

# Epson ePOS SDK for JavaScript

## ユーザーズマニュアル

---

機能

使い方

API リファレンス

デバイス制御スクリプト

機器仕様

サンプルプログラム

活用ガイド

付録

## ご注意

- 本書の内容の一部または全部を無断で転載、複写、複製、改ざんすることは固くお断りします。
- 本書の内容については、予告なしに変更することがあります。最新の情報はお問い合わせください。
- 本書の内容については、万全を期して作成いたしましたが、万一ご不審な点や誤り、記載もれなど、お気づきの点がありましたらご連絡ください。
- 運用した結果の影響については、上項にかかわらず責任を負いかねますのでご了承ください。
- 本製品がお客様により不適切に使用されたり、本書の内容に従わずに取り扱われたり、またはエプソンおよびエプソン指定の者以外の第三者により修理・変更されたことなどに起因して生じた損害などにつきましては、責任を負いかねますのでご了承ください。
- エプソン純正品およびエプソン品質認定品以外のオプションまたは消耗品を装着してトラブルが発生した場合には、責任を負いかねますのでご了承ください。

## 商標について

EPSON、EXCEED YOUR VISION、ESC/POS は、セイコーエプソン株式会社の登録商標です。

Windows<sup>®</sup>、Internet Explorer<sup>®</sup> は、米国 Microsoft Corporation の米国およびその他の国における商標または登録商標です。

Safari<sup>™</sup> は、米国および他の国々で登録された Apple Inc. の商標です。

Android<sup>™</sup>、Google Chrome<sup>™</sup> は、Google Inc. の商標です。

Mozilla<sup>®</sup>、Firefox<sup>®</sup> は、米国 Mozilla Foundation の米国およびその他の国における商標または登録商標です。

IOS<sup>®</sup> は、Cisco の米国およびその他の国における商標または登録商標です。

QR コードは株式会社デンソーウェーブの登録商標です。

その他の製品名および会社名は、各社の商標または登録商標です。



## ESC/POS<sup>®</sup> コマンドシステム

エプソンは、独自の POS プリンターコマンドシステム、ESC/POS により、業界のイニシアチブをとってきました。ESC/POS は特許取得済みのものを含む数多くの独自のコマンドを持ち、高い拡張性で多才な POS システムの構築を実現します。ほとんどのエプソン POS プリンターとディスプレイに互換性を持つ他、この独自の制御システムにはフレキシビリティもあるため、将来アップグレードが行いやすくなります。その機能と利便性は世界中で評価されています。

# 安全のために

## 記号の意味

本書では以下の記号が使われています。それぞれの記号の意味をよく理解してから製品を取り扱ってください。

	ご使用上、必ずお守りいただきたいことを記載しています。この表示を無視して誤った取り扱いをすると、製品の故障や動作不良の原因になる可能性があります。
	補足説明や知っておいていただきたいことを記載しています。

## 使用制限

本製品を航空機・列車・船舶・自動車などの運行に直接関わる装置・防災防犯装置・各種安全装置など機能・精度などにおいて高い信頼性・安全性が必要とされる用途に使用される場合は、これらのシステム全体の信頼性および安全維持のためにフェールセーフ設計や冗長設計の措置を講じるなど、システム全体の安全設計にご配慮いただいた上で弊社製品をご使用いただくようお願いいたします。

本製品は、航空宇宙機器、幹線通信機器、原子力制御機器、医療機器など、きわめて高い信頼性・安全性が必要とされる用途への使用を意図しておりませんので、これらの用途には本製品の適合性をお客様において十分ご確認の上、ご判断ください。

## 本書について

### 本書の目的

Epson ePOS SDK for JavaScript を利用したアプリケーション開発に必要な情報を提供します。

### 本書の構成

本書は次のように構成されています。

第 1 章	<a href="#">機能</a>
第 2 章	<a href="#">使い方</a>
第 3 章	<a href="#">API リファレンス</a>
第 4 章	<a href="#">デバイス制御スクリプト</a>
第 5 章	<a href="#">機器仕様</a>
第 6 章	<a href="#">サンプルプログラム</a>
第 7 章	<a href="#">活用ガイド</a>
付録	<a href="#">オープンソースライセンス</a>

# もくじ

■ 安全のために .....	3
記号の意味 .....	3
■ 使用制限 .....	3
■ 本書について .....	3
本書の目的 .....	3
本書の構成 .....	3
■ もくじ .....	4

---

## 機能 ..... 11

■ Epson ePOS SDK for JavaScript でのアプリケーション開発 .....	11
TM プリンターモデル .....	11
TM インテリジェントプリンターモデル .....	12
■ アプリケーションソフトの動作環境 .....	13
対象プリンター .....	13
■ ダウンロードコンテンツ .....	14
パッケージ .....	14
■ Web コンテンツ .....	14
■ 制限事項 .....	14

---

## 使い方 ..... 15

■ プリンターとの接続 .....	15
■ Epson ePOS SDK for JavaScript の組み込み方法 .....	15
■ プログラミングガイド .....	15
プログラミングフロー .....	15
設定用の命令バッファの有効範囲 .....	22
エラーコード .....	22

---

## API リファレンス ..... 23

■ ePOS SDK API 一覧 .....	23
ePOSDevice オブジェクト .....	23
デバイスオブジェクト共通 .....	24
CashChanger オブジェクト .....	24
Display オブジェクト .....	24
Keyboard オブジェクト .....	25
MSR オブジェクト .....	25
Printer オブジェクト .....	26
HybridPrinter オブジェクト .....	29
Scanner オブジェクト .....	31
SimpleSerial オブジェクト .....	31

DeviceHubTerminal オブジェクト .....	32
CommBoxManager オブジェクト .....	32
CommBox オブジェクト .....	32
ePosDeviceConfiguration オブジェクト .....	32
<b>■ ePOSDevice オブジェクト .....</b>	<b>33</b>
Constructor .....	33
connect メソッド .....	34
disconnect メソッド .....	35
isConnected メソッド .....	36
createDevice メソッド .....	37
deleteDevice メソッド .....	39
getAdmin メソッド .....	40
getLocation メソッド .....	41
sendOfscXml メソッド .....	42
getCommBoxManager メソッド .....	43
onreconnecting イベント .....	44
onreconnect イベント .....	45
ondisconnect イベント .....	46
<b>■ デバイスオブジェクト共通 .....</b>	<b>47</b>
callEvent メソッド .....	47
<b>■ CashChanger オブジェクト .....</b>	<b>48</b>
setConfig メソッド .....	48
readCashCount メソッド .....	49
beginDeposit メソッド .....	50
pauseDeposit メソッド .....	51
restartDeposit メソッド .....	52
endDeposit メソッド .....	53
dispenseChange メソッド .....	54
dispenseCash メソッド .....	55
collectCash メソッド .....	56
openDrawer メソッド .....	57
sendCommand メソッド .....	58
onconfigchange イベント .....	59
oncashcount イベント .....	60
ondeposit イベント .....	61
ondispense イベント .....	63
oncollect イベント .....	64
oncommandreply イベント .....	65
onstatuschange イベント .....	66
<b>■ Display オブジェクト .....</b>	<b>68</b>
createWindow メソッド .....	68
destroyWindow メソッド .....	70
setCurrentWindow メソッド .....	71
clearWindow メソッド .....	72
setCursorPosition メソッド .....	73
moveCursorPosition メソッド .....	74
setCursorType メソッド .....	75
addText メソッド .....	76
addReverseText メソッド .....	77
addMarquee メソッド .....	78
setBlink メソッド .....	80
setBrightness メソッド .....	81
showClock メソッド .....	82
addCommand メソッド .....	83

send メソッド .....	84
reset メソッド .....	85
onreceive イベント .....	86
■ Keyboard オブジェクト .....	87
setPrefix メソッド .....	87
onkeypress イベント .....	88
onstring イベント .....	89
setMSRPrefix メソッド .....	90
ondata イベント .....	91
■ MSR オブジェクト .....	92
ondata イベント .....	92
■ Printer オブジェクト .....	93
addTextAlign メソッド .....	93
addTextLineSpace メソッド .....	94
addTextRotate メソッド .....	95
addText メソッド .....	96
addTextLang メソッド .....	97
addTextFont メソッド .....	99
addTextSmooth メソッド .....	100
addTextDouble メソッド .....	101
addTextSize メソッド .....	102
addTextStyle メソッド .....	103
addTextPosition メソッド .....	105
addTextVPosition メソッド .....	106
addFeedUnit メソッド .....	107
addFeedLine メソッド .....	108
addFeedPosition メソッド .....	109
addFeed メソッド .....	110
addImage メソッド .....	111
addLogo メソッド .....	113
addBarcode メソッド .....	114
addSymbol メソッド .....	118
addHLine メソッド .....	122
addVLineBegin メソッド .....	124
addVLineEnd メソッド .....	126
addPageBegin メソッド .....	127
addPageEnd メソッド .....	128
addPageArea メソッド .....	129
addPageDirection メソッド .....	131
addPagePosition メソッド .....	132
addPageLine メソッド .....	134
addPageRectangle メソッド .....	136
addCut メソッド .....	138
addPulse メソッド .....	139
addSound メソッド .....	140
addLayout メソッド .....	142
recover メソッド .....	145
addRecovery メソッド .....	146
reset メソッド .....	147
addReset メソッド .....	148
addCommand メソッド .....	149
send メソッド .....	150
print メソッド .....	151
getPrintJobStatus メソッド .....	153
startMonitor メソッド .....	154

stopMonitor メソッド .....	155
halftone プロパティ .....	156
brightness プロパティ .....	157
force プロパティ .....	158
timeout プロパティ .....	159
interval プロパティ .....	160
drawerOpenLevel プロパティ .....	161
message プロパティ .....	162
onreceive イベント .....	163
onstatuschange イベント .....	166
onbatterystatuschange イベント .....	167
online イベント .....	168
offline イベント .....	169
onpoweroff イベント .....	170
oncoverok イベント .....	171
oncoveropen イベント .....	172
onpaperok イベント .....	173
onpapernearend イベント .....	174
onpaperend イベント .....	175
ondrawerclosed イベント .....	176
ondraweropen イベント .....	177
onbatteryok イベント .....	178
onbatterylow イベント .....	179
<b>■ HybridPrinter オブジェクト .....</b>	<b>180</b>
lock メソッド .....	180
unlock メソッド .....	181
eject メソッド .....	182
ReceiptPrinter.send メソッド .....	183
ReceiptPrinter.print メソッド .....	184
SlipPrinter.send メソッド .....	185
SlipPrinter.cancel メソッド .....	186
EndorsePrinter.enable40cplMode メソッド .....	187
EndorsePrinter.send メソッド .....	188
EndorsePrinter.cancel メソッド .....	189
MICRReader.read メソッド .....	190
MICRReader.cleaning メソッド .....	191
MICRReader.cancel メソッド .....	192
recover メソッド .....	193
reset メソッド .....	194
startMonitor メソッド .....	195
stopMonitor メソッド .....	196
halftone プロパティ .....	197
brightness プロパティ .....	198
force プロパティ .....	199
SlipPrinter.timeout プロパティ .....	200
EndorsePrinter.timeout プロパティ .....	201
MICRReader.timeout プロパティ .....	202
interval プロパティ .....	203
onreceive イベント .....	204
<b>■ Scanner オブジェクト .....</b>	<b>207</b>
ondata イベント .....	207
<b>■ SimpleSerial オブジェクト .....</b>	<b>208</b>
sendCommand .....	208
oncommandreply イベント .....	209
<b>■ DeviceHubTerminal オブジェクト .....</b>	<b>210</b>

shutdown メソッド .....	210
restart メソッド .....	212
■ CommBoxManager オブジェクト .....	214
openCommBox メソッド .....	214
closeCommBox メソッド .....	216
■ CommBox オブジェクト .....	217
getCommHistory メソッド .....	217
send メソッド .....	219
onreceive イベント .....	220
■ ePosDeviceConfiguration オブジェクト .....	221
Constructor .....	221
getRegisteredDevices メソッド .....	222
■ Error Code 一覧 .....	224
callback パラメーターで取得するエラーコードと対処方法 .....	224
onreceive イベントで取得するエラーコードと対処方法 .....	226
■ Key Code 一覧 .....	231

---

## デバイス制御スクリプト ..... 233

■ プログラミング .....	233
デバイス制御スクリプトの使用について .....	233
デバイス制御スクリプトの構成 .....	234
■ デバイス制御スクリプト API 一覧 .....	236
ClientConnection オブジェクト .....	236
DeviceConnection オブジェクト .....	236
デバイス制御スクリプト名オブジェクト .....	236
■ ClientConnection オブジェクト .....	237
send .....	237
■ DeviceConnection オブジェクト .....	238
send .....	238
■ デバイス制御スクリプト名オブジェクト .....	239
onDeviceData イベント（キー入力デバイス） .....	239
onDeviceData イベント（シリアル通信デバイス） .....	240
任意イベント .....	241

---

## 機器仕様 ..... 242

■ クラス別対応プリンター .....	242
■ サポート API 一覧 .....	243
Printer オブジェクト .....	243
■ プリンター別サポート情報 .....	252
TM-m10 .....	252
TM-m30 .....	254
TM-P20 .....	256
TM-P60II (Peeler) .....	259

TM-T20.....	262
TM-T20II.....	265
TM-T70.....	268
TM-T70II.....	270
TM-T88V.....	273
TM-T90II.....	276
TM-T70-i.....	278
TM-T70II-DT.....	280
TM-T88V-i.....	283
TM-T88V-DT.....	286
TM-T88IV.....	289
TM-T90.....	291
TM-T90KP.....	293
TM-L90.....	295
■ ファームウェア Ver. による使用制限.....	297

## サンプルプログラム..... 299

■ 機能 .....	299
POS Terminal.....	300
Entry Terminal.....	300
Receipt Designer .....	301
Printer Sample.....	301
Hybrid Printer Sample.....	301
Customer Display Sample.....	302
Keyboard Sample .....	302
MSR Sample .....	302
Barcode Scanner Sample.....	303
■ 使用環境 .....	304
■ サンプルプログラムの起動手順.....	304
プリンターの環境設定.....	304
サンプルプログラムの起動.....	308
■ サンプルプログラムの使い方 .....	309
POS Terminal Sample .....	309
Entry Terminal.....	315
Receipt Designer .....	318
Printer Sample.....	324
Hybrid Printer Sample.....	325
Customer Display Sample.....	328
Keyboard Sample .....	330
MSR Sample .....	331
Barcode Scanner Sample.....	332

## 活用ガイド..... 333

■ 常時監視するには .....	333
■ 一台のプリンターを複数のモバイル端末から使用するには.....	334
■ ネットワークを自動で再接続するには .....	335
■ アプリケーション間でデータを送受信するには .....	337

■ 迂回印刷するには .....	339
■ Web ページを頻繁に更新 / 遷移するアプリケーションを開発するには.....	341
ePOS-Print オプションを有効にするには .....	341
制限事項 .....	341

---

## **付録..... 342**

■ オープンソースライセンス.....	342
---------------------	-----

# 機能

Epson ePOS SDK for JavaScript は、Web アプリケーションソフト開発用の SDK です。Epson ePOS SDK for JavaScript を使用すると、シンプルなコードでプリンターを制御できます。また、TM インテリジェントプリンターでは、本体に接続したカスタマーディスプレイやバーコードスキャナーなどの POS 周辺機器、ネットワーク接続した TM プリンターを制御できます。さらに、スプーラー機能やコミュニケーションボックスなどの TM インテリジェントプリンター独自機能を使用したアプリケーションソフトも開発できます。



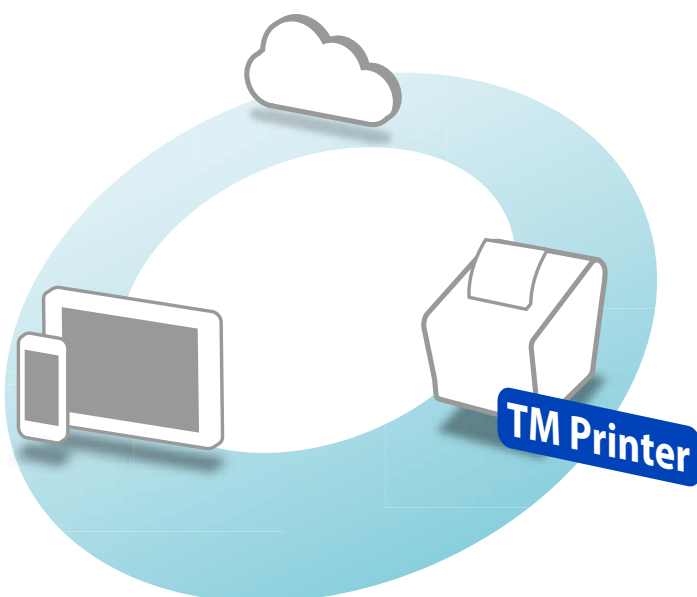
本書では、レシートプリンターのスタンダードモデルとモバイルモデルを「TM プリンター」と総称します。また、TM-DT シリーズと TM-i シリーズを「TM インテリジェントプリンター」と総称します。

## Epson ePOS SDK for JavaScript でのアプリケーション開発

Epson ePOS SDK for JavaScript で開発できるアプリケーションソフトのシステム構成は、以下の通りです。システムの構築方法や使用制限などの詳細は、各プリンターの詳細取扱説明書を参照してください。

### TM プリンターモデル

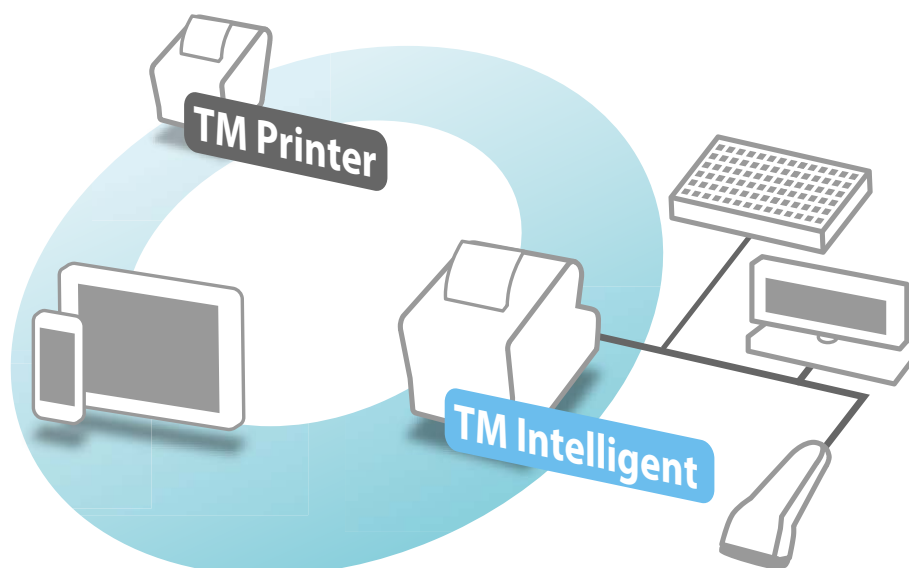
スマートデバイスと TM プリンターをネットワーク（有線 LAN、無線 LAN）で接続するシステムです。スマートデバイス上の HTML5 と互換性のある Web ブラウザーから、TM プリンターを制御することができます。



EpsonNet Config を用いて ePOS-Print メニューを「有効」に設定する必要があります。詳細は、使用する TM プリンター用インターフェースの詳細取扱説明書を参照してください。

## TM インテリジェントプリンターモデル

スマートデバイスと TM インテリジェントプリンターをネットワーク（有線 LAN、無線 LAN）で接続するシステムです。TM インテリジェントプリンターに接続された POS 周辺機器や複数の TM プリンターを制御することができます。



TM インテリジェントプリンターで制御できる POS 周辺機器や TM プリンターは、TM-DT ソフトウェア、TM-i ファームウェアのバージョンにより異なります。使用する POS 周辺機器や TM プリンターは、EPSON TMNet WebConfig を用いて TM インテリジェントプリンターに登録する必要があります。

Epson ePOS SDK for JavaScript で開発したアプリケーションを TM インテリジェントプリンターモデルで利用するには、TM インテリジェントプリンターの sPOS-Device Service を「有効」に設定する必要があります。



TM インテリジェントプリンターの ePOS-Device Service は、初期設定で「有効」に設定されています。

# アプリケーションソフトの動作環境

Epson ePOS SDK for JavaScript を使用して開発するアプリケーションソフトの動作環境は、以下の通りです。  
対象となる Web ブラウザーの最新情報は、Epson ePOS SDK for JavaScript パッケージの README.jp.txt を参照してください。

## 対象プリンター

アプリケーションソフトから制御できるプリンターは、以下の通りです。

---

### TM プリンター

使用できるインターフェイスは TM プリンターによって異なります。詳細は、各プリンターの詳細取扱説明書を参照してください。

- ☐ TM-m10
- ☐ TM-m30
- ☐ TM-P20
- ☐ TM-P60II
- ☐ TM-T20II
- ☐ TM-T70
- ☐ TM-T70II
- ☐ TM-T88V
- ☐ TM-T90II

---

### TM インテリジェントプリンター

- ☐ TM-T70II-DT
- ☐ TM-T88V-DT
- ☐ TM-T70-i (TM-i ファームウェア Ver.4.0 以降)
- ☐ TM-T88V-i (TM-i ファームウェア Ver.4.0 以降)

---

### ネットワークプリンター

TM プリンターで、TM インテリジェントプリンター経由でのみ制御できるプリンターです。

- ☐ TM-L90
- ☐ TM-T88IV
- ☐ TM-T90
- ☐ TM-T90KP

## ダウンロードコンテンツ

### パッケージ

Epson ePOS SDK for JavaScript パッケージには、以下のファイルが含まれています。

ファイル名	説明
epos-x.x.x.js	機能実行用のライブラリーファイルです。
ePOS_SDK_Sample_JavaScript.zip	サンプルプログラムファイルです。
DeviceControlScript_Sample.zip	デバイス制御スクリプト用のサンプルプログラムファイルです。
README.jp.txt	日本語版 README ファイルです。
README.en.txt	英語版 README ファイルです。
EULA.jp.txt	日本語版 SOFTWARE LICENSE AGREEMENT を記載しています。
EULA.en.txt	英語版 SOFTWARE LICENSE AGREEMENT を記載しています。
ePOS_SDK_for_JavaScript_um_ja_revx.pdf	日本語版ユーザーズマニュアルです。
ePOS_SDK_for_JavaScript_um_en_revx.pdf	英語版ユーザーズマニュアルです。
ePOS_SDK_JavaScript_Migration_Guide_ja_revx.pdf	日本語版マイグレーションガイドです。ePOS-Print SDK、ePOS-Device SDK からの移行方法を説明しています。
ePOS_SDK_JavaScript_Migration_Guide_en_revx.pdf	英語版マイグレーションガイドです。

## Web コンテンツ

下記サイトにて、Web リファレンスを公開しています。

<https://reference.epson-biz.com/pos/reference/>

## 制限事項

- 2次元バーコードリーダーを使用する場合、日本語などのマルチバイト文字は正しく取得できません。
- 2次元バーコードのデータに ASCII 制御コード (0x00 ~ 0x1F) が含まれる場合、制御コードは取得できません。
- セキュリティーで保護された Web ページ (HTTPS) からプリンターに印刷することができません。
- JavaScript ライブラリーの prototype.js は、Epson ePOS SDK for JavaScript のライブラリー (epos-x.x.x.js) と競合するため、一緒に使用できません。

# 使い方

Epson ePOS SDK for JavaScript の組み込みと、Epson ePOS SDK for JavaScript を使用した基本的なプログラミング方法を説明します。

## プリンターとの接続

構築するシステム構成に合わせて、スマートデバイスとプリンターを接続します。  
詳細は、各プリンターの詳細取扱説明書を参照してください。

## Epson ePOS SDK for JavaScript の組み込み方法

- 1 epos-x.x.x.js を Web サーバーに配置します。
- 2 HTML の `<script>` タグで Web ページにスクリプトを組み込みます。

組み込み例

```
<script type="text/javascript" src="epos-2.0.0.js"></script>
```

## プログラミングガイド

Epson ePOS SDK for JavaScript を使用した基本的なプログラミング方法を説明します。

### プログラミングフロー

開発するアプリケーションソフトで制御するプリンターや POS 周辺機器別に、プログラミングフローを説明します。

- [プリンターを制御する](#)
- [カスタマーディスプレイを制御する](#)
- [キーボード、バーコードスキャナーを制御する](#)

## プリンターを制御する

TM プリンターや TM インテリジェントプリンターの本体プリンター、TM インテリジェントプリンターとネットワークで繋がった TM プリンターを制御し、レシート印刷するプログラミング方法を説明します。

### 1. ePOSDevice オブジェクトの作成（デバイスと接続 / 通信）



### 2. Printer オブジェクトを取得（プリンターの選択）



### 3. 印刷データの作成（データのバッファリング）



### 4. 印刷データの送信（印刷 / 切断）

#### ePOSDevice オブジェクトの作成(デバイスと接続 / 通信)

ePOSDevice オブジェクトを作成し、初期化します。

```
var ePosDev = new epson.ePOSDevice();

function connect() {
  var ipAddress = '192.168.192.168';
  var port = '8008';

  ePosDev.connect(ipAddress, port, callback_connect);
}
```

#### Printer オブジェクトを取得(プリンターの選択)

Printer オブジェクトを取得し、制御するプリンターを選択します。

```
function callback_connect(resultConnect){
  var deviceId = 'local_printer';
  var options = {'crypto' : false, 'buffer' : false};

  if ((resultConnect == 'OK') || (resultConnect == 'SSL_CONNECT_OK')) {
    //Printer オブジェクトを取得する
    ePosDev.createDevice(deviceID, ePosDev.DEVICE_TYPE_PRINTER, options,
    callback_createDevice);
  }
  else {
    // エラーメッセージ表示
  }
}

var printer = null;

function callback_createDevice(deviceObj, errorCode){
  if (deviceObj === null) {
    // Printer オブジェクト取得失敗、エラーメッセージ表示
    return;
  }
  printer = deviceObj;
```

```
// 印刷完了イベントを登録
printer.onreceive = function(response) {
  if (response.success) {
    // 印刷成功メッセージ表示
  }
  else {
    // エラーメッセージ表示
  }
};
}
```

## 印刷データの作成(データのバッファリング)

Printer オブジェクトの addxx 系の API を使用して、印刷データを作成します。

以下の例では、中央揃えで “Hello World” と印刷するデータを作成しています。

```
function createData() {
  printer.addTextAlign(printer.ALIGN_CENTER);
  printer.addText('Hello World\n');
}
```

## 印刷データの送信(印刷 / 切断)

印刷処理を行い、印刷結果を取得します。

```
function send() {
  if (ePosDev.isConnected) {
    printer.send();
  }
}
```

Printer オブジェクトを破棄し、プリンターと切断します。

```
//Printer オブジェクトを破棄する
ePosDev.deleteDevice(printer, callback_deleteDevice);

function callback_deleteDevice(errorCode) {
  // デバイスと切断
  ePosDev.disconnect();
}
```

## カスタマーディスプレイを制御する

カスタマーディスプレイを制御し、ディスプレイにテキストを表示するプログラミング方法を説明します。

### 1. ePOSDevice オブジェクトの作成（デバイスと接続 / 通信）



### 2. Display オブジェクトを取得（デバイスの選択）



### 3. 表示データの作成（データのバッファリング）



### 4. 表示データの送信（表示 / 切断）

#### ePOSDevice オブジェクトの作成(デバイスと接続 / 通信)

ePOSDevice オブジェクトを作成し、初期化します。

```
var ePosDev = new epos.ePOSDevice();

function connect() {
  var ipAddress = '192.168.192.168';
  var port = '8008';

  ePosDev.connect(ipAddress, port, callback_connect);
}
```

#### Display オブジェクトを取得(デバイスの選択)

Display オブジェクトを取得し、制御するカスタマーディスプレイを選択します。

```
function callback_connect(resultConnect){
  var deviceId = 'local_display';
  var options = {'crypto' : false, 'buffer' : false};

  if ((resultConnect == 'OK') || (resultConnect == 'SSL_CONNECT_OK')) {
    //Display オブジェクトを取得する
    ePosDev.createDevice(deviceId, ePosDev.DEVICE_TYPE_DISPLAY, options,
    callback_createDevice);
  }
  else {
    // エラーメッセージ表示
  }
}

var display = null;

function callback_createDevice(deviceObj, errorCode){
  if (deviceObj === null) {
    // Display オブジェクト取得失敗、エラーメッセージ表示
    return;
  }
  display = deviceObj;
```

```
// ディスプレイ表示完了イベントを登録
display.onreceive = function(response) {
  if (response.success) {
    // ディスプレイ表示成功メッセージ表示
  }
  else {
    // エラーメッセージ表示
  }
};
}
```

### 表示データの作成(データのバッファリング)

Display オブジェクトの addxx 系の API を使用して、表示データを作成します。

以下の例では、“Hello World” と表示するデータを作成しています。

```
function createData() {
  diaplay.addText('Hello World');
}
```

### 表示データの送信(表示 / 切断)

ディスプレイ表示処理を行い、表示結果を取得します。

```
function send(){
  if (ePosDev.isConnected) {
    display.send();
  }
}
```

Display オブジェクトを破棄し、カスタマーディスプレイと切断します。

```
//Display オブジェクトを破棄する
ePosDev.deleteDevice(display, callback_deleteDevice);

function callback_deleteDevice(errorCode) {
  // デバイスと切断
  ePosDev.disconnect();
}
```

## キーボード、バーコードスキャナーを制御する

キーボードやバーコードスキャナーを制御し、入力データを受信するプログラミング方法を説明します。

以下の説明では、キーボードを制御する場合のプログラミングフローを説明します。

### 1. ePOSDevice オブジェクトの作成（デバイスと接続 / 通信）



### 2. Keyboard オブジェクトを取得（デバイスの選択）



### 3. デバイスとの切断

#### ePOSDevice オブジェクトの作成(デバイスと接続 / 通信)

ePOSDevice オブジェクトを作成し、初期化します。

```
var ePosDev = new epson.ePOSDevice();

function connect() {
  var ipAddress = '192.168.192.168';
  var port = '8008';

  ePosDev.connect(ipAddress, port, callback_connect);
}
```

#### Keyboard オブジェクトを取得(デバイスの選択)

Keyboard オブジェクトを取得してキーボードと接続し、入力データを受信するイベントを登録します。

```
function callback_connect(resultConnect){
  var deviceId = 'local_keyboard';
  var options = {'crypto' : false, 'buffer' : false};

  if ((resultConnect == 'OK') || (resultConnect == 'SSL_CONNECT_OK')) {
    //Keyboard オブジェクトを取得する
    ePosDev.createDevice(deviceID, ePosDev.DEVICE_TYPE_KEYBOARD, options,
callback_createDevice);
  }
  else {
    // エラーメッセージ表示
  }
}

var keyboard = null;

function callback_createDevice(deviceObj, errorCode){
  if (deviceObj === null) {
    // Keyboard オブジェクト取得失敗、エラーメッセージ表示
    return;
  }
  keyboard = deviceObj;
```

```
// キー押下イベントを登録
keyboard.onkeypress = function(response) {
  if (response.keycode !== 0) {
    // 受信メッセージ表示
  }
};
```

## デバイスとの切断

Keyboard オブジェクトを破棄して通信を切断し、キーボードからの入力データ受信を終了します。

```
//Keyboard オブジェクトを破棄する
ePosDev.deleteDevice(keyboard, callback_deleteDevice);

function callback_deleteDevice(errorCode) {
  // デバイスと切断
  ePosDev.disconnect();
}
```

## 設定用の命令バッファの有効範囲

設定用に使用される Printer オブジェクトの addXXX の有効範囲は、addXXX 設定後、[send メソッド](#)が実行されるまでです。設定した値は、[send メソッド](#)の実行ごとに初期化されます。以下を参考にしてください。

```
printer.addText('Hello World\n');  
printer.addTextFont(Printer.FONT_B);  
printer.addText('Hello World\n');  
printer.addText('Hello World\n');  
printer.send();  
printer.addText('Hello World\n');  
printer.send();
```

赤字：フォント A

緑字：フォント B

## エラーコード

Epson ePOS SDK for JavaScript には、callback パラメーターで取得するエラーコードと、onreceive イベントで取得するエラーコードがあります。それぞれのエラーコードに対応する要因と対処方法は、以下を参照してください。

- ❑ [callback パラメーターで取得するエラーコードと対処方法](#)
- ❑ [onreceive イベントで取得するエラーコードと対処方法](#)

