition コンスタント

位置揃え指定定数。

### 宣言

```
typedef NS_ENUM(NSUInteger, SCBAlignmentPosition) {
    SCBAlignmentPositionLeft,
    SCBAlignmentPositionCenter,
    SCBAlignmentPositionRight
};
```

### 定数

名称	説明
SCBAlignmentPositionLeft	左揃え
SCBAlignmentPositionCenter	中心揃え
SCBAlignmentPositionRight	右揃え

*ApiFunctions.m* / 各コマンド仕様書を参照ください。

## 8.52 SCBCutPaperAction コンスタント

用紙カット指定定数。

### 宣言

```
typedef NS_ENUM(NSInteger, SCBCutPaperAction) {
    SCBCutPaperActionFullCut,
    SCBCutPaperActionPartialCut,
    SCBCutPaperActionFullCutWithFeed,
    SCBCutPaperActionPartialCutWithFeed
};
```

### 定数

名称	説明
SCBCutPaperActionFullCut	フルカット
SCBCutPaperActionPartialCut	パーシャルカット
SCBCutPaperActionFullCutWithFeed	用紙送り付きフルカット
SCBCutPaperActionPartialCutWithFeed	用紙送り付きパーシャルカット

*ApiFunctions.m* / 各コマンド仕様書を参照ください。

## 8.53 SCBPeripheralChannel 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

```
typedef NS_ENUM(NSInteger, SCBPeripheralChannel) {
    SCBPeripheralChannelNo1,
    SCBPeripheralChannelNo2
};
```

### 定数

名称	説明
SCBPeripheralChannelNo1	チャネル 1
SCBPeripheralChannelNo2	チャネル 2

*ApiFunctions.m* を参照ください。

## 8.54 SCBSoundChannel コンスタント

サウンド/ブザーチャネル指定定数。

### 宣言

```
typedef NS_ENUM(NSInteger, SCBSoundChannel) {
    SCBSoundChannelNo1,
    SCBSoundChannelNo2
};
```

### 定数

名称	説明
SCBSoundChannelNo1	チャンネル 1
SCBSoundChannelNo2	チャンネル 2

*ApiFunctions.m* を参照ください。

## 8.55 SCBBarcodeSymbology コンスタント

バーコードシンボル指定定数。

### 宣言

```
typedef NS_ENUM(NSUInteger, SCBBarcodeSymbology) {
    SCBBarcodeSymbologyUPCE,
    SCBBarcodeSymbologyUPCA,
    SCBBarcodeSymbologyJAN8,
    SCBBarcodeSymbologyJAN13,
    SCBBarcodeSymbologyCode39,
    SCBBarcodeSymbologyITF,
    SCBBarcodeSymbologyCode128,
    SCBBarcodeSymbologyCode93,
    // SCBBarcodeSymbologyCodabar,
    SCBBarcodeSymbologyNW7
};
```

### 定数

名称	説明
SCBBarcodeSymbologyUPCE	UPC-E.
SCBBarcodeSymbologyUPCA	UPC-A.
SCBBarcodeSymbologyJAN8	JAN/EAN8.
SCBBarcodeSymbologyJAN13	JAN/EAN13.
SCBBarcodeSymbologyCode39	Code39.
SCBBarcodeSymbologyITF	ITF.
SCBBarcodeSymbologyCode128	Code128.
SCBBarcodeSymbologyCode93	Code93.
SCBBarcodeSymbologyCodabar	Codabar.
SCBBarcodeSymbologyNW7	NW7.

*ApiFunctions.m* / 各コマンド仕様書を参照ください。

## 8.56 SCBBarcodeWidth コンスタント

バーコード幅指定定数。

ExtMode を指定することで、コマンド仕様に準拠したバーコードモードが指定されます。

### 宣言

```
typedef NS_ENUM(NSUInteger, SCBBarcodeWidth) {
    SCBBarcodeWidthMode1,
    SCBBarcodeWidthMode2,
    SCBBarcodeWidthMode3,
    SCBBarcodeWidthMode4,
    SCBBarcodeWidthMode5,
    SCBBarcodeWidthMode6,
    SCBBarcodeWidthMode7,
    SCBBarcodeWidthMode8,
    SCBBarcodeWidthMode9,
    SCBBarcodeWidthExtMode1,
    SCBBarcodeWidthExtMode2,
    SCBBarcodeWidthExtMode3,
    SCBBarcodeWidthExtMode4,
    SCBBarcodeWidthExtMode5,
    SCBBarcodeWidthExtMode6,
    SCBBarcodeWidthExtMode7,
    SCBBarcodeWidthExtMode8,
    SCBBarcodeWidthExtMode9
};
```

### 定数

名称	説明
SCBBarcodeWidthMode1	モード 1
SCBBarcodeWidthMode2	モード 2
SCBBarcodeWidthMode3	モード 3
SCBBarcodeWidthMode4	モード 4
SCBBarcodeWidthMode5	モード 5
SCBBarcodeWidthMode6	モード 6
SCBBarcodeWidthMode7	モード 7
SCBBarcodeWidthMode8	モード 8
SCBBarcodeWidthMode9	モード 9
SCBBarcodeWidthExtMode1	Ext モード 1
SCBBarcodeWidthExtMode2	Ext モード 2
SCBBarcodeWidthExtMode3	Ext モード 3
SCBBarcodeWidthExtMode4	Ext モード 4
SCBBarcodeWidthExtMode5	Ext モード 5
SCBBarcodeWidthExtMode6	Ext モード 6
SCBBarcodeWidthExtMode7	Ext モード 7
SCBBarcodeWidthExtMode8	Ext モード 8
SCBBarcodeWidthExtMode9	Ext モード 9

*ApiFunctions.m* / 各コマンド仕様書を参照ください。

## 8.57 SCBPdf417Level コンスタント

PDF417 ECC（セキュリティレベル）指定定数。

### 宣言

```
typedef NS_ENUM(NSUInteger, SCBPdf417Level) {
    SCBPdf417LevelECC0,
    SCBPdf417LevelECC1,
    SCBPdf417LevelECC2,
    SCBPdf417LevelECC3,
    SCBPdf417LevelECC4,
    SCBPdf417LevelECC5,
    SCBPdf417LevelECC6,
    SCBPdf417LevelECC7,
    SCBPdf417LevelECC8
};
```

### 定数

名称	説明
SCBPdf417LevelECC0	セキュリティレベル 0
SCBPdf417LevelECC1	セキュリティレベル 1
SCBPdf417LevelECC2	セキュリティレベル 2
SCBPdf417LevelECC3	セキュリティレベル 3
SCBPdf417LevelECC4	セキュリティレベル 4
SCBPdf417LevelECC5	セキュリティレベル 5
SCBPdf417LevelECC6	セキュリティレベル 6
SCBPdf417LevelECC7	セキュリティレベル 7
SCBPdf417LevelECC8	セキュリティレベル 8

*ApiFunctions.m* / 各コマンド仕様書を参照ください。

## 8.58 SCBQRCodeModel コンスタント

QR コードモデル指定定数

### 宣言

```
typedef NS_ENUM(NSUInteger, SCBQRCodeModel) {
    SCBQRCodeModelNo1,
    SCBQRCodeModelNo2
};
```

### 定数

名称	説明
SCBQRCodeModelNo1	モデル 1
SCBQRCodeModelNo2	モデル 2

*ApiFunctions.m* / 各コマンド仕様書を参照ください。

## 8.59 SCBQrCodeLevel コンスタント

QR コード誤り訂正レベル指定定数。

### 宣言

```
typedef NS_ENUM(NSUInteger, SCBQrCodeLevel) {
    SCBQrCodeLevelL,
    SCBQrCodeLevelM,
    SCBQrCodeLevelQ,
    SCBQrCodeLevelH
};
```

### 定数

名称	説明
SCBQrCodeLevelL	誤り訂正レベル L
SCBQrCodeLevelM	誤り訂正レベル M
SCBQrCodeLevelQ	誤り訂正レベル Q
SCBQrCodeLevelH	誤り訂正レベル H

*ApiFunctions.m* / 各コマンド仕様書を参照ください。

## 8.60 SCBBitmapConverterRotation コンスタント

ビットマップ回転指定定数。

### 宣言

```
typedef NS_ENUM(NSUInteger, SCBBitmapConverterRotation) {
    SCBBitmapConverterRotationNormal,
    SCBBitmapConverterRotationRight90,
    SCBBitmapConverterRotationLeft90,
    SCBBitmapConverterRotationRotate180
};
```

### 定数

名称	説明
SCBBitmapConverterRotationNormal	回転なし
SCBBitmapConverterRotationRight90	右 90 度回転
SCBBitmapConverterRotationLeft90	左 90 度回転
SCBBitmapConverterRotationRotate180	180 度回転

*ApiFunctions.m* を参照ください。



## 8.61 SCBBlackMarkType コンスタント

ブラックマーク指定定数。

### 宣言

```
typedef NS_ENUM(NSInteger, SCBBlackMarkType) {
    SCBBlackMarkTypeInvalid,
    SCBBlackMarkTypeValid,
    SCBBlackMarkTypeValidWithDetection
};
```

### 定数

名称	説明
SCBBlackMarkTypeInvalid	ブラックマーク無効
SCBBlackMarkTypeValid	ブラックマーク有効
SCBBlackMarkTypeValidWithDetection	ブラックマーク有効 + ブラックマーク検出動作

*ApiFunctions.m* / 各コマンド仕様書を参照ください。

## 8.62 SCBPrintableAreaType コンスタント

印字領域タイプ指定定数。

### 宣言

```
typedef NS_ENUM(NSInteger, SCBPrintableAreaType) {
    SCBPrintableAreaTypeStandard,
    SCBPrintableAreaTypeType1,
    SCBPrintableAreaTypeType2,
    SCBPrintableAreaTypeType3,
    SCBPrintableAreaTypeType4
};
```

### 定数

名称	説明
SCBPrintableAreaTypeStandard	標準タイプ
SCBPrintableAreaTypeType1	タイプ 1
SCBPrintableAreaTypeType2	タイプ 2
SCBPrintableAreaTypeType3	タイプ 3
SCBPrintableAreaTypeType4	タイプ 4

*ApiFunctions.m* を参照ください。

## 8.63 SCBCjkUnifiedIdeographFont コンスタント

CJK 統合漢字のフォント指定定数。

### 宣言

```
typedef NS_ENUM(NSInteger, SCBCjkUnifiedIdeographFont) {
    SCBCjkUnifiedIdeographFontJapanese      = 1,
    SCBCjkUnifiedIdeographFontSimplifiedChinese = 2,
    SCBCjkUnifiedIdeographFontTraditionalChinese = 3,
    SCBCjkUnifiedIdeographFontHangul        = 4
};
```

### 定数

名称	説明
SCBCjkUnifiedIdeographFontJapanese	日本語フォント
SCBCjkUnifiedIdeographFontSimplifiedChinese	簡体字中国語フォント
SCBCjkUnifiedIdeographFontTraditionalChinese	繁体字中国語フォント
SCBCjkUnifiedIdeographFontHangul	韓国語フォント

*Utf8MultiLanguageReceiptsImpl.m* を参照ください。

## 9 ISDCBBuilder インターフェイス (StarIO\_Extension.framework)

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
appendByte	データ（テキストやコマンド）を commands プロパティに追加します。
appendData	
appendBytes	
appendBackSpace	バックスペースコマンドを生成し、commands プロパティに追加します。
appendHorizontalTab	水平タブコマンドを生成し、commands プロパティに追加します。
appendLineFeed	ラインフィードコマンドを生成し、commands プロパティに追加します。
appendCarriageReturn	キャリッジリターンコマンドを生成し、commands プロパティに追加します。
appendBitmap	グラフィック表示コマンドを生成し、commands プロパティに追加します。
appendInternational	国際文字指定コマンドを生成し、commands プロパティに追加します。
appendCodePage	コードページ指定コマンドを生成し、commands プロパティに追加します。
appendDeleteToEndOfLine	画面を行末までクリアするコマンドを生成し、commands プロパティに追加します。
appendClearScreen	画面をクリアするコマンドを生成し、commands プロパティに追加します。
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、commands プロパティに追加します。
appendTurnOn	バックライトをオン・オフするコマンドを生成し、commands プロパティに追加します。
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、commands プロパティに追加します。
appendCursorMode	カーソルモード変更コマンドを生成し、commands プロパティに追加します。
appendContrastMode	コントラストモード変更コマンドを生成し、commands プロパティに追加します。
appendUserDefinedCharacter	ユーザー定義文字(SBCS)登録コマンドを生成し、commands プロパティに追加します。
appendUserDefinedDbcsCharacter	ユーザー定義文字(DBCS)登録コマンドを生成し、commands プロパティに追加します。

### プロパティ

名称	説明
commands	生成、追加されたコマンド列です。* <i>readonly</i>
passThroughCommands	生成、追加されたコマンド列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です。* <i>readonly</i>

## コンスタント

名称	説明
SDCBIInternationalType	国際文字指定定数。
SDCBCodePageType	コードページ指定定数。
SDCBCursorMode	カーソルモード指定定数。
SDCBContrastMode	コントラストモード指定定数。

### 9.1 モデル : ISDCBBuilder インターフェイス

ISDCBBuilder 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 appendData メソッド

データ（テキストやコマンド）を commands プロパティに追加します。

### 宣言

- (void)appendByte:(unsigned char)data;
- (void)appendData:(NSData \*)otherData;
- (void)appendBytes:(const void \*)bytes length:(NSUInteger)length;