# API リファレンス

本章では、ePOS SDK for JavaScript で用意されている API について説明しています。

## ePOS SDK API 一覧

ePOS SDK for JavaScript には、以下のオブジェクトが用意されています。

プリンターや周辺機器によって使用可能な API やパラメーターが異なります。[機器仕様](#)を参照してください。

- ❑ [ePOSDevice オブジェクト](#) (window.epson.ePOSDevice)
- ❑ [デバイスオブジェクト共通](#)
- ❑ [CashChanger オブジェクト](#)
- ❑ [Display オブジェクト](#)
- ❑ [Keyboard オブジェクト](#)
- ❑ [MSR オブジェクト](#)
- ❑ [Printer オブジェクト](#)
- ❑ [HybridPrinter オブジェクト](#)
- ❑ [Scanner オブジェクト](#)
- ❑ [SimpleSerial オブジェクト](#)
- ❑ [DeviceHubTerminal オブジェクト](#)
- ❑ [CommBoxManager オブジェクト](#)
- ❑ [CommBox オブジェクト](#)
- ❑ [ePosDeviceConfiguration オブジェクト](#)

### ePOSDevice オブジェクト

アプリケーションとプリンターを接続します。

	API	説明
初期化	<a href="#">Constructor</a>	ePOSDevice オブジェクトの初期化
通信経路	<a href="#">connect メソッド</a>	通信経路の確立
	<a href="#">disconnect メソッド</a>	通信経路の切り離し
	<a href="#">isConnected メソッド</a>	通信経路の確立状態を取得
デバイス	<a href="#">createDevice メソッド</a>	デバイスオブジェクトの取得
	<a href="#">deleteDevice メソッド</a>	デバイスオブジェクトの破棄
管理情報	<a href="#">getAdmin メソッド</a>	管理者情報の取得
	<a href="#">getLocation メソッド</a>	設置場所情報の取得
OFSC	<a href="#">sendOfscXml メソッド</a>	OFSC-Print 用 XML データの送信
アプリケーション間通信	<a href="#">getCommBoxManager メソッド</a>	CommBoxManager オブジェクトを取得
再接続開始	<a href="#">onreconnecting イベント</a>	再接続処理開始イベント
再接続終了	<a href="#">onreconnect イベント</a>	再接続終了イベント

API		説明
切断	<code>ondisconnect</code> イベント	ネットワーク切断イベント

## デバイスオブジェクト共通

デバイス制御スクリプト側の任意イベントを呼び出します。

API		説明
イベント呼び出し	<code>callEvent</code> メソッド	対応するデバイス制御スクリプトの任意イベントの呼び出し（SimpleSerial オブジェクトはサポートしていません）

## CashChanger オブジェクト

自動釣り銭機を用いた出入金処理の制御を行います。

API		説明
機器設定	<code>setConfig</code> メソッド	機器の設定を変更
計数	<code>readCashCount</code> メソッド	機器内現金の金額を取得
入金	<code>beginDeposit</code> メソッド	入金処理を開始
	<code>pauseDeposit</code> メソッド	入金処理を一時停止
	<code>restartDeposit</code> メソッド	入金処理を再開
	<code>endDeposit</code> メソッド	入金処理を終了
出金	<code>dispenseCash</code> メソッド	金種を指定して出金
	<code>dispenseChange</code> メソッド	金額を指定して出金
回収	<code>collectCash</code> メソッド	機器内の現金を回収
ドロアー	<code>openDrawer</code> メソッド	キャッシュドロアーを開く
コマンド送信	<code>sendCommand</code> メソッド	任意コマンドを送信
機器設定	<code>onconfigchange</code> イベント	機器の設定変更の結果通知イベント
計数	<code>oncashcount</code> イベント	計数の結果通知イベント
入金	<code>ondeposit</code> イベント	入金金額の通知イベント
出金	<code>ondispense</code> イベント	出金動作の通知イベント
回収	<code>oncollect</code> イベント	回収動作の通知イベント
コマンド結果	<code>oncommandreply</code> イベント	任意コマンドの結果通知イベント
状態変化	<code>onstatuschange</code> イベント	状態変化イベント

## Display オブジェクト

カスタマーディスプレイへの文字表示制御を行います。

API		説明
ウィンドウ	<a href="#">createWindow メソッド</a>	表示領域の定義
	<a href="#">destroyWindow メソッド</a>	表示領域の設定の破棄
	<a href="#">setCurrentWindow メソッド</a>	表示領域の切り替え
	<a href="#">clearWindow メソッド</a>	現在の表示領域を消去
カーソル	<a href="#">setCursorPosition メソッド</a>	カーソル位置の移動
	<a href="#">moveCursorPosition メソッド</a>	表示領域内でカーソル位置を移動
	<a href="#">setCursorType メソッド</a>	カーソル表示の変更
テキスト表示	<a href="#">addText メソッド</a>	テキストの表示
	<a href="#">addReverseText メソッド</a>	反転テキストの表示
	<a href="#">addMarquee メソッド</a>	マーキー表示
表示属性	<a href="#">setBlink メソッド</a>	表示の点滅
	<a href="#">setBrightness メソッド</a>	表示輝度の変更
時計	<a href="#">showClock メソッド</a>	時計の表示
コマンド送信	<a href="#">addCommand メソッド</a>	任意コマンドの実行
送信	<a href="#">send メソッド</a>	制御命令の送信
初期化	<a href="#">reset メソッド</a>	リセット
結果受信	<a href="#">onreceive イベント</a>	制御結果受信イベント


## Keyboard オブジェクト

キーボードによる文字入力の制御を行います。

API		説明
文字列設定	<a href="#">setPrefix メソッド</a>	連続した文字列として扱う条件を設定
キー検出	<a href="#">onkeypress イベント</a>	キー押下検出イベント
文字列検出	<a href="#">onstring イベント</a>	文字列検出イベント
キーコード設定	<a href="#">setMSRPrefix メソッド</a>	MSR データ等を制御する条件を設定
データ検出	<a href="#">ondata イベント</a>	カード情報受信イベント

## MSR オブジェクト

MSR によるカード読み取りを制御します。

	デバイス制御スクリプトの作成により、カスタマイズできます。詳細は、「第4章 デバイス制御スクリプト」を参照してください。
---	--

API		説明
データ検出	<a href="#">ondata イベント</a>	カードデータ検出イベント

## Printer オブジェクト

プリンターへの印刷を制御します。

印字モードには、スタンダードモードとページモードがあります。

- スタンダードモード

1 行単位で印字する印字モードです。文字サイズ、画像、バーコードなどの高さに合わせて改行量が調整されるため、印字量によって用紙の長さが変化するレシート印刷に適しています。

- ページモード

印字領域を設定して印字データ（文字 / 画像 / バーコード等）を展開し、一括印字する印字モードです。

ページモードでの処理を行う際は、[addPageBegin メソッド](#)でページモード処理を開始し、[addPageEnd メソッド](#)でページモード処理を終了してください。

API		説明	スタン ダード モード	ページ モード
テキスト	<a href="#">addTextAlign メソッド</a>	位置揃え設定を命令バッファに追加	○	×
	<a href="#">addTextLineSpace メソッド</a>	改行量設定を命令バッファに追加	○	○
	<a href="#">addTextRotate メソッド</a>	倒立印字設定を命令バッファに追加	○	×
	<a href="#">addText メソッド</a>	文字印字を命令バッファに追加	○	○
	<a href="#">addTextLang メソッド</a>	言語設定を命令バッファに追加	○	○
	<a href="#">addTextFont メソッド</a>	文字フォント設定を命令バッファに追加	○	○
	<a href="#">addTextSmooth メソッド</a>	文字スムージング設定を命令バッファに追加	○	○
	<a href="#">addTextDouble メソッド</a>	文字倍角設定を命令バッファに追加	○	○
	<a href="#">addTextSize メソッド</a>	文字倍率設定を命令バッファに追加	○	○
	<a href="#">addTextStyle メソッド</a>	文字装飾設定を命令バッファに追加	○	○
	<a href="#">addTextPosition メソッド</a>	文字印字位置設定を命令バッファに追加	○	○
紙送り	<a href="#">addTextVPosition メソッド</a>	縦方向の印字開始位置設定を命令バッファに追加	×	○
	<a href="#">addFeedUnit メソッド</a>	ドット単位の紙送りを命令バッファに追加	○	○
	<a href="#">addFeedLine メソッド</a>	行単位の紙送りを命令バッファに追加	○	○
	<a href="#">addFeedPosition メソッド</a>	レシート紙、ラベル紙の制御を命令バッファに追加	○	×
	<a href="#">addFeed メソッド</a>	改行を命令バッファに追加	○	○

API		説明	スタン ダード モード	ページ モード
グラフィック	addImage メソッド	ラスターイメージ印字を命令バッファに追加	○	○
	addLogo メソッド	NV ロゴ印字を命令バッファに追加	○	○
バーコード	addBarcode メソッド	バーコード印字を命令バッファに追加	○	○
	addSymbol メソッド	2次元シンボル印字を命令バッファに追加	○	○
罫線	addHLine メソッド	横罫線印字を命令バッファに追加	○	×
	addVLineBegin メソッド	縦罫線開始を命令バッファに追加	○	×
	addVLineEnd メソッド	縦罫線終了を命令バッファに追加	○	×
ページモード	addPageBegin メソッド	ページモード開始を命令バッファに追加	○	×
	addPageEnd メソッド	ページモード終了を命令バッファに追加	×	○
	addPageArea メソッド	ページモード印字領域設定を命令バッファに追加	×	○
	addPageDirection メソッド	ページモード印字方向設定を命令バッファに追加	×	○
	addPagePosition メソッド	ページモード印字位置設定を命令バッファに追加	×	○
	addPageLine メソッド	ページモード直線描画を命令バッファに追加	×	○
	addPageRectangle メソッド	ページモード四角形描画を命令	×	○
カット	addCut メソッド	用紙カットを命令バッファに追加	○	×
ドロアー	addPulse メソッド	ドロアーキックを命令バッファに追加	○	×
ブザー	addSound メソッド	ブザー鳴動を命令バッファに追加	○	×
レイアウト	addLayout メソッド	用紙レイアウトの設定を命令バッファに追加	○	×
リカバリー	recover メソッド	復帰可能エラーから復帰	○	×
	addRecovery メソッド	エラーからの復帰タグを追加する	○	×
リセット	reset メソッド	プリンターをリセット	○	×
	addReset メソッド	プリンターのリセットタグを追加する	○	×
コマンド送信	addCommand メソッド	コマンドを命令バッファに追加	○	○

API		説明	スタンダードモード	ページモード
送信	send メソッド	<ul style="list-style-type: none"> <li>印刷ドキュメントの送信</li> <li>ジョブ ID を指定して送信</li> </ul>	○	○
	print メソッド	<ul style="list-style-type: none"> <li>HTML5 Canvas の印刷</li> <li>ジョブ ID を指定して印刷</li> </ul>	○	○
印刷ジョブ	getPrintJobStatus メソッド	印刷ジョブの状態を取得	○	○
状態監視	startMonitor メソッド	ステータスイベントの有効化	○	○
	stopMonitor メソッド	ステータスイベントの無効化	○	○

API		説明
イメージ	halftone プロパティ	ラスターイメージのハーフトーン処理方法
	brightness プロパティ	ラスターイメージの明るさ補正值
強制送信	force プロパティ	強制送信モード
タイムアウト	timeout プロパティ	送信タイムアウト時間
監視間隔	interval プロパティ	プリンタステータスの更新間隔
ドロアーオープン	drawerOpenLevel プロパティ	ドロアーの信号線状態
命令バッファ	message プロパティ	命令バッファの直接操作
結果受信	onreceive イベント	<ul style="list-style-type: none"> <li>応答ドキュメント受信イベント</li> <li>ジョブ ID 指定の受信イベント</li> </ul>
	onstatuschange イベント	ステータス変更イベント
	onbatterystatuschange イベント	バッテリーステータス変更イベント
	ononline イベント	オンラインイベント
	onoffline イベント	オフラインイベント
	onpoweroff イベント	無応答イベント
	oncoverok イベント	カバークローズイベント
	oncoveropen イベント	カバーオープンイベント
	onpaperok イベント	用紙ありイベント
	onpaperend イベント	用紙エンドイベント
	onpapernearend イベント	用紙ニアエンドイベント
	ondrawerclosed イベント	ドロアークローズイベント
	ondraweropen イベント	ドロアーオープンイベント
	onbatteryok イベント	バッテリー残量ありイベント
	onbatterylow イベント	バッテリー残量なしイベント

## HybridPrinter オブジェクト

ハイブリッドプリンターへの印刷を制御します。

API		説明	スタンダードモード	ページモード
デバイスロック	lock メソッド	デバイスポートをロック	○	×
	unlock メソッド	デバイスポートのロックを解除	○	×
排紙	eject メソッド	チェック紙を排出	○	×
レシート印字	ReceiptPrinter.send メソッド	印刷ドキュメントの送信	○	○
	ReceiptPrinter.print メソッド	HTML5 Canvas の印刷	○	○
表面スリップ印字	SlipPrinter.send メソッド	印刷ドキュメントの送信	○	○
	SlipPrinter.cancel メソッド	チェック紙挿入待ち状態を解除	○	○
裏面スリップ印字	EndorsePrinter.enable40cplMode メソッド	40cpl モードを設定	○	×
	EndorsePrinter.send メソッド	印刷ドキュメントの送信	○	×
	EndorsePrinter.cancel メソッド	チェック紙挿入待ち状態を解除	○	×
MICR	MICRReader.read メソッド	MICR 読み取り	○	×
	MICRReader.cleaning メソッド	MICR 機構の清掃	○	×
	MICRReader.cancel メソッド	クリーニングシート挿入待ち状態を解除	○	×
リカバリー	recover メソッド	復帰可能エラーから復帰	○	×
リセット	reset メソッド	プリンターをリセット	○	×
状態監視	startMonitor メソッド	ステータスイベントの有効化	○	○
	stopMonitor メソッド	ステータスイベントの無効化	○	○

API		説明
イメージ	halftone プロパティ	ラスタイメージのハーフトーン処理方法
	brightness プロパティ	ラスタイメージの明るさ補正值
強制送信	force プロパティ	強制送信モード
タイムアウト	SlipPrinter.timeout プロパティ	チェック紙挿入待ちタイムアウト時間
	EndorsePrinter.timeout プロパティ	チェック紙挿入待ちタイムアウト時間
	MICRReader.timeout プロパティ	チェック紙 / クリーニングシート挿入待ちタイムアウト時間
監視間隔	interval プロパティ	プリンタースtatusの更新間隔
ドロアーオープン	drawerOpenLevel プロパティ *	ドロアーオープンレベル

API		説明
結果受信	onreceive イベント	応答ドキュメント受信イベント
	onstatuschange イベント *	ステータス変更イベント
	ononline イベント *	オンラインイベント
	onoffline イベント *	オフラインイベント
	onpoweroff イベント *	無応答イベント
	oncoverok イベント *	カバークローズイベント
	oncoveropen イベント *	カバーオープンイベント
	onpaperok イベント *	用紙ありイベント
	onpaperend イベント *	用紙エンドイベント
	onpapernearend イベント *	用紙ニアエンドイベント
	ondrawerclosed イベント *	ドロアークローズイベント
	ondraweropen イベント *	ドロアークオープンイベント

\* [Printer オブジェクト](#)と共通で使用できるイベント。

## HybridPrinter オブジェクトでの印刷ドキュメント作成

HybridPrinter オブジェクトは、[Printer オブジェクト](#)と同じメソッドを使用して印刷ドキュメントを作成します。

以下にメソッドと印字方法の対応一覧表を示します。

API		印字方法			
		レシート	スリップ		
			表面	裏面	裏面 40cpl
テキスト	<a href="#">addTextAlign メソッド</a>	○	○	○	×
	<a href="#">addTextLineSpace メソッド</a>	○	○	○	×
	<a href="#">addTextRotate メソッド</a>	○	○	○	○
	<a href="#">addText メソッド</a>	○	○	○	○
	<a href="#">addTextLang メソッド</a>	○	○	×	×
	<a href="#">addTextFont メソッド</a>	○	○	○	×
	<a href="#">addTextSmooth メソッド</a>	○	×	×	×
	<a href="#">addTextDouble メソッド</a>	○	○	○	×
	<a href="#">addTextSize メソッド</a>	○	○	○	×
	<a href="#">addTextStyle メソッド</a>	○	○	○	×
	<a href="#">addTextPosition メソッド</a>	○	○	○	○
	<a href="#">addTextVPosition メソッド</a>	○	○	×	×

API		印字方法			
		レシート	スリップ		
			表面	裏面	裏面 40cpl
紙送り	<a href="#">addFeedUnit メソッド</a>	○	○	○	○
	<a href="#">addFeedLine メソッド</a>	○	○	○	×
	<a href="#">addFeedPosition メソッド</a>	×	×	×	×
	<a href="#">addFeed メソッド</a>	○	○	○	○
グラフィック	<a href="#">addImage メソッド</a>	○	×	×	×
	<a href="#">addLogo メソッド</a>	○	○	×	×
バーコード	<a href="#">addBarcode メソッド</a>	○	○	×	×
	<a href="#">addSymbol メソッド</a>	○	×	×	×
罫線	<a href="#">addHLine メソッド</a>	×	×	×	×
	<a href="#">addVLineBegin メソッド</a>	×	×	×	×
	<a href="#">addVLineEnd メソッド</a>	×	×	×	×
ページモード	<a href="#">addPageBegin メソッド</a>	○	○	×	×
	<a href="#">addPageEnd メソッド</a>	○	○	×	×
	<a href="#">addPageArea メソッド</a>	○	○	×	×
	<a href="#">addPageDirection メソッド</a>	○	○	×	×
	<a href="#">addPagePosition メソッド</a>	○	○	×	×
	<a href="#">addPageLine メソッド</a>	×	×	×	×
	<a href="#">addPageRectangle メソッド</a>	×	×	×	×
カット	<a href="#">addCut メソッド</a>	○	×	×	×
ドロアー	<a href="#">addPulse メソッド</a>	○	○	○	○
ブザー	<a href="#">addSound メソッド</a>	×	×	×	×
レイアウト	<a href="#">addLayout メソッド</a>	×	×	×	×
コマンド送信	<a href="#">addCommand メソッド</a>	○	○	○	○

## Scanner オブジェクト

バーコードスキャナーによるバーコード読み取りを制御します。

API		説明
データ検出	<a href="#">ondata イベント</a>	バーコードデータ検出イベント

## SimpleSerial オブジェクト

プリンターとデバイスのシリアル通信を制御します。



デバイス制御スクリプトの作成により、カスタマイズできます。詳細は、「第4章 デバイス制御スクリプト」を参照してください。

API		説明
文字列設定	<a href="#">sendCommand</a>	任意コマンドの送信
応答検出	<a href="#">oncommandreply イベント</a>	任意コマンドの送信結果通知イベント

## DeviceHubTerminal オブジェクト

TM-DT シリズを制御します。

API		説明
シャットダウン	<a href="#">shutdown メソッド</a>	プリンターをシャットダウン
再起動	<a href="#">restart メソッド</a>	プリンターを再起動

## CommBoxManager オブジェクト

コミュニケーションボックスのオープン / クローズを制御します。

API		説明
アプリケーション間通信	<a href="#">openCommBox メソッド</a>	コミュニケーションボックスをオープン
	<a href="#">closeCommBox メソッド</a>	コミュニケーションボックスをクローズ

## CommBox オブジェクト

コミュニケーションボックスとアプリケーション間のデータ送受信を制御します。

API		説明
送信履歴	<a href="#">getCommHistory メソッド</a>	データの送信履歴を取得
送信	<a href="#">send メソッド</a>	コミュニケーションボックスにデータを送信
受信	<a href="#">onreceive イベント</a>	コミュニケーションボックスのデータ受信を通知

## ePosDeviceConfiguration オブジェクト

プリンターに登録されているデバイスの情報を取得します。

API		説明
コンストラクター	<a href="#">Constructor</a>	ePosDeviceConfiguration オブジェクトを作成
状態取得	<a href="#">getRegisteredDevices メソッド</a>	デバイスの利用可能状況を取得

# ePOSDevice オブジェクト

## Constructor

ePOSDevice オブジェクトを新しく作成し、初期化します。

### 構文

```
ePOSDevice ();
```

### 戻り値

戻り値	オブジェクト種類
ePOSDevice オブジェクト	ePOSDevice

## connect メソッド

プリンターとの通信を開始します。

### 構文

- ❑ `connect(ipAddress, port, callback);`
- ❑ `connect(ipAddress, port, callback, options);`

### パラメーター

#### ipAddress

設定値	説明
ipAddress	プリンターの IP アドレスを指定

#### port

通信方式を指定します。

設定値	説明
8008	HTTP 通信
8043	SSL 通信

#### callback

実行結果を受け取るコールバック関数を指定します。

第 1 パラメーターに結果を示す以下の文字列が渡されます。

文字列	説明
"OK"	接続が成功した
"SSL_CONNECT_OK"	接続が成功した (SSL 通信)
"ERROR_TIMEOUT"	タイムアウトが発生した
"ERROR_PARAMETER"	パラメーターエラーが発生した

#### options

連想配列を用いてプロパティを指定します。

プロパティ	説明	true/false
eposprint	ePOS-Print オプションを有効にするかどうかを指定します。	true : 有効にする false : 無効にする

### 補足説明

options を指定する構文の機能については、「第 7 章 活用ガイド」を参照してください。

## disconnect メソッド

プリンターとの通信を終了します。

---

### 構文

```
disconnect ( ) ;
```

---

### 補足説明

`ondisconnect` イベント発生時、本 API の呼び出しは不要です。

## isConnected メソッド

`connect` メソッドによる通信状態を取得します。

### 構文

```
isConnected();
```

### 戻り値

戻り値	説明
true	接続済み
false	未接続

### 補足説明

端末がスリープした場合、戻り値に正しい値を返さない場合があります。

## createDevice メソッド

デバイスのインターフェイスとなるデバイスオブジェクトを取得します。取得したデバイスオブジェクトは、コールバック関数に渡されます。

### 構文

- ❑ `createDevice(deviceId, deviceType, options, callback);`
- ❑ `createDevice(deviceId, deviceType, crypto, callback);`

### パラメーター

#### deviceId

設定値	説明
deviceId	デバイス ID を文字列で指定

#### deviceType

設定値	デバイスの種類
DEVICE_TYPE_CASH_CHANGER	自動釣り銭機
DEVICE_TYPE_DISPLAY	カスタマーディスプレイ
DEVICE_TYPE_KEYBOARD	キーボード
DEVICE_TYPE_MSR	MSR
DEVICE_TYPE_PRINTER	プリンター
DEVICE_TYPE_SCANNER	バーコードスキャナー
DEVICE_TYPE_SIMPLE_SERIAL	シリアル通信デバイス
DEVICE_TYPE_DT	TM-DT シリーズ

#### options

連想配列を用いてプロパティを指定します。

プロパティ	説明	true/false
crypto (Boolean)	通信データを暗号化するかどうかを指定します。	true : 暗号化する false : 暗号化しない
buffer (Boolean)	デバイスとの再接続処理中に発生したデータを、再接続後に送信するかどうかを指定します。	true : 送信する false : 送信しない

## crypto

プロパティ	説明	true/false
crypto (Boolean)	通信データを暗号化するかどうかを指定します。	true : 暗号化する false : 暗号化しない

## callback

実行結果を受け取るコールバック関数を指定します。

第 1 パラメーターにデバイスオブジェクト、第 2 パラメーターには、結果を示す以下の文字列が渡されます。

文字列	説明
"OK"	デバイスオブジェクトの取得に成功した
"DEVICE_NOT_FOUND"	デバイスが見つからない
"DEVICE_IN_USE"	デバイスが使用中
"DEVICE_OPEN_ERROR"	デバイスのオープンに失敗した
"DEVICE_TYPE_INVALID"	デバイスの種類が違う
"PARAM_ERROR"	パラメーターエラーが発生した
"SYSTEM_ERROR"	その他のエラーが発生した

## 補足説明

- ❑ createDevice メソッドの実行に成功すると、プリンターを除くデバイスは排他的にロックされます。
- ❑ deviceType に DEVICE\_TYPE\_DISPLAY を指定する場合は、deviceId に local\_display を指定します。
- ❑ deviceType に DEVICE\_TYPE\_DT を指定する場合は、deviceId に local\_dt を指定します。
- ❑ デバイスオブジェクトが取得できなかった場合、第 1 パラメーターにデバイスオブジェクトは渡されず null が渡されます。
- ❑ 排他的にロックされたデバイスを指定して createDevice メソッドを呼び出すと、DEVICE\_IN\_USE が返ります。
- ❑ [ondisconnect イベント](#)発生時の復帰処理の際、createDevice メソッドを実行するタイミングによって DEVICE\_IN\_USE が発生することがあります。この場合、DEVICE\_IN\_USE 以外が取得できるまで、繰り返し createDevice メソッドを実行してください。

deleteDevice メソッド

createDevice メソッドで取得したデバイスオブジェクトを破棄します。破棄したデバイスオブジェクトが制御していたデバイスは解放され、他のアプリケーションから使用可能になります。

構文

```
deleteDevice(deviceObject, callback);
```

パラメーター

deviceObject

設定値	説明
deviceObject	破棄するデバイスオブジェクトを指定

callback

実行結果を受け取るコールバック関数を指定します。  
第 1 パラメーターに結果を示す以下の文字列が渡されます。

文字列	説明
"OK"	デバイスのクローズに成功した
"DEVICE_NOT_OPEN"	デバイスがオープンされていない
"DEVICE_CLOSE_ERROR"	デバイスのクローズに失敗した
"SYSTEM_ERROR"	その他のエラーが発生した

## getAdmin メソッド

TM-DT ソフトウェア / TM-i ファームウェアに設定された管理者名を取得します。

### 構文

```
getAdmin();
```

### 戻り値

戻り値	オブジェクト種類
管理者名の文字列	String

### 補足説明

管理者名は EPSON TMNet WebConfig で設定します。

## getLocation メソッド

TM-DT ソフトウェア / TM-i ファームウェアに設定された設置場所情報を取得します。

### 構文

```
getLocation();
```

### 戻り値

戻り値	オブジェクト種類
設置場所情報の文字列	String

### 補足説明

設置場所情報は EPSON TMNet WebConfig で設定します。

## sendOfscXml メソッド

OFSC-Print サービスに印刷データ（XML データ）を送信します。

### 構文

```
sendOfscXml(xml, timeout, crypto, callback);
```

### パラメーター

#### xml

設定値	説明
xml	OFSC-Print 用 XML データを指定

#### timeout

設定値	説明
timeout	印刷時のタイムアウト時間を msec 単位で指定

#### crypto

設定値	説明
true	通信データを暗号化する
false	通信データを暗号化しない

#### callback

実行結果を受け取るコールバック関数を指定します。

設定値	説明
callback	実行結果 XML データ

### 補足説明

印刷データ（XML データ）の詳細は、OFSC-Print ユーザーズマニュアルを参照してください。

## getCommBoxManager メソッド

アプリケーション間通信を行う [CommBoxManager](#) オブジェクトを取得します。

---

### 構文

```
getCommBoxManager ( ) ;
```

---

### 戻り値

戻り値	オブジェクト種類
CommBoxManager オブジェクト	CommBoxManager

## onreconnecting イベント

ネットワークの再接続処理が開始されたことを受け取ります。onreconnecting イベントが発生した際に、再接続が開始された旨を示すメッセージを表示するなどの処理を行ってください。

---

### 構文

```
Function()
```

## onreconnect イベント

再接続処理が成功したことを受け取ります。onreconnect イベントが発生した際に、再接続処理中に表示していたメッセージを消すなどの処理を行ってください。

---

### 構文

```
Function()
```

## ondisconnect イベント

接続処理に失敗したことを受け取ります。ondisconnect イベントは、以下の場合に発生します。

- ❑ [disconnect メソッド](#)の実行時
- ❑ [onreconnecting イベント](#)発生後の接続処理失敗時

ondisconnect イベントが発生した際に、再度接続処理行うなど、ネットワーク切断対策の処理を行ってください。

---

### 構文

```
Function()
```

# デバイスオブジェクト共通

## callEvent メソッド

デバイス制御スクリプト側の任意イベントを指定し、プロパティ名と値のセットからなる JSON データを渡します。

詳細は、「第4章 デバイス制御スクリプト」を参照してください。

### 構文

```
callEvent(eventName, data);
```

### パラメーター

#### eventName

設定値	説明
String	デバイス制御スクリプト側のイベント名を指定

#### data

設定値	説明
Object	イベントに渡すデータを指定

# CashChanger オブジェクト

## setConfig メソッド

自動釣り銭機の設定を変更します。setConfig メソッドの実行結果は、[onconfigchange イベント](#)のコールバック関数に渡されます。

### 構文

```
setConfig(config, data);
```

### パラメーター

#### config

設定を変更する項目を指定します。

設定値	説明
CONFIG_COUNT_MODE	計数モード
CONFIG_LEFT_CASH	残置指定

#### data

config で指定した項目のプロパティを指定します。

設定項目	プロパティ	説明	値
計数モード	mode	計数モードを設定	MODE_MANUAL_INPUT : 手入力モード MODE_AUTO_COUNT : 自動計数モード
残置指定	coins	硬貨の残置金額を 10 円単位の文字列で指定	< 例 : 10,000 円の場合 > "1000"
	bills	紙幣の残置金額を千円単位の文字列で指定	< 例 : 50,000 円の場合 > "50"

## readCashCount メソッド

機器内の金種、枚数情報を取得します。readCashCount メソッドの実行結果は、[oncashcount イベント](#)と [onstatuschange イベント](#)のコールバック関数に渡されます。

---

### 構文

```
readCashCount ( ) ;
```

## beginDeposit メソッド

入金処理を開始します。beginDeposit メソッドの実行結果は、[ondeposit イベント](#)と [onstatuschange イベント](#)のコールバック関数に渡されます。

---

### 構文

```
beginDeposit();
```

## pauseDeposit メソッド

入金処理を一時停止します。pauseDeposit メソッドの実行結果は、[ondeposit イベント](#)と [onstatuschange イベント](#)のコールバック関数に渡されます。

---

### 構文

```
pauseDeposit();
```

## restartDeposit メソッド

入金処理を再開します。restartDeposit メソッドの実行結果は、[ondeposit イベント](#)と [onstatuschange イベント](#)のコールバック関数に渡されます。

---

### 構文

```
restartDeposit();
```

## endDeposit メソッド

入金処理を終了します。endDeposit メソッドの実行結果は、[ondeposit イベント](#)と [onstatuschange イベント](#) のコールバック関数に渡されます。

### 構文

```
endDeposit (config);
```

### パラメーター

#### config

終了動作を指定します。

設定値	説明
DEPOSIT_CHANGE	釣り銭あり（収納部への収納を保留する）
DEPOSIT_NOCHANGE	釣り銭なし（収納部に収納する）
DEPOSIT_REPAY	預かり金返却

dispenseChange メソッド

紙幣 / 硬貨の枚数を指定して出金します。dispenseChange メソッドの実行結果は、ondispense イベントと onstatuschange イベントのコールバック関数に渡されます。

構文

```
dispenseChange (data) ;
```

パラメーター

data

プロパティ	紙幣 / 硬貨の種類	値
jpy1	1 円硬貨	文字列で指定
jpy5	5 円硬貨	
jpy10	10 円硬貨	
jpy50	50 円硬貨	
jpy100	100 円硬貨	
jpy500	500 円硬貨	
jpy1000	1,000 円紙幣	
jpy2000	2,000 円紙幣	
jpy5000	5,000 円紙幣	
jpy10000	10,000 円紙幣	

## dispenseCash メソッド

指定金額で出金します。dispenseCash メソッドの実行結果は、[ondispense イベント](#)と [onstatuschange イベント](#)のコールバック関数に渡されます。

### 構文

```
dispenseCash(cash);
```

### パラメーター

cash

設定値	説明
cash	出金する金額を文字列で指定

## collectCash メソッド

機器内の現金を回収します。collectCash メソッドの実行結果は、[oncollect イベント](#)と [oncommandreply イベント](#)、[onstatuschange イベント](#)のコールバック関数に渡されます。

### 構文

```
collectCash(type);
```

### パラメーター

type

設定値	説明
COLLECT_ALL_CASH	全額回収
COLLECT_PART_OF_CASH	残置回収

## openDrawer メソッド

キャッシュドローア接続時に、キャッシュドローアを開きます。openDrawer メソッドの実行結果は、[onstatuschange イベント](#)のコールバック関数に渡されます。

---

### 構文

```
openDrawer ( ) ;
```

## sendCommand メソッド

任意コマンドを送信します。sendCommand メソッドの実行結果は、[onstatuschange イベント](#)のコールバック関数に渡されます。

---

### 構文

```
sendCommand (data) ;
```

---

### パラメーター

data

設定値	説明
data	コマンドを文字列で指定

## onconfigchange イベント

setConfig メソッドによる設定変更の結果を受け取ります。

### 構文

```
Function(data);
```

### コールバック関数のパラメーター

パラメーター：data

オブジェクト種類：Object

### data オブジェクトのプロパティ

プロパティ	説明	オブジェクト種類
status	設定変更の実行結果	String

#### status

ステータス	説明
"SUCCESS"	取得成功
"BUSY"	動作中のため取得できない
"DISCREPANCY"	金額不一致の可能性あり
"DEVICE_ERROR"	デバイスエラー
"SYSTEM_ERROR"	システムエラー

## oncashcount イベント

`readCashCount` メソッドが取得した情報を受け取ります。

### 構文

```
Function(data);
```

### コールバック関数のパラメーター

パラメーター：data

オブジェクト種類：Object

### data オブジェクトのプロパティ

プロパティ	説明	オブジェクト種類
status	取得実行結果	String
jpy1	1 円の枚数	String
jpy5	5 円の枚数	String
jpy10	10 円の枚数	String
jpy50	50 円の枚数	String
jpy100	100 円の枚数	String
jpy500	500 円の枚数	String
jpy1000	1,000 円の枚数	String
jpy2000	2,000 円の枚数	String
jpy5000	5,000 円の枚数	String
jpy10000	10,000 円の枚数	String

### status

ステータス	説明
"SUCCESS"	取得成功
"BUSY"	動作中のため取得できない
"DISCREPANCY"	金額不一致の可能性あり
"DEVICE_ERROR"	デバイスエラー
"SYSTEM_ERROR"	システムエラー

## ondeposit イベント

以下のメソッドの実行結果を受け取ります。

- [beginDeposit メソッド](#)
- [pauseDeposit メソッド](#)
- [restartDeposit メソッド](#)
- [endDeposit メソッド](#)

### 構文

```
Function(data);
```

### コールバック関数のパラメーター

パラメーター：data

オブジェクト種類：Object

### data オブジェクトのプロパティ

プロパティ	説明	オブジェクト種類
status	入金処理実行結果	String
amount	入金金額	String
jpy1	1 円の枚数	String
jpy5	5 円の枚数	String
jpy10	10 円の枚数	String
jpy50	50 円の枚数	String
jpy100	100 円の枚数	String
jpy500	500 円の枚数	String
jpy1000	1,000 円の枚数	String
jpy2000	2,000 円の枚数	String
jpy5000	5,000 円の枚数	String
jpy10000	10,000 円の枚数	String

### status

ステータス	説明
"SUCCESS"	入金成功
"BUSY"	計数処理の動作中
"PAUSE"	計数処理の一時停止
"END"	計数処理の停止
"CASH_IN_TRAY_ERROR"	現金抜き取り待ち

ステータス	説明
"DEVICE_ERROR"	デバイスエラー
"SYSTEM_ERROR"	システムエラー

## ondispense イベント

以下のメソッドの実行結果を受け取ります。

- [dispenseChange メソッド](#)
- [dispenseCash メソッド](#)

### 構文

```
Function (data) ;
```

### コールバック関数のパラメーター

パラメーター：data

オブジェクト種類：Object

### data オブジェクトのプロパティ

プロパティ	説明	オブジェクト種類
status	出金処理実行結果	String

#### status

ステータス	説明
"SUCCESS"	出金の成功
"BUSY"	装置動作中のため出金不可
"SHORTAGE"	現金不足による失敗
"CASH_IN_TRAY_ERROR"	現金抜き取り待ち
"DEVICE_ERROR"	デバイスエラー
"SYSTEM_ERROR"	システムエラー

## oncollect イベント

collectCash メソッドの実行結果を受け取ります。

### 構文

```
Function(data);
```

### コールバック関数のパラメーター

パラメーター：data

オブジェクト種類：Object

### data オブジェクトのプロパティ

プロパティ	説明	オブジェクト種類
status	collectCash メソッド実行結果	String

### status

ステータス	説明
"SUCCESS"	出金の成功
"BUSY"	装置動作中のため出金不可
"SHORTAGE"	出金に成功したが、装置内現金が空、もしくは空に近い状態
"CASH_IN_TRAY_ERROR"	現金抜き取り待ち
"DEVICE_ERROR"	デバイスエラー
"SYSTEM_ERROR"	システムエラー

## oncommandreply イベント

[collectCash メソッド](#)の実行結果を受け取ります。

### 構文

```
Function(data);
```

### コールバック関数のパラメーター

パラメーター：data

オブジェクト種類：Object

### data オブジェクトのプロパティ

プロパティ	説明	オブジェクト種類
data	応答データ文字列	String

## onstatuschange イベント

以下のメソッド実行時に収納庫ステータスの変化を受け取ります。

- ❑ [readCashCount](#) メソッド
- ❑ [beginDeposit](#) メソッド
- ❑ [pauseDeposit](#) メソッド
- ❑ [restartDeposit](#) メソッド
- ❑ [endDeposit](#) メソッド
- ❑ [dispenseChange](#) メソッド
- ❑ [dispenseCash](#) メソッド
- ❑ [collectCash](#) メソッド
- ❑ [openDrawer](#) メソッド
- ❑ [sendCommand](#) メソッド

### 構文

```
Function(data) ;
```

### コールバック関数のパラメーター

パラメーター：data

オブジェクト種類：Object

### data オブジェクトのプロパティ

プロパティ	説明	オブジェクト種類
status	入金処理実行結果	String
st1	1 円収納庫ステータス	String
st5	5 円収納庫ステータス	String
st10	10 円収納庫ステータス	String
st50	50 円収納庫ステータス	String
st100	100 円収納庫ステータス	String
st500	500 円収納庫ステータス	String
st1000	1,000 円収納庫ステータス	String
st2000	2,000 円収納庫ステータス	String
st5000	5,000 円収納庫ステータス	String
st10000	10,000 円収納庫ステータス	String

**status**

ステータス	説明
"OK"	正常
"DISCREPANCY"	金額不一致の可能性あり

**st1 ~ st10000**

ステータス	説明
"EMPTY"	空
"NEAREMPTY"	空に近い状態
"OK"	適量
"NEARFULL"	満杯に近い状態
"FULL"	満杯

# Display オブジェクト

## createWindow メソッド

ウィンドウ設定を命令バッファに追加します。  
ウィンドウは最大で 4 つまで設定可能です。

### 構文

```
createWindow(number, x, y, width, height, scrollMode);
```

### パラメーター

number

設定値	説明
1 ~ 4	ウィンドウ番号を指定

x

設定値	説明
1 ~ 20	x 座標原点位置を指定

y

設定値	説明
1 ~ 2	y 座標原点位置を指定

width

設定値	説明
1 ~ 20	ウィンドウの幅を指定

height

設定値	説明
1 ~ 2	ウィンドウの高さを指定

scrollMode

設定値	説明
SCROLL_OVERWRITE	<ul style="list-style-type: none"> <li>表示位置が上段右端の時に文字表示すると、表示位置を下段左端に移動する。</li> <li>表示位置が下段右端の時に文字表示すると、表示位置を上段左端に移動する。</li> <li>表示位置の移動先に表示されている文字は、上書きされる。</li> </ul>

設定値	説明
SCROLL_VERTICAL	<ul style="list-style-type: none"><li>表示位置が上段右端の時に文字表示すると、表示位置を下段左端に移動する。</li><li>表示位置が下段右端の時に文字表示すると、下段の表示文字を上段へ移動させて下段の表示文字を消去し、表示位置を下段左端へ移動する。</li></ul>
SCROLL_HORIZONTAL	<ul style="list-style-type: none"><li>表示位置が右端の時に文字表示すると、行全体を 1 文字分左へ移動させ、右端に新しく文字を表示する。</li><li>改行は行わない。</li></ul>

戻り値

戻り値	オブジェクト種類
Display オブジェクト	Display

補足説明

追加でウィンドウを設定する際は、設定済みのウィンドウとカスタマーディスプレイ上の設定範囲が重ならないように注意してください。

## destroyWindow メソッド

[createWindow メソッド](#)で設定した任意のウィンドウ番号を指定し、指定したウィンドウの破棄設定を命令バッファーに追加します。

### 構文

```
destroyWindow(number);
```

### パラメーター

number

設定値	説明
1 ~ 4	ウィンドウ番号を指定

### 戻り値

戻り値	オブジェクト種類
Display オブジェクト	Display

## setCurrentWindow メソッド

カレントウィンドウの変更設定を命令バッファーに追加します。

指定ウィンドウをカレントウィンドウとし、カーソルをカレントウィンドウの原点位置に移動させます。

### 構文

```
setCurrentWindow (number) ;
```

### パラメーター

number

設定値	説明
1 ~ 4	ウィンドウ番号を指定

### 戻り値

戻り値	オブジェクト種類
Display オブジェクト	Display

## clearWindow メソッド

カレントウィンドウ内の表示消去を命令バッファに追加します。

### 構文

```
clearWindow();
```

### 戻り値

戻り値	オブジェクト種類
Display オブジェクト	Display

## setCursorPosition メソッド

座標を指定し、カーソル位置設定を命令バッファに追加します。

### 構文

```
setCursorPosition(x, y);
```

### パラメーター

x

設定値	説明
1 ~ 20	x 座標を指定

y

設定値	説明
1 ~ 2	y 座標を指定

### 戻り値

戻り値	オブジェクト種類
Display オブジェクト	Display

## moveCursorPosition メソッド

カレントウィンドウ内でのカーソル移動先を指定し、カーソルの位置設定を命令バッファに追加します。

### 構文

```
moveCursorPosition(position);
```

### パラメーター

position

設定値	説明
MOVE_TOP_LEFT	上段左端へ移動
MOVE_TOP_RIGHT	上段右端へ移動
MOVE_BOTTOM_LEFT	下段左端へ移動
MOVE_BOTTOM_RIGHT	下段右端へ移動

### 戻り値

戻り値	オブジェクト種類
Display オブジェクト	Display

setCursorType メソッド

カーソルの表示設定を命令バッファに追加します。

構文

```
setCursorType (type) ;
```

パラメーター

type

設定値	説明
CURSOR_NONE	表示なし
CURSOR_UNDERLINE	アンダーライン表示

戻り値

戻り値	オブジェクト種類
Display オブジェクト	Display

addText メソッド

カスタマーディスプレイに表示する文字の表示設定を命令バッファーに追加します。

構文

- ❑ addText (data) ;
- ❑ addText (data, lang) ;
- ❑ addText (data, x, y) ;
- ❑ addText (data, x, y, lang) ;

パラメーター

data

設定値	説明
data	表示文字を文字列で指定

lang

設定値	説明
"en" (初期値)	英語
"ja"	日本語 (カナ)

x

設定値	説明
1 ~ 20	表示位置の x 座標を指定

y

設定値	説明
1 ~ 2	表示位置の y 座標を指定

戻り値

戻り値	オブジェクト種類
Display オブジェクト	Display

補足説明

表示位置の座標指定を省略すると、addText メソッド実行時のカーソル位置座標が指定されます。

## addReverseText メソッド

カスタマーディスプレイに表示する白黒反転文字の表示設定を命令バッファーに追加します。

### 構文

- `addReverseText (data) ;`
- `addReverseText (data, lang) ;`
- `addReverseText (data, x, y) ;`
- `addReverseText (data, x, y, lang) ;`

### パラメーター

data

設定値	説明
data	表示文字を文字列で指定

lang

設定値	説明
"en" (初期値)	英語
"ja"	日本語 (加)

x

設定値	説明
1 ~ 20	表示位置の x 座標を指定

y

設定値	説明
1 ~ 2	表示位置の y 座標を指定

### 戻り値

戻り値	オブジェクト種類
Display オブジェクト	Display

### 補足説明

表示位置の座標指定を省略すると、addReverseText メソッド実行時のカーソル位置座標が指定されます。

addMarquee メソッド

マーキー表示設定を命令バッファに追加します。  
マーキー表示設定は、createWindow メソッドで指定したウィンドウのスクロールモードの影響を受けず、1 行の文字列を水平スクロールモードで表示します。

構文

```
addMarquee(data, format, unitWait, repeatWait, repeatCount, lang);
```

パラメーター

data

設定値	説明
data	表示文字を文字列で指定

format

設定値	説明
MARQUEE_WALK	ウィンドウ右端から文字を表示
MARQUEE_PLACE	ウィンドウ左端から文字を表示

unitWait

設定値	説明
0 ~ 2000	1 文字ごとの表示間隔を msec 単位で指定

repeatWait

設定値	説明
100 ~ 2000	文字の繰り返し間隔を msec 単位で指定

repeatCount

設定値	説明
0 ~ 127	文字の繰り返し回数を指定

lang

設定値	説明
"en" (初期値)	英語
"ja"	日本語 (カナ)

---

## 戻り値

戻り値	オブジェクト種類
Display オブジェクト	Display

---

## 補足説明

repeatCount に 0 を指定すると無制限に繰り返します。

## setBlink メソッド

カスタマーディスプレイの点滅設定を命令バッファに追加します。

### 構文

```
setBlink(interval);
```

### パラメーター

interval

設定値	説明
0 ~ 12700	点滅時間を msec 単位で指定

### 戻り値

戻り値	オブジェクト種類
Display オブジェクト	Display

### 補足説明

- 0 を指定すると点灯状態になります。
- 設定値は 50 msec 単位で切り上げられます。  
例：1 ⇒ 50、51 ⇒ 100、101 ⇒ 150

## setBrightness メソッド

カスタマーディスプレイの輝度設定を命令バッファーに追加します。

### 構文

```
setBrightness(brightness);
```

### パラメーター

brightness

設定値	説明
BRIGHTNESS_20	輝度値を 20%に設定
BRIGHTNESS_40	輝度値を 40%に設定
BRIGHTNESS_60	輝度値を 60%に設定
BRIGHTNESS_100	輝度値を 100%に設定

### 戻り値

戻り値	オブジェクト種類
Display オブジェクト	Display

## showClock メソッド

時刻の表示を命令バッファに追加します。

時刻表示を実行すると、ウィンドウ内に表示中の文字はすべて消去されます。また、時刻表示中に他の命令を送信すると、時刻表示を終了します。

---

### 構文

```
showClock();
```

---

### 戻り値

戻り値	オブジェクト種類
Display オブジェクト	Display

## addCommand メソッド

ESC/POS コマンドを命令バッファに追加します。

### 構文

```
addCommand (data) ;
```

### パラメーター

data

設定値	説明
data	ESC/POS コマンドを文字列で指定

### 戻り値

戻り値	オブジェクト種類
Display オブジェクト	Display

### 補足説明

カスタマーディスプレイの ESC/POS コマンド情報の入手方法はエプソンのホームページをご覧ください。  
<http://partner.epson.jp/support/details/contents038/>

## send メソッド

命令バッファー内のデータをプリンターへ送信します。send メソッドの実行結果は、[onreceive イベント](#)のコールバック関数に渡されます。

---

### 構文

```
send();
```

reset メソッド

カスタマーディスプレイの初期化設定を命令バッファに追加します。

reset メソッドの実行後は、以下の状態になります。

- 表示文字や登録したウィンドウは全て破棄されます。
- カーソルの設定は初期状態に戻り、カーソルの位置はカスタマーディスプレイ原点へ移動します。
- 表示点滅と表示輝度の設定が初期状態に戻ります。

構文

```
reset();
```

戻り値

戻り値	オブジェクト種類
Display オブジェクト	Display

## onreceive イベント

`send` メソッドで送信した命令の実行結果を受け取ります。

### 構文

```
Function(response);
```

### コールバック関数のパラメーター

パラメーター：response

オブジェクト種類：Object

### response オブジェクトのプロパティ

プロパティ	説明	オブジェクト種類
success	命令の実行結果	Boolean
code	エラーコード	String

#### success

値	説明
true	処理に成功
false	処理に失敗

#### code

値	説明
'EDSP_NOT_FOUND'	デバイスが見つからない
'EDSP_NOT_OPEN'	デバイスのオープンに失敗した
'EDSP_INVALID_WINDOW'	未登録のウィンドウが指定された
'EX_BADPORT'	デバイスとの内部通信エラーが発生した
'EX_TIMEOUT'	デバイスとの通信でタイムアウトエラーが発生した
'EX_INVALID_VALUE'	パラメーター不正を検出した

# Keyboard オブジェクト

## setPrefix メソッド

キーボード入力を文字列としてまとめて受け取る際、文字列の始まりと判断するためのキーコードを設定します。

### 構文

```
setPrefix(data);
```

### パラメーター

data

設定値	説明
data	キーコードの配列を指定

### 補足説明

指定するキーコードは、[Key Code 一覧](#)を参照してください。

## onkeypress イベント

キー押下イベントを受け取ります。

### 構文

```
Function(data);
```

### コールバック関数のパラメーター

パラメーター：data

オブジェクト種類：Object

### data オブジェクトのプロパティ

プロパティ	説明	オブジェクト種類
keyCode	キーコード	number
ascii	キーコードに対応する文字	String

### 補足説明

キーコードに対応する文字がない場合、ascii には undefined がセットされます。

## onstring イベント

[setPrefix メソッド](#)で指定したキーコードのいずれかで始まり、Enter キーを検出するまでの間に検出した入力文字を文字列として受け取ります。

### 構文

```
Function(data);
```

### コールバック関数のパラメーター

パラメーター：data

オブジェクト種類：Object

### data オブジェクトのプロパティ

プロパティ	説明	オブジェクト種類
input	検出した文字列	String
prefix	文字列の開始と判断したキーコード	number

setMSRPrefix メソッド

MSR 付きプログラマブルキーボードの使用を開始するためのキーコードを指定します。拡張メソッドである setMSRPrefix メソッドの呼び出しには、コールバックイベントを使用してください。

構文

```
callEvent('setPrefix', data);
```

パラメーター

'setPrefix'

設定値	説明
'setPrefix'	固定

data

設定値	説明
keycode	カード情報の受信開始を判断するためのキーコードを指定

戻り値

戻り値	説明	オブジェクト種類
sq	処理のシーケンス番号	String

補足説明

指定するキーコードは、[Key Code 一覧](#)を参照してください。

## ondata イベント

MSR 付きプログラマブルキーボードのカード読み取り情報を受け取ります。

### 構文

```
Function(data);
```

### コールバック関数のパラメーター

パラメーター：data

オブジェクト種類：Object

### data オブジェクトのプロパティ

プロパティ	説明	オブジェクト種類
track1	トラック 1 のデータ全体	String
track2	トラック 2 のデータ全体	String
track4	トラック 4 (JIS2 トラック) のデータ全体	String
account_number	アカウント番号	String
expiration_date	有効期限 ("YYMM" 形式)	String
surname	姓	String
first_name	名	String
middle_initial	ミドルネーム / イニシャル	String
title	タイトル	String
service_code	サービスコード	String
track1_dd	トラック 1 の任意データ	String
track2_dd	トラック 2 の任意データ	String

# MSR オブジェクト

## ondata イベント

MSR デバイスのカード読み取り情報を受け取ります。

### 構文

```
Function (data) ;
```

### コールバック関数のパラメーター

- パラメーター：data
- オブジェクト種類：Object

### data オブジェクトのプロパティ

プロパティ	説明	オブジェクト種類
track1	トラック 1 のデータ全体	String
track2	トラック 2 のデータ全体	String
track4	トラック 4 (JIS2 トラック) のデータ全体	String
account_number	アカウント番号	String
expiration_date	有効期限 ("YYMM" 形式)	String
surname	姓	String
first_name	名	String
middle_initial	ミドルネーム / イニシャル	String
title	タイトル	String
service_code	サービスコード	String
track1_dd	トラック 1 の任意データ	String
track2_dd	トラック 2 の任意データ	String

# Printer オブジェクト

## addTextAlign メソッド

印字位置設定を命令バッファーに追加します。

### 構文

```
addTextAlign (align) ;
```

### パラメーター

align

定数	説明
ALIGN_LEFT (初期値)	左揃え
ALIGN_CENTER	中央揃え
ALIGN_RIGHT	右揃え

### 戻り値

戻り値	オブジェクト種類
Printer オブジェクト	Printer
HybridPrinter オブジェクト	HybridPrinter

### 例外

例外	オブジェクト種類
Parameter " ... " is invalid	Error

### 補足説明

- ❑ addTextAlign メソッドの指定は「行の先頭」で行ってください。先頭以外で使用した場合、無効になります。
- ❑ ページモードでは機能しません。
- ❑ addTextAlign メソッドの設定は、ラスターイメージ/NV ロゴ/バーコード/2次元シンボルにも適用されません。
- ❑ ページモードでの横方向印字の設定は、[addTextPosition メソッド](#)を使用してください。

## addTextLineSpace メソッド

改行量設定を命令バッファーに追加します。

### 構文

```
addTextLineSpace (linespc) ;
```

### パラメーター

linespc

設定値	説明
0 ~ 255 の整数値	改行量をドット単位で指定

### 戻り値

戻り値	オブジェクト種類
Printer オブジェクト	Printer
HybridPrinter オブジェクト	HybridPrinter

### 例外

例外	オブジェクト種類
Parameter " ..." is invalid	Error

## addTextRotate メソッド

倒立印字設定を命令バッファーに追加します。

### 構文

```
addTextRotate (rotate) ;
```

### パラメーター

rotate

設定値	説明
true	倒立印字を設定
false (初期値)	倒立印字設定を解除

### 戻り値

戻り値	オブジェクト種類
Printer オブジェクト	Printer
HybridPrinter オブジェクト	HybridPrinter

### 例外

例外	オブジェクト種類
Parameter " ..." is invalid	Error

### 補足説明

- ❑ addTextRotate メソッドの指定は「行の先頭」で行ってください。先頭以外で使用した場合、無効になります。
- ❑ ページモードでは機能しません。
- ❑ addTextRotate メソッドの設定は、バーコード / 2 次元シンボルにも適用されます。
- ❑ ページモードでの倒立印字の設定は、[addPageDirection メソッド](#)を使用してください。

## addText メソッド

印字する文字を命令バッファに追加します。

### 構文

```
addText (data) ;
```

### パラメーター

data

設定値	説明
String	印字する文字を文字列で指定

水平タブと改行、バックスラッシュは、以下のエスケープシーケンスを使用してください。

文字列	説明
\t	水平タブ (HT)
\n	改行 (LF)
\\	バックスラッシュ

### 戻り値

戻り値	オブジェクト種類
Printer オブジェクト	Printer
HybridPrinter オブジェクト	HybridPrinter

### 例外

例外	オブジェクト種類
Parameter " ... " is invalid	Error

### 補足説明

- ❑ addText メソッドによる文字の印字後に、文字以外を印字するには、改行または紙送りを実行してください。
- ❑ ページモードでは、文字のベースラインドット（[機器仕様](#)）を基準に現在の印字位置へ印字文字を展開します。

## addTextLang メソッド

言語設定を命令バッファーに追加します。  
addText メソッドで指定した文字列の言語を設定します。

### 構文

```
addTextLang (lang) ;
```

### パラメーター

lang

設定値	説明
en (初期値)	英語 (ANK 仕様)
de	ドイツ語 (ANK 仕様)
fr	フランス語 (ANK 仕様)
it	イタリア語 (ANK 仕様)
es	スペイン語 (ANK 仕様)
ja	日本語 (国際文字セットも日本語に変更されます。)
ja-jp	
ko	韓国語 (国際文字セットも韓国語に変更されます。)
ko-kr	
zh-hans	簡体語 (国際文字セットも中国語に変更されます。)
zh-cn	
zh-hant	繁体語
zh-tw	
th	タイ語 (南アジアモデルのみ)
上記以外の言語コード	英語 (ANK 仕様)

指定した言語によって、一部の文字は以下のように印字されます。

言語	Characters \$ (U+0024)	Characters \ (U+005C)
日本語	\$	¥
韓国語	\$	₩
簡体語	¥	\
繁体語	\$	\

---

## 戻り値

戻り値	オブジェクト種類
Printer オブジェクト	Printer
HybridPrinter オブジェクト	HybridPrinter

---

## 例外

例外	オブジェクト種類
Parameter "... " is invalid	Error

---

## 補足説明

- ❑ プリンターに搭載されていない文字は印字できません。
- ❑ 印字できる文字コード及び国際文字セットについては、各プリンターの詳細取扱説明書を参照してください。

## addTextFont メソッド

文字のフォント設定を命令バッファに追加します。

### 構文

```
addTextFont (font) ;
```

### パラメーター

font

定数	説明
FONT_A (初期値)	フォント A
FONT_B	フォント B
FONT_C	フォント C
FONT_D	フォント D
FONT_E	フォント E
FONT_SPECIAL_A	特殊フォント A
FONT_SPECIAL_B	特殊フォント B

### 戻り値

戻り値	オブジェクト種類
Printer オブジェクト	Printer
HybridPrinter オブジェクト	HybridPrinter

### 例外

例外	オブジェクト種類
Parameter " ..." is invalid	Error

## addTextSmooth メソッド

スムージング設定を命令バッファーに追加します。

### 構文

```
addTextSmooth (smooth) ;
```

### パラメーター

smooth

設定値	説明
true	スムージングを設定
false (初期値)	スムージング設定を解除

### 戻り値

戻り値	オブジェクト種類
Printer オブジェクト	Printer
HybridPrinter オブジェクト	HybridPrinter

### 例外

例外	オブジェクト種類
Parameter "... " is invalid	Error

## addTextDouble メソッド

文字の倍角設定を命令バッファに追加します。

### 構文

```
addTextDouble(dw, dh);
```

### パラメーター

dw

設定値	説明
true	横倍角を設定
false (初期値)	横倍角設定を解除
undefined	設定を変更しない

dh

設定値	説明
true	縦倍角を設定
false (初期値)	縦倍角設定を解除
undefined	設定を変更しない

### 戻り値

戻り値	オブジェクト種類
Printer オブジェクト	Printer
HybridPrinter オブジェクト	HybridPrinter

### 例外

例外	オブジェクト種類
Parameter " ..." is invalid	Error

### 補足説明

4 倍角文字は、dw と dh に true を指定します。

## addTextSize メソッド

文字の倍率設定を命令バッファに追加します。

### 構文

```
addTextSize(width, height);
```

### パラメーター

#### width

設定値	説明
1 ~ 8 の整数値（初期値：1）	横方向の倍率を指定
undefined	設定を変更しない

#### height

設定値	説明
1 ~ 8 の整数値（初期値：1）	縦方向の倍率を指定
undefined	設定を変更しない

### 戻り値

戻り値	オブジェクト種類
Printer オブジェクト	Printer
HybridPrinter オブジェクト	HybridPrinter

### 例外

例外	オブジェクト種類
Parameter "... " is invalid	Error

## addTextStyle メソッド

文字の装飾設定を命令バッファに追加します。

### 構文

```
addTextStyle(reverse, ul, em, color);
```

### パラメーター

#### reverse

設定値	説明
true	白黒反転文字を設定
false (初期値)	白黒反転文字設定を解除
undefined	設定を変更しない

#### ul

設定値	説明
true	アンダーラインを設定
false (初期値)	アンダーライン設定を解除
undefined	設定を変更しない

#### em

設定値	説明
true	太字を設定
false (初期値)	太字設定を解除
undefined	設定を変更しない

#### color

設定値	説明
COLOR_NONE	非印字
COLOR_1 (初期値)	第1色
COLOR_2	第2色
COLOR_3	第3色
COLOR_4	第4色
undefined	設定を変更しない

---

## 戻り値

戻り値	オブジェクト種類
Printer オブジェクト	Printer
HybridPrinter オブジェクト	HybridPrinter

---

## 例外

例外	オブジェクト種類
Parameter "... " is invalid	Error

addTextPosition メソッド

横方向の印字開始位置設定を命令バッファーに追加します。

構文

```
addTextPosition(x);
```

パラメーター

x

初期値：0

設定値	説明
0 ~ 65535 の整数値	横方向の印字開始位置をドット単位で指定

戻り値

戻り値	オブジェクト種類
Printer オブジェクト	Printer
HybridPrinter オブジェクト	HybridPrinter

例外

例外	オブジェクト種類
Parameter " ..." is invalid	Error

補足説明

- ❑ 本 API により、プリンターは『行の先頭』ではない状態になります。  
x に 0 を指定した場合でも『行の先頭』ではない状態になります。
- ❑ 本 API 実行後、[addTextAlign メソッド](#)、[addTextRotate メソッド](#) は使用できません。

## addTextVPosition メソッド

縦方向の印字開始位置設定を命令バッファーに追加します。

### 構文

```
addTextVPosition(y);
```

### パラメーター

y

初期値：21

設定値	説明
0 ~ 65535 の整数値	縦方向の印字開始位置をドット単位で指定

### 戻り値

戻り値	オブジェクト種類
Printer オブジェクト	Printer
HybridPrinter オブジェクト	HybridPrinter

### 例外

例外	オブジェクト種類
Parameter "... " is invalid	Error

### 補足説明

スタンダードモードでは機能しません。

## addFeedUnit メソッド

ドット単位の紙送り量設定を命令バッファーに追加します。

### 構文

```
addFeedUnit(unit);
```

### パラメーター

unit

設定値	説明
0 ~ 255 の整数値	紙送り量をドット単位で指定

### 戻り値

戻り値	オブジェクト種類
Printer オブジェクト	Printer
HybridPrinter オブジェクト	HybridPrinter

### 例外

例外	オブジェクト種類
Parameter "... " is invalid	Error

## addFeedLine メソッド

行単位の紙送り量設定を命令バッファーに追加します。

### 構文

```
addFeedLine(line);
```

### パラメーター

line

設定値	説明
0 ~ 255 の整数値	紙送り量を行単位で指定

### 戻り値

戻り値	オブジェクト種類
Printer オブジェクト	Printer
HybridPrinter オブジェクト	HybridPrinter

### 例外

例外	オブジェクト種類
Parameter " ..." is invalid	Error

## addFeedPosition メソッド

レシート紙とラベル紙の紙送り位置設定を命令バッファに追加します。

### 構文

```
addFeedPosition(pos);
```

### パラメーター

pos

設定値	説明
FEED_PEELING	剥離位置まで紙送り
FEED_CUTTING	カット位置まで紙送り
FEED_CURRENT_TOF	現在ラベルの頭出し位置まで紙送り
FEED_NEXT_TOF	次ラベルの頭出し位置まで紙送り

### 戻り値

戻り値	オブジェクト種類
Printer オブジェクト	Printer
HybridPrinter オブジェクト	HybridPrinter

### 例外

例外	オブジェクト種類
Parameter "... " is invalid	Error

### 補足説明

ページモードでは機能しません。

## addFeed メソッド

改行を命令バッファーに追加します。

### 構文

```
addFeed();
```

### 戻り値

戻り値	オブジェクト種類
Printer オブジェクト	Printer
HybridPrinter オブジェクト	HybridPrinter

## addImage メソッド

HTML5 Canvas で描画したイメージデータの印字を命令バッファに追加します。

HTML5 Canvas の RGBA フルカラー画像の指定範囲を、addImage メソッドのカラーモード設定と [halftone プロパティ](#)、[brightness プロパティ](#) の設定に従って、ラスターイメージデータに変換します。画像の 1 ピクセルがプリンターの 1 ドットに相当し、透明色が含まれている場合は、画像の背景を白とみなします。

### 構文

```
addImage(context, x, y, width, height, color, mode);
```

### パラメーター

#### context

設定値	説明
context	HTML5 Canvas の 2D コンテキストを指定

#### x

設定値	説明
0 ~ 65535 の整数値	印字範囲の横方向開始位置を指定

#### y

設定値	説明
0 ~ 65535 の整数値	印字範囲の縦方向開始位置を指定

#### width

設定値	説明
0 ~ 65535 の整数値	印字範囲の幅を指定

#### height

設定値	説明
0 ~ 65535 の整数値	印字範囲の高さを指定

#### color

設定値	説明
COLOR_NONE	非印字
COLOR_1 (初期値)	第 1 色
COLOR_2	第 2 色
COLOR_3	第 3 色

設定値	説明
COLOR_4	第 4 色
undefined	設定を変更しない

## mode

設定値	説明
MODE_MONO	モノクロ (2 階調)
MODE_GRAY16	多階調 (16 階調)
undefined	モノクロ (2 階調)

## 戻り値

戻り値	オブジェクト種類
Printer オブジェクト	Printer
HybridPrinter オブジェクト	HybridPrinter

## 例外

例外	オブジェクト種類
Parameter "... " is invalid	Error

## 補足説明

- ❑ 異なるドメインからダウンロードした画像が含まれた HTML5 Canvas は印字できません。  
JavaScript の同一出身ポリシーにより、セキュリティエラーが発生します。
- ❑ 多階調印字はスタンダードモードでは使用できますが、ページモードでは使用できません。
- ❑ イメージデータを高速で印字するためには、[addTextAlign メソッド](#)のパラメーター定数を ALIGN\_LEFT に指定し、addImage メソッドの width パラメーター値をプリンターの用紙幅を超えない 8 の倍数に指定してください。
- ❑ ページモードでは、イメージデータの左下ドットを基準として現在の印字位置にイメージデータを展開するため、印字位置は移動しません。

## addLogo メソッド

プリンターの NV メモリーに登録されている NV ロゴ印字を命令バッファに追加します。

NV ロゴは、あらかじめプリンターに登録しておく必要があります。NV ロゴの登録方法は各プリンターの詳細取扱説明書を参照してください。

### 構文

```
addLogo(key1, key2);
```

### パラメーター

key1

設定値	説明
0 ~ 255 の整数値	NV ロゴのキーコード 1 を指定

key2

設定値	説明
0 ~ 255 の整数値	NV ロゴのキーコード 2 を指定

### 戻り値

戻り値	オブジェクト種類
Printer オブジェクト	Printer
HybridPrinter オブジェクト	HybridPrinter

### 例外

例外	オブジェクト種類
Parameter " ..." is invalid	Error

### 補足説明

- ❑ 多階調印字はスタンダードモードでは使用できますが、ページモードでは使用できません。
- ❑ ページモードでは、NV ロゴの左下ドットを基準として、現在の印字位置に NV ロゴを展開します。