### 引数

名称	説明	型
data	データ（テキストやコマンド）	unsigned char
otherData		NSData
bytes		const void *
length		NSUInteger

### 戻り値

説明	型
-	-

### Example

```
+ (void)appendTextPattern:(ISDCBBUILDER *)builder number:(int)number {
// [builder appendClearScreen];
[builder appendCursorMode:SDCBCursorModeOff];
[builder appendSpecifiedPosition:1 y:1];

unsigned char pattern1[] =
"\x020\x021\x022\x023\x024\x025\x026\x027\x028\x029\x02a\x02b\x02c\x02d\x02e\x02f\x030\x031\x032\x033"
"\x034\x035\x036\x037\x038\x039\x03a\x03b\x03c\x03d\x03e\x03f\x040\x041\x042\x043\x044\x045\x046\x047";
...

switch (number) {
    default : [builder appendBytes:pattern1 length:sizeof(pattern1)]; break;    // 0
    case 1 : [builder appendBytes:pattern2 length:sizeof(pattern2)]; break;
    case 2 : [builder appendBytes:pattern3 length:sizeof(pattern3)]; break;
    case 3 : [builder appendBytes:pattern4 length:sizeof(pattern4)]; break;
    case 4 : [builder appendBytes:pattern5 length:sizeof(pattern5)]; break;
    case 5 : [builder appendBytes:pattern6 length:sizeof(pattern6)]; break;
}
}
```

DisplayFunctions.m 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、commands プロパティに追加します。

#### 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、commands プロパティに追加します。

#### 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、commands プロパティに追加します。

#### 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、commands プロパティに追加します。

### 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、commands プロパティに追加します。

### 宣言

```
- (void)appendBitmap:(UIImage *)image diffusion:(BOOL)diffusion;
```

### 引数

名称	説明	型
image	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。	UIImage *
diffusion	誤差拡散 <ul style="list-style-type: none"> <li>true ... 誤差拡散を行う</li> <li>false ... 誤差拡散を行わない</li> </ul>	BOOL

### 戻り値

説明	型
-	-

### Example

```
+ (void)appendGraphicPattern:(ISDCBBuilder *)builder number:(int)number {
// [builder appendClearScreen];
[builder appendCursorMode:SDCBCursorModeOff];
// [builder appendSpecifiedPosition:1 y:1];

UIImage *image;

switch (number) {
    default : image = [UIImage imageNamed:@"DisplayImage1.png"]; break;    // 0
    case 1 : image = [UIImage imageNamed:@"DisplayImage2.png"]; break;
    case 2 : image = [UIImage imageNamed:@"DisplayImage3.png"]; break;
    case 3 : image = [UIImage imageNamed:@"DisplayImage4.png"]; break;
}

[builder appendBitmap:image diffusion:YES];
}
```

*DisplayFunctions.m* 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、commands プロパティに追加します。

### 宣言

- (void)appendInternational:(SDCBInternationalType)type;

### 引数

名称	説明	型
type	国際文字指定定数	SDCBInternationalType

### 戻り値

説明	型
-	-

### Example

```
+ (void)appendCharacterSet:(ISDCBBuilder *)builder internationalType:(SDCBInternationalType)internationalType
codePageType:(SDCBCodePageType)codePageType {
// [builder appendClearScreen];
[builder appendCursorMode:SDCBCursorModeOff];
[builder appendSpecifiedPosition:1 y:1];

[builder appendInternational:internationalType];
[builder appendCodePage :codePageType];

    unsigned char pattern1[] =
    "\x02d\x020\x020\x020\x023\x024\x040\x05b\x05c\x05d\x05e\x060\x07b\x07c\x07d\x07e\x020\x020\x020\x02d"

    "\x0a0\x0a1\x0a2\x0a3\x0a4\x0a5\x0a6\x0a7\x0a8\x0a9\x0aa\x0ab\x0ac\x0ad\x0ae\x0af\x0b0\x0b1\x0b2\x0b3";

    ...

    switch (codePageType) {
        default : [builder appendBytes:pattern1 length:sizeof(pattern1)]; break; //
CP437,Katakana,CP850,CP860,CP863,CP865,CP1252,CP866,CP852,CP858
        case SDCBCodePageTypeJapanese : [builder appendBytes:pattern2 length:sizeof(pattern2)]; break;
        case SDCBCodePageTypeSimplifiedChinese : [builder appendBytes:pattern3 length:sizeof(pattern3)]; break;
        case SDCBCodePageTypeTraditionalChinese : [builder appendBytes:pattern4 length:sizeof(pattern4)]; break;
        case SDCBCodePageTypeHangul : [builder appendBytes:pattern5 length:sizeof(pattern5)]; break;
    }
}
```

DisplayFunctions.m 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、commands プロパティに追加します。

### 宣言

- (void)appendCodePage:(SDCBCCodePageType)type;

### 引数

名称	説明	型
type	コードページ指定定数	SDCBCCodePageType

### 戻り値

説明	型
-	-

### Example

```
+ (void)appendCharacterSet:(ISDCBBuilder *)builder internationalType:(SDCBInternationalType)internationalType
codePageType:(SDCBCCodePageType)codePageType {
// [builder appendClearScreen];
[builder appendCursorMode:SDCBCursorModeOff];
[builder appendSpecifiedPosition:1 y:1];

[builder appendInternational:internationalType];
[builder appendCodePage :codePageType];

unsigned char pattern1[] =
"\x02d\x020\x020\x020\x023\x024\x040\x05b\x05c\x05d\x05e\x060\x07b\x07c\x07d\x07e\x020\x020\x020\x02d"

"\x0a0\x0a1\x0a2\x0a3\x0a4\x0a5\x0a6\x0a7\x0a8\x0a9\x0aa\x0ab\x0ac\x0ad\x0ae\x0af\x0b0\x0b1\x0b2\x0b3";

...

switch (codePageType) {
default : [builder appendBytes:pattern1 length:sizeof(pattern1)]; break; //
CP437,Katakana,CP850,CP860,CP863,CP865,CP1252,CP866,CP852,CP858
case SDCBCCodePageTypeJapanese : [builder appendBytes:pattern2 length:sizeof(pattern2)]; break;
case SDCBCCodePageTypeSimplifiedChinese : [builder appendBytes:pattern3 length:sizeof(pattern3)]; break;
case SDCBCCodePageTypeTraditionalChinese : [builder appendBytes:pattern4 length:sizeof(pattern4)]; break;
case SDCBCCodePageTypeHangul : [builder appendBytes:pattern5 length:sizeof(pattern5)]; break;
}
}
```

DisplayFunctions.m 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、commands プロパティに追加します。

### 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、commands プロパティに追加します。

### 宣言

- (void)appendClearScreen;

### 引数

名称	説明	型
-	-	-

### 戻り値

説明	型
-	-

### Example

```
+ (void)appendClearScreen:(ISDCBBuilder *)builder {
    [builder appendClearScreen];
}
```

DisplayFunctions.m 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、commands プロパティに追加します。

### 宣言

- (void)appendHomePosition;

### 引数

名称	説明	型
-	-	-

### 戻り値

説明	型
-	-

## 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、commands プロパティに追加します。

### 宣言

- (void)appendTurnOn:(BOOL)turnOn;

### 引数

名称	説明	型
turnOn	バックライトのオン・オフ	BOOL

### 戻り値

説明	型
-	-

### Example

```
+ (void)appendTurnOn:(ISDCBBuilder *)builder turnOn:(BOOL)turnOn {
    ...
    [builder appendTurnOn:turnOn];
}
```

DisplayFunctions.m 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、commands プロパティに追加します。

### 宣言

- (void)appendSpecifiedPosition:(int)x y:(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

```
+ (void)appendCursorMode:(ISDCBBuilder *)builder cursorMode:(SDCBCursorMode)cursorMode {
// [builder appendClearScreen];
[builder appendCursorMode:SDCBCursorModeOff];
[builder appendSpecifiedPosition:1 y:1];

    unsigned char pattern[] = "Star Micronics    "
        "Total :    12345";

    [builder appendBytes:pattern length:sizeof(pattern)];

    [builder appendSpecifiedPosition:20 y:2];

    [builder appendCursorMode:cursorMode];
}
```

DisplayFunctions.m 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、commands プロパティに追加します。

### 宣言

- (void)appendCursorMode:(SDCBCursorMode)cursorMode;

### 引数

名称	説明	型
cursorMode	カーソルモード指定定数	SDCBCursorMode

### 戻り値

説明	型
-	-

### Example

```
+ (void)appendCursorMode:(ISDCBBuilder *)builder cursorMode:(SDCBCursorMode)cursorMode {
// [builder appendClearScreen];
[builder appendCursorMode:SDCBCursorModeOff];
[builder appendSpecifiedPosition:1 y:1];

    unsigned char pattern[] = "Star Micronics    "
        "Total :    12345";

    [builder appendBytes:pattern length:sizeof(pattern)];

    [builder appendSpecifiedPosition:20 y:2];

    [builder appendCursorMode:cursorMode];
}
```

DisplayFunctions.m 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、commands プロパティに追加します。

### 宣言

- (void)appendContrastMode:(SDCBCContrastMode)contrastMode;

### 引数

名称	説明	型
contrastMode	コントラストモード指定定数	SDCBCContrastMode

### 戻り値

説明	型
-	-

### Example

```
+ (void)appendContrastMode:(ISDCBBuilder *)builder contrastMode:(SDCBCContrastMode)contrastMode {
...
    [builder appendContrastMode:contrastMode];
}
```

DisplayFunctions.m 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、commands プロパティに追加します。

### 宣言

- (void)appendUserDefinedCharacter:(int)index code:(int)code font:(unsigned char \*)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 $20h \leq \text{code} \leq 7Fh$ appendCodePage メソッドで SDCBCodePageTypeJapanese 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。	unsigned char *

### 戻り値

説明	型
-	-

### Example

```
+ (void)appendUserDefinedCharacter:(ISDCBBuilder *)builder set:(BOOL)set {
// [builder appendClearScreen];
[builder appendCursorMode:SDCBCursorModeOff];
[builder appendSpecifiedPosition:1 y:1];

[builder appendInternational:SDCBInternationalTypeUSA];
[builder appendCodePage :SDCBCodePageTypeJapanese];

if (set) {
    [builder appendUserDefinedCharacter:0 code:0x20 font:(unsigned char *)
    "\x000\x000\x032\x000\x049\x000\x049\x07f\x026\x048\x000\x048\x000\x030\x000\x000"];

    [builder appendUserDefinedDbcsCharacter:0 code:0x8140 font:(unsigned char *)
    "\x000\x000\x000\x000\x000\x000\x000\x000\x003\x020\x004\x090\x004\x090\x002\x060"
    "\x000\x000\x007\x0f0\x004\x080\x004\x080\x003\x000\x000\x000\x000\x000\x000"];
}
else {
    [builder appendUserDefinedCharacter:0 code:0x00 font:nil];

    [builder appendUserDefinedDbcsCharacter:0 code:0x0000 font:nil];
}

unsigned char pattern[] =
"\x05b\x020\x020\x053\x074\x061\x072\x020\x04d\x069\x063\x072\x06f\x06e\x069\x063\x073\x020\x020\x05d"
"\x05b\x081\x040\x081\x040\x083\x058\x083\x05e\x081\x05b\x090\x0b8\x096\x0a7\x081\x040\x081\x040\x05d";

[builder appendBytes:pattern length:sizeof(pattern)];
}
```

DisplayFunctions.m 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、commands プロパティに追加します。

### 宣言

- (void)appendUserDefinedDbcsCharacter:(int)index code:(int)code font:(unsigned char \*)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 $20h \leq \text{code} \leq 7Fh$ appendCodePage メソッドで SDCBCodePageTypeJapanese 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。	unsigned char *

### 戻り値

説明	型
-	-

### Example

```
+ (void)appendUserDefinedCharacter:(ISDCBBuilder *)builder set:(BOOL)set {
// [builder appendClearScreen];
[builder appendCursorMode:SDCBCursorModeOff];
[builder appendSpecifiedPosition:1 y:1];

[builder appendInternational:SDCBInternationalTypeUSA];
[builder appendCodePage :SDCBCodePageTypeJapanese];

if (set) {
[builder appendUserDefinedCharacter:0 code:0x20 font:(unsigned char *)
"\x000\x000\x032\x000\x049\x000\x049\x07f\x026\x048\x000\x048\x000\x030\x000\x000"];

[builder appendUserDefinedDbcsCharacter:0 code:0x8140 font:(unsigned char *)
"\x000\x000\x000\x000\x000\x000\x000\x000\x003\x020\x004\x090\x004\x090\x002\x060"
"\x000\x000\x007\x0f0\x004\x080\x004\x080\x003\x000\x000\x000\x000\x000\x000"];
}
else {
[builder appendUserDefinedCharacter:0 code:0x00 font:nil];

[builder appendUserDefinedDbcsCharacter:0 code:0x0000 font:nil];
}

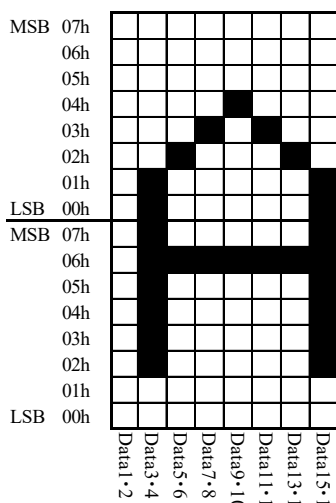
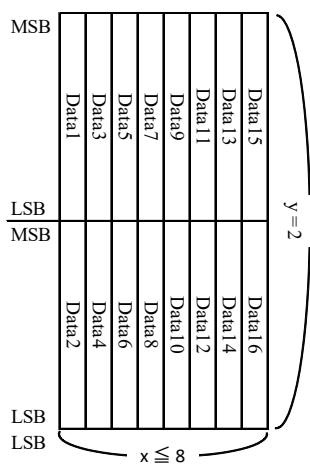
unsigned char pattern[] =
"\x05b\x020\x020\x053\x074\x061\x072\x020\x04d\x069\x063\x072\x06f\x06e\x069\x063\x073\x020\x020\x05d"
"\x05b\x081\x040\x081\x040\x083\x058\x083\x05e\x081\x05b\x090\x0b8\x096\x0a7\x081\x040\x081\x040\x05d";

[builder appendBytes:pattern length:sizeof(pattern)];
}
```