## addBarcode メソッド

バーコード印字を命令バッファに追加します。

### 構文

```
addBarcode(data, type, hri, font, width, height);
```

### パラメーター

#### data

文字列	説明
String	バーコードデータを文字列で指定

文字列で表現できないバイナリーデータは、以下のエスケープシーケンスで指定します。

文字列	説明
\xnn	コントロールコード
\\	バックスラッシュ

#### type

定数	バーコードの種類
BARCODE_UPC_A	UPC-A
BARCODE_UPC_E	UPC-E
BARCODE_EAN13	EAN13
BARCODE_JAN13	JAN13
BARCODE_EAN8	EAN8
BARCODE_JAN8	JAN8
BARCODE_CODE39	CODE39
BARCODE_ITF	ITF
BARCODE_CODABAR	CODABAR
BARCODE_CODE93	CODE93
BARCODE_CODE128	CODE128
BARCODE_GS1_128	GS1-128
BARCODE_GS1_DATABAR_OMNIDIRECTIONAL	GS1 DataBar Omnidirectional
BARCODE_GS1_DATABAR_TRUNCATED	GS1 DataBar Truncated
BARCODE_GS1_DATABAR_LIMITED	GS1 DataBar Limited
BARCODE_GS1_DATABAR_EXPANDED	GS1 DataBar Expanded

バーコードの種類	説明
UPC-A	11 桁の数字を指定した場合、チェックデジットを自動で付加します。 12 桁の数字を指定した場合、12 桁目をチェックデジットとして処理しますが、チェックデジットの検算は行いません。
UPC-E	最初の桁に 0 を指定してください。 2 ～ 6 桁目にメーカーコードを指定してください。 7 ～ 11 桁目にアイテムコードを右詰めで指定してください。アイテムコードの桁数はメーカーコードにより異なります。使用しない桁は 0 を指定してください。 11 桁の数字を指定した場合、チェックデジットを自動で付加します。 12 桁の数字を指定した場合、12 桁目をチェックデジットとして処理しますが、チェックデジットの検算は行いません。
EAN13	12 桁の数字を指定した場合、チェックデジットを自動で付加します。
JAN13	13 桁の数字を指定した場合、13 桁目をチェックデジットとして処理しますが、チェックデジットの検算は行いません。
EAN8	7 桁の数字を指定した場合、チェックデジットを自動で付加します。
JAN8	8 桁の数字を指定した場合、8 桁目をチェックデジットとして処理しますが、チェックデジットの検算は行いません。
CODE39	先頭の文字が * の場合、この文字をスタートキャラクターとして処理します。それ以外の場合、スタートキャラクターを自動で付加します。
ITF	スタートコードおよびストップコードを自動で付加します。 チェックデジットの付加および検算は行いません。
CODABAR	スタートキャラクター (A ～ D, a ～ d) を指定してください。 ストップキャラクター (A ～ D, a ～ d) を指定してください。 チェックデジットの付加および検算は行いません。
CODE93	スタートキャラクターおよびストップキャラクターを自動で付加します。 チェックデジットを計算して自動で付加します。
CODE128	スタートキャラクター (CODE A, CODE B, CODE C) を指定してください。 ストップキャラクターを自動で付加します。 チェックデジットを計算して自動で付加します。 以下の文字をエンコードするには、文字 { で始まる 2 文字を指定してください。 FNC1 : {1 FNC2 : {2 FNC3 : {3 FNC4 : {4 CODE A : {A CODE B : {B CODE C : {C SHIFT : {S { : {{

バーコードの種類	説明
GS1-128	<p>スタートキャラクター、FNC1、チェックデジット、ストップキャラクターを自動で付加します。</p> <p>アプリケーション識別子 (AI) とそれに続くデータのチェックデジットを計算して自動で付加するには、チェックデジットの位置に文字 * を指定します。</p> <p>アプリケーション識別子 (AI) を括弧で囲むことができます。括弧は HRI の印字文字として使用し、データとしてエンコードしません。</p> <p>アプリケーション識別子 (AI) とデータの間に空白を挿入することができます。空白は HRI の印字文字として使用し、データとしてエンコードしません。</p> <p>以下の文字をエンコードするには、文字 { で始まる 2 文字を指定してください。</p> <p>FNC1 : {1  FNC3 : {3  (: {(  ): }  *: {*  {: {{</p>
GS1 DataBar Omnidirectional	アプリケーション識別子 (AI) とチェックデジットを除く 13 桁の商品識別番号 (GTIN) を指定してください。
GS1 DataBar Truncated	
GS1 DataBar Limited	
GS1 DataBar Expanded	<p>アプリケーション識別子 (AI) を括弧で囲むことができます。括弧は HRI の印字文字として使用し、データとしてエンコードしません。</p> <p>以下の文字をエンコードするには、文字 { で始まる 2 文字を指定してください。</p> <p>FNC1 : {1  (: {(  ): }</p>

## hri

定数	説明
HRI_NONE (初期値)	印字しない
HRI_ABOVE	バーコードの上
HRI_BELOW	バーコードの下
HRI_BOTH	バーコードの上と下の両方

## font

定数	説明
FONT_A (初期値)	フォント A
FONT_B	フォント B
FONT_C	フォント C
FONT_D	フォント D
FONT_E	フォント E

**width**

初期値：3

設定値	説明
2 ～ 6 の整数値	1 モジュールの幅をドット単位で指定

**height**

初期値：162

設定値	説明
1 ～ 255 の整数値	バーコードの高さをドット単位で指定

**戻り値**

戻り値	オブジェクト種類
Printer オブジェクト	Printer
HybridPrinter オブジェクト	HybridPrinter

**例外**

例外	オブジェクト種類
Parameter " ..." is invalid	Error

**補足説明**

- ❑ addBarcode メソッドの指定は「行の先頭」で行ってください。
- ❑ ページモードでは、バーコードの左下ドット（HRI を除く）を基準として、現在の印字位置に展開します。
- ❑ type で指定するバーコードの種類に従った文字列を指定してください。data で指定した文字列と type で指定するバーコードの種類が一致しないとバーコードは印字されません。

## addSymbol メソッド

2次元シンボル印字を命令バッファに追加します。

### 構文

```
addSymbol(data, type, level, width, height, size);
```

### パラメーター

#### data

文字列	説明
String	2次元シンボルデータを文字列で指定

文字列で表現できないバイナリーデータは、以下のエスケープシーケンスで指定します。

文字列	説明
\xnn	コントロールコード
\\	バックスラッシュ

#### type

定数	2次元シンボルの種類
SYMBOL_PDF417_STANDARD	Standard PDF417
SYMBOL_PDF417_TRUNCATED	Truncated PDF417
SYMBOL_QRCODE_MODEL_1	QR Code Model 1
SYMBOL_QRCODE_MODEL_2	QR Code Model 2
SYMBOL_QRCODE_MICRO *2	Micro QR Code
SYMBOL_MAXICODE_MODE_2	MaxiCode Mode 2
SYMBOL_MAXICODE_MODE_3	MaxiCode Mode 3
SYMBOL_MAXICODE_MODE_4	MaxiCode Mode 4
SYMBOL_MAXICODE_MODE_5	MaxiCode Mode 5
SYMBOL_MAXICODE_MODE_6	MaxiCode Mode 6
SYMBOL_GS1_DATABAR_STACKED	GS1 DataBar Stacked
SYMBOL_GS1_DATABAR_STACKED_OMNIDIRECTIONAL	GS1 DataBar Stacked Omnidirectional
SYMBOL_GS1_DATABAR_EXPANDED_STACKED	GS1 DataBar Expanded Stacked
SYMBOL_AZTECCODE_FULLRANGE	Aztec Code Full-Range モード
SYMBOL_AZTECCODE_COMPACT	Aztec Code Compact モード
SYMBOL_DATAMATRIX_SQUARE	DataMatrix ECC200 正方形

定数	2次元シンボルの種類
SYMBOL_DATAMATRIX_RECTANGLE_8	DataMatrix ECC200 長方形 行数 8
SYMBOL_DATAMATRIX_RECTANGLE_12	DataMatrix ECC200 長方形 行数 12
SYMBOL_DATAMATRIX_RECTANGLE_16	DataMatrix ECC200 長方形 行数 16

2次元シンボルの種類	説明
Standard PDF417	文字列を UTF-8 に変換後、エスケープシーケンスの処理を行い、エンコードします。 データ領域の最大コードワード数は 928 個、1 段あたりのデータ領域の最大コードワード数は 30 個、最大段数は 90 段です。
Truncated PDF417	
QR Code Model 1	文字列をシフト JIS に変換後、エスケープシーケンスの処理を行い、データの種類を以下の中から選択してエンコードします。 数字 : 0 ~ 9 英数字 : 0 ~ 9, A ~ Z, スペース, \$, %, *, +, -, ., /, : 漢字 : シフト JIS 値 8 ビットバイトデータ : 0x00 ~ 0xff
QR Code Model 2	
Micro QR Code2	
MaxiCode Mode 2	文字列を UTF-8 に変換後、エスケープシーケンスの処理を行い、エンコードします。 モード 2 およびモード 3 は、最初のデータが 0>\x1e01\x1dyy (yy は 2 桁の数字) の場合、これをメッセージヘッダーとして処理し、次のデータからプライマリメッセージとして処理します。それ以外の場合、最初のデータからプライマリメッセージとして処理します。 モード 2 の場合、以下の形式でプライマリメッセージを指定してください。 <ul style="list-style-type: none"> <li>郵便コード (1 ~ 9 桁の数字) GS:(\x1d)</li> <li>ISO 国名コード (1 ~ 3 桁の数字) GS:(\x1d)</li> <li>サービスクラスコード (1 ~ 3 桁の数字)</li> </ul> モード 3 の場合、以下の形式でプライマリメッセージを指定してください。 <ul style="list-style-type: none"> <li>郵便コード (1 ~ 6 個のコードセット A で変換可能なデータ) GS(\x1d) ISO</li> <li>国名コード (1 ~ 3 桁の数字) GS(\x1d)</li> <li>サービスクラスコード (1 ~ 3 桁の数字)</li> </ul>
MaxiCode Mode 3	
MaxiCode Mode 4	
MaxiCode Mode 5	
MaxiCode Mode 6	
GS1 DataBar Stacked	文字列を UTF-8 に変換後、エスケープシーケンスの処理を行い、エンコードします。 アプリケーション識別子 (AI) とチェックデジットを除く 13 桁の商品識別番号 (GTIN) を指定してください。
GS1 DataBar Stacked Omnidirectional	
GS1 DataBar Expanded Stacked	文字列を UTF-8 に変換後、エスケープシーケンスの処理を行い、エンコードします。 アプリケーション識別子 (AI) を括弧で囲むことができます。括弧は HRI の印字文字として使用し、データとしてエンコードしません。 以下の文字をエンコードするには、文字 { で始まる 2 文字を指定してください。 FNC1 : { ( : {( ) : } }
Aztec Code	文字列を UTF-8 に変換後、エスケープシーケンスの処理を行い、エンコードします。

2次元シンボルの種類	説明
DataMatrix	文字列を UTF-8 に変換後、エスケープシーケンスの処理を行い、エンコードします。

**level**

定数	説明
LEVEL_0	PDF417 エラー訂正レベル 0
LEVEL_1	PDF417 エラー訂正レベル 1
LEVEL_2	PDF417 エラー訂正レベル 2
LEVEL_3	PDF417 エラー訂正レベル 3
LEVEL_4	PDF417 エラー訂正レベル 4
LEVEL_5	PDF417 エラー訂正レベル 5
LEVEL_6	PDF417 エラー訂正レベル 6
LEVEL_7	PDF417 エラー訂正レベル 7
LEVEL_8	PDF417 エラー訂正レベル 8
LEVEL_L	QR Code エラー訂正レベル L
LEVEL_M	QR Code エラー訂正レベル M
LEVEL_Q	QR Code エラー訂正レベル Q
LEVEL_H	QR Code エラー訂正レベル H
整数値 (5 ~ 95)	Aztec Code エラー訂正レベル (初期値 : 23)
LEVEL_DEFAULT	既定レベル

**width**

2次元シンボルの種類	有効な値	初期値
PDF417	2 ~ 8	3
QR Code	3 ~ 16	3
MaxiCode	未使用	
2次元 GS1 DataBar	2 ~ 8	2
Aztec Code	2 ~ 16	3
DataMatrix	2 ~ 16	3

**height**

2次元シンボルの種類	有効な値	初期値
PDF417	2 ~ 8 (width に対する倍率)	3

2次元シンボルの種類	有効な値	初期値
QR Code	未使用	
MaxiCode		
2次元 GS1 DataBar		
Aztec Code		
DataMatrix		

## size

2次元シンボルの種類	初期値	説明
PDF417	0 (自動)	1段あたりのコードワード数を指定
QR Code	未使用	
MaxiCode	未使用	
2次元 GS1 DataBar		
Expanded Stacked	0 (自動)	バーコードの最大幅を指定 (106 以上)
その他	未使用	
Aztec Code	未使用	
DataMatrix	未使用	

## 戻り値

戻り値	オブジェクト種類
Printer オブジェクト	Printer
HybridPrinter オブジェクト	HybridPrinter

## 例外

例外	オブジェクト種類
Parameter " ... " is invalid	Error

## 補足説明

- ❑ addSymbol メソッドの指定は「行の先頭」で行ってください。
- ❑ ページモードでは、2次元シンボルの左上ドットを基準として、現在の印字位置に展開します。
- ❑ type で指定する2次元シンボルの種類に従った文字列を指定してください。data で指定した文字列と type で指定する2次元シンボルの種類が一致しないと2次元シンボルは印字されません。
- ❑ Micro QR Code は、LEVEL\_H をサポートしていません。
- ❑ type で指定した2次元シンボルの種類に合わせて、定数を指定してください。
- ❑ data にて MaxiCode、2次元 GS1 DataBar を指定した場合は、LEVEL\_DEFAULT を指定してください。

addHLine メソッド

横罫線を描画し、横罫線印字を命令バッファに追加します。

構文

```
addHLine(x1, x2, style);
```

パラメーター

x1

設定値	説明
0 ~ 65535 の整数値	横罫線の描画開始位置をドット単位で指定

x2

設定値	説明
0 ~ 65535 の整数値	横罫線の描画終了位置をドット単位で指定

style

定数	説明
LINE_THIN	実線：細
LINE_MEDIUM	実線：中太
LINE_THICK	実線：太
LINE_THIN_DOUBLE	二重線：細
LINE_MEDIUM_DOUBLE	二重線：中太
LINE_THICK_DOUBLE	二重線：太
undefined	実線：細

戻り値

戻り値	オブジェクト種類
Printer オブジェクト	Printer
HybridPrinter オブジェクト	HybridPrinter

例外

例外	オブジェクト種類
Parameter " ... " is invalid	Error

---

## 補足説明

- ページモードでは機能しません。
- ページモードでの横罫線は、[addPageLine メソッド](#)を使用してください。

addVLineBegin メソッド

縦罫線の描画開始を命令バッファーに追加します。

構文

```
addVLineBegin(x, style);
```

パラメーター

x

設定値	説明
0 ~ 65535 の整数値	縦罫線の描画開始位置をドット単位で指定

style

定数	説明
LINE_THIN	実線：細
LINE_MEDIUM	実線：中太
LINE_THICK	実線：太
LINE_THIN_DOUBLE	二重線：細
LINE_MEDIUM_DOUBLE	二重線：中太
LINE_THICK_DOUBLE	二重線：太
undefined	実線：細

戻り値

戻り値	オブジェクト種類
Printer オブジェクト	Printer
HybridPrinter オブジェクト	HybridPrinter

例外

例外	オブジェクト種類
Parameter " ..." is invalid	Error

補足説明

- ❑ ページモードでは機能しません。
- ❑ ページモードでの横罫線は、[addPageLine メソッド](#)を使用してください。

- ❑ `addVLineEnd` メソッドで終了を指定するまで縦罫線の描画を行います。`addVLineBegin` メソッドは、`addVLineEnd` メソッドと組み合わせて使用してください。

addVLineEnd メソッド

縦罫線の描画終了を命令バッファーに追加します。

構文

```
addVLineEnd(x, style);
```

パラメーター

x

設定値	説明
0 ~ 65535 の整数値	縦罫線の描画終了位置をドット単位で指定

style

定数	説明
LINE_THIN	実線：細
LINE_MEDIUM	実線：中太
LINE_THICK	実線：太
LINE_THIN_DOUBLE	二重線：細
LINE_MEDIUM_DOUBLE	二重線：中太
LINE_THICK_DOUBLE	二重線：太
undefined	実線：細

戻り値

戻り値	オブジェクト種類
Printer オブジェクト	Printer
HybridPrinter オブジェクト	HybridPrinter

例外

例外	オブジェクト種類
Parameter " ..." is invalid	Error

補足説明

- ❑ ページモードでは機能しません。
- ❑ ページモードでの縦罫線は、[addPageLine メソッド](#)を使用してください。
- ❑ [addPageBegin メソッド](#)と組み合わせて使用してください。

addPageBegin メソッド

ページモードの開始を命令バッファーに追加します。  
addPageBegin メソッドを実行すると、ページモードでの処理を開始します。

構文

```
addPageBegin();
```

戻り値

戻り値	オブジェクト種類
Printer オブジェクト	Printer
HybridPrinter オブジェクト	HybridPrinter

補足説明

- ❑ ページモードでは、多階調印字をサポートしていません。
- ❑ [addPageEnd メソッド](#)と組み合わせて使用してください。

addPageEnd メソッド

ページモードの終了を命令バッファに追加します。  
addPageEnd メソッドを実行すると、ページモードでの処理を終了します。

構文

```
addPageEnd ( ) ;
```

戻り値

戻り値	オブジェクト種類
Printer オブジェクト	Printer
HybridPrinter オブジェクト	HybridPrinter

補足説明

[addPageBegin メソッド](#)と組み合わせて使用してください。

addPageArea メソッド

ページモード印字領域を命令バッファに追加します。  
addPageArea メソッドでページモード印字領域（座標）を指定し、addText メソッドなどの印字データを指定します。

構文

```
addPageArea(x, y, width, height);
```

パラメーター

x

0 は印字可能領域の左端です。

設定値	説明
0 ～ 65535 の整数値	横方向の原点をドット単位で指定

y

0 は紙送り前の位置です。

設定値	説明
0 ～ 65535 の整数値	縦方向の原点をドット単位で指定

width

設定値	説明
0 ～ 65535 の整数値	印字領域の幅をドット単位で指定

height

設定値	説明
0 ～ 65535 の整数値	印字領域の高さをドット単位で指定

戻り値

戻り値	オブジェクト種類
Printer オブジェクト	Printer
HybridPrinter オブジェクト	HybridPrinter

例外

例外	オブジェクト種類
Parameter " ... " is invalid	Error

---

## 補足説明

- ❑ 印字する内容に合わせて、印字領域を指定してください。印字領域からはみ出した印字データは、印字されません。
- ❑ スタンダードモードでは機能しません。
- ❑ 印字領域の幅と高さは、印字方向の設定に合わせて指定してください。印字領域の幅と高さが印字方向の設定と一致しないと、印字領域からはみ出した印字データは印字されません。印字方向の設定は、[addPageDirection メソッド](#)で指定します。

## addPageDirection メソッド

ページモード印字方向設定を命令バッファに追加します。

### 構文

```
addPageDirection(dir);
```

### パラメーター

dir

定数	説明
DIRECTION_LEFT_TO_RIGHT (初期値)	回転しない (左上を始点に右方向へ印字)
DIRECTION_BOTTOM_TO_TOP	反時計回り 90 度回転 (左下を始点に上方向へ印字)
DIRECTION_RIGHT_TO_LEFT	180 度回転 (右下を始点に左方向へ印字)
DIRECTION_TOP_TO_BOTTOM	時計回り 90 度回転 (右上を始点に下方向へ印字)

### 戻り値

戻り値	オブジェクト種類
Printer オブジェクト	Printer
HybridPrinter オブジェクト	HybridPrinter

### 例外

例外	オブジェクト種類
Parameter " ..." is invalid	Error

### 補足説明

スタンダードモードでは機能しません。

addPagePosition メソッド

addPageArea メソッドで指定した印字領域内での印字開始位置（座標）を指定し、ページモード印字領域内の印字位置設定を命令バッファに追加します。

構文

```
addPagePosition(x, y);
```

パラメーター

x

初期値：0

設定値	説明
0 ～ 65535 の整数値	横方向の印字位置をドット単位で指定

y

初期値：21

設定値	説明
0 ～ 65535 の整数値	縦方向の印字位置をドット単位で指定

戻り値

戻り値	オブジェクト種類
Printer オブジェクト	Printer
HybridPrinter オブジェクト	HybridPrinter

例外

例外	オブジェクト種類
Parameter " ... " is invalid	Error

補足説明

- ❑ スタンダードモードでは機能しません。
- ❑ 印字内容に合わせて、印字開始位置（座標）を指定してください。

印字内容	指定方法
文字列	最初の文字のベースライン左端を指定します。 標準の大きさで左詰め印字する場合は省略可能です。高さが 2 倍の文字を印刷する場合は、y を 42 以上に指定します。

印字内容	指定方法
バーコード	シンボルの左下を指定します。y にバーコードの高さを指定してください。
グラフィック / ロゴ	グラフィックデータの左下を指定します。y にグラフィックデータの高さを指定します。
2次元シンボル	シンボルの左上を指定します。左上から印字する場合は、省略可能です。

## addPageLine メソッド

ページモードの直線描画設定を命令バッファに追加します。

### 構文

```
addPageLine(x1, y1, x2, y2, style);
```

### パラメーター

x1

設定値	説明
0 ~ 65535 の整数値	横方向の描画開始位置をドット単位で指定

y1

設定値	説明
0 ~ 65535 の整数値	縦方向の描画開始位置をドット単位で指定

x2

設定値	説明
0 ~ 65535 の整数値	横方向の描画終了位置をドット単位で指定

y2

設定値	説明
0 ~ 65535 の整数値	縦方向の描画終了位置をドット単位で指定

style

定数	説明
LINE_THIN	実線：細
LINE_MEDIUM	実線：中太
LINE_THICK	実線：太
LINE_THIN_DOUBLE	二重線：細
LINE_MEDIUM_DOUBLE	二重線：中太
LINE_THICK_DOUBLE	二重線：太
undefined	実線：細

---

## 戻り値

戻り値	オブジェクト種類
Printer オブジェクト	Printer
HybridPrinter オブジェクト	HybridPrinter

---

## 例外

例外	オブジェクト種類
Parameter "... " is invalid	Error

---

## 補足説明

- スタンダードモードでは機能しません。
- 斜線は描画できません。描画する直線が縦罫線 / 横罫線になるようにパラメーターを指定してください。

## addPageRectangle メソッド

ページモードの四角形描画設定を命令バッファーに追加します。

### 構文

```
addPageRectangle(x1, y1, x2, y2, style);
```

### パラメーター

x1

設定値	説明
0 ~ 65535 の整数値	横方向の描画開始位置をドット単位で指定

y1

設定値	説明
0 ~ 65535 の整数値	縦方向の描画開始位置をドット単位で指定

x2

設定値	説明
0 ~ 65535 の整数値	横方向の描画終了位置をドット単位で指定

y2

設定値	説明
0 ~ 65535 の整数値	縦方向の描画終了位置をドット単位で指定

style

定数	説明
LINE_THIN	実線：細
LINE_MEDIUM	実線：中太
LINE_THICK	実線：太
LINE_THIN_DOUBLE	二重線：細
LINE_MEDIUM_DOUBLE	二重線：中太
LINE_THICK_DOUBLE	二重線：太
undefined	実線：細

---

## 戻り値

戻り値	オブジェクト種類
Printer オブジェクト	Printer
HybridPrinter オブジェクト	HybridPrinter

---

## 例外

例外	オブジェクト種類
Parameter "... " is invalid	Error

---

## 補足説明

スタンダードモードでは機能しません。

addCut メソッド

用紙カットを命令バッファに追加します。

構文

```
addCut (type) ;
```

パラメーター

type

設定値	説明
CUT_NO_FEED	フィードなしカット（紙送りせずにカット）
CUT_FEED	フィードカット（紙送り後カット）
CUT_RESERVE	カット予約（後に続く印字を実行後、カット位置でカット）
undefined	フィードカット（紙送り後カット）

戻り値

戻り値	オブジェクト種類
Printer オブジェクト	Printer
HybridPrinter オブジェクト	HybridPrinter

例外

例外	オブジェクト種類
Parameter " ... " is invalid	Error

補足説明

- ❑ ページモードでは機能しません。
- ❑ 本 API は「行の先頭」で使用してください。先頭以外で使用的場合、無効になります。

## addPulse メソッド

ドロアーキック設定を命令バッファに追加します。

### 構文

```
addPulse(drawer, time);
```

### パラメーター

**drawer**

設定値	説明
DRAWER_1	ドロアーキックコネクタ 2 番ピン
DRAWER_2	ドロアーキックコネクタ 5 番ピン
undefined	ドロアーキックコネクタ 2 番ピン

**time**

設定値	説明
PULSE_100	100 msec の信号
PULSE_200	200 msec の信号
PULSE_300	300 msec の信号
PULSE_400	400 msec の信号
PULSE_500	500 msec の信号
undefined	100 msec の信号

### 戻り値

戻り値	オブジェクト種類
Printer オブジェクト	Printer
HybridPrinter オブジェクト	HybridPrinter

### 例外

例外	オブジェクト種類
Parameter "... " is invalid	Error

### 補足説明

- ページモードでは機能しません。
- ドロアー機能は、ブザー機能と同時に使用できません。

## addSound メソッド

ブザーの鳴動設定を命令バッファに追加します。

### 構文

```
addSound(pattern, repeat, cycle);
```

### パラメーター

#### pattern

設定値	説明
PATTERN_NONE	停止
PATTERN_A	パターン A (外付けオプションブザー)
PATTERN_B	パターン B (外付けオプションブザー)
PATTERN_C	パターン C (外付けオプションブザー)
PATTERN_D	パターン D (外付けオプションブザー)
PATTERN_E	パターン E (外付けオプションブザー)
PATTERN_ERROR	エラー鳴動パターン (外付けオプションブザー)
PATTERN_PAPER_END	用紙なし鳴動パターン (外付けオプションブザー)
PATERN_0	パターン 0 (内蔵ブザー)
PATERN_1	パターン 1 (内蔵ブザー)
PATERN_2	パターン 2 (内蔵ブザー)
PATERN_3	パターン 3 (内蔵ブザー)
PATERN_4	パターン 4 (内蔵ブザー)
PATERN_5	パターン 5 (内蔵ブザー)
PATERN_6	パターン 6 (内蔵ブザー)
PATERN_7	パターン 7 (内蔵ブザー)
PATERN_8	パターン 8 (内蔵ブザー)
PATERN_9	パターン 9 (内蔵ブザー)
PATERN_10	パターン 10 (内蔵ブザー)
undefined	パターン A (外付けオプションブザー)

#### repeat

設定値	説明
0	無限
1 ~ 255	1 ~ 255 回

設定値	説明
undefined	1 回

**cycle**

設定値	説明
1000 ~ 25500	ブザーの鳴動周期を msec 単位で指定

**戻り値**

戻り値	オブジェクト種類
Printer オブジェクト	Printer
HybridPrinter オブジェクト	HybridPrinter

**例外**

例外	オブジェクト種類
Parameter " ... " is invalid	Error

**補足説明**

- ❑ ページモードでは機能しません。
- ❑ ブザー機能は、ドロアー機能と同時に使用できません。
- ❑ cycle は、pattern にて PATTERN\_0 ~ PATTERN\_10 を指定した場合のみ有効です。

## addLayout メソッド

用紙レイアウト設定を命令バッファに追加します。

### 構文

```
addLayout(type, width, height, margin_top,
margin_bottom, offset_cut, offset_label);
```

### パラメーター

#### type

設定値	説明
LAYOUT_RECEIPT	レシート紙
LAYOUT_RECEIPT_BM	レシート紙（ブラックマーク付）
LAYOUT_LABEL	ラベル紙
LAYOUT_LABEL_BM	ラベル紙（ブラックマーク付）

#### width

設定値	説明
290 ～ 600	用紙の幅を 0.1 mm 単位で指定

#### height

用紙の種類	有効な値	説明
レシート紙	0	指定不要
レシート紙（ブラックマーク付）	<ul style="list-style-type: none"> <li>0（自動）</li> <li>284 ～ 1550（手動）</li> </ul>	ブラックマーク上端位置から次のブラックマーク上端位置までの距離を 0.1 mm 単位で指定
ラベル紙		ラベル上端位置から次のラベル上端位置までの距離を 0.1 mm 単位で指定
ラベル紙（ブラックマーク付）		ブラックマーク下端位置から次のブラックマーク下端位置までの距離を 0.1 mm 単位で指定

#### margin\_top

用紙の種類	有効な値	説明
レシート紙	0	指定不要
レシート紙（ブラックマーク付）	-150 ～ 1500	ブラックマーク上端位置から頭出し位置までの距離を 0.1 mm 単位で指定
ラベル紙	0 ～ 1500	ラベル上端位置から頭出し位置までの距離を 0.1 mm 単位で指定

用紙の種類	有効な値	説明
ラベル紙（ブラックマーク付）	-15 ~ 1500	ブラックマーク下端位置から頭出し位置までの距離を 0.1 mm 単位で指定

**margin\_bottom**

用紙の種類	有効な値	説明
レシート紙	0	指定不要
レシート紙（ブラックマーク付）	0	指定不要
ラベル紙	-15 ~ 0	ラベル下端位置から印字領域下端までの距離を 0.1 mm 単位で指定 （正数方向は余白が拡がり、負数方向は余白が縮む）
ラベル紙（ブラックマーク付）	-15 ~ 15	ブラックマーク上端位置から印字領域下端までの距離を 0.1 mm 単位で指定 （正数方向は余白が拡がり、負数方向は余白が縮む）

**offset\_cut**

用紙の種類	有効な値	説明
レシート紙	0	指定不要
レシート紙（ブラックマーク付）	-290 ~ 50	ブラックマーク上端位置からカット位置までの距離を 0.1 mm 単位で指定
ラベル紙	0 ~ 50	ラベル下端位置からカット位置までの距離を 0.1 mm 単位で指定
ラベル紙（ブラックマーク付）	0 ~ 50	ブラックマーク上端位置からカット位置までの距離を 0.1 mm 単位で指定

**offset\_label**

用紙の種類	有効な値	説明
レシート紙	0	指定不要
レシート紙（ブラックマーク付）	0	指定不要
ラベル紙	0	指定不要
ラベル紙（ブラックマーク付）	0 ~ 15	ブラックマーク上端位置からラベル下端位置までの距離を 0.1 mm 単位で指定

**戻り値**

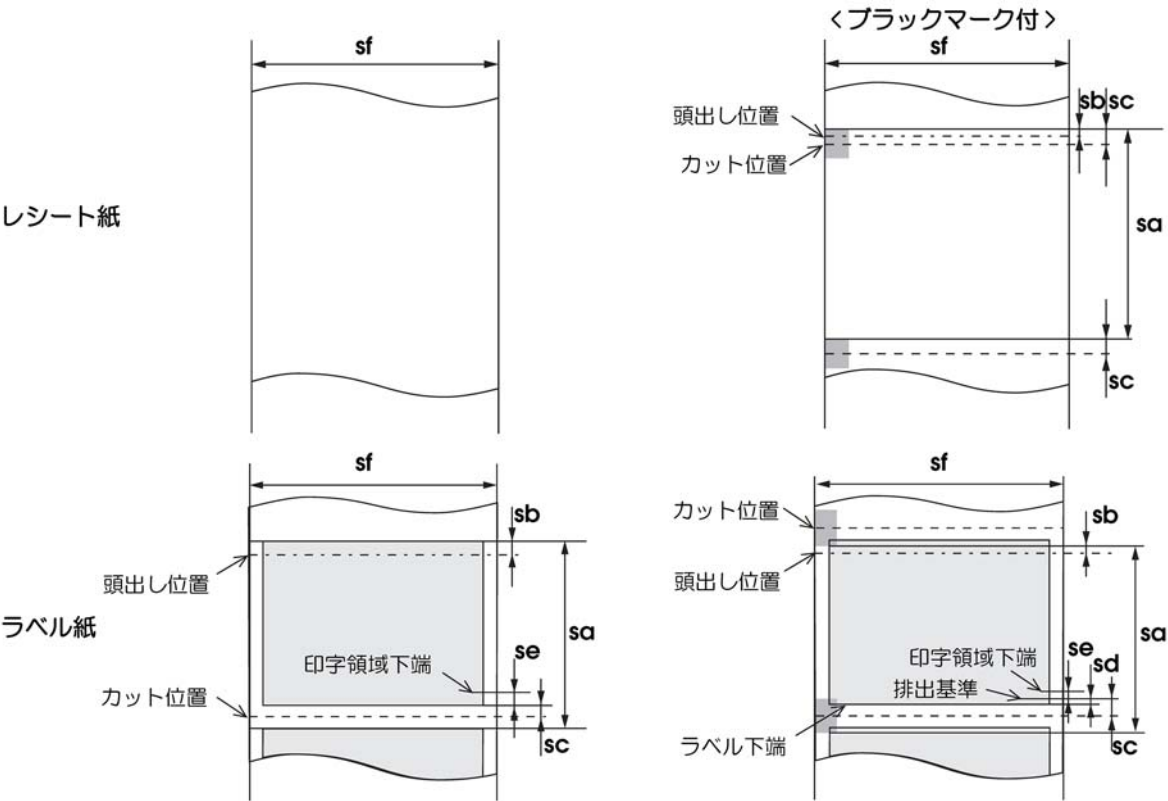
戻り値	オブジェクト種類
Printer オブジェクト	Printer
HybridPrinter オブジェクト	HybridPrinter

例外

例外	オブジェクト種類
Parameter "... " is invalid	Error

詳細説明

用紙ごと指定可能なパラメーターの位置は、以下を参照してください。



補足説明

ページモードでは機能しません。

## recover メソッド

プリンター本体をエラー状態から復帰させます。

プリンター本体に復帰可能エラーが発生した時、プリンター本体のエラー要因を除去しただけでは、プリンター本体はエラー状態から復帰しません。recover メソッドを実行することで、プリンター本体はエラー状態から復帰し、印字可能な状態に戻ります。

---

### 構文

```
recover();
```

addRecovery メソッド

プリンター本体のエラー状態からの復帰命令を命令バッファに追加します。

プリンター本体に復帰可能エラーが発生した時、addRecovery メソッドを実行することで、プリンター本体はエラー状態から復帰し、印字可能な状態に戻ります。

addRecovery メソッドを実行することによって得られる結果は、recover メソッドと同じです。

プリンター本体をエラー状態から復帰させる際は、recover メソッドの使用を推奨します。

構文

```
addRecovery();
```

戻り値

戻り値	オブジェクト種類
Printer オブジェクト	Printer

例外

例外	オブジェクト種類
Parameter " ... " is invalid	Error

補足説明

- ❑ 強制送信モード（force プロパティ）を有効にして使用してください。
- ❑ 復帰可能エラーから復帰すると、プリンター本体のバッファはリセットされます。

## reset メソッド

プリンター本体をリセットします。

プリンター本体は初期状態に戻り、プリンターバッファに残っている印字データや、印刷設定などのプリンター本体に記憶されないデータは、全て失われます。

---

### 構文

```
reset();
```

addReset メソッド

プリンター本体のリセット命令を命令バッファに追加します。

プリンター本体は初期状態に戻り、プリンターバッファに残っている印字データや、印刷設定などのプリンター本体に記憶されないデータは、全て失われます。

addReset メソッドを実行することによって得られる結果は、[reset メソッド](#)と同じです。

addReset メソッドを命令バッファに追加してプリンターのリセットを行うと、命令バッファに含まれている他の印字命令は実行されません。プリンター本体のリセットには、reset メソッドの使用を推奨します。

構文

```
addReset ( ) ;
```

戻り値

戻り値	オブジェクト種類
Printer オブジェクト	Printer

例外

例外	オブジェクト種類
Parameter " ... " is invalid	Error

## addCommand メソッド

ESC/POS コマンドを命令バッファに追加します。

### 構文

```
addCommand(data);
```

### パラメーター

data

設定値	説明
String	ESC/POS コマンドを文字列で指定

### 戻り値

戻り値	オブジェクト種類
Printer オブジェクト	Printer
HybridPrinter オブジェクト	HybridPrinter

### 例外

例外	オブジェクト種類
Parameter " ..." is invalid	Error

### 補足説明

ESC/POS コマンドの詳細は以下の URL を参照してください。  
[https://reference.epson-biz.com/modules/ref\\_escpos\\_ja/](https://reference.epson-biz.com/modules/ref_escpos_ja/)

## send メソッド

命令バッファーのデータをプリンターへ送信します。  
印刷ジョブ ID を指定できます。

### 構文

- ❑ `send()` ;
- ❑ `send(printjobid)` ;

### パラメーター

**printjobid**

設定値	説明
String	1 ~ 30 文字の英数字、アンダースコア、ハイフン、ピリオドを使用して指定

### サンプルプログラム

印刷ジョブ ID を指定して送信する

```
var printjobid = 'ABC123';

printer.addText('Hello, World!\n');
printer.addCut();
printer.onreceive = function (res) { alert(res.printjobid); };
printer.onerror = function (err) { alert(err.status); };
printer.send(printjobid);
```

### 補足説明

- ❑ send メソッドの実行結果は、[onreceive イベント](#)に渡されます。
- ❑ send メソッドは強制送信モード ([force プロパティ](#)) を無効にします。

## print メソッド

HTML5 Canvas で描画したイメージデータを印字します。

印刷ジョブ ID を指定できます。

HTML5 Canvas の RGBA フルカラー画像の指定範囲を、[halftone プロパティ](#)と [brightness プロパティ](#)の設定に従って、ラスターイメージデータに変換します。画像の 1 ピクセルがプリンターの 1 ドットに相当します。透明色が含まれている場合は、画像の背景を白とみなします。

### 構文

- ❑ `print(canvas, cut, mode);`
- ❑ `print(canvas, cut, mode, printjobid);`

### パラメーター

#### canvas

設定値	説明
canvas	HTML5 Canvas オブジェクトを指定

#### cut

設定値	説明
true	オートカットする
false	オートカットしない
undefined	オートカットしない

#### mode

設定値	説明
MODE_MONO	モノクロ (2 階調)
MODE_GRAY16	多階調 (16 階調)
undefined	モノクロ (2 階調)

#### printjobid

設定値	説明
String	1 ～ 30 文字の英数字、アンダースコア、ハイフン、ピリオドを使用して指定

---

## 例外

例外	オブジェクト種類
Parameter "... " is invalid	Error

---

## 補足説明

- ❑ 異なるドメインからダウンロードした画像が含まれた HTML5 Canvas は印字できません。JavaScript の同一出身ポリシーにより、セキュリティエラーが発生します。
- ❑ 多階調印字はスタンダードモードでは使用できますが、ページモードでは使用できません。

## getPrintJobStatus メソッド

指定した印刷ジョブ ID の状態を取得します。

getPrintJobStatus メソッドで取得した印刷ジョブ ID の状態は、[onreceive イベント](#)へ渡されます。

---

### 構文

```
getPrintJobStatus (printjobid) ;
```

---

### パラメーター

printjobid

設定値	説明
String	1 ～ 30 文字の英数字、アンダースコア、ハイフン、ピリオドを使用して指定

---

### 補足説明

同じ印刷ジョブ ID で複数の印字処理を実行した場合、最新の印刷ジョブ ID の状態を取得します。

## startMonitor メソッド

Printer オブジェクトによるプリンターステータスの取得、更新を有効にします。

startMonitor メソッドを実行すると、Printer オブジェクトは [interval プロパティ](#) で指定した間隔でプリンターステータスを取得、更新し、以下のイベントに取得、更新したプリンターステータスを渡します。

- ❑ [onstatuschange イベント](#)
- ❑ [onbatterystatuschange イベント](#)
- ❑ [ononline イベント](#)
- ❑ [onoffline イベント](#)
- ❑ [onpoweroff イベント](#)
- ❑ [oncoverok イベント](#)
- ❑ [oncoveropen イベント](#)
- ❑ [onpaperok イベント](#)
- ❑ [onpapernearend イベント](#)
- ❑ [onpaperend イベント](#)
- ❑ [ondrawerclosed イベント](#)
- ❑ [ondraweropen イベント](#)
- ❑ [onbatteryok イベント](#)
- ❑ [onbatterylow イベント](#)

---

## 構文

```
startMonitor();
```

## stopMonitor メソッド

[startMonitor メソッド](#)で有効にした Printer オブジェクトによるプリンタステータスの取得、更新を無効にします。

---

### 構文

```
stopMonitor();
```

## halftone プロパティ

ハーフトーンの処理方法を指定します。  
[addImage メソッド](#)と [print メソッド](#)のモノクロ（2 階調）印字に適用されます。

### オブジェクト種類

String

定数	処理方法	説明
HALFTONE_DITHER（初期値）	ディザ	グラフィックのみの印刷
HALFTONE_ERROR_DIFFUSION	誤差拡散	文字とグラフィックが混在する印刷
HALFTONE_THRESHOLD	しきい値	文字のみの印刷

## brightness プロパティ

明るさ補正値を指定します。

[addImage メソッド](#)と [print メソッド](#)の印字に適用されます。

---

## オブジェクト種類

### Number

初期値：1.0

設定値	説明
0.1 ~ 10.0	明るさ補正値を指定

force プロパティ

強制送信モードを指定します。  
強制送信モードを有効にすると、プリンター本体へ強制的に印字命令を送信します。

オブジェクト種類

Boolean

設定値	説明
true	有効
false（初期値）	無効

補足説明

- ❑ 強制送信モードは、プリンター本体がオフラインの状態で使用してください。
- ❑ プリンター本体がオンラインの状態で使用した場合、[onreceive イベント](#)の code にエラーコード 'EX\_BADPORT' が渡されます。
- ❑ 強制送信モードで有効な機能は、以下の通りです。
  - ドロアーキック ([addPulse メソッド](#))
  - ブザーの停止 ([addSound メソッド](#))
  - 復帰可能エラーからの復帰 ([addRecovery メソッド](#))
  - リセット ([addReset メソッド](#))
  - リアルタイムコマンドの送信 ([addCommand メソッド](#))

timeout プロパティ

[send メソッド](#)で送信するメッセージのタイムアウト時間を指定します。  
機種仕様や通信インターフェイス、送信データサイズに合わせてタイムアウト時間を調整します。

オブジェクト種類

Number

初期値：10000

設定値	説明
0 ~ 600000	タイムアウト時間を msec 単位で指定

interval プロパティ

[startMonitor メソッド](#)で有効にしたプリンターステータスの取得、更新間隔を指定します。

オブジェクト種類

Number

初期値：3000  
不正な値を指定した場合は、初期値が指定されます。

設定値	説明
1000 ~ 6000	プリンターステータスの更新間隔を msec 単位で指定

drawerOpenLevel プロパティ

[ondraweropen イベント](#)と [ondrawerclosed イベント](#)を発生させるドロアーの信号線状態を指定します。  
使用するドロアーの仕様に合わせて変更してください。

オブジェクト種類

Integer

設定値	信号線状態	説明
DRAWER_OPEN_LEVEL_LOW（初期値）	HIGH → LOW	<a href="#">ondraweropen イベント</a> 発生
	LOW → HIGH	<a href="#">ondrawerclosed イベント</a> 発生
DRAWER_OPEN_LEVEL_HIGH	HIGH → LOW	<a href="#">ondrawerclosed イベント</a> 発生
	LOW → HIGH	<a href="#">ondraweropen イベント</a> 発生

## message プロパティ

Printer オブジェクトの API で追加したコマンドを、コードで表示します。  
本プロパティからコードを直接操作し、コマンドの確認や追加、削除ができます。

---

### オブジェクト種類

String

## onreceive イベント

印刷結果を受け取ります。

### 構文

Function (response)

### コールバック関数のパラメーター

パラメーター：response

オブジェクト種類：Object

### response オブジェクトのプロパティ

プロパティ	説明	オブジェクト種類
success	印字結果	Boolean
code	エラーコード	String
status	ステータス	Number
battery	バッテリーステータス	Number
printjobid	印刷ジョブ ID	String

#### success

値	説明
true	<ul style="list-style-type: none"> <li>印字成功</li> <li>処理に成功（スプーラー機能有効時）</li> </ul>
false	<ul style="list-style-type: none"> <li>印字失敗</li> <li>処理に失敗（スプーラー機能有効時）</li> </ul>

#### code

値	説明
'EPTR_AUTOMATICAL'	自動復帰エラー発生
'EPTR_BATTERY_LOW'	バッテリー残量なし
'EPTR_COVER_OPEN'	カバーオープンエラー発生
'EPTR_CUTTER'	オートカッターエラー発生
'EPTR_MECHANICAL'	メカニカルエラー発生
'EPTR_REC_EMPTY'	ロール紙エンド
'EPTR_UNRECOVERABLE'	復帰不可能エラー発生
'SchemaError'	印刷要求 XML の構文に誤りがある
'DeviceNotFound'	デバイス ID で指定したプリンターが存在しない

値	説明
'PrintSystemError'	印刷システムにエラーが発生
'EX_BADPORT'	通信ポートに異常を検出
'EX_TIMEOUT'	印刷タイムアウト発生
'EX_SPOOLER'	印刷キューに空き容量がない
'JobNotFound'	指定のジョブ ID が存在しない
'Printing'	印刷中

## status

定数との AND 演算を行ってください。

定数 (status)	説明
ASB_NO_RESPONSE	プリンター無応答
ASB_PRINT_SUCCESS	印刷終了 / スプール完了
ASB_DRAWER_KICK	ドロアーキックコネクタ 3 番ピンの状態 = "H"
ASB_BATTERY_OFFLINE	バッテリー残量によるオフライン状態
ASB_OFF_LINE	オフライン状態
ASB_COVER_OPEN	カバーオープン
ASB_PAPER_FEED	紙送りスイッチによる紙送信中
ASB_WAIT_ON_LINE	オンライン復帰待ち中
ASB_PANEL_SWITCH	パネルスイッチ操作中
ASB_MECHANICAL_ERR	メカニカルエラー発生
ASB_AUTOCUTTER_ERR	オートカッターエラー発生
ASB_UNRECOVER_ERR	復帰不可能エラー発生
ASB_AUTORECOVER_ERR	自動復帰エラー発生
ASB_RECEIPT_NEAR_END	ロール紙ニアエンド
ASB_RECEIPT_END	ロール紙エンド
ASB_BUZZER	ブザー鳴動中（対応機種のみ）
ASB_SPOOLER_IS_STOPPED	スプーラーの停止

## battery

□ AC アダプター接続状態

定数 (battery)	説明
0x30XX	接続されている
0x31XX	接続されていない

□ バッテリー残量

定数 (battery)	説明
0xXX36	残量 6
0xXX35	残量 5
0xXX34	残量 4
0xXX33	残量 3
0xXX32	残量 2
0xXX31	残量 1 (ニアエンド)
0xXX30	残量 0 (リアルエンド)

### printjobid

値	説明
printjobid	印刷ジョブ ID

### 補足説明

- ❑ バッテリー非搭載機種の場合、バッテリー残量の定数は 0xXX30 がセットされます。
- ❑ 印刷ジョブ ID が無い場合、printjobid には “ ” 空文字列がセットされます。

## onstatuschange イベント

Printer オブジェクトが取得したプリンターステータスを受け取ります。

---

### 構文

```
Function (status)
```

---

### コールバック関数のパラメーター

パラメーター：status

オブジェクト種類：Number

## onbatterystatuschange イベント

Printer オブジェクトが取得したバッテリーステータスを受け取ります。

---

### 構文

```
Function (battery)
```

---

### コールバック関数のパラメーター

パラメーター：battery

オブジェクト種類：Number

## ononline イベント

Printer オブジェクトが取得したオンラインステータスを受け取ります。

---

### 構文

```
Function ()
```

## onoffline イベント

Printer オブジェクトが取得したオフラインステータスを受け取ります。

---

### 構文

```
Function ()
```

## onpoweroff イベント

Printer オブジェクトが取得した無応答ステータスを受け取ります。

---

### 構文

```
Function ()
```

## oncoverok イベント

Printer オブジェクトが取得したカバークローズステータスを受け取ります。

---

### 構文

```
Function ()
```

## oncoveropen イベント

Printer オブジェクトが取得したカバーオープンステータスを受け取ります。

---

### 構文

```
Function ()
```

## onpaperok イベント

Printer オブジェクトが取得した用紙ありステータスを受け取ります。

---

### 構文

```
Function ()
```

## onpapernearend イベント

Printer オブジェクトが取得した用紙ニアエンドステータスを受け取ります。

---

### 構文

```
Function ()
```

## onpaperend イベント

Printer オブジェクトが取得した用紙なしステータスを受け取ります。

---

### 構文

```
Function ()
```

## ondrawerclosed イベント

Printer オブジェクトが取得したドロアークローズステータスを受け取ります。

---

### 構文

```
Function ()
```

---

### 補足説明

ondrawerclosed イベントを発生させるドロアーの信号線状態は、[drawerOpenLevel](#) プロパティで指定します。

## ondraweropen イベント

Printer オブジェクトが取得したドロアーオープンステータスを受け取ります。

---

### 構文

```
Function ()
```

---

### 補足説明

ondraweropen イベントを発生させるドロアーの信号線状態は、[drawerOpenLevel](#) プロパティで指定します。

## onbatteryok イベント

Printer オブジェクトが取得したバッテリー残量ありステータスを受け取ります。

---

### 構文

```
Function ()
```

## onbatterylow イベント

Printer オブジェクトが取得したバッテリー残量なしステータスを受け取ります。

---

### 構文

```
Function ()
```

# HybridPrinter オブジェクト

## lock メソッド

デバイスポートをロックします。MICR 制御とスリップ印字の連続動作を行う場合等に使用し、動作中に他のプロセスから割り込まれないようにします。

lock メソッドの実行結果は、[onreceive イベント](#)のコールバック関数に渡されます。

---

## 構文

```
lock();
```

---

## 補足説明

lock メソッド実行後は、アプリケーションを閉じる前に [unlock メソッド](#)を実行してください。実行しなかった場合、TM-DT ソフトウェアによってポートのロックが解除されるまでの約 5 分間、ポートをロックできず、[onreceive イベント](#)に ERROR\_DEVICE\_BUSY が渡されます。

## unlock メソッド

デバイスポートのロックを解除します。unlock メソッドの実行結果は、[onreceive イベント](#)のコールバック関数に渡されます。

---

### 構文

```
unlock();
```

## eject メソッド

チェック紙を排出します。eject メソッドの実行結果は、[onreceive イベント](#)のコールバック関数に渡されます。

---

### 構文

```
eject();
```

## ReceiptPrinter.send メソッド

命令バッファのデータをプリンターへ送信し、レシート印字を実行します。ReceiptPrinter.send メソッドの実行結果は、[onreceive イベント](#)のコールバック関数に渡されます。

---

### 構文

```
ReceiptPrinter.send();
```

## ReceiptPrinter.print メソッド

HTML5 Canvas で描画したイメージデータを印字します。  
HTML5 Canvas の RGBA フルカラー画像の指定範囲を、[halftone プロパティ](#)と [brightness プロパティ](#)の設定に従って、ラスタイメージデータに変換します。画像の 1 ピクセルがプリンターの 1 ドットに相当します。透明色が含まれている場合は、画像の背景を白とみなします。

### 構文

```
print(canvas, cut, mode);
```

### パラメーター

**canvas**

設定値	説明
canvas	HTML5 Canvas オブジェクトを指定

**cut**

設定値	説明
true	オートカットする
false	オートカットしない
undefined	オートカットしない

**mode**

設定値	説明
MODE_MONO	モノクロ（2 階調）
MODE_GRAY16	多階調（16 階調）
undefined	モノクロ（2 階調）

### 補足説明

- ❑ 異なるドメインからダウンロードした画像が含まれた HTML5 Canvas は印字できません。JavaScript の同一出身ポリシーにより、セキュリティエラーが発生します。
- ❑ 多階調印字はスタンダードモードでは使用できますが、ページモードでは使用できません。

## SlipPrinter.send メソッド

命令バッファーのデータをプリンターへ送信し、チェック紙表面のスリップ印字を実行します。SlipPrinter.send メソッドの実行結果は、[onreceive イベント](#)のコールバック関数に渡されます。

---

### 構文

```
SlipPrinter.send();
```

## SlipPrinter.cancel メソッド

[SlipPrinter.send メソッド](#)実行後のチェック紙挿入待ち状態を解除します。SlipPrinter.cancel メソッドの実行結果は、[onreceive イベント](#)のコールバック関数に渡されます。

---

### 構文

```
SlipPrinter.cancel();
```

## EndorsePrinter.enable40cplMode メソッド

40cpl モードでのチェック紙裏面スリップ印字を設定します。

### 構文

```
EndorsePrinter.enable40cplMode(enable);
```

### パラメーター

enable

設定値	説明
true	40cpl モード有効
false	40cpl モード無効

### 補足説明

ページモードでは機能しません。

## EndorsePrinter.send メソッド

命令バッファのデータをプリンターへ送信し、チェック紙裏面のスリップ印字を実行します。  
EndorsePrinter.send メソッドの実行結果は、[onreceive イベント](#)のコールバック関数に渡されます。

---

### 構文

```
EndorsePrinter.send();
```

---

### 補足説明

ページモードでは機能しません。

## EndorsePrinter.cancel メソッド

[EndorsePrinter.send メソッド](#)実行後のチェック紙挿入待ち状態を解除します。EndorsePrinter.cancel メソッドの実行結果は、[onreceive イベント](#)のコールバック関数に渡されます。

---

### 構文

```
EndorsePrinter.cancel();
```

---

### 補足説明

ページモードでは機能しません。

MICRReader.read メソッド

MICR 読み取りを行います。MICRReader.read メソッド実行後にチェック紙を挿入すると、MICR 読み取りが実行されます。MICRReader.read メソッドの実行結果は、[onreceive イベント](#)のコールバック関数に渡されます。

構文

```
MICRReader.read(ignoreerror, font);
```

パラメーター

ignoreerror

設定値	説明
true	読み取り不能文字を ? に置き換え、処理を継続
false	読み取り不能時に処理を停止

font

設定値	説明
"MICR_E13B"	読み取りフォントに E13B を指定
"MICR_CMC7"	読み取りフォントに CMC7 を指定

## MICRReader.cleaning メソッド

MICR 機構を清掃します。MICRReader.cleaning メソッド実行後にクリーニングシートを挿入すると、MICR 機構の清掃を実行します。MICRReader.cleaning メソッドの実行結果は、[onreceive イベント](#)のコールバック関数に渡されます。

---

### 構文

```
MICRReader.cleaning();
```

## MICRReader.cancel メソッド

[MICRReader.cleaning メソッド](#)実行後のクリーニングシート挿入待ち状態を解除します。MICRReader.cancel メソッドの実行結果は、[onreceive イベント](#)のコールバック関数に渡されます。

---

### 構文

```
MICRReader.cancel();
```

## recover メソッド

プリンター本体をエラー状態から復帰させます。

プリンター本体に復帰可能エラーが発生した時、プリンター本体のエラー要因を除去しただけでは、プリンター本体はエラー状態から復帰しません。recover メソッドを実行することで、プリンター本体はエラー状態から復帰し、印字可能な状態に戻ります。

---

### 構文

```
recover();
```

## reset メソッド

プリンター本体をリセットします。

プリンター本体は初期状態に戻り、プリンターバッファに残っている印字データや、印刷設定などのプリンター本体に記憶されないデータは、全て失われます。

---

### 構文

```
reset();
```

## startMonitor メソッド

HybridPrinter オブジェクトによるプリンターステータスの取得、更新を有効にします。

startMonitor メソッドを実行すると、HybridPrinter オブジェクトは [interval プロパティ](#) で指定した間隔でプリンターステータスを取得、更新し、以下のイベントに取得、更新したプリンターステータスを渡します。

- ❑ [onstatuschange イベント](#)
- ❑ [ononline イベント](#)
- ❑ [onoffline イベント](#)
- ❑ [onpoweroff イベント](#)
- ❑ [oncoverok イベント](#)
- ❑ [oncoveropen イベント](#)
- ❑ [onpaperok イベント](#)
- ❑ [onpapernearend イベント](#)
- ❑ [onpaperend イベント](#)
- ❑ [ondrawerclosed イベント](#)
- ❑ [ondraweropen イベント](#)

---

## 構文

```
startMonitor();
```

## stopMonitor メソッド

[startMonitor メソッド](#)で有効にした HybridPrinter オブジェクトによるプリンタステータスの取得、更新を無効にします。

---

### 構文

```
stopMonitor();
```

halftone プロパティ

ハーフトーンの処理方法を指定します。  
[ReceiptPrinter.print メソッド](#)と [addImage メソッド](#)のレシート紙モノクロ（2 階調）印字に適用されます。

オブジェクト種類

String

定数	処理方法	説明
HALFTONE_DITHER（初期値）	ディザ	グラフィックのみの印刷
HALFTONE_ERROR_DIFFUSION	誤差拡散	文字とグラフィックが混在する印刷
HALFTONE_THRESHOLD	しきい値	文字のみの印刷

## brightness プロパティ

明るさ補正値を指定します。

[ReceiptPrinter.print メソッド](#)と [addImage メソッド](#)のレシート印字に適用されます。

---

## オブジェクト種類

### Number

初期値：1.0

設定値	説明
0.1 ~ 10.0	明るさ補正値を指定

## force プロパティ

強制送信モードを指定します。  
強制送信モードを有効にすると、プリンター本体へ強制的に印字命令を送信します。

## オブジェクト種類

### Boolean

設定値	説明
true	有効
false（初期値）	無効

## 補足説明

- ❑ 強制送信モードは、プリンター本体がオフラインの状態で使用してください。
- ❑ プリンター本体がオンラインの状態で使用した場合、[onreceive イベント](#)の code にエラーコード 'EX\_BADPORT' が渡されます。
- ❑ 強制送信モードは、プリンター本体がオフラインの状態で使用してください。
- ❑ プリンター本体がオンラインの状態で使用した場合、[onreceive イベント](#)の code にエラーコード 'EX\_BADPORT' が渡されます。
- ❑ 強制送信モードで有効な機能は、以下の通りです。
  - ドロアーキック ([addPulse メソッド](#))
  - ブザーの停止 ([addSound メソッド](#))
  - 復帰可能エラーからの復帰 ([recover メソッド](#))
  - リセット ([reset メソッド](#))
  - リアルタイムコマンドの送信 ([addCommand メソッド](#))

## SlipPrinter.timeout プロパティ

SlipPrinter.send メソッド実行後のチェック紙挿入待ち状態のタイムアウト時間を msec 単位で指定します。

---

### オブジェクト種類

Number

初期値：10000

## EndorsePrinter.timeout プロパティ

`EndorsePrinter.send` メソッド実行後のチェック紙挿入待ち状態のタイムアウト時間を msec 単位で指定します。

---

### オブジェクト種類

Number

初期値：10000

## MICRReader.timeout プロパティ

[MICRReader.read メソッド](#)や[MICRReader.cleaning メソッド](#)実行後のチェック紙 / クリーニングシート挿入待ち状態のタイムアウト時間を msec 単位で指定します。

---

### オブジェクト種類

#### Number

初期値：10000

interval プロパティ

[startMonitor メソッド](#)で有効にしたプリンターステータスの取得、更新間隔を指定します。

オブジェクト種類

Number

初期値：3000  
不正な値を指定した場合は、初期値が指定されます。

設定値	説明
1000 ~ 6000	プリンターステータスの更新間隔を msec 単位で指定

## onreceive イベント

HybridPrinter オブジェクトの実行結果を受け取ります。

### 構文

```
Function(data, sq);
```

### コールバック関数のパラメーター

パラメーター：data、sq

オブジェクト種類：Object、Number

### data オブジェクトのプロパティ

プロパティ	説明	オブジェクト種類
eventtype	イベントを返すメソッド名	String
success	イベントを返すメソッドの処理結果	Boolean
code	イベントを返すメソッドの実行結果	String
status	プリンターステータス	String
data	MICR 読み取り結果	String

#### success

値	説明
true	成功
false	失敗

#### code

値	説明
'SUCCESS'	成功
'EPTR_AUTOMATICAL'	自動復帰エラー発生
'EPTR_COVER_OPEN'	カバーオープンエラー発生
'EPTR_CUTTER'	オートカッターエラー発生
'EPTR_MECHANICAL'	メカニカルエラー発生
'EPTR_REC_EMPTY'	ロール紙エンド
'EPTR_UNRECOVERABLE'	復帰不可能エラー発生
'SchemaError'	印刷要求 XML の構文に誤りがある
'DeviceNotFound'	デバイス ID で指定したプリンターが存在しない
'PrintSystemError'	印刷システムにエラーが発生

値	説明
'EX_BADPORT'	通信ポートに異常を検出
'EX_TIMEOUT'	印刷タイムアウト発生
'ERROR_DEVICE_NOT_FOUND'	デバイスが見つからない
'ERROR_DEVICE_BUSY'	ポートのオープンに失敗
'CANCEL'	チェック紙挿入待ち状態を解除
'ERROR_TIMEOUT'	チェック紙挿入待ちタイムアウト
'ERROR_PARAMETER'	パラメーターエラーが発生した
'ERROR_NOT_SUPPORTED'	ハイブリッドプリンターが接続されていない
'EPTR_SCHEMAERROR'	印刷要求 XML の構文でエラー発生
'ERROR_COMMAND'	チェック紙挿入待ち状態を解除できない
'EMICR_ILLEGAL_LENGTH'	挿入したチェック紙が適切な長さではない
'EMICR_NO_MICR'	MICR データを検出できない
'EMICR_RECOGNITION'	読み取り不能な MICR テキストを検出した
'EMICR_READ'	MICR 読み取り中にエラー発生
'EMICR_NOISE_DETECTED'	ノイズエラーを検出
'EMICR_COVER_OPEN'	MICR 読み取り中にカバーが開けられた
'EMICR_PAPER_JAM'	紙ジャムエラー発生

## status

定数との AND 演算を行ってください。

定数 (status)	説明
ASB_NO_RESPONSE	プリンター無応答
ASB_PRINT_SUCCESS	印刷終了
ASB_DRAWER_KICK	ドロアーキックコネクタ 3 番ピンの状態 = "H"
ASB_OFF_LINE	オフライン状態
ASB_COVER_OPEN	カバーオープン
ASB_PAPER_FEED	紙送りスイッチによる紙送り中
ASB_WAIT_ON_LINE	オンライン復帰待ち中
ASB_PANEL_SWITCH	紙送りスイッチが押下中
ASB_MECHANICAL_ERR	メカニカルエラー発生
ASB_AUTOCUTTER_ERR	オートカッターエラー発生
ASB_UNRECOVER_ERR	復帰不可能エラー発生
ASB_AUTORECOVER_ERR	自動復帰エラー発生
ASB_SLIP_INSERT_WAIT	スリップ印字のチェック紙挿入待ち状態
ASB_RECEIPT_NEAR_END	ロール紙ニアエンド

定数 (status)	説明
ASB_SLIP_REMOVE_WAIT	スリップ印字のチェック紙排出待ち状態
ASB_RECEIPT_END	ロール紙エンド
ASB_TOF_NOPAPER	チェック紙が奥まで挿入されていない
ASB_BOF_NOPAPER	チェック紙が用紙ガイドの右端に正しくセットされていない
ASB_BUZZER	ブザー鳴動中（対応機種のみ）
ASB_SLIP_NO_SELECT	スリップ印字が選択されていない
ASB_SLIP_IMPOSSIBLE_PRINT	チェック紙に印字できない
ASB_SPOOLER_IS_STOPPED	スプーラー停止中（未使用）

## sq オブジェクトのプロパティ

プロパティ	説明	オブジェクト種類
sq	処理のシーケンス番号	Number

## 補足説明

[MICRReader.read メソッド](#)以外が実行された場合、data プロパティには null がセットされます。

# Scanner オブジェクト

## ondata イベント

バーコードスキャナーがバーコードから読み取った情報を受け取ります。

### 構文

```
Function(data);
```

### コールバック関数のパラメーター

パラメーター：data

オブジェクト種類：Object

### data オブジェクトのプロパティ

プロパティ	説明	オブジェクト種類
input	検出された文字列	String

# SimpleSerial オブジェクト

## sendCommand

任意コマンドを送信します。sendCommand 実行によるデバイスからの応答データは、[oncommandreply イベント](#)のコールバック関数に渡されます。

### 構文

```
sendCommand(data);
```

### パラメーター

data

設定値	説明
data	コマンドを文字列で指定

## oncommandreply イベント

`sendCommand` 実行によるデバイスからの応答データを受け取ります。

### 構文

```
Function(data);
```

### コールバック関数のパラメーター

パラメーター：data

オブジェクト種類：Object

### data オブジェクトのプロパティ

プロパティ	説明	オブジェクト種類
data	応答データの文字列	String

### 補足説明

デバイスからの応答データが分割されて通知されることがあるため、必ず `oncommandreply` イベントで全ての応答データを受信したかどうかを確認してください。