DisplayFunctions.m を参照ください。

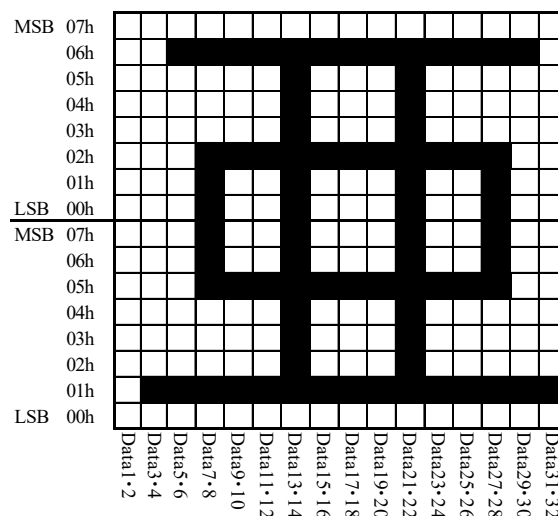
## ユーザ定義文字のフォントデータのフォーマット

### ユーザ一定義文字 (SBCD)



Byte	1	3	5	7	9	11	13	15
Data	00h	03h	04h	08h	10h	08h	04h	03h
Byte	2	4	6	8	10	12	14	16
Data	00h	FCh	40h	40h	40h	40h	40h	FCh

ユーザ一定義文字 (DBCS)



Byte	1	3	5	7	9	11	13	15	17	19	21	23	25	27	29	31
Data	00h	00h	40h	47h	44h	44h	7Fh	44h	44h	44h	7Fh	44h	44h	47h	44h	00h
Byte	2	4	6	8	10	12	14	16	18	20	22	24	26	28	30	32
Data	00h	02h	02h	E2h	22h	22h	FEh	22h	22h	22h	FEh	22h	22h	E2h	22h	02h



## 9.19 commands プロパティ

生成、追加されたコマンド列です。

### 宣言

```
@property (nonatomic, readonly) NSMutableData *commands;
```

### 値

説明	型
生成、追加されたコマンド列	NSMutableData

## 9.20 passThroughCommands プロパティ

生成、追加されたコマンド列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です。

### 宣言

```
@property (nonatomic, readonly) NSMutableData *passThroughCommands;
```

### 値

説明	型
生成、追加されたコマンド列	NSMutableData

### Example

```
ISDCBBuilder *builder = [StarloExt createDisplayCommandBuilder:StarloExtDisplayModelISCD222];
```

```
...
```

```
NSData *commands = [builder.passThroughCommands copy];
```

*DisplayViewController.m / DisplayExtViewController.m を参照ください。*

## 9.21 SDCBInternationalType コンスタント

国際文字指定定数。

### 宣言

```
typedef NS_ENUM(NSInteger, SDCBInternationalType) {
    SDCBInternationalTypeUSA      = 0x00,
    SDCBInternationalTypeFrance   = 0x01,
    SDCBInternationalTypeGermany  = 0x02,
    SDCBInternationalTypeUK       = 0x03,
    SDCBInternationalTypeDenmark  = 0x04,
    SDCBInternationalTypeSweden   = 0x05,
    SDCBInternationalTypeItaly    = 0x06,
    SDCBInternationalTypeSpain    = 0x07,
    SDCBInternationalTypeJapan    = 0x08,
    SDCBInternationalTypeNorway   = 0x09,
    SDCBInternationalTypeDenmark2 = 0x0a,
    SDCBInternationalTypeSpain2   = 0x0b,
    SDCBInternationalTypeLatinAmerica = 0x0c,
    SDCBInternationalTypeKorea    = 0x0d
};
```

### 定数

名称	説明
SDCBInternationalTypeUSA	USA
SDCBInternationalTypeFrance	フランス
SDCBInternationalTypeGermany	ドイツ
SDCBInternationalTypeUK	イギリス
SDCBInternationalTypeDenmark	デンマーク
SDCBInternationalTypeSweden	スウェーデン
SDCBInternationalTypeItaly	イタリア
SDCBInternationalTypeSpain	スペイン
SDCBInternationalTypeJapan	日本
SDCBInternationalTypeNorway	ノルウェー
SDCBInternationalTypeDenmark2	デンマーク II
SDCBInternationalTypeSpain2	スペイン II
SDCBInternationalTypeLatinAmerica	ラテンアメリカ
SDCBInternationalTypeKorea	韓国

## 9.22 SDCBCodePageType コンスタント

コードページ指定定数。

### 宣言

```
typedef NS_ENUM(NSInteger, SDCBCodePageType) {
    SDCBCodePageTypeCP437      = 0x00,
    SDCBCodePageTypeKatakana   = 0x01,
    SDCBCodePageTypeCP850      = 0x02,
    SDCBCodePageTypeCP860      = 0x03,
    SDCBCodePageTypeCP863      = 0x04,
    SDCBCodePageTypeCP865      = 0x05,
    SDCBCodePageTypeCP1252     = 0x06,
    SDCBCodePageTypeCP866      = 0x07,
    SDCBCodePageTypeCP852      = 0x08,
    SDCBCodePageTypeCP858      = 0x09,
    SDCBCodePageTypeJapanese   = 0x0a,
    SDCBCodePageTypeSimplifiedChinese = 0x0b,
    SDCBCodePageTypeTraditionalChinese = 0x0c,
    SDCBCodePageTypeHangul     = 0x0d
};
```

### 定数

名称	説明
SDCBCodePageTypeCP437	CodePage437 (USA, Std. Europe).
SDCBCodePageTypeKatakana	Katakana.
SDCBCodePageTypeCP850	PC850 (Multilingual)
SDCBCodePageTypeCP860	PC860 (Portuguese)
SDCBCodePageTypeCP863	PC863 (Canadian-French)
SDCBCodePageTypeCP865	PC865 (Norwegian)
SDCBCodePageTypeCP1252	WPC1252
SDCBCodePageTypeCP866	PC866 [Cyrillic #2]
SDCBCodePageTypeCP852	PC852 [Latin 2]
SDCBCodePageTypeCP858	Page 19 [PC858]
SDCBCodePageTypeJapanese	Japanese font (shift JIS)
SDCBCodePageTypeSimplifiedChinese	Simplified Chinese (GB2312)
SDCBCodePageTypeTraditionalChinese	Traditional Chinese (Big5)
SDCBCodePageTypeHangul	Hangul (KSC5601)

## 9.23 SDCBCursorMode コンスタント

カーソルモード指定定数。

### 宣言

```
typedef NS_ENUM(NSInteger, SDCBCursorMode) {
    SDCBCursorModeOff = 0x00,
    SDCBCursorModeBlink = 0x01,
    SDCBCursorModeOn = 0x02
};
```

### 定数

名称	説明
SDCBCursorModeOff	カーソル消灯
SDCBCursorModeBlink	カーソル点滅
SDCBCursorModeOn	カーソル点灯

## 9.24 SDCBContrastMode コンスタント

コントラストモード指定定数。

### 宣言

```
typedef NS_ENUM(NSInteger, SDCBContrastMode) {
    SDCBContrastModeMinus3 = 0x00,
    SDCBContrastModeMinus2 = 0x01,
    SDCBContrastModeMinus1 = 0x02,
    SDCBContrastModeDefault = 0x03,
    SDCBContrastModePlus1 = 0x04,
    SDCBContrastModePlus2 = 0x05,
    SDCBContrastModePlus3 = 0x06
};
```

### 定数

名称	説明
SDCBContrastModeMinus3	コントラストマイナス 3
SDCBContrastModeMinus2	コントラストマイナス 2
SDCBContrastModeMinus1	コントラストマイナス 1
SDCBContrastModeDefault	デフォルト
SDCBContrastModePlus1	コントラストプラス 1
SDCBContrastModePlus2	コントラストプラス 2
SDCBContrastModePlus3	コントラストプラス 3

## 10 ISMCBBuilder インターフェイス (StarIO\_Extension.framework)

メロディスピーカー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
appendSoundWithSetting	メロディスピーカーに登録されたサウンドを再生するコマンドを生成し、commands プロパティに追加します。
appendSoundWithSound	引数で指定されたサウンドデータを再生するコマンドを生成し、commands プロパティに追加します。

### プロパティ

名称	説明
commands	生成、追加されたコマンド列です。* <i>readonly</i>

### コンスタント

名称	説明
SMCBMelodySpeakerSoundStorageArea	サウンド格納領域指定定数。

## 10.1 モデル : ISMCBBuilder インターフェイス

ISMCBBuilder インターフェイスはメロディスピーカーに対応しているモデルまたは FVP10 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 appendSoundWithSetting メソッド

メロディスピーカーに登録されたサウンドを再生するコマンドを生成し、commands プロパティに追加します。

### 宣言

```
-(void)appendSoundWithSetting:(nullable SMSoundSetting *)setting error:(nullable NSError **)error;
```

### 引数

名称	説明	型
setting	再生するサウンドの設定情報（サウンド格納領域、サウンド番号など）	SMSoundSetting
error	コマンド生成に失敗した際のエラー情報	NSError

setting 引数に設定可能なプロパティおよびパラメータは以下の通りです。

StarloExtMelody SpeakerModel	プロパティ	説明	設定可能パラメータ	デフォルト（未指定時の動作）
StarloExtMelody SpeakerModel MCS10	soundStorageArea	サウンド格納領域	SMCBMelodySpeakerSoundStorageArea1 SMCBMelodySpeakerSoundStorageArea2	MCS10 本体のディップスイッチ設定を使用
	soundNumber	サウンド番号	0 - 7	MCS10 本体のディップスイッチ設定を使用
	volume	音量	0 - 15 SMSoundSetting.volumeOff SMSoundSetting.volumeMax SMSoundSetting.volumeMin	MCS10 本体のディップスイッチ設定を使用
StarloExtMelody SpeakerModel FVP10	soundStorageArea	サウンド格納領域	SMCBMelodySpeakerSoundStorageArea1 SMCBMelodySpeakerSoundStorageArea2	SMCBMelodySpeakerSoundStorageArea1
	soundNumber	サウンド番号	1 - 255	1
	count	回数	1 - 65535	1
	delay	ディレイ時間（1 秒単位）	0 - 65535	0
	interval	インターバル時間（1 秒単位）	0 - 65535	0

soundStorageArea プロパティと soundNumber プロパティに指定するパラメータの組み合わせにより再生されるサウンドは以下の通りです。

StarloExtMelodySpeakerModel	soundStorageArea プロパティ	soundNumber プロパティ	再生されるサウンド
StarloExtMelodySpeakerModelMCS10	SMCBMelodySpeakerSoundStorageArea1	0	サウンド 1
		1	サウンド 2
		2	サウンド 3
		3	サウンド 4
		4	サウンド 5
		5	サウンド 6
		6	サウンド 7
		7	サウンド 8
	SMCBMelodySpeakerSoundStorageArea2	0	サウンド 9
		1	サウンド 10
		2	サウンド 11
		3	サウンド 12
		4	サウンド 13
		5	サウンド 14
		6	サウンド 15
		7	サウンド 16
StarloExtMelodySpeakerModelFVP10	SMCBMelodySpeakerSoundStorageArea1	1 - 255	デフォルトエリアの指定されたサウンド番号に格納されたサウンド
	SMCBMelodySpeakerSoundStorageArea2	1 - 255	ユーザーエリアの指定されたサウンド番号に格納されたサウンド

## 戻り値

説明	型
-	-



## エラーコード

値	説明
SMResultCodeInvalidArgument	以下のいずれか 1. 指定されたモデルでは設定できないプロパティが設定されている場合 2. soundStorageArea プロパティと soundNumber プロパティのどちらか一方のみが設定されている場合
SMResultCodeArgumentOutOfRange	設定可能範囲外のパラメータが指定されている場合

### Example

```
+ (NSData *)createPlayingRegisteredSound:(StarIoExtMelodySpeakerModel)model
    specifySound:(BOOL)specifySound
    soundStorageArea:(SMCBSoundStorageArea)soundStorageArea
    soundNumber:(NSInteger)soundNumber
    specifyVolume:(BOOL)specifyVolume
    volume:(NSInteger)volume
    error:(NSError * _Nullable * _Nullable) error {
    ISMCBBUILDER *builder = [StarIoExt createMelodySpeakerCommandBuilder:model];

    SMSoundSetting *setting = [SMSoundSetting new];

    if (specifySound == YES) {
        [setting setSoundStorageArea:soundStorageArea];
        [setting setSoundNumber:soundNumber];
    }

    if (specifyVolume == YES) {
        [setting setVolume:volume];
    }

    [builder appendSoundWithSetting:setting error:error];

    if (*error != nil) {
        return nil;
    }

    return [builder.commands copy];
}
```