応答データが分割されて通知された場合、`oncommandreply` イベントが複数回発生します。後続データを待って処理を進めてください。

# DeviceHubTerminal オブジェクト

## shutdown メソッド

プリンターの電源をシャットダウンします。

### 構文

```
shutdown(password, callback);
```

### パラメーター

#### password

設定値	説明
password	EPSON TMNet WebConfig に設定されているパスワードを文字列で指定

#### callback

設定値	説明
callback	実行結果を受け取るコールバック関数を指定

### コールバック関数のパラメーター

パラメーター：data

オブジェクト種類：Object

### data オブジェクトのプロパティ

プロパティ	説明	オブジェクト種類
status	シャットダウンの実行結果	String

#### status

値	説明
"SUCCESS"	成功
"AUTHENTICATION_ERROR"	パスワード認証エラー
"ACCESS_DENIED"	シャットダウン設定が「許可しない」設定の状態で、シャットダウンを実行した
"SYSTEM_ERROR"	システムエラー

---

## 補足説明

- EPSON TMNet WebConfig を使用して、プリンターのシャットダウン設定を「許可する」に設定する必要があります。詳細は、各プリンターの詳細取扱説明書を参照してください。
- EPSON TMNet WebConfig のシャットダウン設定にて、パスワード認証設定を「認証する」に設定した場合、password に指定した文字列でパスワード認証を行います。

restart メソッド

プリンターを再起動します。

構文

```
restart (password, callback);
```

パラメーター

password

設定値	説明
password	EPSON TMNet WebConfig に設定されているパスワードを文字列で指定

callback

設定値	説明
callback	実行結果を受け取るコールバック関数を指定

コールバック関数のパラメーター

パラメーター：data  
オブジェクト種類：Object

data オブジェクトのプロパティ

プロパティ	説明	オブジェクト種類
status	再起動の実行結果	String

status

値	説明
"SUCCESS"	成功
"AUTHENTICATION_ERROR"	パスワード認証エラー
"ACCESS_DENIED"	シャットダウン設定が「許可しない」設定の状態で、再起動を実行した
"SYSTEM_ERROR"	システムエラー

## 戻り値

戻り値	説明	オブジェクト種類
sq	処理のシーケンス番号	String

## 補足説明

- ❑ EPSON TMNet WebConfig を使用して、プリンターのシャットダウン設定を「許可する」に設定する必要があります。詳細は、各プリンターの詳細取扱説明書を参照してください。
- ❑ EPSON TMNet WebConfig のシャットダウン設定にて、パスワード認証設定を「認証する」に設定した場合、password に指定した文字列でパスワード認証を行います。

# CommBoxManager オブジェクト

## openCommBox メソッド

コミュニケーションボックスをオープンします。

TM-DT ソフトウェア /TM-i ファームウェアはコミュニケーションボックスを作成し、アプリケーションをコミュニケーションボックスのメンバーに追加します。

### 構文

```
openCommBox(boxID, option, callback);
```

### パラメーター

boxID

設定値	説明
boxID	オープンするボックス ID を文字列で指定

option

プロパティ	説明	オブジェクト種類
memberID	コミュニケーションボックス内で、自身を識別するためのメンバー ID を指定	String

callback

オープン処理結果を受け取るコールバック関数を指定します。

第 1 パラメーターに commBox オブジェクト、第 2 パラメーターには、結果を示す以下の文字列が渡されます。

第 3 パラメーターには戻り値と同じシーケンス番号が渡されます。

文字列	説明
"OK"	コミュニケーションボックスのオープンに成功
"ALREADY_OPENED"	すでにコミュニケーションボックスがオープンしている
"MEMBERID_ALREADY_USED"	指定したメンバー ID はすでに使用されている
"BOX_COUNT_OVER"	コミュニケーションボックスを作成可能な最大数を超えている
"BOX_CLIENT_OVER"	コミュニケーションボックスに所属可能な最大数を超えている
"PARAMETER_ERROR"	パラメーターエラーが発生した
"SYSTEM_ERROR"	その他のエラーが発生した

---

## 戻り値

戻り値	説明	オブジェクト種類
sq	処理のシーケンス番号	String

---

## 補足説明

openCommBox メソッドの処理に失敗すると、第 1 パラメーターには null が渡されます。

closeCommBox メソッド

コミュニケーションボックスをクローズします。  
TM-DT ソフトウェア /TM-i ファームウェアはアプリケーションをコミュニケーションボックスのメンバーから削除し、メンバーのいなくなったコミュニケーションボックスは破棄されます。

構文

```
closeCommBox(commBoxObj, callback);
```

パラメーター

commBoxObj

設定値	説明
commBoxObj	クローズする commBox オブジェクトを指定

callback

クローズ処理結果を受け取るコールバック関数を指定します。  
第 1 パラメーターに結果を示す以下の文字列が渡され、第 2 パラメーターには戻り値と同じシーケンス番号が渡されます。

文字列	説明
"OK"	コミュニケーションボックスのクローズに成功
"NOT_OPENED"	コミュニケーションボックスがオープンしていない
"PARAMETER_ERROR"	パラメーターエラーが発生した
"SYSTEM_ERROR"	その他のエラーが発生した

戻り値

戻り値	説明	オブジェクト種類
sq	処理のシーケンス番号	String

# CommBox オブジェクト

## getCommHistory メソッド

コミュニケーションボックスの送信履歴を取得します。

### 構文

- ❑ `getCommHistory(callback);`
- ❑ `getCommHistory(option, callback);`

### パラメーター

#### option

設定値	説明
true	コミュニケーションボックス内全ての送信履歴を取得
false	自身のメンバー ID 宛に送信されたデータの送信履歴のみを取得

#### callback

コミュニケーションボックスの送信履歴を受け取るコールバック関数を指定します。

第 1 パラメーターに結果を示す以下の文字列が渡されます。

文字列	説明
"OK"	送信履歴の取得に成功
"NOT_OPENED"	コミュニケーションボックスが開いていない
"SYSTEM_ERROR"	その他のエラーが発生した

第 2 パラメーターに以下の historyList が渡されます。

historyList.(index)	説明	オブジェクト種類
senderId	送信元の memberId	String
receiverId	送信先の memberId	String
message	受信したメッセージ	Hash

第 3 パラメーターには戻り値と同じシーケンス番号が渡されます。

### 戻り値

戻り値	説明	オブジェクト種類
sq	処理のシーケンス番号	String

---

## 補足説明

option を省略すると、false が指定されます。

## send メソッド

コミュニケーションボックスにデータを送信します。

### 構文

```
send(message, memberId, callback);
```

### パラメーター

#### message

設定値	説明
message	送信するデータを指定

#### memberId

設定値	説明
memberId	送信先のメンバー ID を指定

#### callback

send メソッドの実行結果を受け取るコールバック関数を指定します。

第 1 パラメーターに結果を示す以下の文字列が渡されます。

文字列	説明
"OK"	メッセージの送信に成功
"NOT_OPENED"	コミュニケーションボックスが開いていない
"MEMBER_NOT_FOUND"	指定したメンバーが見つからない
"SYSTEM_ERROR"	その他のエラーが発生した

第 2 パラメーターにデータの送信を試みたクライアント数が渡され、第 3 パラメーターには戻り値と同じシーケンス番号が渡されます。

### 戻り値

戻り値	説明	オブジェクト種類
sq	処理のシーケンス番号	String

### 補足説明

memberId に null もしくは "" (空文字) を指定すると、送信者を除くコミュニケーションボックス内全てのメンバーに送信します。

## onreceive イベント

コミュニケーションボックスが受信した内容をアプリケーションに通知します。

### 構文

```
Function(data);
```

### コールバック関数のパラメーター

パラメーター：data

オブジェクト種類：Object

### data オブジェクトのプロパティ

プロパティ	説明	オブジェクト種類
senderId	送信元のメンバー ID	String
receiverId	送信先のメンバー ID	String
message	受信データ	String

### 補足説明

- ❑ [openCommBox メソッド](#) 実行時にメンバー ID の指定がない場合、senderId には ""（空文字）がセットされます。
- ❑ コミュニケーションボックス内全てのメンバーに送信した場合、receiverId には ""（空文字）がセットされます。

# ePosDeviceConfiguration オブジェクト

## Constructor

ePosDeviceConfiguration オブジェクトを作成します。

## 構文

```
ePosDeviceConfiguration (address) ;
```

## パラメーター

address

設定値	説明
address	接続先プリンターの IP アドレスを指定

## getRegisteredDevices メソッド

プリンターに登録されているデバイスの利用可能状況を取得し、アプリケーションに通知します。

### 構文

```
getRegisteredDevices(deviceGroup, callback);
```

### パラメーター

#### deviceGroup

設定値	説明
DEVICE_GROUP_ALL	全てのデバイスを取得

#### callback

getRegisteredDevices メソッドの実行結果を受け取るコールバック関数を指定します。

第 1 パラメーターに ePosDeviceConfiguration オブジェクトが渡され、第 2 パラメーターには結果を示す以下の値が渡されます。

プロパティ	説明
deviceId	デバイス ID
deviceType	デバイスの種類
status	デバイスの状態

- deviceType

値	説明
type_cash_changer	自動釣り銭機
type_display	カスタマーディスプレイ
type_keyboard	キーボード
type_msr	MSR
type_printer	プリンター
type_hybrid_printer	ハイブリッドプリンター
type_scanner	バーコードスキャナー
type_simple_serial	シリアル通信デバイス

- status

値	説明
online	利用可能
offline	利用不可能
unknown	状態不明

---

## 補足説明

deviceGroup パラメーターの指定は、DEVICE\_GROUP\_ALL 固定です。

# Error Code一覧

## callback パラメーターで取得するエラーコードと対処方法

エラーコード	要因	対処方法
PARAM_ERROR	不正なパラメーターが渡された。 < 例 > サポート範囲外の数値が指定された。	パラメーターの指定値を確認してください。
ERROR_PARAMETER		
PARAMETER_ERROR		
ERROR_TIMEOUT	指定時間内にデバイスと通信できなかった。	タイムアウト時間を確認してください。 タイムアウト時間は、印刷所要時間以上に設定してください。
DEVICE_NOT_FOUND	指定対象を発見できなかった。 < 例 > 指定した接続先のプリンターが存在しない。	接続形式や IP アドレス等が間違っていないか、確認してください。
DEVICE_IN_USE	指定されたデバイスが使用中。 < 例 > 他のアプリケーションソフトで使われている。	他のアプリケーションソフトからの使用を終了してください。
ERR_TYPE_INVALID	デバイスのオブジェクトが違う (Printer、Display)。	接続しているデバイスの種類を確認し、適切なデバイスオブジェクトで接続してください。
	ESC/POS をサポートしていない TM プリンターです。	TM インテリジェントプリンターとして使用してください。
DEVICE_OPEN_ERROR	デバイスの接続に失敗した。	<ul style="list-style-type: none"> <li>接続形式や IP アドレス等が間違っていないか、確認してください。</li> <li>デバイスとの接続状態を確認してください。</li> </ul>
DEVICE_NOT_OPEN	<ul style="list-style-type: none"> <li>デバイスと接続されいない。</li> <li><a href="#">createDevice メソッド</a>が呼ばれていない状態で、デバイスを制御しようとした。</li> </ul>	<ul style="list-style-type: none"> <li>通信状態を確認し、デバイスと接続してください。</li> <li><a href="#">createDevice メソッド</a>が呼ばれているか確認してください。</li> </ul>
DEVICE_CLOSE_ERROR	デバイスの切断に失敗した。	デバイスとの通信状態を確認してください。
ALREADY_OPENED	既にコミュニケーションボックスがオープンしている。	コミュニケーションボックスとの通信を終了してください。
MEMBERID_ALREADY_USED	指定したメンバー ID は既に使用されている。	異なるメンバー ID を指定してください。

エラーコード	要因	対処方法
MEMBER_NOT_FOUND	指定対象を発見できなかった。 < 例 > コミュニケーションボックスで、 指定したデータ送信先のメンバー が存在しない。	指定したメンバー ID と、送信先デバ イスの通信状態を確認してください。
BOX_COUNT_OVER	コミュニケーションボックスを作 成可能な最大数を超過している。	不要なコミュニケーションボックス を削除してください。
BOX_CLIENT_OVER	コミュニケーションボックスに所 属可能な最大数を超過している。	コミュニケーションボックスを使用 していないメンバーを除名してくだ さい。
NOT_OPENED	コミュニケーションボックスが オープンしていない。	コミュニケーションボックスをオー プンしてください。
SYSTEM_ERROR	その他のエラーが発生した。	実行環境に問題がないか確認してく ださい。

## onreceive イベントで取得するエラーコードと対処方法

## エラーコード

□ Printer オブジェクト

エラーコード	要因	対処方法
EPTR_AUTOMATICAL	自動復帰エラーが発生した。	プリンター復帰後に処理を実行してください。
EPTR_BATTERY_LOW	バッテリー残量がなくなった。	バッテリーを交換するか、AC アダプターを接続してください。
EPTR_COVER_OPEN	カバーが開いている。	プリンターのカバーを閉じてください。
EPTR_CUTTER	オートカッターエラーが発生した。	エラー原因を取り除き、プリンターの電源を入れ直してください。
EPTR_MECHANICAL	メカニカルエラーが発生した。	エラー原因を取り除き、プリンターの電源を入れ直してください。
EPTR_REC_EMPTY	用紙なし。	用紙を補充してください。
EPTR_UNRECOVERABLE	復帰不可能エラーが発生した。	プリンターの電源を入れ直してください。 再発する場合は、購入先かサービスセンターに連絡してください。
SchemaError	印刷要求 XML の構文に誤りがある	印刷要求 XML の構文を確認してください。
DeviceNotFound	<ul style="list-style-type: none"> <li>接続形式や IP アドレスなどの指定が間違っている。</li> <li>指定したデバイスが接続されていない。</li> <li>指定したデバイスがプリンターに登録されていない。</li> </ul>	<ul style="list-style-type: none"> <li>接続形式や IP アドレスなどの指定値を確認してください。</li> <li>デバイスの接続状況を確認してください。</li> <li>該当デバイスのプリンターへの登録状態を確認してください。</li> </ul>
PrintSystemError	TM-i ファームウェア、TM-DT ソフトウェアに異常が発生した。	TM インテリジェントプリンターの電源を入れ直し、OS を再起動してください。
EX_BADPORT	オンライン状態で強制送信処理を実行した。	オフライン状態で強制送信処理を実行してください。
EX_TIMEOUT	処理の実行時間が、指定したタイムアウト時間を超過した。	タイムアウト時間を調整してください。
EX_SPOOLER	スプーラー容量を超えて、印刷データが送信された。	プリンターとの通信が切れていないか、接続状態を確認してください。
JobNotFound	存在しない印刷ジョブ ID が指定された。	印刷ジョブ ID の指定先を確認してください。
Printing	印刷中の印刷ジョブ ID を指定して、 <a href="#">getPrintJobStatus メソッド</a> が実行された。	印刷完了後に処理を実行してください。

## □ HybridPrinter オブジェクト

エラーコード	要因	対処方法
EPTR_AUTOMATIC	自動復帰エラーが発生した。	プリンター復帰後に処理を実行してください。
EPTR_COVER_OPEN	カバーが開いている。	プリンターのカバーを閉じてください。
EPTR_CUTTER	オートカッターエラーが発生した。	エラー原因を取り除き、プリンターの電源を入れ直してください。
EPTR_MECHANICAL	メカニカルエラーが発生した。	エラー原因を取り除き、プリンターの電源を入れ直してください。
EPTR_REC_EMPTY	用紙なし。	用紙を補充してください。
EPTR_UNRECOVERABLE	復帰不可能エラーが発生した。	プリンターの電源を入れ直してください。 再発する場合は、購入先かサービスセンターに連絡してください。
SchemaError	印刷要求 XML の構文に誤りがある	印刷要求 XML の構文を確認してください。
DeviceNotFound	<ul style="list-style-type: none"> <li>接続形式や IP アドレスなどの指定が間違っている。</li> <li>指定したデバイスが接続されていない。</li> </ul>	<ul style="list-style-type: none"> <li>接続形式や IP アドレスなどの指定値を確認してください。</li> <li>デバイスの接続状況を確認してください。</li> </ul>
PrintSystemError	TM-i ファームウェア、TM-DT ソフトウェアに異常が発生した。	TM インテリジェントプリンターの電源を入れ直し、OS を再起動してください。
EX_BADPORT	オンライン状態で強制送信処理を実行した。	オフライン状態で強制送信処理を実行してください。
EX_TIMEOUT	処理の実行時間が、指定したタイムアウト時間を超過した。	タイムアウト時間を調整してください。
ERROR_DEVICE_NOT_FOUND	<ul style="list-style-type: none"> <li>接続形式や IP アドレスなどの指定が間違っている。</li> <li>指定したデバイスが接続されていない。</li> <li>指定したデバイスがプリンターに登録されていない。</li> </ul>	<ul style="list-style-type: none"> <li>接続形式や IP アドレスなどの指定値を確認してください。</li> <li>デバイスの接続状況を確認してください。</li> <li>該当デバイスのプリンターへの登録状態を確認してください。</li> </ul>
ERROR_DEVICE_BUSY	<ul style="list-style-type: none"> <li>指定されたデバイスが使用中。</li> <li>スリップ印刷中。</li> </ul>	<ul style="list-style-type: none"> <li>他のアプリケーションソフトからの使用を終了してください。</li> <li>スリップ印刷完了後に操作してください。</li> </ul>
ERROR_TIMEOUT	チェック紙挿入待ち時間が、指定したタイムアウト時間を超過した。	タイムアウト時間を調整してください。
ERROR_PARAMETER	不正なパラメーターが渡された。	パラメーターの指定値を確認してください。
ERROR_NOT_SUPPORTED	デバイスでサポートしていない機能を指定した。	接続デバイスのサポート機能を確認してください。

エラーコード	要因	対処方法
EPTR_SCHEMAERROR	印刷要求 XML の構文に誤りがある	印刷要求 XML の構文を確認してください。
ERROR_COMMAND	無効なコマンドが渡された。	デバイスの状態を確認してください。
EMICR_ILLEGAL_LENGTH	用紙の長さが適切でない。	用紙の長さを確認してください。
EMICR_NO_MICR	MICR データを検出できない。	チェック紙を確認し、必要に応じて MICR 機構のクリーニングを実施してください。
EMICR_RECOGNITION	読み取り不能な MICR テキストを検出した。	チェック紙の内容を確認してください。
EMICR_READ	MICR 読み取りエラーが発生した。	エラー原因を取り除き、MICR 読み取りを実施してください。再度エラーが発生する場合は、プリンターを再起動してください。
EMICR_NOISE_DETECTED	ノイズ計測でエラーが発生した。	エラー原因を取り除き、ノイズ計測を実施してください。再度エラーが発生する場合は、プリンターを再起動してください。
EMICR_COVER_OPEN	MICR 読み取り中にカバーが開けられた。	カバーを閉じてください。
EMICR_PAPER_JAM	紙ジャムエラーが発生した。	詰まった用紙を取り除いてください。

## □ Display オブジェクト

エラーコード	要因	対処方法
EDSP_NOT_FOUND	<ul style="list-style-type: none"> <li>接続形式や IP アドレスなどの指定が間違っている。</li> <li>指定したデバイスが接続されていない。</li> </ul>	<ul style="list-style-type: none"> <li>接続形式や IP アドレスなどの指定値を確認してください。</li> <li>デバイスの接続状況を確認してください。</li> </ul>
EDSP_NOT_OPEN	デバイスが破損している可能性がある。	プリンターを再起動し、再度処理を行ってください。再発する場合は、購入先またはサービスサポートまで問い合わせてください。
EDSP_INVALID_WINDOW	未登録のウィンドウが指定された。	指定するウィンドウ番号を確認してください。
EX_BADPORT	EPSON TMNet WebConfig で、ディスプレイの使用が無効に設定されている。	EPSON TMNet WebConfig の設定を確認してください。
EX_TIMEOUT	処理の実行時間が、指定したタイムアウト時間を超過した。	タイムアウト時間を調整してください。
EX_INVALID_VALUE	不正なパラメーターが渡された。	パラメーターの指定値を確認してください。

## ステータス

### □ プリンターステータス

定数 (status)	要因	対処方法
ASB_NO_RESPONSE	<ul style="list-style-type: none"> <li>プリンターの電源が入っていない。</li> <li>プリンターと接続されていない。</li> </ul>	プリンターの電源や通信状態を確認してください。
ASB_BATTERY_OFFLINE	バッテリー残量少のため、オフライン状態。	バッテリーを交換するか、AC アダプターに接続してください。
ASB_OFF_LINE	オフライン状態。	カバーオープンや用紙切れなど、オフラインとなる要因を取り除いてください。
ASB_COVER_OPEN	カバーが開いている。	プリンターのカバーを閉じてください。
ASB_PAPER_FEED	紙送りボタンによる紙送信中。	紙送り停止後に処理を実行してください。
ASB_WAIT_ON_LINE	オンライン復帰待ち状態。	プリンターをオンラインに復帰させてください。
ASB_PANEL_SWITCH	パネルスイッチ操作中。	操作完了後に処理を実行してください。
ASB_MECHANICAL_ERR	メカニカルエラーが発生した。	エラー原因を取り除き、プリンターの電源を入れ直してください。
ASB_AUTOCUTTER_ERR	オートカッターエラーが発生した。	エラー原因を取り除き、プリンターの電源を入れ直してください。
ASB_UNRECOVER_ERR	復帰不可能エラーが発生した。	プリンターの電源を入れ直してください。 再発する場合は、購入先かサービスセンターに連絡してください。
ASB_AUTORECOVER_ERR	自動復帰エラーが発生した。	プリンター復帰後に処理を実行してください。
ASB_RECEIPT_NEAR_END	用紙が残り少ない。	用紙を交換してください。
ASB_RECEIPT_END	用紙なし。	用紙を補充してください。
ASB_BUZZER	ブザー鳴動中。	各プリンターのブザー鳴動条件を確認し、ブザー鳴動原因を取り除いてから処理を実行してください。
ASB_SPOOLER_IS_STOPPED	スプーラー容量を超えて、印刷データが送信された。	プリンターとの通信が切れていないか、接続状態を確認してください。

### □ バッテリーステータス

定数 (battery)	要因	対処方法
0x31XX	AC アダプターが接続されていない。	AC アダプターの接続状態を確認してください。

定数 (battery)	要因	対処方法
0xXX31	バッテリー残量少。	バッテリーを充電するか、AC アダプターに接続してください。
0xXX30	バッテリー残量なし。	バッテリーを充電するか、AC アダプターに接続してください。

## Key Code一覧

定数	コード	定数	コード
VK_BACK	0x08	VK_L	0x4C
VK_TAB	0x09	VK_M	0x4D
VK_RETURN	0x0D	VK_N	0x4E
VK_SHIFT	0x10	VK_O	0x4F
VK_CONTROL	0x11	VK_P	0x50
VK_MENU	0x12	VK_Q	0x51
VK_CAPITAL	0x14	VK_R	0x52
VK_ESCAPE	0x1B	VK_S	0x53
VK_CONVERT	0x1C	VK_T	0x54
VK_NONCONVERT	0x1D	VK_U	0x55
VK_SPACE	0x20	VK_V	0x56
VK_PRIOR	0x21	VK_W	0x57
VK_NEXT	0x22	VK_X	0x58
VK_END	0x23	VK_Y	0x59
VK_HOME	0x24	VK_Z	0x5A
VK_LEFT	0x25	VK_MULTIPLY	0x6A
VK_UP	0x26	VK_ADD	0x6B
VK_RIGHT	0x27	VK_SUBTRACT	0x6D
VK_DOWN	0x28	VK_F1	0x70
VK_INSERT	0x2D	VK_F2	0x71
VK_DELETE	0x2E	VK_F3	0x72
VK_0	0x30	VK_F4	0x73
VK_1	0x31	VK_F5	0x74
VK_2	0x32	VK_F6	0x75
VK_3	0x33	VK_F7	0x76
VK_4	0x34	VK_F8	0x77
VK_5	0x35	VK_F9	0x78
VK_6	0x36	VK_F10	0x79
VK_7	0x37	VK_F11	0x7A
VK_8	0x38	VK_F12	0x7B
VK_9	0x39	VK_OEM_1	0xBA
VK_A	0x41	VK_OEM_PLUS	0xBB
VK_B	0x42	VK_OEM_COMMA	0xBC

定数	コード	定数	コード
VK_C	0x43	VK_OEM_MINUS	0xBD
VK_D	0x44	VK_OEM_PERIOD	0xBE
VK_E	0x45	VK_OEM_2	0xBF
VK_F	0x46	VK_OEM_3	0xC0
VK_G	0x47	VK_OEM_4	0xDB
VK_H	0x48	VK_OEM_5	0xDC
VK_I	0x49	VK_OEM_6	0xDD
VK_J	0x4A	VK_OEM_7	0xDE
VK_K	0x4B	VK_OEM_ATTN	0xF0

# デバイス制御スクリプト

本章では、お客様がデバイス制御スクリプトを開発するために必要な情報について説明しています。

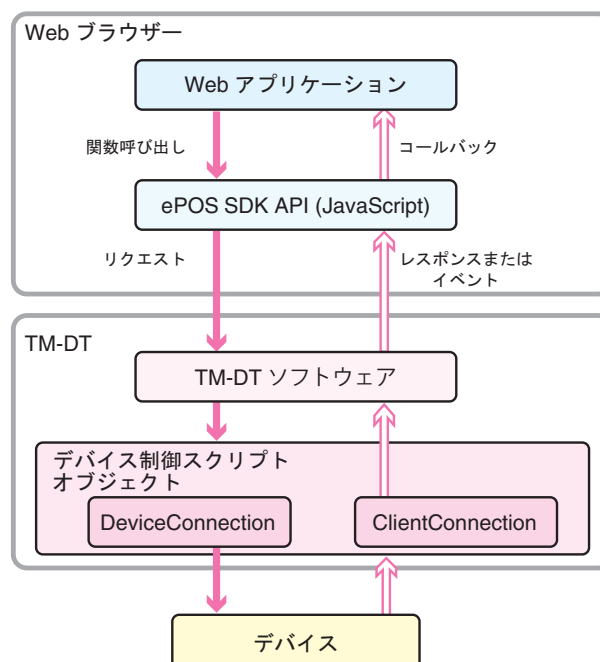


TM インテリジェントプリンターの TM-DT シリーズのみ使用できます。

## プログラミング

### デバイス制御スクリプトの使用について

Epson ePOS SDK for JavaScript で提供するデバイス制御スクリプト用 API を利用すると、デバイスデータ処理のカスタマイズや、新たなデバイスを Web アプリケーションから使用できます。



ePOSDevice オブジェクトの [createDevice メソッド](#) が実行されると、TM-DT ソフトウェアが要求されたデバイスに対応するデバイス制御スクリプトを使用できるようにオブジェクトを生成します。

生成されたオブジェクトにより、デバイスを制御できます。

### デバイス制御スクリプトのオブジェクト

デバイス制御スクリプトには、TM-DT ソフトウェアから以下のオブジェクトが渡されます。これらのオブジェクトを使用して、デバイス制御スクリプトは Web アプリケーションおよびデバイスと通信します。

オブジェクト	説明
ClientConnection	Web ブラウザー側のデバイスオブジェクトにデータを送信するためのオブジェクト
DeviceConnection	デバイスとデータを送受信するためのオブジェクト

## デバイス制御スクリプトのオブジェクト使用した機能

デバイス制御スクリプト用 API を利用して以下の機能を使用できます。

- Web ブラウザー側デバイスオブジェクトの任意のイベントを呼び出し
- デバイスにデータを送信
- デバイスから発生したデータを受信

## デバイス制御スクリプトの構成

デバイス制御スクリプトは、以下の条件でコーディングしてください。

- デバイス制御に必要なコードは、1 つのファイルにまとめて記述すること。  
(EPSON TMNet WebConfig を使用したデバイスの設定は、1 つのファイルしか登録できません。)
- ファイル名の最初のドット "." までの名称と、コンストラクターの名称を同じにすること。  
例) ファイル名: Keyboard\_Generic.ver1.0.js → コンストラクターの名称: Keyboard\_Generic
- コンストラクターの外部参照のために、exports 宣言すること。  
例) exports.Keyboard\_Generic = Keyboard\_Generic;
- デバイス制御スクリプトには、以下のプロパティが必要です。コンストラクターで適切な名称を設定すること。
  - DEVICE\_TYPE プロパティ (オブジェクト種類: String)

設定値	説明
type_cash_changer	自動釣り銭機を使用する場合に指定します。
type_keyboard	キーボードデバイスを使用する場合に指定します。
type_msr	MSR を使用する場合に指定します。
type_scanner	バーコードスキャナーを使用する場合に指定します。
type_simple_serial	単純なシリアル通信する場合に指定します。

- DEVICE\_GROUP プロパティ (オブジェクト種類: String)

設定値	説明
group_hid	HID ドライバーで動作可能なキー入力デバイスを使用する場合に指定します。
group_serial	シリアル通信デバイスを使用する場合に指定します。
group_other	その他のデバイスを使用する場合に指定します。

- デバイスで発生したデータを受け取るため onDeviceData メソッドを用意すること。  
詳細は、[デバイス制御スクリプト名オブジェクト](#)を参照してください。
- Web ブラウザー側で動作するデバイスオブジェクトのメソッドに対応するメソッドを用意すること。  
詳細は、[任意イベント](#)を参照してください。

## デバイス制御スクリプトの構成例

```
// exports 宣言
exports.Keyboard_Generic = Keyboard_Generic;

// 2つの引数を持ち、ファイル名と同名
function Keyboard_Generic(clientConn, deviceConn){
// DEVICE_TYPE プロパティを持つ
  this.DEVICE_TYPE = 'type_keyboard';
// DEVICE_GROUP プロパティを持つ
  this.DEVICE_GROUP = 'group_hid';
  this.clientConn = clientConn;
  this.deviceConn = deviceConn;
  .....
}

Keyboard_Generic.prototype = {
// onDataData メソッドを持つ
  onDataData : function(event, keycode, ascii){...},
// デバイスオブジェクト対応したメソッドがある
  setprefix : function(data){...}
}
```

## デバイス制御スクリプト API 一覧

デバイス制御スクリプト API には、以下のオブジェクトが用意されています。

- ❑ [ClientConnection](#) オブジェクト
- ❑ [DeviceConnection](#) オブジェクト
- ❑ [デバイス制御スクリプト名オブジェクト](#)

### ClientConnection オブジェクト

デバイス制御スクリプトのコンストラクターの第 1 パラメーターに渡されるオブジェクトです。

API		説明
送信	<a href="#">send</a>	Web ブラウザーで動作するデバイスオブジェクトにデータを送信

### DeviceConnection オブジェクト

デバイス制御スクリプトのコンストラクターの第 2 パラメーターに渡されるオブジェクトです。

API		説明
送信	<a href="#">send</a>	シリアル通信デバイスにデータを送信

### デバイス制御スクリプト名オブジェクト

デバイスからのデータ受信用のオブジェクトです。

API		説明
結果受信	<a href="#">onDeviceData イベント (キー入力デバイス)</a>	キー入力デバイスからのデータ受信イベント
	<a href="#">onDeviceData イベント (シリアル通信デバイス)</a>	シリアル通信デバイスからのデータ受信イベント
	<a href="#">任意イベント</a>	Web ブラウザーで動作するデバイスオブジェクトの API 実行結果受信イベント

# ClientConnection オブジェクト

## send

Web ブラウザーで動作するデバイスオブジェクトにデータを送信します。

### 構文

```
send(event, data);
```

### パラメーター

#### event

設定値	説明
String	デバイスオブジェクトのイベント名を指定

#### data

設定値	説明
Object	デバイスオブジェクトのイベントに渡すデータを指定

### サンプルプログラム

デバイスオブジェクトの onkeypress イベントを呼び出し、onkeypress イベントの data パラメーターを使用して data.keycode から 49、data.ascii から '1' を取得する。

```
data = {'keycode' : 49, 'ascii' : '1'};  
clientConn.send('onkeypress', data)
```

## DeviceConnection オブジェクト

### send

シリアル通信デバイスにデータを送信します。

#### 構文

```
send(data);
```

#### パラメーター

data

設定値	説明
Buffer	デバイスに送信するデータを指定

#### 補足説明

HID ドライバーで動作可能な入力デバイスには送信できません。

# デバイス制御スクリプト名オブジェクト

## onDeviceData イベント(キー入力デバイス)

HID ドライバーで動作可能なキー入力デバイスから検出されたデータを受信するイベントです。  
キー入力デバイス用のデバイス制御スクリプトの場合、本形式のイベントを記述してください。

### 構文

```
onDeviceData(event, keycode, ascii);
```

### パラメーター

#### event

キー操作の方向を受け取ります。

値	説明
1	キーダウン
2	キーアップ

#### keycode

値	説明
Number	キーコード

#### ascii

値	説明
String	操作されたキーに対応する文字

### 補足説明

- ❑ keycode で受け取るキーコードについては、[Key Code 一覧](#)を参照してください。
- ❑ ascii で受け取ったキーコードに対応する文字が無い場合、undefined が入ります。

## onDeviceData イベント(シリアル通信デバイス)

シリアル通信デバイスから検出されたデータを受信するイベントです。

シリアル通信デバイス用のデバイス制御スクリプトの場合、本形式のイベントを記述してください。

---

### 構文

```
onDeviceData (data) ;
```

---

### パラメーター

data

値	説明
Buffer	シリアル通信デバイスから受信したデータ

## 任意イベント

Web ブラウザーで動作するデバイスオブジェクトの API 実行結果を受信するイベントです。

### 構文

```
Name specified with callEvent(data);
```

### パラメーター

data

値	説明
Object	デバイスオブジェクトの <a href="#">callEvent メソッド</a> でパラメーター指定したオブジェクト

# 機器仕様

プリンターや POS 周辺機器などの機器側に依存する API 使用制限や、パラメーターの設定値についての情報を掲載します。

## クラス別対応プリンター

クラス別の対応プリンターは、以下の通りです。

クラス	対応プリンター
<a href="#">ePOSDevice オブジェクト</a>	全プリンター
<a href="#">デバイスオブジェクト共通</a>	TM-DT シリーズ
<a href="#">CashChanger オブジェクト</a>	TM-DT シリーズ
<a href="#">Display オブジェクト</a>	以下のプリンターとカスタマーディスプレイの組み合わせで使用できます。 <ul style="list-style-type: none"> <li>• TM インテリジェントプリンター +DM-D110</li> <li>• TM-m30+DM-D30</li> </ul>
<a href="#">Keyboard オブジェクト</a>	TM インテリジェントプリンター
<a href="#">MSR オブジェクト</a>	TM-DT シリーズ
<a href="#">Printer オブジェクト</a>	全プリンター プリンターごとに API の使用可否や、使用制限があります。 <a href="#">サポート API 一覧</a> を参照してください。
<a href="#">HybridPrinter オブジェクト</a>	ハイブリットプリンター
<a href="#">Scanner オブジェクト</a>	TM インテリジェントプリンター
<a href="#">SimpleSerial オブジェクト</a>	TM インテリジェントプリンター
<a href="#">DeviceHubTerminal オブジェクト</a>	TM-DT シリーズ
<a href="#">CommBoxManager オブジェクト</a>	TM インテリジェントプリンター
<a href="#">CommBox オブジェクト</a>	TM インテリジェントプリンター
<a href="#">ePosDeviceConfiguration オブジェクト</a>	TM-DT ソフトウェア Ver.3.0 以降の TM-DT シリーズ

## サポート API 一覧

各プリンターのサポート API を一覧表で掲載します。

表内記号の意味は、以下の通りです。

- ：対応している。
- △：対応しているが、指定できるパラメーター設定値に制限がある。
- ：対応していない。

「△」の制限については、[プリンター別サポート情報](#)を参照してください。

### Printer オブジェクト

#### TM プリンター

API	TM-m10	TM-m30	TM-P20	TM-P60II (Peeler)	TM-T20	TM-T20II	TM-T70	TM-T70II	TM-T88V	TM-T90II
<a href="#">addTextAlign メソッド</a>	○	○	○	○	○	○	○	○	○	○
<a href="#">addTextLineSpace メソッド</a>	○	○	○	○	○	○	○	○	○	○
<a href="#">addTextRotate メソッド</a>	○	○	○	○	○	○	○	○	○	○
<a href="#">addText メソッド</a>	○	○	○	○	○	○	○	○	○	○
<a href="#">addTextLang メソッド</a>	○	○	○	○	○	○	○	○	○	○
<a href="#">addTextFont メソッド</a>	△	△	△	△	△	△	△	△	△	△
<a href="#">addTextSmooth メソッド</a>	○	○	○	○	○	○	○	○	○	○
<a href="#">addTextDouble メソッド</a>	○	○	○	○	○	○	○	○	○	○
<a href="#">addTextSize メソッド</a>	○	○	○	○	○	○	○	○	○	○
<a href="#">addTextStyle メソッド</a>	△	△	△	△	△	△	△	△	△	△
<a href="#">addTextPosition メソッド</a>	○	○	○	○	○	○	○	○	○	○
<a href="#">addTextVPosition メソッド</a>	○	○	○	○	○	○	○	○	○	○
<a href="#">addFeedUnit メソッド</a>	○	○	○	○	○	○	○	○	○	○
<a href="#">addFeedLine メソッド</a>	○	○	○	○	○	○	○	○	○	○
<a href="#">addFeedPosition メソッド</a>	-	-	△	○	-	-	-	-	-	-

API	TM-m10	TM-m30	TM-P20	TM-P60II (Peeler)	TM-T20	TM-T20II	TM-T70	TM-T70II	TM-T88V	TM-T90II
addFeed メソッド	○	○	○	○	○	○	○	○	○	○
addImage メソッド	△	△	△	△	△	△	△	△	△	△
addLogo メソッド	○	○	○	○	○	○	○	○	○	○
addBarcode メソッド	△	△	○	△	△	△	△	△	△	△
addSymbol メソッド	△	△	△	△	△	△	△	△	△	△
addHLine メソッド	-	-	-	-	-	-	-	-	-	-
addVLineBegin メソッド	-	-	-	-	-	-	-	-	-	-
addVLineEnd メソッド	-	-	-	-	-	-	-	-	-	-
addPageBegin メソッド	○	○	○	○	○	○	○	○	○	○
addPageEnd メソッド	○	○	○	○	○	○	○	○	○	○
addPageArea メソッド	○	○	○	○	○	○	○	○	○	○
addPageDirection メソッド	○	○	○	○	○	○	○	○	○	○
addPagePosition メソッド	○	○	○	○	○	○	○	○	○	○
addPageLine メソッド	-	-	△	△	-	-	-	-	-	-
addPageRectangle メソッド	-	-	△	△	-	-	-	-	-	-
addCut メソッド	△	△	△	△	△	△	△	△	△	△
addPulse メソッド	○	○	-	-	○	○	○	○	○	○
addSound メソッド	△	△	△	△	△	△	-	△	△	-
addLayout メソッド	-	-	△	○	-	-	-	-	-	-
recover メソッド	○	○	○	○	○	○	○	○	○	○
addRecovery メソッド	○	○	○	○	○	○	○	○	○	○
reset メソッド	○	○	○	○	○	○	○	○	○	○
addReset メソッド	○	○	○	○	○	○	○	○	○	○
addCommand メソッド	○	○	○	○	○	○	○	○	○	○

API	TM-m10	TM-m30	TM-P20	TM-P60II (Peeler)	TM-T20	TM-T20II	TM-T70	TM-T70II	TM-T88V	TM-T90II
send メソッド	○	○	○	○	○	○	○	○	○	○
print メソッド	△	△	△	△	△	△	△	△	○	○
getPrintJobStatus メソッド	○	○	○	○	○	○	○	○	○	○
startMonitor メソッド	○	○	○	○	○	○	○	○	○	○
stopMonitor メソッド	○	○	○	○	○	○	○	○	○	○
halftone プロパティ	○	○	○	○	○	○	○	○	○	○
brightness プロパティ	○	○	○	○	○	○	○	○	○	○
force プロパティ	○	○	○	○	○	○	○	○	○	○
timeout プロパティ	○	○	○	○	○	○	○	○	○	○
interval プロパティ	○	○	○	○	○	○	○	○	○	○
drawerOpenLevel プロパティ	○	○	-	-	○	○	○	○	○	○
message プロパティ	○	○	○	○	○	○	○	○	○	○
onreceive イベント	○	○	○	○	○	○	○	○	○	○
onstatuschange イベント	○	○	○	○	○	○	○	○	○	○
onbatterystatuschange イベント	-	-	○	○	-	-	-	-	-	-
ononline イベント	○	○	○	○	○	○	○	○	○	○
onoffline イベント	○	○	○	○	○	○	○	○	○	○
onpoweroff イベント	○	○	○	○	○	○	○	○	○	○
oncoverok イベント	○	○	○	○	○	○	○	○	○	○
oncoveropen イベント	○	○	○	○	○	○	○	○	○	○
onpaperok イベント	○	○	○	○	○	○	○	○	○	○
onpapernearend イベント	○	○	○	○	○	○	○	○	○	○
onpaperend イベント	○	○	○	○	○	○	○	○	○	○
ondrawerclosed イベント	○	○	-	-	○	○	○	○	○	○
ondraweropen イベント	○	○	-	-	○	○	○	○	○	○
onbatteryok イベント	-	-	○	○	-	-	-	-	-	-

API	TM-m10	TM-m30	TM-P20	TM-P60II (Peeler)	TM-T20	TM-T20II	TM-T70	TM-T70II	TM-T88V	TM-T90II
onbatterylow イベント	-	-	○	○	-	-	-	-	-	-

## TM インテリジェントプリンター

API	TM-T70-i	TM-T70II-DT	TM-T88V-i	TM-T88V-DT
addTextAlign メソッド	○	○	○	○
addTextLineSpace メソッド	○	○	○	○
addTextRotate メソッド	○	○	○	○
addText メソッド	○	○	○	○
addTextLang メソッド	○	○	○	○
addTextFont メソッド	△	△	△	△
addTextSmooth メソッド	○	○	○	○
addTextDouble メソッド	○	○	○	○
addTextSize メソッド	○	○	○	○
addTextStyle メソッド	△	△	△	△
addTextPosition メソッド	○	○	○	○
addTextVPosition メソッド	○	○	○	○
addFeedUnit メソッド	○	○	○	○
addFeedLine メソッド	○	○	○	○
addFeedPosition メソッド	-	-	-	-
addFeed メソッド	○	○	○	○
addImage メソッド	△	△	△	△
addLogo メソッド	○	○	○	○
addBarcode メソッド	△	△	△	△

API	TM-T70-i	TM-T70II-DT	TM-T88V-i	TM-T88V-DT
addSymbol メソッド	△	△	△	△
addHLine メソッド	-	-	-	-
addVLineBegin メソッド	-	-	-	-
addVLineEnd メソッド	-	-	-	-
addPageBegin メソッド	○	○	○	○
addPageEnd メソッド	○	○	○	○
addPageArea メソッド	○	○	○	○
addPageDirection メソッド	○	○	○	○
addPagePosition メソッド	○	○	○	○
addPageLine メソッド	-	-	-	-
addPageRectangle メソッド	-	-	-	-
addCut メソッド	△	△	△	△
addPulse メソッド	○	○	○	○
addSound メソッド	-	△	△	△
addLayout メソッド	-	-	-	-
recover メソッド	○	○	○	○
addRecovery メソッド	○	○	○	○
reset メソッド	○	○	○	○
addReset メソッド	○	○	○	○
addCommand メソッド	○	○	○	○
send メソッド	○	○	○	○
print メソッド	△	○	○	○
getPrintJobStatus メソッド	○	○	○	○
startMonitor メソッド	○	○	○	○

API	TM-T70-i	TM-T70II-DT	TM-T88V-i	TM-T88V-DT
stopMonitor メソッド	○	○	○	○
halftone プロパティ	○	○	○	○
brightness プロパティ	○	○	○	○
force プロパティ	○	○	○	○
timeout プロパティ	○	○	○	○
interval プロパティ	○	○	○	○
drawerOpenLevel プロパティ	○	○	○	○
message プロパティ	○	○	○	○
onreceive イベント	○	○	○	○
onstatuschange イベント	○	○	○	○
onbatterystatuschange イベント	-	-	-	-
ononline イベント	○	○	○	○
onoffline イベント	○	○	○	○
onpoweroff イベント	○	○	○	○
oncoverok イベント	○	○	○	○
oncoveropen イベント	○	○	○	○
onpaperok イベント	○	○	○	○
onpapernearend イベント	○	○	○	○
onpaperend イベント	○	○	○	○
ondrawerclosed イベント	○	○	○	○
ondraweropen イベント	○	○	○	○
onbatteryok イベント	-	-	-	-
onbatterylow イベント	-	-	-	-

## ネットワークプリンター

API	TM-T88IV	TM-T90	TM-T90KP	TM-L90
addTextAlign メソッド	○	○	○	○
addTextLineSpace メソッド	○	○	○	○
addTextRotate メソッド	○	○	○	○
addText メソッド	○	○	○	○
addTextLang メソッド	○	○	○	○
addTextFont メソッド	△	△	△	△
addTextSmooth メソッド	○	○	○	○
addTextDouble メソッド	○	○	○	○
addTextSize メソッド	○	○	○	○
addTextStyle メソッド	△	△	△	△
addTextPosition メソッド	○	○	○	○
addTextVPosition メソッド	○	○	○	○
addFeedUnit メソッド	○	○	○	○
addFeedLine メソッド	○	○	○	○
addFeedPosition メソッド	-	-	-	○
addFeed メソッド	○	○	○	○
addImage メソッド	△	△	△	△
addLogo メソッド	○	○	○	○
addBarcode メソッド	△	△	△	△
addSymbol メソッド	△	△	△	△
addHLine メソッド	-	-	○	-
addVLineBegin メソッド	-	-	○	-
addVLineEnd メソッド	-	-	○	-
addPageBegin メソッド	○	○	○	○

API	TM-T88IV	TM-T90	TM-T90KP	TM-L90
addPageEnd メソッド	○	○	○	○
addPageArea メソッド	○	○	○	○
addPageDirection メソッド	○	○	○	○
addPagePosition メソッド	○	○	○	○
addPageLine メソッド	-	-	○	-
addPageRectangle メソッド	-	-	○	-
addCut メソッド	△	△	○	△
addPulse メソッド	○	○	○	○
addSound メソッド	-	-	△	-
addLayout メソッド	-	-	-	-
recover メソッド	○	○	○	○
addRecovery メソッド	○	○	○	○
reset メソッド	○	○	○	○
addReset メソッド	○	○	○	○
addCommand メソッド	○	○	○	○
send メソッド	○	○	○	○
print メソッド	○	○	△	○
getPrintJobStatus メソッド	○	○	○	○
startMonitor メソッド	○	○	-	○
stopMonitor メソッド	○	○	-	○
halftone プロパティ	○	○	○	○
brightness プロパティ	○	○	○	○
force プロパティ	○	○	○	○
timeout プロパティ	○	○	○	○
interval プロパティ	○	○	○	○
drawerOpenLevel プロパティ	○	○	-	○

API	TM-T88IV	TM-T90	TM-T90KP	TM-L90
message プロパティ	○	○	○	○
onreceive イベント	○	○	○	○
onstatuschange イベント	○	○	○	○
onbatterystatuschange イベント	-	-	-	-
ononline イベント	○	○	○	○
onoffline イベント	○	○	○	○
onpoweroff イベント	○	○	○	○
oncoverok イベント	○	○	○	○
oncoveropen イベント	○	○	○	○
onpaperok イベント	○	○	○	○
onpapernearend イベント	○	○	○	○
onpaperend イベント	○	○	○	○
ondrawerclosed イベント	○	○	-	○
ondraweropen イベント	○	○	-	○
onbatteryok イベント	-	-	-	-
onbatterylow イベント	-	-	-	-

# プリンター別サポート情報

## TM-m10

### プリンター仕様

項目	仕様
初期改行量	30 ドット
ページモード初期領域	420 x 2400 ドット
ページモード最大領域	420 x 2400 ドット
Font A のベースライン	文字の上端から 21 ドット目
Font B のベースライン	文字の上端から 21 ドット目
Font C のベースライン	文字の上端から 16 ドット目

### パラメーター制限

API	パラメーター	指定可能な設定値
addTextFont メソッド	font	<ul style="list-style-type: none"> <li>• FONT_A (初期値)</li> <li>• FONT_B</li> <li>• FONT_C</li> </ul>
addTextStyle メソッド	color	<ul style="list-style-type: none"> <li>• COLOR_NONE</li> <li>• COLOR_1 (初期値)</li> <li>• undefined</li> </ul>
addImage メソッド	color	<ul style="list-style-type: none"> <li>• COLOR_NONE</li> <li>• COLOR_1 (初期値)</li> <li>• undefined</li> </ul>
	mode	<ul style="list-style-type: none"> <li>• MODE_MONO</li> <li>• undefined</li> </ul>
addBarcode メソッド	font	<ul style="list-style-type: none"> <li>• FONT_A (初期値)</li> <li>• FONT_B</li> <li>• FONT_C</li> </ul>

API	パラメーター	指定可能な設定値
addSymbol メソッド	type	<ul style="list-style-type: none"> <li>• SYMBOL_PDF417_STANDARD</li> <li>• SYMBOL_PDF417_TRUNCATED</li> <li>• SYMBOL_QRCODE_MODEL_1</li> <li>• SYMBOL_QRCODE_MODEL_2</li> <li>• SYMBOL_MAXICODE_MODE_2</li> <li>• SYMBOL_MAXICODE_MODE_3</li> <li>• SYMBOL_MAXICODE_MODE_4</li> <li>• SYMBOL_MAXICODE_MODE_5</li> <li>• SYMBOL_MAXICODE_MODE_6</li> <li>• SYMBOL_GS1_DATABAR_STACKED</li> <li>• SYMBOL_GS1_DATABAR_STACKED_OMNIDIRECTIONAL</li> <li>• SYMBOL_GS1_DATABAR_EXPANDED_STACKED</li> <li>• SYMBOL_AZTECCODE_FULLRANGE</li> <li>• SYMBOL_AZTECCODE_COMPACT</li> <li>• SYMBOL_DATAMATRIX_SQUARE</li> <li>• SYMBOL_DATAMATRIX_RECTANGLE_8</li> <li>• SYMBOL_DATAMATRIX_RECTANGLE_12</li> <li>• SYMBOL_DATAMATRIX_RECTANGLE_16</li> </ul>
addCut メソッド	type	<ul style="list-style-type: none"> <li>• CUT_NO_FEED</li> <li>• CUT_FEED</li> <li>• undefined</li> </ul>
addSound メソッド	pattern	<ul style="list-style-type: none"> <li>• PATTERN_NONE</li> <li>• PATTERN_A</li> <li>• PATTERN_B</li> <li>• PATTERN_C</li> <li>• PATTERN_D</li> <li>• PATTERN_E</li> <li>• PATTERN_ERROR</li> <li>• PATTERN_PAPER_END</li> <li>• undefined</li> </ul>
	repeat	<ul style="list-style-type: none"> <li>• 1 ~ 255</li> <li>• undefined</li> </ul>
print メソッド	mode	<ul style="list-style-type: none"> <li>• MODE_MONO</li> <li>• undefined</li> </ul>

## TM-m30

## プリンター仕様

項目	58mm 仕様	80mm 仕様
初期改行量	30 ドット	
ページモード初期領域	420 x 2400 ドット	576 x 2400 ドット
ページモード最大領域	420 x 2400 ドット	576 x 2400 ドット
Font A のベースライン	文字の上端から 21 ドット目	
Font B のベースライン	文字の上端から 21 ドット目	
Font C のベースライン	文字の上端から 16 ドット目	

## パラメーター制限

API	パラメーター	指定可能な設定値
addTextFont メソッド	font	<ul style="list-style-type: none"> <li>• FONT_A (初期値)</li> <li>• FONT_B</li> <li>• FONT_C</li> </ul>
addTextStyle メソッド	color	<ul style="list-style-type: none"> <li>• COLOR_NONE</li> <li>• COLOR_1 (初期値)</li> <li>• undefined</li> </ul>
addImage メソッド	color	<ul style="list-style-type: none"> <li>• COLOR_NONE</li> <li>• COLOR_1 (初期値)</li> <li>• undefined</li> </ul>
	mode	<ul style="list-style-type: none"> <li>• MODE_MONO</li> <li>• undefined</li> </ul>
addBarcode メソッド	font	<ul style="list-style-type: none"> <li>• FONT_A (初期値)</li> <li>• FONT_B</li> <li>• FONT_C</li> </ul>

API	パラメーター	指定可能な設定値
addSymbol メソッド	type	<ul style="list-style-type: none"> <li>• SYMBOL_PDF417_STANDARD</li> <li>• SYMBOL_PDF417_TRUNCATED</li> <li>• SYMBOL_QRCODE_MODEL_1</li> <li>• SYMBOL_QRCODE_MODEL_2</li> <li>• SYMBOL_MAXICODE_MODE_2</li> <li>• SYMBOL_MAXICODE_MODE_3</li> <li>• SYMBOL_MAXICODE_MODE_4</li> <li>• SYMBOL_MAXICODE_MODE_5</li> <li>• SYMBOL_MAXICODE_MODE_6</li> <li>• SYMBOL_GS1_DATABAR_STACKED</li> <li>• SYMBOL_GS1_DATABAR_STACKED_OMNIDIRECTIONAL</li> <li>• SYMBOL_GS1_DATABAR_EXPANDED_STACKED</li> <li>• SYMBOL_AZTECCODE_FULLRANGE</li> <li>• SYMBOL_AZTECCODE_COMPACT</li> <li>• SYMBOL_DATAMATRIX_SQUARE</li> <li>• SYMBOL_DATAMATRIX_RECTANGLE_8</li> <li>• SYMBOL_DATAMATRIX_RECTANGLE_12</li> <li>• SYMBOL_DATAMATRIX_RECTANGLE_16</li> </ul>
addCut メソッド	type	<ul style="list-style-type: none"> <li>• CUT_NO_FEED</li> <li>• CUT_FEED</li> <li>• undefined</li> </ul>
addSound メソッド	pattern	<ul style="list-style-type: none"> <li>• PATTERN_NONE</li> <li>• PATTERN_A</li> <li>• PATTERN_B</li> <li>• PATTERN_C</li> <li>• PATTERN_D</li> <li>• PATTERN_E</li> <li>• PATTERN_ERROR</li> <li>• PATTERN_PAPER_END</li> <li>• undefined</li> </ul>
	repeat	<ul style="list-style-type: none"> <li>• 1 ~ 255</li> <li>• undefined</li> </ul>
print メソッド	mode	<ul style="list-style-type: none"> <li>• MODE_MONO</li> <li>• undefined</li> </ul>

## TM-P20

## プリンター仕様

項目	仕様
初期改行量	30 ドット
ページモード初期領域	384 x 2400 ドット
ページモード最大領域	384 x 2400 ドット
Font A のベースライン	文字の上端から 21 ドット目
Font B のベースライン	文字の上端から 21 ドット目
Font C のベースライン	文字の上端から 16 ドット目
Font D のベースライン	文字の上端から 21 ドット目
Font E のベースライン	文字の上端から 15 ドット目

## パラメーター制限

API	パラメーター	指定可能な設定値
addTextFont メソッド	font	<ul style="list-style-type: none"> <li>• FONT_A (初期値)</li> <li>• FONT_B</li> <li>• FONT_C</li> <li>• FONT_D</li> <li>• FONT_E</li> </ul>
addTextStyle メソッド	color	<ul style="list-style-type: none"> <li>• COLOR_NONE</li> <li>• COLOR_1 (初期値)</li> <li>• undefined</li> </ul>
addFeedPosition メソッド	pos	<ul style="list-style-type: none"> <li>• FEED_CUTTING</li> <li>• FEED_CURRENT_TOF</li> <li>• FEED_NEXT_TOF</li> </ul>
addImage メソッド	color	<ul style="list-style-type: none"> <li>• COLOR_NONE</li> <li>• COLOR_1 (初期値)</li> <li>• undefined</li> </ul>
	mode	<ul style="list-style-type: none"> <li>• MODE_MONO</li> <li>• undefined</li> </ul>

API	パラメーター	指定可能な設定値
addSymbol メソッド	type	<ul style="list-style-type: none"> <li>• SYMBOL_PDF417_STANDARD</li> <li>• SYMBOL_PDF417_TRUNCATED</li> <li>• SYMBOL_QRCODE_MODEL_1</li> <li>• SYMBOL_QRCODE_MODEL_2</li> <li>• SYMBOL_MAXICODE_MODE_2</li> <li>• SYMBOL_MAXICODE_MODE_3</li> <li>• SYMBOL_MAXICODE_MODE_4</li> <li>• SYMBOL_MAXICODE_MODE_5</li> <li>• SYMBOL_MAXICODE_MODE_6</li> <li>• SYMBOL_GS1_DATABAR_STACKED</li> <li>• SYMBOL_GS1_DATABAR_STACKED_OMNIDIRECTIONAL</li> <li>• SYMBOL_GS1_DATABAR_EXPANDED_STACKED</li> <li>• SYMBOL_AZTECCODE_FULLRANGE</li> <li>• SYMBOL_AZTECCODE_COMPACT</li> <li>• SYMBOL_DATAMATRIX_SQUARE</li> <li>• SYMBOL_DATAMATRIX_RECTANGLE_8</li> <li>• SYMBOL_DATAMATRIX_RECTANGLE_12</li> <li>• SYMBOL_DATAMATRIX_RECTANGLE_16</li> </ul>
addPageLine メソッド	style	<ul style="list-style-type: none"> <li>• LINE_THIN</li> <li>• LINE_MEDIUM</li> <li>• LINE_THICK</li> <li>• undefined</li> </ul>
addPageRectangle メソッド	style	<ul style="list-style-type: none"> <li>• LINE_THIN</li> <li>• LINE_MEDIUM</li> <li>• LINE_THICK</li> <li>• undefined</li> </ul>
addCut メソッド	type	<ul style="list-style-type: none"> <li>• CUT_NO_FEED</li> <li>• CUT_FEED</li> <li>• undefined</li> </ul>

API	パラメーター	指定可能な設定値
addSound メソッド	pattern	<ul style="list-style-type: none"> <li>• PATTERN_NONE</li> <li>• PATERN_1</li> <li>• PATERN_2</li> <li>• PATERN_3</li> <li>• PATERN_4</li> <li>• PATERN_5</li> <li>• PATERN_6</li> <li>• PATERN_7</li> <li>• PATERN_8</li> <li>• PATERN_9</li> <li>• PATERN_10</li> <li>• undefined</li> </ul>
	repeat	<ul style="list-style-type: none"> <li>• 1 ~ 255</li> <li>• undefined</li> </ul>
addLayout メソッド	type	<ul style="list-style-type: none"> <li>• LAYOUT_RECEIPT</li> <li>• LAYOUT_RECEIPT_BM</li> </ul>
	height	<ul style="list-style-type: none"> <li>• レシート紙</li> <li>• レシート紙（ブラックマーク付）</li> </ul>
	margin_top	<ul style="list-style-type: none"> <li>• レシート紙</li> <li>• レシート紙（ブラックマーク付）</li> </ul>
	margin_bottom	<ul style="list-style-type: none"> <li>• レシート紙</li> <li>• レシート紙（ブラックマーク付）</li> </ul>
	offset_cut	<ul style="list-style-type: none"> <li>• レシート紙</li> <li>• レシート紙（ブラックマーク付）</li> </ul>
	offset_label	<ul style="list-style-type: none"> <li>• レシート紙</li> <li>• レシート紙（ブラックマーク付）</li> </ul>
print メソッド	mode	<ul style="list-style-type: none"> <li>• MODE_MONO</li> <li>• undefined</li> </ul>

## TM-P60II (Peeler)

## プリンター仕様

項目	レシート仕様	ダイカットラベル仕様
初期改行量	30 ドット	
ページモード初期領域	432 x 1624 ドット	400 x 1624 ドット
ページモード最大領域	432 x 1624 ドット	400 x 1624 ドット
Font A のベースライン	文字の上端から 21 ドット目	
Font B のベースライン	文字の上端から 16 ドット目	
Font C のベースライン	文字の上端から 15 ドット目	

## パラメーター制限

API	パラメーター	指定可能な設定値
addTextFont メソッド	font	<ul style="list-style-type: none"> <li>• FONT_A (初期値)</li> <li>• FONT_B</li> <li>• FONT_C</li> </ul>
addTextStyle メソッド	color	<ul style="list-style-type: none"> <li>• COLOR_NONE</li> <li>• COLOR_1 (初期値)</li> <li>• undefined</li> </ul>
addImage メソッド	color	<ul style="list-style-type: none"> <li>• COLOR_NONE</li> <li>• COLOR_1 (初期値)</li> <li>• undefined</li> </ul>
	mode	<ul style="list-style-type: none"> <li>• MODE_MONO</li> <li>• undefined</li> </ul>
addBarcode メソッド	font	<ul style="list-style-type: none"> <li>• FONT_A (初期値)</li> <li>• FONT_B</li> <li>• FONT_C</li> </ul>

API	パラメーター	指定可能な設定値
addSymbol メソッド	type	<ul style="list-style-type: none"> <li>• SYMBOL_PDF417_STANDARD</li> <li>• SYMBOL_PDF417_TRUNCATED</li> <li>• SYMBOL_QRCODE_MODEL_1</li> <li>• SYMBOL_QRCODE_MODEL_2</li> <li>• SYMBOL_MAXICODE_MODE_2</li> <li>• SYMBOL_MAXICODE_MODE_3</li> <li>• SYMBOL_MAXICODE_MODE_4</li> <li>• SYMBOL_MAXICODE_MODE_5</li> <li>• SYMBOL_MAXICODE_MODE_6</li> <li>• SYMBOL_GS1_DATABAR_STACKED</li> <li>• SYMBOL_GS1_DATABAR_STACKED_OMNIDIRECTIONAL</li> <li>• SYMBOL_GS1_DATABAR_EXPANDED_STACKED</li> <li>• SYMBOL_AZTECCODE_FULLRANGE</li> <li>• SYMBOL_AZTECCODE_COMPACT</li> <li>• SYMBOL_DATAMATRIX_SQUARE</li> <li>• SYMBOL_DATAMATRIX_RECTANGLE_8</li> <li>• SYMBOL_DATAMATRIX_RECTANGLE_12</li> <li>• SYMBOL_DATAMATRIX_RECTANGLE_16</li> </ul>
addPageLine メソッド	style	<ul style="list-style-type: none"> <li>• LINE_THIN</li> <li>• LINE_MEDIUM</li> <li>• LINE_THICK</li> <li>• undefined</li> </ul>
addPageRectangle メソッド	style	<ul style="list-style-type: none"> <li>• LINE_THIN</li> <li>• LINE_MEDIUM</li> <li>• LINE_THICK</li> <li>• undefined</li> </ul>
addCut メソッド	type	<ul style="list-style-type: none"> <li>• CUT_NO_FEED</li> <li>• CUT_FEED</li> <li>• undefined</li> </ul>

API	パラメーター	指定可能な設定値
addSound メソッド	pattern	<ul style="list-style-type: none"><li>• PATTERN_NONE</li><li>• PATTERN_1</li><li>• PATTERN_2</li><li>• PATTERN_3</li><li>• PATTERN_4</li><li>• PATTERN_5</li><li>• PATTERN_6</li><li>• PATTERN_7</li><li>• PATTERN_8</li><li>• PATTERN_9</li><li>• PATTERN_10</li><li>• undefined</li></ul>
	repeat	<ul style="list-style-type: none"><li>• 1 ~ 255</li><li>• undefined</li></ul>
print メソッド	mode	<ul style="list-style-type: none"><li>• MODE_MONO</li><li>• undefined</li></ul>

## TM-T20

## プリンター仕様

項目	58mm 仕様	80mm 仕様
初期改行量	30 ドット	
ページモード初期領域	420 x 831 ドット	576 x 831 ドット
ページモード最大領域	420 x 1662 ドット	576 x 1662 ドット
Font A のベースライン	文字の上端から 21 ドット目	
Font B のベースライン	文字の上端から 16 ドット目	

## パラメーター制限

API	パラメーター	指定可能な設定値
addTextFont メソッド	font	<ul style="list-style-type: none"> <li>• FONT_A (初期値)</li> <li>• FONT_B</li> </ul>
addTextStyle メソッド	color	<ul style="list-style-type: none"> <li>• COLOR_NONE</li> <li>• COLOR_1 (初期値)</li> <li>• undefined</li> </ul>
addImage メソッド	color	<ul style="list-style-type: none"> <li>• COLOR_NONE</li> <li>• COLOR_1 (初期値)</li> <li>• undefined</li> </ul>
	mode	<ul style="list-style-type: none"> <li>• MODE_MONO</li> <li>• undefined</li> </ul>
addBarcode メソッド	font	<ul style="list-style-type: none"> <li>• FONT_A (初期値)</li> <li>• FONT_B</li> </ul>

API	パラメーター	指定可能な設定値
addSymbol メソッド	type	<ul style="list-style-type: none"> <li>• SYMBOL_PDF417_STANDARD</li> <li>• SYMBOL_PDF417_TRUNCATED</li> <li>• SYMBOL_QRCODE_MODEL_1</li> <li>• SYMBOL_QRCODE_MODEL_2</li> <li>• SYMBOL_MAXICODE_MODE_2</li> <li>• SYMBOL_MAXICODE_MODE_3</li> <li>• SYMBOL_MAXICODE_MODE_4</li> <li>• SYMBOL_MAXICODE_MODE_5</li> <li>• SYMBOL_MAXICODE_MODE_6</li> <li>• SYMBOL_GS1_DATABAR_STACKED</li> <li>• SYMBOL_GS1_DATABAR_STACKED_OMNIDIRECTIONAL</li> <li>• SYMBOL_GS1_DATABAR_EXPANDED_STACKED</li> </ul>
	level	<ul style="list-style-type: none"> <li>• LEVEL_0</li> <li>• LEVEL_1</li> <li>• LEVEL_2</li> <li>• LEVEL_3</li> <li>• LEVEL_4</li> <li>• LEVEL_5</li> <li>• LEVEL_6</li> <li>• LEVEL_7</li> <li>• LEVEL_8</li> <li>• LEVEL_L</li> <li>• LEVEL_M</li> <li>• LEVEL_Q</li> <li>• LEVEL_H</li> <li>• LEVEL_DEFAULT</li> </ul>
	width	<ul style="list-style-type: none"> <li>• PDF417</li> <li>• QR Code</li> <li>• MaxiCode</li> <li>• 2次元 GS1 DataBar</li> </ul>
	height	<ul style="list-style-type: none"> <li>• PDF417</li> <li>• QR Code</li> <li>• MaxiCode</li> <li>• 2次元 GS1 DataBar</li> </ul>
	size	<ul style="list-style-type: none"> <li>• PDF417</li> <li>• QR Code</li> <li>• MaxiCode</li> <li>• 2次元 GS1 DataBar</li> </ul>
addCut メソッド	type	<ul style="list-style-type: none"> <li>• CUT_NO_FEED</li> <li>• CUT_FEED</li> <li>• undefined</li> </ul>