MelodySpeakerFunctions.m を参照ください。

## 10.3 appendSoundWithSound メソッド

引数で指定されたサウンドデータを再生するコマンドを生成し、commands プロパティに追加します。

### 宣言

```
- (void)appendSoundWithSound:(nonnull NSData *)data (nullable SMSoundSetting *)setting error:(nullable NSError **)error;
```

### 引数

名称	説明	型
data	サウンドデータ 対応サウンドフォーマットは対応サウンドフォーマット表を参照	NSData
setting	再生するサウンドの設定情報（サウンド格納領域、サウンド番号など）	SMSoundSetting
error	コマンド生成に失敗した際のエラー情報	NSError

### 対応サウンドフォーマット表

StarloExtMelodySpeakerModel	サウンドフォーマット
StarloExtMelodySpeakerModelMCS10	サウンドファイルフォーマット: WAV WAV ファイルフォーマット: リニア PCM サンプリングレート: 12800Hz ビット深度: 16bit または 8 bit チャンネル: モノラル 再生時間: ビット深度 16bit の場合 5 秒以内 ビット深度 8bit の場合 10 秒以内
StarloExtMelodySpeakerModelFVP10	非対応

1. 本パッケージ同梱の Star Sound Converter アプリケーションを利用することで、サウンドファイルのフォーマットを対応サウンドフォーマット表のフォーマットに変換することができます。詳細は”Others/StarSoundConverter/(Mac or Windows)/Documents”フォルダ内の”FileFormatList\_jp.htm”を参照ください。

setting 引数に設定可能なプロパティおよびパラメータは以下の通りです。

StarIoExtMelody SpeakerModel	プロパティ	説明	設定可能パラメータ	デフォルト（未指定時の動作）
StarIoExtMelody SpeakerModelMCS10	volume	音量	0 – 15 SMSoundSetting.volumeOff SMSoundSetting.volumeMax SMSoundSetting.volumeMin	MCS10 本体のディップスイッチ設定を使用
StarIoExtMelody SpeakerModelFVP10	非対応			

## 戻り値

説明	型
-	-

## エラーコード

値	説明
StarIoExtErrorFormatError	対応サウンドフォーマット以外のデータが入力された場合
SMResultCodeInvalidArgument	指定されたモデルでは設定できないプロパティが設定されている場合
SMResultCodeArgumentOutOfRange	設定可能範囲外のパラメータが指定されている場合
StarIoExtErrorInvalidOperation	StarIoExtMelodySpeakerModelFVP10 指定時に本メソッドが呼ばれた場合

## Example

```
+ (NSData *)createPlayingReceivedData:(StarIoExtMelodySpeakerModel)model
    filePath:(NSString *)filePath
    specifyVolume:(BOOL)specifyVolume
    volume:(NSInteger)volume
    error:(NSError * _Nullable * _Nullable)error {
    ISMCBBUILDER *builder = [StarIoExt createMelodySpeakerCommandBuilder:model];

    SMSoundSetting *setting = [SMSoundSetting new];

    NSURL *fileUrl = [[NSURL alloc] initWithString:filePath];

    NSData *fileData = [[NSData alloc] initWithContentsOfURL:fileUrl];

    if (specifyVolume == YES) {
        [setting setVolume:volume];
    }

    [builder appendSoundWithSound:fileData
                     setting:setting
                     error:error];

    if (*error != nil) {
        return nil;
    }

    return [builder.commands copy];
}
```

*MelodySpeakerFunctions.m* を参照ください。

## 10.4 commands プロパティ

生成、追加されたコマンド列です。

### 宣言

```
@property (nonatomic, readonly) NSMutableData *commands;
```

### 値

説明	型
生成、追加されたコマンド列	NSMutableData

## 10.5 SMCBMelodySpeakerSoundStorageArea コンスタント

サウンド格納領域指定定数。

### 宣言

```
typedef NS_ENUM(NSUInteger, SMCBMelodySpeakerSoundStorageArea) {
    SMCBMelodySpeakerSoundStorageArea1,
    SMCBMelodySpeakerSoundStorageArea2
};
```

定数に対応するサウンド格納領域は以下の通りです。

StarIoExtMelody SpeakerModel	定数	サウンド格納領域
StarIoExtMelody SpeakerModelMCS10	SMCBMelodySpeakerSoundStorageArea1	サウンド格納領域 1
	SMCBMelodySpeakerSoundStorageArea2	サウンド格納領域 2
StarIoExtMelody SpeakerModelFVP10	SMCBMelodySpeakerSoundStorageArea1	デフォルトエリア
	SMCBMelodySpeakerSoundStorageArea2	ユーザーエリア

*MelodySpeakerFunctions.m* を参照ください。

## 11 SMSoundSetting クラス (StarIO\_Extension.framework)

サウンドに関する設定を定義するクラスです。

### プロパティ

名称	説明
sounStorageArea	サウンド格納領域
soundNumber	サウンド番号
volume	音量
count	再生回数
delay	ディレイ時間 (1 秒単位)
interval	インターバル時間 (1 秒単位)

### コンスタント

名称	説明
SMSoundSettingVolume	ボリューム指定定数。

## 11.1 soundStorageArea プロパティ

サウンド格納領域を指定するプロパティです。

### 宣言

@property (nonatomic) SMCBMelodySpeakerSoundStorageArea soundStorageArea;

### 値

説明	型
サウンド格納領域	SMCBMelodySpeakerSoundStorageArea

## 11.2 soundNumber プロパティ

サウンド番号を指定するプロパティです。

### 宣言

@property (nonatomic) NSInteger soundNumber;

### 値

説明	型
サウンド番号	NSInteger

## 11.3 volume プロパティ

音量を指定するプロパティです。

### 宣言

@property (nonatomic) NSInteger volume;

### 値

説明	型
音量	NSInteger

## 11.4 count プロパティ

サウンドの再生回数を指定するプロパティです。

### 宣言

@property (nonatomic) NSInteger count;

### 値

説明	型
サウンドの再生回数	NSInteger



## 11.5 delay プロパティ

ディレイ時間を指定するプロパティです。

### 宣言

```
@property (nonatomic) NSInteger delay;
```

### 値

説明	型
ディレイ時間（1 秒単位）	NSInteger

## 11.6 interval プロパティ

インターバル時間を指定するプロパティです。

### 宣言

```
@property (nonatomic) NSInteger interval;
```

### 値

説明	型
インターバル時間（1 秒単位）	NSInteger

## 11.7 SMSoundSettingVolume コンスタント

ボリューム指定定数。

### 宣言

```
typedef NS_ENUM(NSInteger, SMSoundSettingVolume) {
    SMSoundSettingVolumeOff = -2,
    SMSoundSettingVolumeMin = -3,
    SMSoundSettingVolumeMax = -4
};
```

### 定数

名称	説明
volumeOff	音量オフを示す定数です
volumeMin	最小音量を示す定数です。
volumeMax	最大音量を示す定数です。

SMSoundSetting クラスのプロパティに使用したときに設定されるパラメータは以下の通りです。

StarloExtMelody SpeakerModel	定数	プロパティ	パラメータ
StarloExtMelody SpeakerModelMCS10	SMSoundSettingVolumeOff	volume	0
	SMSoundSettingVolumeMin	volume	1
	SMSoundSettingVolumeMax	volume	15
StarloExtMelody SpeakerModelFVP10	SMSoundSettingVolumeOff	非対応	
	SMSoundSettingVolumeMin	非対応	
	SMSoundSettingVolumeMax	非対応	

## 12 ISCPParser インターフェイス (StarIO\_Extension.framework)

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・バーコードリーダー)からの応答を受け取るコマンドを生成します。

### プロパティ

名称	説明
completionHandler	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・バーコードリーダー)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
StarIoExtParserCompletionResult	解析結果定数。

### 12.1 モデル : ISCPParser インターフェイス

ISCPParser 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。

## 12.2 createSendCommands メソッド

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

### 宣言

```
-(NSData *)createSendCommands;
```

### 引数

名称	説明	型
-	-	-

### 戻り値

説明	型
生成されたコマンド列	NSData *

## Example

```
+ (BOOL)parseDoNotCheckCondition:(ISCPParser *)parser
    port:(SMPort *)port
    completionHandler:(SendCompletionHandler)completionHandler {
    CommResult result = CommResultErrorOpenPort;
    NSInteger code = SMStarIOResultCodeFailedError;

    NSData *commands = [parser createSendCommands];

    while (YES) {
        NSError *error = nil;

        ...

        StarPrinterStatus_2 printerStatus;

        ...

        NSDate *startDate = [NSDate date];

        uint32_t total = 0;

        while (total < (uint32_t) commands.length) {
            uint32_t written = [port writePort:(unsigned char *) commands.bytes :total :(uint32_t) commands.length -
            total :&error];

            ...
        }

        ...

        startDate = [NSDate date];    // Restart

        NSMutableData *receivedData = [NSMutableData data];

        while (YES) {
            uint8_t buffer[1024 + 8] = {0};

            ...

            uint32_t readLength = [port readPort:buffer :0 :1024 :&error];

            ...

            [receivedData appendBytes:buffer length:readLength];

            int recvDataLength = (int) receivedData.length;

            uint8_t *recvDataBytes = (uint8_t *) receivedData.bytes;

            if (parser.completionHandler(recvDataBytes, &recvDataLength) ==
StarIoExtParserCompletionResultSuccess) {
                result = CommResultSuccess;
                code = SMStarIOResultCodeSuccess;

                break;
            }
        }

        break;
    }

    ...
}
```

Communication.m を参照ください。

## 12.3 completionHandler プロパティ

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・バーコードリーダー)からのコマンド応答を解析します。

### 宣言

```
typedef StarloExtParserCompletionResult (^StarloExtParserCompletionHandler)(uint8_t *buffer, int *length);
```

```
@property (nonatomic, copy) StarloExtParserCompletionHandler completionHandler;
```

### 値

説明	型
解析ハンドラ	StarloExtParserCompletionHandler

### Example

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

## 12.4 StarloExtParserCompletionResult コンスタント

解析結果定数。

### 宣言

```
typedef NS_ENUM(NSInteger, StarloExtParserCompletionResult) {
    StarloExtParserCompletionResultInvalid = 0,
    StarloExtParserCompletionResultSuccess,
    StarloExtParserCompletionResultFailure
};
```

### 定数

名称	説明
StarloExtParserCompletionResultInvalid	不定 (コマンド応答未完了)
StarloExtParserCompletionResultSuccess	成功
StarloExtParserCompletionResultFailure	失敗

## 13 ISCPConnectParser インターフェイス (StarIO\_Extension.framework)

外部機器(カスタマーディスプレイ・バーコードリーダー・メロディスピーカー)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

@interface ISCPConnectParser : ISCPParser

### メソッド

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

### 13.1 モデル : ISCPConnectParser インターフェイス

ISCPConnectParser 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・メロディスピーカー)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。

## 13.2 connect 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・メロディスピーカー)の接続・切断ステータスを取得します。

completionHandler が StarloExtParserCompletionResultSuccess を返した後に実行してください。

### 宣言

- (BOOL)connect;

### 引数

名称	説明	型
-	-	-

### 戻り値

説明	型
接続・切断ステータス	BOOL

### Example

```
ISCPConnectParser *parser = [StarloExt createDisplayConnectParser:StarloExtDisplayModelSCD222];

[Communication parseDoNotCheckCondition:parser port:port completionHandler:^(BOOL result, NSString *title, NSString *message) {
    if (result == YES) {
        if (parser.connect == YES) {
            UIAlertView *alertView = [[UIAlertView alloc] initWithTitle:@"Check Status" message:@"Display Connect."
            delegate:nil cancelButtonTitle:@"OK" otherButtonTitles:nil];

            [alertView show];
        }
        else {
            UIAlertView *alertView = [[UIAlertView alloc] initWithTitle:@"Check Status" message:@"Display Disconnect."
            delegate:nil cancelButtonTitle:@"OK" otherButtonTitles:nil];

            [alertView show];
        }
    }
    else {
        // UIAlertView *alertView = [[UIAlertView alloc] initWithTitle:@"Failure" message:@"Display Impossible." delegate:nil
        cancelButtonTitle:@"OK" otherButtonTitles:nil];
        UIAlertView *alertView = [[UIAlertView alloc] initWithTitle:@"Failure" message:@"Printer Impossible." delegate:nil
        cancelButtonTitle:@"OK" otherButtonTitles:nil];

        [alertView show];
    }
}];
```

DisplayViewController.m / DisplayExtViewController.m / MelodySpeakerViewController.m を参照ください。

## 14 StarPRNT iOS SDK Sample

### 14.1 Communication

StarIO iOS 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 (StarIO iOS SDK 相当)

```
+ (BOOL)sendCommands:(NSData *)commands portName:(NSString *)portName portSettings:(NSString
*)portSettings timeout:(NSInteger)timeout completionHandler:(SendCompletionHandler)completionHandler {
    ...
    SMPort *port = nil;

    while (YES) {
        NSError *error = nil;

        port = [SMPort getPort:portName :portSettings :(uint32_t) timeout :&error];

        if (port == nil) { break; }

        StarPrinterStatus_2 printerStatus;

        [port beginCheckedBlock:&printerStatus :2 :&error];

        if (error != nil) { break; }

        if (printerStatus.offline == SM_TRUE) {
            break;
        }

        NSDate *startDate = [NSDate date];

        uint32_t total = 0;

        while (total < (uint32_t) commands.length) {
            uint32_t written = [port writePort:(unsigned char *) commands.bytes :total :(uint32_t) commands.length -
total :&error];

            if (error != nil) { break; }

            total += written;

            if ([NSDate date] timeIntervalSinceDate:startDate] >= 30.0) { // 30000mS!!!
                break;
            }
        }

        if (total < (uint32_t) commands.length) {
            break;
        }

        port.endCheckedBlockTimeoutMillis = 30000; // 30000mS!!!

        [port endCheckedBlock:&printerStatus :2 :&error];

        if (error != nil) { break; }

        if (printerStatus.offline == SM_TRUE) {
            break;
        }

        break;
    }

    if (port != nil) {
        [SMPort releasePort:port];

        port = nil;
    }
    ...
}
```

### Example (StarIoExtManager オブジェクトを使用)

```
+ (BOOL)sendCommands:(NSData *)commands
    port:(SMPort *)port
completionHandler:(SendCompletionHandler)completionHandler {
    ...

    while (YES) {
        NSError *error = nil;

        if (port == nil) {
            break;
        }

        StarPrinterStatus_2 printerStatus;

        [port beginCheckedBlock:&printerStatus :2 :&error];

        if (error != nil) {
            break;
        }

        if (printerStatus.offline == SM_TRUE) {
            break;
        }

        NSDate *startDate = [NSDate date];

        uint32_t total = 0;

        while (total < (uint32_t) commands.length) {
            uint32_t written = [port writePort:(unsigned char *) commands.bytes :total :(uint32_t) commands.length -
total :&error];

            if (error != nil) {
                break;
            }

            total += written;

            if ([NSDate date] timeIntervalSinceDate:startDate] >= 30.0) {    // 30000mS!!!
                break;
            }
        }

        if (total < (uint32_t) commands.length) {
            break;
        }

        port.endCheckedBlockTimeoutMillis = 30000;    // 30000mS!!!

        [port endCheckedBlock:&printerStatus :2 :&error];

        if (error != nil) {
            break;
        }

        if (printerStatus.offline == SM_TRUE) {
            break;
        }

        break;
    }
    ...
}
```

## 15 StarIoExtManager クラス (StarIO\_Extension.framework)

### メソッド

名称	説明
initWithType	StarIoExtManager を初期化します。
connectAsync	マネージメントを開始します。
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。

### プロパティ

名称	説明
port	SMPort オブジェクト。 * <i>readonly</i>
lock	port プロパティによる通信権利の排他制御オブジェクト。 * <i>readonly</i>
delegate	StarIoExtManager のデリゲートオブジェクト。
printerStatus	プリンタステータス。 * <i>readonly</i>
printerPaperStatus	プリンタ用紙ステータス。 * <i>readonly</i>
printerCoverStatus	プリンタカバーステータス。 * <i>readonly</i>
cashDrawerStatus	キャッシュドロアーステータス。 * <i>readonly</i>
barcodeReaderStatus	バーコードリーダーステータス。 * <i>readonly</i>
cashDrawerOpenActiveHigh	キャッシュドロア開閉センサのモード。

### コンスタント

名称	説明
StarIoExtManagerType	マネージャタイプ定数。
StarIoExtManagerPrinterStatus	プリンタステータス定数。
StarIoExtManagerPrinterPaperStatus	プリンタ用紙ステータス定数。
StarIoExtManagerPrinterCoverStatus	プリンタカバーステータス定数。
StarIoExtManagerCashDrawerStatus	キャッシュドロアーステータス定数。
StarIoExtManagerBarcodeReaderStatus	バーコードリーダーステータス定数。

-Bluetooth 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。

## 15.1 initWithType メソッド

StarloExtManager を初期化します。

### 宣言

```
- (id)initWithType:(StarloExtManagerType)type portName:(NSString *)portName portSettings:(NSString *)portSettings ioTimeoutMillis:(NSUInteger)ioTimeoutMillis;
```

### 引数

名称	説明	型
type	マネージャタイプ <ul style="list-style-type: none"> <li>StarloExtManagerTypeStandard ... プリンタ、キャッシュドロワのマネージメント</li> <li>StarloExtManagerTypeWithBarcodeReader ... 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</li> <li>StarloExtManagerTypeOnlyBarcodeReader ... バーコードリーダーのマネージメント</li> </ul>	StarloExtManagerType
portName	プリンタポート名	NSString *
portSettings	ポートセッティング	NSString *
ioTimeoutMillis	内部制御および API のタイムアウト値	NSUInteger

### 戻り値

説明	型
StarloExtManager オブジェクト	StarloExtManager

### Example

```
- (void)viewDidLoad {
    [super viewDidLoad];
    // Do any additional setup after loading the view.

    _starloExtManager = [[StarloExtManager alloc] initWithType:StarloExtManagerTypeStandard
                                                             portName:[AppDelegate getPortName]
                                                             portSettings:[AppDelegate getPortSettings]
                                                             ioTimeoutMillis:10000]; // 10000mS!!!

    _starloExtManager.delegate = self;
}
```

PrinterExtViewController.m / CashDrawerExtViewController.m / BarcodeReaderExtViewController.m / CombinationExtViewController.m を参照ください。

## 15.2 connectAsync メソッド

マネージメントを開始します。

本メソッドは非同期的に実行され、接続結果は delegate プロパティにセットしたオブジェクトに通知されます。プリンタとの接続に成功した場合 didConnectPort メソッドが呼ばれ、失敗した場合 didFailToConnectPort メソッドが呼ばれます。

### 宣言

- (void)connectAsync;

### 引数

名称	説明	型
-	-	-

### 戻り値

説明	型
-	-

### Example

```
- (void)refreshPrinter {
    self.blind = YES;

    [self.starIoExtManager disconnect];

    [self.starIoExtManager connectAsync];
}
```

PrinterExtWithConnectAsyncViewController.m を参照ください。

## 15.3 connect メソッド

マネージメントを開始します。  
本メソッドは同期的に実行されます。

### 宣言

```
-(BOOL)connect;
```

### 引数

名称	説明	型
-	-	-

### 戻り値

説明	型
リザルト <ul style="list-style-type: none"> <li>YES ... 成功</li> <li>NO ... 失敗</li> </ul>	BOOL

### Example

```
-(void)viewWillAppear:(BOOL)animated {
    [super viewWillAppear:animated];

    if (_starloExtManager.port != nil) {
        [_starloExtManager disconnect];
    }

    if ([_starloExtManager connect] == NO) {
        UIAlertView *alert = [[UIAlertView alloc] initWithTitle:@"Fail to Open Port."
                                                             message:@""
                                                             delegate:self
                                                             cancelButtonTitle:@"OK"
                                                             otherButtonTitles:nil];

        [alert show];
    }
}
```

*PrinterExtViewController.m / CashDrawerExtViewController.m / BarcodeReaderExtViewController.m / CombinationExtViewController.m* を参照ください。

## 15.4 disconnect メソッド

マネージメントを停止します。

### 宣言

- (BOOL)disconnect;

### 引数

名称	説明	型
-	-	-

### 戻り値

説明	型
リザルト	BOOL
• YES ... 成功	
• NO ... 失敗	

### Example

```
- (void)viewWillDisappear:(BOOL)animated {
    [super viewWillDisappear:animated];
    [starIoExtManager disconnect];
}
```

*PrinterExtViewController.m / CashDrawerExtViewController.m / BarcodeReaderExtViewController.m / CombinationExtViewController.m を参照ください。*

## 15.5 port プロパティ

SMPort オブジェクト。\*readonly

### 宣言

@property (readonly, nonatomic) SMPort \*port;

## 15.6 lock プロパティ

port プロパティによる通信権利の排他制御オブジェクト。\*readonly

lock プロパティを使用する事で、アプリは StarloExtManager クラスの port プロパティの使用権を獲得することができます。

StarloExtManager クラスは、connect メソッド成功後に自動的にプリンタ状態の監視を開始します。プリンタ状態の監視処理は、プリンタとの通信に StarloExtManager クラスの port プロパティを使用します。上記の監視中にアプリが port プロパティのメソッドを実行すると、port プロパティは同時にニカ所から利用される事になり、port プロパティのメソッドは正常に動作しません。

アプリが lock プロパティの lock()メソッドを実行すると、StarloExtManager クラスはプリンタ状態の監視を停止し、アプリは StarloExtManager クラスから port プロパティの使用権を獲得します。その結果、アプリは port プロパティのメソッドを正常に実行できるようになります。

アプリが port プロパティのメソッドを使い終わった後は、必ず lock プロパティの unlock()メソッドを実行してください。アプリから StarloExtManager クラスに port プロパティの使用権が戻り、StarloExtManager クラスはプリンタ状態の監視を再開します。

### 宣言

```
@property (readonly, nonatomic) NSRecursiveLock *lock;
```

### Example

```
- (IBAction)touchUpInsidePrintButton:(id)sender {  
    ...  
    [_starloExtManager.lock lock];  
    [Communication sendCommands:commands port:[_starloExtManager port]];  
    [_starloExtManager.lock unlock];  
    ...  
}
```

PrinterExtViewController.m / CashDrawerExtViewController.m / CombinationExtViewController.m を参照ください。



## 15.7 delegate プロパティ

StarloExtManager のデリゲートオブジェクト。

### 宣言

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

### Example

```
- (void)viewDidLoad {
    [super viewDidLoad];
    // Do any additional setup after loading the view.

    _starloExtManager = [[StarloExtManager alloc] initWithType:StarloExtManagerTypeOnlyBarcodeReader
        portName:[AppDelegate getPortName]
        portSettings:[AppDelegate getPortSettings]
        ioTimeoutMillis:10000]; // 10000mS!!!

    _starloExtManager.delegate = self;
}
```

*PrinterExtViewController.m / CashDrawerExtViewController.m / BarcodeReaderExtViewController.m / CombinationExtViewController.m* を参照ください。

## 15.8 printerOnlineStatus プロパティ

プリンタオンラインステータス。\* *readonly*

### 宣言

@property (readonly, nonatomic) StarloExtManagerPrinterStatus printerOnlineStatus;

### 値

説明	型
プリンタステータス <ul style="list-style-type: none"> <li>StarloExtManagerPrinterStatusInvalid ... 不定</li> <li>StarloExtManagerPrinterStatusImpossible ... プリンタ使用不能</li> <li>StarloExtManagerPrinterStatusOnline ... プリンタオンライン検出</li> <li>StarloExtManagerPrinterStatusOffline ... プリンタオフライン検出</li> </ul>	StarloExtManagerPrinterStatus

## 15.9 printerPaperStatus プロパティ

プリンタ用紙ステータス。\* *readonly*

### 宣言

@property (readonly, nonatomic) StarloExtManagerPrinterPaperStatus printerPaperStatus;

### 値

説明	型
プリンタ用紙ステータス <ul style="list-style-type: none"> <li>StarloExtManagerPrinterPaperStatusInvalid ... 不定</li> <li>StarloExtManagerPrinterPaperStatusImpossible ... 機器使用不能</li> <li>StarloExtManagerPrinterPaperStatusReady ... プリンタ用紙レディ検出</li> <li>StarloExtManagerPrinterPaperStatusNearEmpty ... プリンタ用紙ニアエンド検出</li> <li>StarloExtManagerPrinterPaperStatusEmpty ... プリンタ用紙エンpty検出</li> </ul>	StarloExtManagerPrinterPaperStatus

## 15.10 printerCoverStatus プロパティ

プリンタカバーステータス。\* *readonly*

### 宣言

@property (readonly, nonatomic) StarloExtManagerPrinterCoverStatus printerCoverStatus;

### 値

説明	型
プリンタカバーステータス <ul style="list-style-type: none"> <li>StarloExtManagerPrinterCoverStatusInvalid ... 不定</li> <li>StarloExtManagerPrinterCoverStatusImpossible ... 機器使用不能</li> <li>StarloExtManagerPrinterCoverStatusOpen ... プリンタカバーオープン検出</li> <li>StarloExtManagerPrinterCoverStatusClose ... プリンタカバークローズ検出</li> </ul>	StarloExtManagerPrinterCoverStatus

## 15.11 cashDrawerStatus プロパティ

キャッシュドロアーステータス。\* *readonly*

### 宣言

@property (readonly, nonatomic) StarloExtManagerCashDrawerStatus cashDrawerStatus;

### 値

説明	型
キャッシュドロアーステータス <ul style="list-style-type: none"> <li>StarloExtManagerCashDrawerStatusInvalid ... 不定</li> <li>StarloExtManagerCashDrawerStatusImpossible ... 機器使用不能</li> <li>StarloExtManagerCashDrawerStatusOpen ... キャッシュドロアオープン検出</li> <li>StarloExtManagerCashDrawerStatusClose ... キャッシュドロアークローズ検出</li> </ul>	StarloExtManagerCashDrawerStatus

## 15.12 barcodeReaderStatus プロパティ

バーコードリーダーステータス。\* *readonly*

### 宣言

@property (readonly, nonatomic) StarloExtManagerBarcodeReaderStatus barcodeReaderStatus;

### 値

説明	型
バーコードリーダーステータス <ul style="list-style-type: none"> <li>StarloExtManagerBarcodeReaderStatusInvalid ... 不定</li> <li>StarloExtManagerBarcodeReaderStatusImpossible ... 機器使用不能</li> <li>StarloExtManagerBarcodeReaderStatusConnect ... バーコードリーダー接続検出</li> <li>StarloExtManagerBarcodeReaderStatusDisconnect ... バーコードリーダー切断検出</li> </ul>	StarloExtManagerBarcodeReaderStatus

## 15.13 cashDrawerOpenActiveHigh プロパティ

キャッシュドロアー開閉センサのモード。

### 宣言

@property (nonatomic) BOOL cashDrawerOpenActiveHigh;

### 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"> <li>YES ... High アクティブ</li> <li>NO ... Low アクティブ</li> </ul>	BOOL

## 15.14 StarloExtManagerType コンスタント

マネージャタイプ定数。

### 宣言

```
typedef NS_ENUM(NSUInteger, StarloExtManagerType) {
    StarloExtManagerTypeStandard = 0,
    StarloExtManagerTypeWithBarcodeReader,
    StarloExtManagerTypeOnlyBarcodeReader,
};
```

### 定数

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

## 15.15 StarloExtManagerPrinterStatus コンスタント

プリンタステータス定数。

### 宣言

```
typedef NS_ENUM(NSUInteger, StarloExtManagerPrinterStatus) {
    StarloExtManagerPrinterStatusInvalid = 0,
    StarloExtManagerPrinterStatusImpossible,
    StarloExtManagerPrinterStatusOnline,
    StarloExtManagerPrinterStatusOffline
};
```

### 定数

名称	説明
StarloExtManagerPrinterStatusInvalid	不定
StarloExtManagerPrinterStatusImpossible	プリンタ使用不能
StarloExtManagerPrinterStatusPrinterOnline	プリンタオンライン検出
StarloExtManagerPrinterStatusPrinterOffline	プリンタオフライン検出

## 15.16 StarloExtManagerPrinterPaperStatus コンスタント

プリンタ用紙ステータス定数。

### 宣言

```
typedef NS_ENUM(NSInteger, StarloExtManagerPrinterPaperStatus) {
    StarloExtManagerPrinterPaperStatusInvalid = 0,
    StarloExtManagerPrinterPaperStatusImpossible,
    StarloExtManagerPrinterPaperStatusReady,
    StarloExtManagerPrinterPaperStatusNearEmpty,
    StarloExtManagerPrinterPaperStatusEmpty
};
```

### 定数

名称	説明
StarloExtManagerPrinterPaperStatusInvalid	不定
StarloExtManagerPrinterPaperStatusImpossible	プリンタ使用不能
StarloExtManagerPrinterPaperStatusReady	プリンタ用紙レディ検出
StarloExtManagerPrinterPaperStatusNearEmpty	プリンタ用紙ニアエンド検出
StarloExtManagerPrinterPaperStatusEmpty	プリンタ用紙エンプティ検出

## 15.17 StarloExtManagerPrinterCoverStatus コンスタント

プリンタカバーステータス定数。

### 宣言

```
typedef NS_ENUM(NSInteger, StarloExtManagerPrinterCoverStatus) {
    StarloExtManagerPrinterCoverStatusInvalid = 0,
    StarloExtManagerPrinterCoverStatusImpossible,
    StarloExtManagerPrinterCoverStatusOpen,
    StarloExtManagerPrinterCoverStatusClose
};
```

### 定数

名称	説明
StarloExtManagerPrinterCoverStatusInvalid	不定
StarloExtManagerPrinterCoverStatusImpossible	プリンタ使用不能
StarloExtManagerPrinterCoverStatusOpen	プリンタカバーオープン検出
StarloExtManagerPrinterCoverStatusClose	プリンタカバークローズ検出

## 15.18 StarloExtManagerCashDrawerStatus コンスタント

キャッシュドロアステータス定数。

### 宣言

```
typedef NS_ENUM(NSInteger, StarloExtManagerCashDrawerStatus) {
    StarloExtManagerCashDrawerStatusInvalid = 0,
    StarloExtManagerCashDrawerStatusImpossible,
    StarloExtManagerCashDrawerStatusOpen,
    StarloExtManagerCashDrawerStatusClose
};
```

### 定数

名称	説明
StarloExtManagerCashDrawerStatusInvalid	不定
StarloExtManagerCashDrawerStatusImpossible	キャッシュドロア使用不能
StarloExtManagerCashDrawerStatusOpen	キャッシュドロアオープン検出
StarloExtManagerCashDrawerStatusClose	キャッシュドロアクローズ検出

## 15.19 StarloExtManagerBarcodeReaderStatus コンスタント

バーコードリーダーステータス定数。

### 宣言

```
typedef NS_ENUM(NSInteger, StarloExtManagerBarcodeReaderStatus) {
    StarloExtManagerBarcodeReaderStatusInvalid = 0,
    StarloExtManagerBarcodeReaderStatusImpossible,
    StarloExtManagerBarcodeReaderStatusConnect,
    StarloExtManagerBarcodeReaderStatusDisconnect
};
```

### 定数

名称	説明
StarloExtManagerBarcodeReaderStatusInvalid	不定
StarloExtManagerBarcodeReaderStatusImpossible	バーコードリーダー使用不能
StarloExtManagerBarcodeReaderStatusConnect	バーコードリーダー接続検出
StarloExtManagerBarcodeReaderStatusDisconnect	バーコードリーダー切断検出

## 16 StarIoExtManagerDelegate プロトコル (StarIO\_Extension.framework)

### メソッド

名称	説明
didConnectPort	接続成功を通知。 * <i>optional</i>
didFailToConnectPort	接続失敗を通知。 * <i>optional</i>
didPrinterImpossible	プリンタ使用不能を通知。 * <i>optional</i>
didPrinterOnline	プリンタオンラインを通知。 * <i>optional</i>
didPrinterOffline	プリンタオフラインを通知。 * <i>optional</i>
didPrinterPaperReady	プリンタ用紙レディを通知。 * <i>optional</i>
didPrinterPaperNearEmpty	プリンタ用紙ニアエンプティを通知。 * <i>optional</i>
didPrinterPaperEmpty	プリンタ用紙エンプティを通知。 * <i>optional</i>
didPrinterCoverOpen	プリンタカバーオープンを通知。 * <i>optional</i>
didPrinterCoverClose	プリンタカバークローズを通知。 * <i>optional</i>
didCashDrawerOpen	キャッシュドロアーオープンを通知。 * <i>optional</i>
didCashDrawerClose	キャッシュドロアークローズを通知。 * <i>optional</i>
didBarcodeReaderImpossible	バーコードリーダー使用不能を通知。 * <i>optional</i>
didBarcodeReaderConnect	バーコードリーダー接続を通知。 * <i>optional</i>
didBarcodeReaderDisconnect	バーコードリーダー切断を通知。 * <i>optional</i>
didBarcodeDataReceive	バーコードデータ受信を通知。 * <i>optional</i>
didAccessoryConnectSuccess	Bluetooth アクセサリー接続成功を通知。 * <i>optional</i>
didAccessoryConnectFailure	Bluetooth アクセサリー接続失敗を通知。 * <i>optional</i>
didAccessoryDisconnect	Bluetooth アクセサリー切断を通知。 * <i>optional</i>
didStatusUpdate	ステータスアップデートを通知。 * <i>optional</i>



## 16.1 didConnectPort メソッド

接続成功を通知。\* *optional*

### 宣言

```
- (void)manager:(nonnull StarloExtManager *)manager
didConnectPort:(nonnull NSString *)portName;
```

### 引数

名称	説明	型
manager	呼び出し元の StarloExtManager オブジェクト	StarloExtManager *
portName	プリンタポート名	NSString *

### 戻り値

説明	型
-	-

### Example

```
- (void)manager:(StarloExtManager *)manager didConnectPort:(NSString *)portName {
    dispatch_async(GlobalQueueManager.sharedManager.serialQueue, ^{
        dispatch_async(dispatch_get_main_queue(), ^{
            if ((self.didAppear == NO) && (self.starloExtManager.port != nil)) {
                [self.printButton sendActionsForControlEvents:UIControlEventTouchUpInside];
            } else {
                self.blind = NO;
            }

            self.didAppear = YES;
        });
    });
}
```

*PrinterExtWithConnectAsyncViewController.m* を参照ください。

## 16.2 didFailToConnectPort メソッド

接続失敗を通知。\* optional

### 宣言

```
- (void)manager:(nonnull StarloExtManager *)manager
didFailToConnectPort:(nonnull NSString *)portName
error:(nullable NSError *)error;
```

### 引数

名称	説明	型
manager	呼び出し元の StarloExtManager オブジェクト	StarloExtManager *
portName	プリンタポート名	NSString *
error	接続失敗のエラー情報 code プロパティにエラーコードがセットされています。	NSError *

### エラーコード

値	説明
SMResultCodeFailed	何らかのエラーが発生した
SMResultCodeInUse	プリンタから接続拒否された (他ホストが接続中など)
SMResultCodeInProcess	既に接続処理を実行中

### 戻り値

説明	型
-	-

### Example

```
- (void)manager:(StarloExtManager *)manager didFailToConnectPort:(NSString *)portName error:(NSError *)error
{
    if (error != nil) {
        [self showSimpleAlertWithTitle:@"Fail to Open Port."
        message:nil
        buttonTitle:@"OK"
        buttonStyle:UIAlertActionStyleCancel
        completion:^(UIAlertController *alertController) {
            self.commentLabel.text = [NSString stringWithFormat:@"%s@\n"
            "\n"
            "Check the device. (Power and Bluetooth pairing)\n"
            "Then touch up the Refresh button.\n", error.localizedDescription];

            self.commentLabel.textColor = UIColor.redColor;

            [self beginAnimationCommantLabel];

            self.blind = NO;
        }];
    }

    self.didAppear = YES;
}
```

PrinterExtWithConnectAsyncViewController.m を参照ください。

## 16.3 didPrinterImpossible メソッド

プリンタ使用不能を通知。\* *optional*

### 宣言

```
- (void)didPrinterImpossible;
```

```
- (void)didPrinterImpossible:(StarloExtManager *)manager;
```

### 引数

名称	説明	型
manager	呼び出し元の StarloExtManager オブジェクト	StarloExtManager

### 戻り値

説明	型
-	-

### Example

```
- (void)didPrinterImpossible:(StarloExtManager *)manager {
    _commentLabel.text = @"Printer Impossible.";
    _commentLabel.textColor = [UIColor redColor];
}
```

*PrinterExtViewController.m / CashDrawerExtViewController.m / CombinationExtViewController.m* を参照ください。

## 16.4 didPrinterOnline メソッド

プリンタオンラインを通知。\* *optional*

### 宣言

- (void)didPrinterOnline;

- (void)didPrinterOnline:(StarloExtManager \*)manager;

### 引数

名称	説明	型
manager	呼び出し元の StarloExtManager オブジェクト	StarloExtManager

### 戻り値

説明	型
-	-

### Example

```
- (void)didPrinterOnline:(StarloExtManager *)manager {
    _commentLabel.text = @"Printer Online.";
    _commentLabel.textColor = [UIColor blueColor];
}
```

PrinterExtViewController.m / CombinationExtViewController.m を参照ください。

## 16.5 didPrinterOffline メソッド

プリンタオフラインを通知。\* *optional*

### 宣言

- (void)didPrinterOffline;

- (void)didPrinterOffline:(StarloExtManager \*)manager;

### 引数

名称	説明	型
manager	呼び出し元の StarloExtManager オブジェクト	StarloExtManager

### 戻り値

説明	型
-	-

### Example

```
- (void)didPrinterOffline:(StarloExtManager *)manager {
    _commentLabel.text = @"Printer Offline.";
    _commentLabel.textColor = [UIColor redColor];
}
```

PrinterExtViewController.m / CombinationExtViewController.m を参照ください。

## 16.6 didPrinterPaperReady メソッド

プリンタ用紙レディを通知。\* optional

### 宣言

- (void)didPrinterPaperReady;
- (void)didPrinterPaperReady:(StarloExtManager \*)manager;

### 引数

名称	説明	型
manager	呼び出し元の StarloExtManager オブジェクト	StarloExtManager

### 戻り値

説明	型
-	-

### Example

```
- (void)didPrinterPaperReady:(StarloExtManager *)manager {
    _commentLabel.text = @"Printer Paper Ready.";
    _commentLabel.textColor = [UIColor blueColor];
}
```

PrinterExtViewController.m / CombinationExtViewController.m を参照ください。

## 16.7 didPrinterPaperNearEmpty メソッド

プリンタ用紙ニアエンptyを通知。\* optional

### 宣言

- (void)didPrinterPaperNearEmpty;
- (void)didPrinterPaperNearEmpty:(StarloExtManager \*)manager;

### 引数

名称	説明	型
manager	呼び出し元の StarloExtManager オブジェクト	StarloExtManager

### 戻り値

説明	型
-	-

### Example

```
- (void)didPrinterPaperNearEmpty:(StarloExtManager *)manager {
    _commentLabel.text = @"Printer Paper Near Empty.";
    _commentLabel.textColor = [UIColor orangeColor];
}
```

PrinterExtViewController.m / CombinationExtViewController.m を参照ください。

## 16.8 didPrinterPaperEmpty メソッド

プリンタ用紙エンプティを通知。\* optional

### 宣言

```
- (void)didPrinterPaperEmpty;

- (void)didPrinterPaperEmpty:(StarloExtManager *)manager;
```

### 引数

名称	説明	型
manager	呼び出し元の StarloExtManager オブジェクト	StarloExtManager

### 戻り値

説明	型
-	-

### Example

```
- (void)didPrinterPaperEmpty:(StarloExtManager *)manager {
    _commentLabel.text = @"Printer Paper Empty.";
    _commentLabel.textColor = [UIColor redColor];
}
```

PrinterExtViewController.m / CombinationExtViewController.m を参照ください。

## 16.9 didPrinterCoverOpen メソッド

プリンタカバーオープンを通知。\* optional

### 宣言

```
- (void)didPrinterCoverOpen;

- (void)didPrinterCoverOpen:(StarloExtManager *)manager;
```

### 引数

名称	説明	型
manager	呼び出し元の StarloExtManager オブジェクト	StarloExtManager

### 戻り値

説明	型
-	-

### Example

```
- (void)didPrinterCoverOpen:(StarloExtManager *)manager {
    _commentLabel.text = @"Printer Cover Open.";
    _commentLabel.textColor = [UIColor redColor];
}
```

PrinterExtViewController.m / CombinationExtViewController.m を参照ください。

## 16.10 didPrinterCoverClose メソッド

プリンタカバークローズを通知。\* optional

### 宣言

```
- (void)didPrinterCoverClose;

- (void)didPrinterCoverClose:(StarloExtManager *)manager;
```

### 引数

名称	説明	型
manager	呼び出し元の StarloExtManager オブジェクト	StarloExtManager

### 戻り値

説明	型
-	-

### Example

```
- (void)didPrinterCoverClose:(StarloExtManager *)manager {
    _commentLabel.text = @"Printer Cover Close.";
    _commentLabel.textColor = [UIColor blueColor];
}
```

PrinterExtViewController.m / CombinationExtViewController.m を参照ください。

## 16.11 didCashDrawerOpen メソッド

キャッシュドロアーオープンを通知。\* optional

### 宣言

```
- (void)didCashDrawerOpen;

- (void)didCashDrawerOpen:(StarloExtManager *)manager;
```

### 引数

名称	説明	型
manager	呼び出し元の StarloExtManager オブジェクト	StarloExtManager

### 戻り値

説明	型
-	-

### Example

```
- (void)didCashDrawerOpen:(StarloExtManager *)manager {
    _commentLabel.text = @"Cash Drawer Open.";
    // _commentLabel.textColor = [UIColor redColor];
    _commentLabel.textColor = [UIColor magentaColor];
}
```

CashDrawerExtViewController.m / CombinationExtViewController.m を参照ください。

## 16.12 didCashDrawerClose メソッド

キャッシュドロアークローズを通知。\* optional

### 宣言

```
- (void)didCashDrawerClose;

- (void)didCashDrawerClose:(StarloExtManager *)manager;
```

### 引数

名称	説明	型
manager	呼び出し元の StarloExtManager オブジェクト	StarloExtManager

### 戻り値

説明	型
-	-

### Example

```
- (void)didCashDrawerClose:(StarloExtManager *)manager {
    _commentLabel.text = @"Cash Drawer Close.";
    _commentLabel.textColor = [UIColor blueColor];
}
```

CashDrawerExtViewController.m / CombinationExtViewController.m を参照ください。

## 16.13 didBarcodeReaderImpossible メソッド

バーコードリーダー使用不能を通知。\* optional

### 宣言

```
- (void)didBarcodeReaderImpossible;

- (void)didBarcodeReaderImpossible:(StarloExtManager *)manager;
```

### 引数

名称	説明	型
manager	呼び出し元の StarloExtManager オブジェクト	StarloExtManager

### 戻り値

説明	型
-	-

### Example

```
- (void)didBarcodeReaderImpossible:(StarloExtManager *)manager {
    _commentLabel.text = @"Barcode Reader Impossible.";
    _commentLabel.textColor = [UIColor redColor];
}
```

BarcodeReaderExtViewController.m / CombinationExtViewController.m を参照ください。



## 16.14 didBarcodeReaderConnect メソッド

バーコードリーダー接続を通知。\* *optional*

### 宣言

- (void)didBarcodeReaderConnect;

- (void)didBarcodeReaderConnect:(StarloExtManager \*)manager;

### 引数

名称	説明	型
manager	呼び出し元の StarloExtManager オブジェクト	StarloExtManager

### 戻り値

説明	型
-	-

### Example

```
- (void)didBarcodeReaderConnect:(StarloExtManager *)manager {
    _commentLabel.text = @"Barcode Reader Connect.";
    _commentLabel.textColor = [UIColor blueColor];
}
```

BarcodeReaderExtViewController.m / CombinationExtViewController.m を参照ください。

## 16.15 didBarcodeReaderDisconnect メソッド

バーコードリーダー切断を通知。\* *optional*

### 宣言

- (void)didBarcodeReaderDisconnect;

- (void)didBarcodeReaderDisconnect:(StarloExtManager \*)manager;

### 引数

名称	説明	型
manager	呼び出し元の StarloExtManager オブジェクト	StarloExtManager

### 戻り値

説明	型
-	-

### Example

```
- (void)didBarcodeReaderDisconnect:(StarloExtManager *)manager {
    _commentLabel.text = @"Barcode Reader Disconnect.";
    _commentLabel.textColor = [UIColor redColor];
}
```

BarcodeReaderExtViewController.m / CombinationExtViewController.m を参照ください。

## 16.16 didBarcodeDataReceive メソッド

バーコードデータ受信を通知。\* optional

### 宣言

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

```
- (void)didBarcodeDataReceive:(StarloExtManager *)manager data:(NSData *)data;
```

### 引数

名称	説明	型
data	受信したバーコードデータ	NSData
manager	呼び出し元の StarloExtManager オブジェクト	StarloExtManager

### 戻り値

説明	型
-	-

### Example

```
- (void)didBarcodeDataReceive:(StarloExtManager *)manager data:(NSData *)data {
    NSMutableString *text = [NSMutableString stringWithString:@""];

    const uint8_t *p = [data bytes];

    for (int i = 0; i < data.length; i++) {
        uint8_t ch = *(p + i);

        if (ch >= 0x20 && ch <= 0x7f) {
            [text appendFormat:@"%c", (char) ch];
        }
        else if (ch == 0x0d) {
            ...

            text = [NSMutableString stringWithString:@""];
        }
    }
}
```

BarcodeReaderExtViewController.m / CombinationExtViewController.m を参照ください。

## 16.17 didAccessoryConnectSuccess メソッド

Bluetooth アクセサリー接続成功を通知。\* optional

### 宣言

- (void)didAccessoryConnectSuccess;

- (void)didAccessoryConnectSuccess:(StarloExtManager \*)manager;

### 引数

名称	説明	型
manager	呼び出し元の StarloExtManager オブジェクト	StarloExtManager

### 戻り値

説明	型
-	-

### Example

```
- (void)didAccessoryConnectSuccess:(StarloExtManager *)manager {
    _commentLabel.text = @"Accessory Connect Success.";
    _commentLabel.textColor = [UIColor blueColor];
}
```

PrinterExtViewController.m / CashDrawerExtViewController.m / BarcodeReaderExtViewController.m / CombinationExtViewController.m を参照ください。

## 16.18 didAccessoryConnectFailure メソッド

Bluetooth アクセサリー接続失敗を通知。\* optional

### 宣言

- (void)didAccessoryConnectFailure;

- (void)didAccessoryConnectFailure:(StarloExtManager \*)manager;

### 引数

名称	説明	型
manager	呼び出し元の StarloExtManager オブジェクト	StarloExtManager

### 戻り値

説明	型
-	-

### Example

```
- (void)didAccessoryConnectFailure:(StarloExtManager *)manager {
    _commentLabel.text = @"Accessory Connect Failure.";
    _commentLabel.textColor = [UIColor redColor];
}
```

PrinterExtViewController.m / CashDrawerExtViewController.m / BarcodeReaderExtViewController.m / CombinationExtViewController.m を参照ください。

## 16.19 didAccessoryDisconnect メソッド

Bluetooth アクセサリー切断を通知。\* *optional*

### 宣言

```
- (void)didAccessoryDisconnect;

- (void)didAccessoryDisconnect:(StarloExtManager *)manager;
```

### 引数

名称	説明	型
manager	呼び出し元の StarloExtManager オブジェクト	StarloExtManager

### 戻り値

説明	型
-	-

### Example

```
- (void)didAccessoryDisconnect:(StarloExtManager *)manager {
    _commentLabel.text = @"Accessory Disconnect.";
    _commentLabel.textColor = [UIColor redColor];
}
```

*PrinterExtViewController.m / CashDrawerExtViewController.m / BarcodeReaderExtViewController.m / CombinationExtViewController.m* を参照ください。

## 16.20 didStatusUpdate メソッド

ステータスアップデートを通知。\* *optional*

### 宣言

```
- (void)didStatusUpdate:(NSString *)status;
```

```
- (void)didStatusUpdate:(StarloExtManager *)manager status:(NSString *)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"> <li>ステータス変化時</li> <li>ステータス未変化時、5 分毎</li> </ul> <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>	NSString *
manager	呼び出し元の StarloExtManager オブジェクト	StarloExtManager

### 戻り値

説明	型
-	-

### Example

```
- (void)didStatusUpdate:(StarloExtManager *)manager status:(NSString *)status {
    _commentLabel.text = status;

    _commentLabel.textColor = [UIColor greenColor];
}
```

AllReceiptsExtViewController.m を参照ください。

## 17 SMBluetoothManagerFactory クラス (StarIO\_Extension.framework)

プリンタのエミュレーションに対して適切な SMBluetoothManager 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	SMBluetoothManager クラスのオブジェクトを取得します。

### 17.1 getManager メソッド

引数で渡すエミュレーション用の SMBluetoothManager クラスのオブジェクトを取得します。

#### 宣言

```
-(SMBluetoothManager *)getManager:(NSString *)portName emulation:(StarIoExtEmulation)emulation;
```

#### 引数

名称	説明	型
portName	プリンタポート名	NSString *
emulation	エミュレーション	StarIoExtEmulation

#### 戻り値

説明	型
SMBluetoothManager オブジェクト	SMBluetoothManager *

Bluetooth 設定変更の手順は [SMBluetoothManager 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 StarIoExtEmulation コンスタント定数は [StarIoExtEmulation コンスタント](#) を参照ください。

## 18 SMCloudServices クラス (SMCloudServices.framework)

### 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。

### 18.1 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

```
+ (void)showRegistrationView:(void (^)(BOOL isRegistered))completion;
```

### 引数

名称	説明	型
completion	ビューが閉じられた際、アプリケーションの登録状況を以下の引数によって捉えることができます。 -isRegistered- • YES ... 登録されている • NO ... 登録されていない	void (^)(BOOL isRegistered)

### 戻り値

説明	型
-	-

### Example

```
- (void)tableView:(UITableView *)tableView didSelectRowAtIndexPath:(NSIndexPath *)indexPath {
    ...

    [SMCloudServices showRegistrationView:^(BOOL isRegistration) {
        [_tableView reloadData];
    }];

    ...
}
```

AllReceiptsViewController.m を参照ください。

## 18.2 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

+ (BOOL)isRegistered;

### 引数

名称	説明	型
-	-	-

### 戻り値

説明	型
リザルト <ul style="list-style-type: none"> <li>YES ... 登録されている</li> <li>NO ... 登録されていない</li> </ul>	BOOL

### Example

```
- (UITableViewCell *)tableView:(UITableView *)tableView cellForRowAtIndexPath:(NSIndexPath *)indexPath {
    ...

    BOOL userInteractionEnabled = YES;

    if ([SMCloudServices isRegistered] == NO) {
        userInteractionEnabled = NO;
    }

    ...

    return cell;
}
```

*AllReceiptsViewController.m* を参照ください。



## 19 SMCSAllReceipts クラス (SMCloudServices.framework)

### 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。

## 19.1 モデル : SMCSAllReceipts 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	SM-T400i	BSC10	SM-S210i StarPRNT	SM-S220i StarPRNT	SM-S230i StarPRNT	SM-T300i StarPRNT	SM-T400i StarPRNT	SM-L200	SM-L300	SP700
Upload	uploadBitmap	✓	✓	✓	✓	✓	✓	✓	✓	-	-	-	-	-	✓	✓	✓	✓	✓	✓	✓	✓	-
	uploadData	✓	✓	✓	✓	-	✓	✓	✓	-	-	-	-	-	✓	✓	✓	✓	✓	✓	✓	✓	-
Update	updateStatus	✓	✓	✓	✓	✓	✓	✓	✓	-	-	-	-	-	✓	✓	✓	✓	✓	✓	✓	✓	-
Generate	generateAllReceipts	✓	✓	✓	✓	✓	✓	✓	✓	-	-	-	-	-	✓	✓	✓	✓	✓	✓	✓	✓	-

- : 対象外

## 19.2 uploadBitmap メソッド

ビットマップ(UImage 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

```
+ (NSString *)uploadBitmap:(UImage *)image completion:(void (^)(NSInteger statusCode, NSError *error))completion;
```

### 引数

名称	説明	型
image	アップロードするビットマップ	UImage
completion	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	void (^)(NSInteger statusCode, NSError *error)

### 戻り値

説明	型
アップロード URL	NSString *

### Example

```
+ (NSData *)createData:(StarIoExtEmulation)emulation image:(UImage *)image {
    NSString *urlString = [SMCSAllReceipts uploadBitmap:image completion:nil];

    ISCBBuilder *builder = [StarIoExt createCommandBuilder:emulation];

    [builder beginDocument];

    [builder appendBitmap:image diffusion:NO];

    NSData *data = [SMCSAllReceipts generateAllReceipts:urlString emulation:emulation info:YES qrCode:YES];

    [builder appendRawData:data];

    [builder appendCutPaper:SCBCutPaperActionPartialCutWithFeed];

    [builder endDocument];

    return [builder.commands copy];
}
```

*AllReceiptsFunctions.m* を参照ください。

## 19.3 uploadData メソッド

データ(NSData 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

```
+ (NSString *)uploadData:(NSData *)data emulation:(StarloExtEmulation)emulation
characterCode:(StarloExtCharacterCode)characterCode width:(NSInteger)width completion:(void
(^)(NSInteger statusCode, NSError *error))completion;
```

### 引数

名称	説明	型
data	アップロードするデータ	NSData
emulation	エミュレーション指定 <ul style="list-style-type: none"> <li>StarloExtEmulationStarPRNT ... StarPRNT エミュレーション</li> <li>StarloExtEmulationStarLine ... STAR Line Mode エミュレーション</li> <li>StarloExtEmulationStarGraphic ... STAR Graphic Mode エミュレーション</li> <li>StarloExtEmulationEscPos ... ESC/POS エミュレーション</li> <li>StarloExtEmulationEscPosMobile ... ESC/POS Mobile エミュレーション</li> <li>StarloExtEmulationStarDotImpact ... STAR ドットインパクトエミュレーション</li> </ul>	StarloExtEmulation
characterCode	文字コード指定 <ul style="list-style-type: none"> <li>StarloExtCharacterCodeStandard ... スタンダード</li> <li>StarloExtCharacterCodeJapanese ... 日本語</li> <li>StarloExtCharacterCodeSimplifiedChinese ... 簡体字中国語</li> <li>StarloExtCharacterCodeTraditionalChinese ... 繁体字中国語</li> </ul>	StarloExtCharacterCode
width	印刷領域幅 (ドット数単位)	NSInteger
completion	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"> <li>HTTP ステータスコード</li> </ul> -error- <ul style="list-style-type: none"> <li>アップロードに失敗した際のエラー情報</li> </ul>	void (^)(NSInteger statusCode, NSError *error)

### 戻り値

説明	型
アップロード URL	NSString *

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

### Example

```
+ (NSData *)createData:(StarloExtEmulation)emulation characterCode:(StarloExtCharacterCode)characterCode
data:(NSData *)data width:(NSInteger)width {
    ISCBBuilder *builder = [StarloExt createCommandBuilder:emulation];

    [builder beginDocument];

    [builder appendData:data];

    [builder endDocument];

    NSData *receiptData = [builder.commands copy];

    NSString *urlString = [SMCSAllReceipts uploadData:receiptData emulation:emulation
characterCode:characterCode width:width completion:nil];

    builder = [StarloExt createCommandBuilder:emulation];

    [builder beginDocument];

    [builder appendData:data];

    NSData *allReceiptsData = [SMCSAllReceipts generateAllReceipts:urlString emulation:emulation info:YES
qrCode:YES];

    [builder appendRawData:allReceiptsData];

    [builder appendCutPaper:SCBCutPaperActionPartialCutWithFeed];

    [builder endDocument];

    return [builder.commands copy];
}
```

*AllReceiptsFunctions.m* を参照ください。

## 19.4 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

```
+ (void)updateStatus:(NSString *)status completion:(void (^)(NSInteger statusCode, NSError *error))completion;
```

### 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>※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>	NSString *
completion	<p>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。</p> <p>-statusCode-</p> <ul style="list-style-type: none"> <li>HTTP ステータスコード</li> </ul> <p>-error-</p> <ul style="list-style-type: none"> <li>アップデートに失敗した際のエラー情報</li> </ul>	void (^)(NSInteger statusCode, NSError *error)

### 戻り値

説明	型
-	-

### Example

```
- (void)didStatusUpdate:(StarIoExtManager *)manager status:(NSString *)status {
    [SMCSAllReceipts updateStatus:status completion:^(NSInteger statusCode, NSError *error) {
        ...
    }];
}
```

AllReceiptsExtViewController.m を参照ください。

## 19.5 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

```
+ (NSData *)generateAllReceipts:(NSString *)urlString emulation:(StarloExtEmulation)emulation
info:(BOOL)info qrCode:(BOOL)qrCode;
+ (NSData *)generateAllReceipts:(NSString *)urlString emulation:(StarloExtEmulation)emulation
info:(BOOL)info qrCode:(BOOL)qrCode width:(NSInteger)width;
```

### 引数

名称	説明	型
urlString	アップロード URL	NSString *
emulation	エミュレーション <ul style="list-style-type: none"> <li>StarloExtEmulationStarPRNT ... StarPRNT エミュレーション</li> <li>StarloExtEmulationStarLine ... STAR Line Mode エミュレーション</li> <li>StarloExtEmulationStarGraphic ... STAR Graphic Mode エミュレーション</li> <li>StarloExtEmulationEscPos ... ESC/POS エミュレーション</li> <li><del>StarloExtEmulationEscPosMobile ... ESC/POS Mobile エミュレーション</del></li> <li><del>StarloExtEmulationStarDotImpact ... STAR ドットインパクトエミュレーション</del></li> </ul>	StarloExtEmulation
info	インフォメーションロゴ <ul style="list-style-type: none"> <li>YES ... 印刷データに含める</li> <li>NO ... 印刷データに含めない</li> </ul>	BOOL
qrCode	QR コード <ul style="list-style-type: none"> <li>YES ... 印刷データに含める</li> <li>NO ... 印刷データに含めない</li> </ul>	BOOL
width	印刷領域幅 (ドット数単位)	NSInteger

### 戻り値

説明	型
生成された印刷データ	NSData

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

### Example

```
+ (NSData *)createData:(StarloExtEmulation)emulation image:(UIImage *)image {
    NSString *urlString = [SMCSAllReceipts uploadBitmap:image completion:nil];

    ISCBBuilder *builder = [StarloExt createCommandBuilder:emulation];

    [builder beginDocument];

    [builder appendBitmap:image diffusion:NO];

    NSData *data;

    if (emulation == StarloExtEmulationStarGraphic) {
        data = [SMCSAllReceipts generateAllReceipts:urlString emulation:emulation info:info qrCode:qrCode width:
width]; // Support to centering in Star Graphic.
    }
    else {
        data = [SMCSAllReceipts generateAllReceipts:urlString emulation:emulation info:info qrCode:qrCode]; //
Non support to centering in Star Graphic.
    }

    [builder appendRawData:data];

    [builder appendCutPaper:SCBCutPaperActionPartialCutWithFeed];

    [builder endDocument];

    return [builder.commands copy];
}
```

*AllReceiptsFunctions.m* を参照ください。



## 20 非推奨 API

### 20.1 SMPort クラス

#### 20.1.1 getPort メソッド

このメソッドは非推奨となりました。代わりに [getPort メソッド](#) をご利用ください。

SMPort オブジェクトを生成し、指定されたポートへの接続をオープンします。

#### 宣言

```
+ (SMPort *) getPort: (NSString *) portName : (NSString *) portSettings : (u_int32_t) timeoutMillis;
```

#### 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	NSString *
portSettings	プリンタとの接続設定情報 <ul style="list-style-type: none"> <li>● エミュレーションの指定</li> <li>● データタイムアウト機能の設定</li> </ul>	NSString *
timeoutMillis	内部制御および API のタイムアウト値（ミリ秒指定）	u_int32_t

#### 戻り値

説明	型
SMPort オブジェクト ポートオープンに失敗した場合、既にオープン済みであった場合は、 nil が返されます。	SMPort *

## 20.1.2 searchPrinter メソッド

このメソッドは非推奨となりました。代わりに [searchPrinter メソッド](#) をご利用ください。

iOS デバイスと接続できるプリンタを検索します。

### 宣言

```
+ (NSArray *) searchPrinter;
```

```
+ (NSArray *) searchPrinter: (NSString *) target;
```

### 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 target を指定しない場合、すべてのインターフェイスタイプを検索します。	NSString *
	Bluetooth の場合	
	Bluetooth Low Energy の場合	
	Ethernet / Wireless LAN の場合	
	USB の場合	

### 戻り値

説明	型
Star プリンタの検索結果 戻り値には、PortInfo クラスのインスタンスが含まれます。 取得できる情報については <a href="#">PortInfo クラス</a> を参照ください。	NSArray *

### 20.1.3 writePort メソッド

このメソッドは非推奨となりました。代わりに [writePort メソッド](#) をご利用ください。

プリンタにデータを書き込みます。コマンドや印刷データの送信に使用します。  
印字終了の確認を行うため、このメソッドの前後で beginCheckedBlock/endCheckedBlock を使用してください。

#### 宣言

```
- (u_int32_t) writePort:(u_int8_t const *) writeBuffer :(u_int32_t) offset :(u_int32_t) size;
```

#### 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	u_int8_t const *
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	u_int32_t
size	書き込む byte 数	u_int32_t

#### 戻り値

説明	型
<b>Bluetooth/Ethernet/Wireless LAN/USB インターフェイス</b> 実際に書き込まれたバイト数。 データが全て書き込めなかった時でも、writePort 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、writePort 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。	u_int32_t
<b>Bluetooth Low Energy インターフェイス</b> 成功時には、送信データのサイズが返ります。失敗時は 0 が返ります。	

#### 例外

説明	型
通信エラーが発生したとき	PortException

## 20.1.4 readPort メソッド

このメソッドは非推奨となりました。代わりに [readPort メソッド](#) をご利用ください。

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

```
-(u_int32_t) readPort:(u_int8_t *) readBuffer :(u_int32_t) offSet :(u_int32_t) size;
```

### 引数

名称	説明	型
readBuffer	データが読み込まれる byte 配列	u_int8_t *
offset	readBuffer にデータを書き込み始めるインデックス	u_int32_t
size	読み込む byte 数	u_int32_t

### 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。	u_int32_t

### 例外

説明	型
通信エラーが発生したとき	PortException

## 20.1.5 getParsedStatus メソッド

このメソッドは非推奨となりました。代わりに [getParsedStatus メソッド](#) をご利用ください。

プリンタのステータスを取得します。

### 宣言

```
- (void) getParsedStatus:(void *) starPrinterStatus :(u_int32_t) level;
```

### 引数

名称	説明	型
starPrinterStatus	現在のプリンタのステータスが格納されます。 取得できるステータスの種類は <a href="#">StarPrinterStatus 構造体</a> を参照ください。	void *
level	StarPrinterStatus 構造体のレベル (0,1,2 の指定が可能だが、通常は 2 を指定)	u_int32_t

### 戻り値

説明	型
-	-

### 例外

説明	型
通信エラーが発生したとき	PortException

## 20.1.6 beginCheckedBlock メソッド

このメソッドは非推奨となりました。代わりに [beginCheckedBlock メソッド](#) をご利用ください。

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

```
- (void) beginCheckedBlock:(void *) starPrinterStatus :(u_int32_t) level;
```

### 引数

名称	説明	型
starPrinterStatus	StarPrinterStatus 構造体へのポインタ (StarPrinterStatus, StarPrinterStatus_1, StarPrinterStatus_2 の指定が可能だが、通常は 2 を指定) このメソッドが成功したとき、現在のプリンタのステータスが格納されます。	void *
level	StarPrinterStatus 構造体のレベル (0,1,2 の指定が可能だが、通常は 2 を指定)	u_int32_t

### 戻り値

説明	型
-	-

### 例外

説明	型
通信エラーが発生したとき	PortException

## 20.1.7 endCheckedBlock メソッド

このメソッドは非推奨となりました。代わりに [endCheckedBlock メソッド](#) をご利用ください。

beginCheckedBlock 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。  
 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。

### 宣言

```
- (void) endCheckedBlock:(void *) starPrinterStatus :(u_int32_t) level;
```

### 引数

名称	説明	型
starPrinterStatus	StarPrinterStatus 構造体へのポインタ（StarPrinterStatus, StarPrinterStatus_1, StarPrinterStatus_2 の指定が可能だが、通常は 2 を指定） このメソッドが成功したとき、現在のプリンタのステータスが格納されます。	void *
level	StarPrinterStatus 構造体のレベル（0,1,2 の指定が可能だが、通常は 2 を指定）	u_int32_t

### 戻り値

説明	型
-	-

### 例外

説明	型
通信エラー（プリンタオフライン等でコマンド送信自体が失敗した場合）が発生したとき	PortException
タイムアウト時間内にプリンタからの応答がないとき	

## 20.1.8 disconnect メソッド

このメソッドは非推奨となりました。代わりに [disconnect メソッド](#) をご利用ください。

指定された Bluetooth デバイスへの接続を切断します。  
 コネクションの切断後、Bluetooth デバイスは再び他の iOS 端末から接続することができるようになります。

### 宣言

- (BOOL) disconnect;

### 引数

名称	説明	型
-	-	-

### 戻り値

説明	型
切断成功・失敗 このメソッドは、以下の場合に失敗となります。 <ul style="list-style-type: none"> <li>● getPort で指定したタイムアウト時間内に切断が完了しなかった場合</li> <li>● プリンタが切断機能に対応していない場合（モバイルプリンタ等）</li> </ul> Ethernet デバイスに対しては何も行いません。 Ethernet デバイスに対して実行した場合は常に YES を返します。	BOOL



## 20.1.9 getFirmwareInformation メソッド

このメソッドは非推奨となりました。代わりに [getFirmwareInformation メソッド](#) をご利用ください。

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

-(NSDictionary \*) getFirmwareInformation;

### 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 NSDictionary 戻り値の NSDictionary 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。	NSDictionary *

### 例外

説明	型
取得に失敗したとき	PortException

## 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](http://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](http://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。