API	パラメーター	指定可能な設定値
addSound メソッド	pattern	<ul style="list-style-type: none"><li>• PATTERN_NONE</li><li>• PATTERN_A</li><li>• PATTERN_B</li><li>• PATTERN_C</li><li>• PATTERN_D</li><li>• PATTERN_E</li><li>• PATTERN_ERROR</li><li>• PATTERN_PAPER_END</li><li>• undefined</li></ul>
	repeat	<ul style="list-style-type: none"><li>• 1 ~ 255</li><li>• undefined</li></ul>
print メソッド	mode	<ul style="list-style-type: none"><li>• MODE_MONO</li><li>• undefined</li></ul>

## TM-T20II

## プリンター仕様

項目	58mm 仕様	80mm 仕様
初期改行量	30 ドット	
ページモード初期領域	420 x 831 ドット	576 x 831 ドット
ページモード最大領域	420 x 1662 ドット	576 x 1662 ドット
Font A のベースライン	文字の上端から 21 ドット目	
Font B のベースライン	文字の上端から 16 ドット目	

## パラメーター制限

API	パラメーター	指定可能な設定値
addTextFont メソッド	font	<ul style="list-style-type: none"> <li>• FONT_A (初期値)</li> <li>• FONT_B</li> </ul>
addTextStyle メソッド	color	<ul style="list-style-type: none"> <li>• COLOR_NONE</li> <li>• COLOR_1 (初期値)</li> <li>• undefined</li> </ul>
addImage メソッド	color	<ul style="list-style-type: none"> <li>• COLOR_NONE</li> <li>• COLOR_1 (初期値)</li> <li>• undefined</li> </ul>
	mode	<ul style="list-style-type: none"> <li>• MODE_MONO</li> <li>• undefined</li> </ul>
addBarcode メソッド	font	<ul style="list-style-type: none"> <li>• FONT_A (初期値)</li> <li>• FONT_B</li> </ul>

API	パラメーター	指定可能な設定値
addSymbol メソッド	type	<ul style="list-style-type: none"> <li>• SYMBOL_PDF417_STANDARD</li> <li>• SYMBOL_PDF417_TRUNCATED</li> <li>• SYMBOL_QRCODE_MODEL_1</li> <li>• SYMBOL_QRCODE_MODEL_2</li> <li>• SYMBOL_MAXICODE_MODE_2</li> <li>• SYMBOL_MAXICODE_MODE_3</li> <li>• SYMBOL_MAXICODE_MODE_4</li> <li>• SYMBOL_MAXICODE_MODE_5</li> <li>• SYMBOL_MAXICODE_MODE_6</li> <li>• SYMBOL_GS1_DATABAR_STACKED</li> <li>• SYMBOL_GS1_DATABAR_STACKED_OMNIDIRECTIONAL</li> <li>• SYMBOL_GS1_DATABAR_EXPANDED_STACKED</li> </ul>
	level	<ul style="list-style-type: none"> <li>• LEVEL_0</li> <li>• LEVEL_1</li> <li>• LEVEL_2</li> <li>• LEVEL_3</li> <li>• LEVEL_4</li> <li>• LEVEL_5</li> <li>• LEVEL_6</li> <li>• LEVEL_7</li> <li>• LEVEL_8</li> <li>• LEVEL_L</li> <li>• LEVEL_M</li> <li>• LEVEL_Q</li> <li>• LEVEL_H</li> <li>• LEVEL_DEFAULT</li> </ul>
	width	<ul style="list-style-type: none"> <li>• PDF417</li> <li>• QR Code</li> <li>• MaxiCode</li> <li>• 2次元 GS1 DataBar</li> </ul>
	height	<ul style="list-style-type: none"> <li>• PDF417</li> <li>• QR Code</li> <li>• MaxiCode</li> <li>• 2次元 GS1 DataBar</li> </ul>
	size	<ul style="list-style-type: none"> <li>• PDF417</li> <li>• QR Code</li> <li>• MaxiCode</li> <li>• 2次元 GS1 DataBar</li> </ul>
addCut メソッド	type	<ul style="list-style-type: none"> <li>• CUT_NO_FEED</li> <li>• CUT_FEED</li> <li>• undefined</li> </ul>

API	パラメーター	指定可能な設定値
addSound メソッド	pattern	<ul style="list-style-type: none"><li>• PATTERN_NONE</li><li>• PATTERN_A</li><li>• PATTERN_B</li><li>• PATTERN_C</li><li>• PATTERN_D</li><li>• PATTERN_E</li><li>• PATTERN_ERROR</li><li>• PATTERN_PAPER_END</li><li>• undefined</li></ul>
	repeat	<ul style="list-style-type: none"><li>• 1 ~ 255</li><li>• undefined</li></ul>
print メソッド	mode	<ul style="list-style-type: none"><li>• MODE_MONO</li><li>• undefined</li></ul>

## TM-T70

## プリンター仕様

項目	58mm 仕様	80mm 仕様
初期改行量	30 ドット	
ページモード初期領域	416 x 1662 ドット	576 x 1662 ドット
ページモード最大領域	416 x 1662 ドット	576 x 1662 ドット
Font A のベースライン	文字の上端から 21 ドット目	
Font B のベースライン	文字の上端から 15 ドット目	

## パラメーター制限

API	パラメーター	指定可能な設定値
addTextFont メソッド	font	<ul style="list-style-type: none"> <li>• FONT_A (初期値)</li> <li>• FONT_B</li> </ul>
addTextStyle メソッド	color	<ul style="list-style-type: none"> <li>• COLOR_NONE</li> <li>• COLOR_1 (初期値)</li> <li>• undefined</li> </ul>
addImage メソッド	color	<ul style="list-style-type: none"> <li>• COLOR_NONE</li> <li>• COLOR_1 (初期値)</li> <li>• undefined</li> </ul>
	mode	<ul style="list-style-type: none"> <li>• MODE_MONO</li> <li>• undefined</li> </ul>
addBarcode メソッド	type	<ul style="list-style-type: none"> <li>• BARCODE_UPC_A</li> <li>• BARCODE_UPC_E</li> <li>• BARCODE_EAN13</li> <li>• BARCODE_JAN13</li> <li>• BARCODE_EAN8</li> <li>• BARCODE_JAN8</li> <li>• BARCODE_CODE39</li> <li>• BARCODE_ITF</li> <li>• BARCODE_CODABAR</li> <li>• BARCODE_CODE93</li> <li>• BARCODE_CODE128</li> </ul>
	font	<ul style="list-style-type: none"> <li>• FONT_A (初期値)</li> <li>• FONT_B</li> </ul>

API	パラメーター	指定可能な設定値
addSymbol メソッド	type	<ul style="list-style-type: none"><li>• SYMBOL_QRCODE_MODEL_1</li><li>• SYMBOL_QRCODE_MODEL_2</li></ul>
	level	<ul style="list-style-type: none"><li>• LEVEL_L</li><li>• LEVEL_M</li><li>• LEVEL_Q</li><li>• LEVEL_H</li><li>• LEVEL_DEFAULT</li></ul>
	width	<ul style="list-style-type: none"><li>• QR Code</li></ul>
	height	<ul style="list-style-type: none"><li>• QR Code</li></ul>
	size	<ul style="list-style-type: none"><li>• QR Code</li></ul>
addCut メソッド	type	<ul style="list-style-type: none"><li>• CUT_NO_FEED</li><li>• CUT_FEED</li><li>• undefined</li></ul>
print メソッド	mode	<ul style="list-style-type: none"><li>• MODE_MONO</li><li>• undefined</li></ul>

## TM-T70II

## プリンター仕様

項目	58mm 仕様	80mm 仕様
初期改行量	30 ドット	
ページモード初期領域	416 x 1662 ドット	576 x 1662 ドット
ページモード最大領域	416 x 1662 ドット	576 x 1662 ドット
Font A のベースライン	文字の上端から 21 ドット目	
Font B のベースライン	文字の上端から 15 ドット目	

## パラメーター制限

API	パラメーター	指定可能な設定値
addTextFont メソッド	font	<ul style="list-style-type: none"> <li>• FONT_A (初期値)</li> <li>• FONT_B</li> </ul>
addTextStyle メソッド	color	<ul style="list-style-type: none"> <li>• COLOR_NONE</li> <li>• COLOR_1 (初期値)</li> <li>• undefined</li> </ul>
addImage メソッド	color	<ul style="list-style-type: none"> <li>• COLOR_NONE</li> <li>• COLOR_1 (初期値)</li> <li>• undefined</li> </ul>
	mode	<ul style="list-style-type: none"> <li>• MODE_MONO</li> <li>• undefined</li> </ul>
addBarcode メソッド	font	<ul style="list-style-type: none"> <li>• FONT_A (初期値)</li> <li>• FONT_B</li> </ul>

API	パラメーター	指定可能な設定値
addSymbol メソッド	type	<ul style="list-style-type: none"> <li>• SYMBOL_PDF417_STANDARD</li> <li>• SYMBOL_PDF417_TRUNCATED</li> <li>• SYMBOL_QRCODE_MODEL_1</li> <li>• SYMBOL_QRCODE_MODEL_2</li> <li>• SYMBOL_MAXICODE_MODE_2</li> <li>• SYMBOL_MAXICODE_MODE_3</li> <li>• SYMBOL_MAXICODE_MODE_4</li> <li>• SYMBOL_MAXICODE_MODE_5</li> <li>• SYMBOL_MAXICODE_MODE_6</li> <li>• SYMBOL_GS1_DATABAR_STACKED</li> <li>• SYMBOL_GS1_DATABAR_STACKED_OMNIDIRECTIONAL</li> <li>• SYMBOL_GS1_DATABAR_EXPANDED_STACKED</li> </ul>
	level	<ul style="list-style-type: none"> <li>• LEVEL_0</li> <li>• LEVEL_1</li> <li>• LEVEL_2</li> <li>• LEVEL_3</li> <li>• LEVEL_4</li> <li>• LEVEL_5</li> <li>• LEVEL_6</li> <li>• LEVEL_7</li> <li>• LEVEL_8</li> <li>• LEVEL_L</li> <li>• LEVEL_M</li> <li>• LEVEL_Q</li> <li>• LEVEL_H</li> <li>• LEVEL_DEFAULT</li> </ul>
	width	<ul style="list-style-type: none"> <li>• PDF417</li> <li>• QR Code</li> <li>• MaxiCode</li> <li>• 2次元 GS1 DataBar</li> </ul>
	height	<ul style="list-style-type: none"> <li>• PDF417</li> <li>• QR Code</li> <li>• MaxiCode</li> <li>• 2次元 GS1 DataBar</li> </ul>
	size	<ul style="list-style-type: none"> <li>• PDF417</li> <li>• QR Code</li> <li>• MaxiCode</li> <li>• 2次元 GS1 DataBar</li> </ul>
addCut メソッド	type	<ul style="list-style-type: none"> <li>• CUT_NO_FEED</li> <li>• CUT_FEED</li> <li>• undefined</li> </ul>

API	パラメーター	指定可能な設定値
addSound メソッド	pattern	<ul style="list-style-type: none"><li>• PATTERN_NONE</li><li>• PATTERN_A</li><li>• PATTERN_B</li><li>• PATTERN_C</li><li>• PATTERN_D</li><li>• PATTERN_E</li><li>• PATTERN_ERROR</li><li>• PATTERN_PAPER_END</li><li>• undefined</li></ul>
	repeat	<ul style="list-style-type: none"><li>• 1 ~ 255</li><li>• undefined</li></ul>
print メソッド	mode	<ul style="list-style-type: none"><li>• MODE_MONO</li><li>• undefined</li></ul>

## TM-T88V

## プリンター仕様

項目		58mm 仕様	80mm 仕様
初期改行量		30 ドット	
ページモード初期領域		360 x 831 ドット	512 x 831 ドット
ページモード最大領域		360 x 1662 ドット	512 x 1662 ドット
Font A のベースライン		文字の上端から 21 ドット目	
Font B のベースライン	ANK	文字の上端から 16 ドット目	
	漢字	文字の上端から 15 ドット目	

## パラメーター制限

API	パラメーター	指定可能な設定値
addTextFont メソッド	font	<ul style="list-style-type: none"> <li>• FONT_A (初期値)</li> <li>• FONT_B</li> </ul>
addTextStyle メソッド	color	<ul style="list-style-type: none"> <li>• COLOR_NONE</li> <li>• COLOR_1 (初期値)</li> <li>• undefined</li> </ul>
addImage メソッド	color	<ul style="list-style-type: none"> <li>• COLOR_NONE</li> <li>• COLOR_1 (初期値)</li> <li>• undefined</li> </ul>
addBarcode メソッド	font	<ul style="list-style-type: none"> <li>• FONT_A (初期値)</li> <li>• FONT_B</li> </ul>

API	パラメーター	指定可能な設定値
addSymbol メソッド	type	<ul style="list-style-type: none"> <li>• SYMBOL_PDF417_STANDARD</li> <li>• SYMBOL_PDF417_TRUNCATED</li> <li>• SYMBOL_QRCODE_MODEL_1</li> <li>• SYMBOL_QRCODE_MODEL_2</li> <li>• SYMBOL_MAXICODE_MODE_2</li> <li>• SYMBOL_MAXICODE_MODE_3</li> <li>• SYMBOL_MAXICODE_MODE_4</li> <li>• SYMBOL_MAXICODE_MODE_5</li> <li>• SYMBOL_MAXICODE_MODE_6</li> <li>• SYMBOL_GS1_DATABAR_STACKED</li> <li>• SYMBOL_GS1_DATABAR_STACKED_OMNIDIRECTIONAL</li> <li>• SYMBOL_GS1_DATABAR_EXPANDED_STACKED</li> </ul>
	level	<ul style="list-style-type: none"> <li>• LEVEL_0</li> <li>• LEVEL_1</li> <li>• LEVEL_2</li> <li>• LEVEL_3</li> <li>• LEVEL_4</li> <li>• LEVEL_5</li> <li>• LEVEL_6</li> <li>• LEVEL_7</li> <li>• LEVEL_8</li> <li>• LEVEL_L</li> <li>• LEVEL_M</li> <li>• LEVEL_Q</li> <li>• LEVEL_H</li> <li>• LEVEL_DEFAULT</li> </ul>
	width	<ul style="list-style-type: none"> <li>• PDF417</li> <li>• QR Code</li> <li>• MaxiCode</li> <li>• 2次元 GS1 DataBar</li> </ul>
	height	<ul style="list-style-type: none"> <li>• PDF417</li> <li>• QR Code</li> <li>• MaxiCode</li> <li>• 2次元 GS1 DataBar</li> </ul>
	size	<ul style="list-style-type: none"> <li>• PDF417</li> <li>• QR Code</li> <li>• MaxiCode</li> <li>• 2次元 GS1 DataBar</li> </ul>
addCut メソッド	type	<ul style="list-style-type: none"> <li>• CUT_NO_FEED</li> <li>• CUT_FEED</li> <li>• undefined</li> </ul>

API	パラメーター	指定可能な設定値
addSound メソッド	pattern	<ul style="list-style-type: none"><li>• PATTERN_NONE</li><li>• PATTERN_A</li><li>• PATTERN_B</li><li>• PATTERN_C</li><li>• PATTERN_D</li><li>• PATTERN_E</li><li>• PATTERN_ERROR</li><li>• PATTERN_PAPER_END</li><li>• undefined</li></ul>
	repeat	<ul style="list-style-type: none"><li>• 1 ~ 255</li><li>• undefined</li></ul>

## TM-T90II

## プリンター仕様

項目		58mm 仕様	80mm 仕様
初期改行量		30 ドット	
ページモード初期領域	モノクロ印字	420 x 738 ドット	576 x 738 ドット
	2色印字	420 x 369 ドット	576 x 369 ドット
ページモード最大領域	モノクロ印字	420 x 1476 ドット	576 x 1476 ドット
	2色印字	420 x 738 ドット	576 x 738 ドット
Font A のベースライン		文字の上端から 21 ドット目	
Font B のベースライン		文字の上端から 21 ドット目	
Font C のベースライン		文字の上端から 15 ドット目	

## パラメーター制限

API	パラメーター	指定可能な設定値
addTextFont メソッド	font	<ul style="list-style-type: none"> <li>• FONT_A (初期値)</li> <li>• FONT_B</li> <li>• FONT_C</li> </ul>
addTextStyle メソッド	color	<ul style="list-style-type: none"> <li>• COLOR_NONE</li> <li>• COLOR_1 (初期値)</li> <li>• COLOR_2</li> <li>• undefined</li> </ul>
addImage メソッド	color	<ul style="list-style-type: none"> <li>• COLOR_NONE</li> <li>• COLOR_1 (初期値)</li> <li>• undefined</li> </ul>
addBarcode メソッド	font	<ul style="list-style-type: none"> <li>• FONT_A (初期値)</li> <li>• FONT_B</li> <li>• FONT_C</li> </ul>

API	パラメーター	指定可能な設定値
addSymbol メソッド	type	<ul style="list-style-type: none"> <li>• SYMBOL_PDF417_STANDARD</li> <li>• SYMBOL_PDF417_TRUNCATED</li> <li>• SYMBOL_QRCODE_MODEL_1</li> <li>• SYMBOL_QRCODE_MODEL_2</li> <li>• SYMBOL_MAXICODE_MODE_2</li> <li>• SYMBOL_MAXICODE_MODE_3</li> <li>• SYMBOL_MAXICODE_MODE_4</li> <li>• SYMBOL_MAXICODE_MODE_5</li> <li>• SYMBOL_MAXICODE_MODE_6</li> <li>• SYMBOL_GS1_DATABAR_STACKED</li> <li>• SYMBOL_GS1_DATABAR_STACKED_OMNIDIRECTIONAL</li> <li>• SYMBOL_GS1_DATABAR_EXPANDED_STACKED</li> </ul>
	level	<ul style="list-style-type: none"> <li>• LEVEL_0</li> <li>• LEVEL_1</li> <li>• LEVEL_2</li> <li>• LEVEL_3</li> <li>• LEVEL_4</li> <li>• LEVEL_5</li> <li>• LEVEL_6</li> <li>• LEVEL_7</li> <li>• LEVEL_8</li> <li>• LEVEL_L</li> <li>• LEVEL_M</li> <li>• LEVEL_Q</li> <li>• LEVEL_H</li> <li>• LEVEL_DEFAULT</li> </ul>
	width	<ul style="list-style-type: none"> <li>• PDF417</li> <li>• QR Code</li> <li>• MaxiCode</li> <li>• 2次元 GS1 DataBar</li> </ul>
	height	<ul style="list-style-type: none"> <li>• PDF417</li> <li>• QR Code</li> <li>• MaxiCode</li> <li>• 2次元 GS1 DataBar</li> </ul>
	size	<ul style="list-style-type: none"> <li>• PDF417</li> <li>• QR Code</li> <li>• MaxiCode</li> <li>• 2次元 GS1 DataBar</li> </ul>
addCut メソッド	type	<ul style="list-style-type: none"> <li>• CUT_NO_FEED</li> <li>• CUT_FEED</li> <li>• undefined</li> </ul>

## TM-T70-i

## プリンター仕様

項目	58mm 仕様	80mm 仕様
初期改行量	30 ドット	
ページモード初期領域	416 x 1662 ドット	576 x 1662 ドット
ページモード最大領域	416 x 1662 ドット	576 x 1662 ドット
Font A のベースライン	文字の上端から 21 ドット目	
Font B のベースライン	文字の上端から 15 ドット目	

## パラメーター制限

API	パラメーター	指定可能な設定値
addTextFont メソッド	font	<ul style="list-style-type: none"> <li>• FONT_A (初期値)</li> <li>• FONT_B</li> </ul>
addTextStyle メソッド	color	<ul style="list-style-type: none"> <li>• COLOR_NONE</li> <li>• COLOR_1 (初期値)</li> <li>• undefined</li> </ul>
addImage メソッド	color	<ul style="list-style-type: none"> <li>• COLOR_NONE</li> <li>• COLOR_1 (初期値)</li> <li>• undefined</li> </ul>
	mode	<ul style="list-style-type: none"> <li>• MODE_MONO</li> <li>• undefined</li> </ul>
addBarcode メソッド	type	<ul style="list-style-type: none"> <li>• BARCODE_UPC_A</li> <li>• BARCODE_UPC_E</li> <li>• BARCODE_EAN13</li> <li>• BARCODE_JAN13</li> <li>• BARCODE_EAN8</li> <li>• BARCODE_JAN8</li> <li>• BARCODE_CODE39</li> <li>• BARCODE_ITF</li> <li>• BARCODE_CODABAR</li> <li>• BARCODE_CODE93</li> <li>• BARCODE_CODE128</li> </ul>
	font	<ul style="list-style-type: none"> <li>• FONT_A (初期値)</li> <li>• FONT_B</li> </ul>

API	パラメーター	指定可能な設定値
addSymbol メソッド	type	<ul style="list-style-type: none"><li>• SYMBOL_QRCODE_MODEL_1</li><li>• SYMBOL_QRCODE_MODEL_2</li></ul>
	level	<ul style="list-style-type: none"><li>• LEVEL_L</li><li>• LEVEL_M</li><li>• LEVEL_Q</li><li>• LEVEL_H</li><li>• LEVEL_DEFAULT</li></ul>
	width	<ul style="list-style-type: none"><li>• QR Code</li></ul>
	height	<ul style="list-style-type: none"><li>• QR Code</li></ul>
	size	<ul style="list-style-type: none"><li>• QR Code</li></ul>
addCut メソッド	type	<ul style="list-style-type: none"><li>• CUT_NO_FEED</li><li>• CUT_FEED</li><li>• undefined</li></ul>
print メソッド	mode	<ul style="list-style-type: none"><li>• MODE_MONO</li><li>• undefined</li></ul>

## TM-T70II-DT

## プリンター仕様

項目	58mm 仕様	80mm 仕様
初期改行量	30 ドット	
ページモード初期領域	416 x 1662 ドット	576 x 1662 ドット
ページモード最大領域	416 x 1662 ドット	576 x 1662 ドット
Font A のベースライン	文字の上端から 21 ドット目	
Font B のベースライン	文字の上端から 15 ドット目	

## パラメーター制限

API	パラメーター	指定可能な設定値
addTextFont メソッド	font	<ul style="list-style-type: none"> <li>• FONT_A (初期値)</li> <li>• FONT_B</li> </ul>
addTextStyle メソッド	color	<ul style="list-style-type: none"> <li>• COLOR_NONE</li> <li>• COLOR_1 (初期値)</li> <li>• undefined</li> </ul>
addImage メソッド	color	<ul style="list-style-type: none"> <li>• COLOR_NONE</li> <li>• COLOR_1 (初期値)</li> <li>• undefined</li> </ul>
addBarcode メソッド	font	<ul style="list-style-type: none"> <li>• FONT_A (初期値)</li> <li>• FONT_B</li> </ul>

API	パラメーター	指定可能な設定値
addSymbol メソッド	type	<ul style="list-style-type: none"> <li>• SYMBOL_PDF417_STANDARD</li> <li>• SYMBOL_PDF417_TRUNCATED</li> <li>• SYMBOL_QRCODE_MODEL_1</li> <li>• SYMBOL_QRCODE_MODEL_2</li> <li>• SYMBOL_MAXICODE_MODE_2</li> <li>• SYMBOL_MAXICODE_MODE_3</li> <li>• SYMBOL_MAXICODE_MODE_4</li> <li>• SYMBOL_MAXICODE_MODE_5</li> <li>• SYMBOL_MAXICODE_MODE_6</li> <li>• SYMBOL_GS1_DATABAR_STACKED</li> <li>• SYMBOL_GS1_DATABAR_STACKED_OMNIDIRECTIONAL</li> <li>• SYMBOL_GS1_DATABAR_EXPANDED_STACKED</li> </ul>
	level	<ul style="list-style-type: none"> <li>• LEVEL_0</li> <li>• LEVEL_1</li> <li>• LEVEL_2</li> <li>• LEVEL_3</li> <li>• LEVEL_4</li> <li>• LEVEL_5</li> <li>• LEVEL_6</li> <li>• LEVEL_7</li> <li>• LEVEL_8</li> <li>• LEVEL_L</li> <li>• LEVEL_M</li> <li>• LEVEL_Q</li> <li>• LEVEL_H</li> <li>• LEVEL_DEFAULT</li> </ul>
	width	<ul style="list-style-type: none"> <li>• PDF417</li> <li>• QR Code</li> <li>• MaxiCode</li> <li>• 2次元 GS1 DataBar</li> </ul>
	height	<ul style="list-style-type: none"> <li>• PDF417</li> <li>• QR Code</li> <li>• MaxiCode</li> <li>• 2次元 GS1 DataBar</li> </ul>
	size	<ul style="list-style-type: none"> <li>• PDF417</li> <li>• QR Code</li> <li>• MaxiCode</li> <li>• 2次元 GS1 DataBar</li> </ul>
addCut メソッド	type	<ul style="list-style-type: none"> <li>• CUT_NO_FEED</li> <li>• CUT_FEED</li> <li>• undefined</li> </ul>

API	パラメーター	指定可能な設定値
addSound メソッド	pattern	<ul style="list-style-type: none"><li>• PATTERN_NONE</li><li>• PATTERN_A</li><li>• PATTERN_B</li><li>• PATTERN_C</li><li>• PATTERN_D</li><li>• PATTERN_E</li><li>• PATTERN_ERROR</li><li>• PATTERN_PAPER_END</li><li>• undefined</li></ul>

## TM-T88V-i

## プリンター仕様

項目		58mm 仕様	80mm 仕様
初期改行量		30 ドット	
ページモード初期領域		360 x 831 ドット	512 x 831 ドット
ページモード最大領域		360 x 1662 ドット	512 x 1662 ドット
Font A のベースライン		文字の上端から 21 ドット目	
Font B のベースライン	ANK	文字の上端から 16 ドット目	
	漢字	文字の上端から 15 ドット目	

## パラメーター制限

API	パラメーター	指定可能な設定値
addTextFont メソッド	font	<ul style="list-style-type: none"> <li>• FONT_A (初期値)</li> <li>• FONT_B</li> </ul>
addTextStyle メソッド	color	<ul style="list-style-type: none"> <li>• COLOR_NONE</li> <li>• COLOR_1 (初期値)</li> <li>• undefined</li> </ul>
addImage メソッド	color	<ul style="list-style-type: none"> <li>• COLOR_NONE</li> <li>• COLOR_1 (初期値)</li> <li>• undefined</li> </ul>
addBarcode メソッド	font	<ul style="list-style-type: none"> <li>• FONT_A (初期値)</li> <li>• FONT_B</li> </ul>

API	パラメーター	指定可能な設定値
addSymbol メソッド	type	<ul style="list-style-type: none"> <li>• SYMBOL_PDF417_STANDARD</li> <li>• SYMBOL_PDF417_TRUNCATED</li> <li>• SYMBOL_QRCODE_MODEL_1</li> <li>• SYMBOL_QRCODE_MODEL_2</li> <li>• SYMBOL_MAXICODE_MODE_2</li> <li>• SYMBOL_MAXICODE_MODE_3</li> <li>• SYMBOL_MAXICODE_MODE_4</li> <li>• SYMBOL_MAXICODE_MODE_5</li> <li>• SYMBOL_MAXICODE_MODE_6</li> <li>• SYMBOL_GS1_DATABAR_STACKED</li> <li>• SYMBOL_GS1_DATABAR_STACKED_OMNIDIRECTIONAL</li> <li>• SYMBOL_GS1_DATABAR_EXPANDED_STACKED</li> </ul>
	level	<ul style="list-style-type: none"> <li>• LEVEL_0</li> <li>• LEVEL_1</li> <li>• LEVEL_2</li> <li>• LEVEL_3</li> <li>• LEVEL_4</li> <li>• LEVEL_5</li> <li>• LEVEL_6</li> <li>• LEVEL_7</li> <li>• LEVEL_8</li> <li>• LEVEL_L</li> <li>• LEVEL_M</li> <li>• LEVEL_Q</li> <li>• LEVEL_H</li> <li>• LEVEL_DEFAULT</li> </ul>
	width	<ul style="list-style-type: none"> <li>• PDF417</li> <li>• QR Code</li> <li>• MaxiCode</li> <li>• 2次元 GS1 DataBar</li> </ul>
	height	<ul style="list-style-type: none"> <li>• PDF417</li> <li>• QR Code</li> <li>• MaxiCode</li> <li>• 2次元 GS1 DataBar</li> </ul>
	size	<ul style="list-style-type: none"> <li>• PDF417</li> <li>• QR Code</li> <li>• MaxiCode</li> <li>• 2次元 GS1 DataBar</li> </ul>
addCut メソッド	type	<ul style="list-style-type: none"> <li>• CUT_NO_FEED</li> <li>• CUT_FEED</li> <li>• undefined</li> </ul>

API	パラメーター	指定可能な設定値
addSound メソッド	pattern	<ul style="list-style-type: none"><li>• PATTERN_NONE</li><li>• PATTERN_A</li><li>• PATTERN_B</li><li>• PATTERN_C</li><li>• PATTERN_D</li><li>• PATTERN_E</li><li>• PATTERN_ERROR</li><li>• PATTERN_PAPER_END</li><li>• undefined</li></ul>

## TM-T88V-DT

## プリンター仕様

項目		58mm 仕様	80mm 仕様
初期改行量		30 ドット	
ページモード初期領域		360 x 831 ドット	512 x 831 ドット
ページモード最大領域		360 x 1662 ドット	512 x 1662 ドット
Font A のベースライン		文字の上端から 21 ドット目	
Font B のベースライン	ANK	文字の上端から 16 ドット目	
	漢字	文字の上端から 15 ドット目	

## パラメーター制限

API	パラメーター	指定可能な設定値
addTextFont メソッド	font	<ul style="list-style-type: none"> <li>• FONT_A (初期値)</li> <li>• FONT_B</li> </ul>
addTextStyle メソッド	color	<ul style="list-style-type: none"> <li>• COLOR_NONE</li> <li>• COLOR_1 (初期値)</li> <li>• undefined</li> </ul>
addImage メソッド	color	<ul style="list-style-type: none"> <li>• COLOR_NONE</li> <li>• COLOR_1 (初期値)</li> <li>• undefined</li> </ul>
addBarcode メソッド	font	<ul style="list-style-type: none"> <li>• FONT_A (初期値)</li> <li>• FONT_B</li> </ul>

API	パラメーター	指定可能な設定値
addSymbol メソッド	type	<ul style="list-style-type: none"> <li>• SYMBOL_PDF417_STANDARD</li> <li>• SYMBOL_PDF417_TRUNCATED</li> <li>• SYMBOL_QRCODE_MODEL_1</li> <li>• SYMBOL_QRCODE_MODEL_2</li> <li>• SYMBOL_MAXICODE_MODE_2</li> <li>• SYMBOL_MAXICODE_MODE_3</li> <li>• SYMBOL_MAXICODE_MODE_4</li> <li>• SYMBOL_MAXICODE_MODE_5</li> <li>• SYMBOL_MAXICODE_MODE_6</li> <li>• SYMBOL_GS1_DATABAR_STACKED</li> <li>• SYMBOL_GS1_DATABAR_STACKED_OMNIDIRECTIONAL</li> <li>• SYMBOL_GS1_DATABAR_EXPANDED_STACKED</li> </ul>
	level	<ul style="list-style-type: none"> <li>• LEVEL_0</li> <li>• LEVEL_1</li> <li>• LEVEL_2</li> <li>• LEVEL_3</li> <li>• LEVEL_4</li> <li>• LEVEL_5</li> <li>• LEVEL_6</li> <li>• LEVEL_7</li> <li>• LEVEL_8</li> <li>• LEVEL_L</li> <li>• LEVEL_M</li> <li>• LEVEL_Q</li> <li>• LEVEL_H</li> <li>• LEVEL_DEFAULT</li> </ul>
	width	<ul style="list-style-type: none"> <li>• PDF417</li> <li>• QR Code</li> <li>• MaxiCode</li> <li>• 2 次元 GS1 DataBar</li> </ul>
	height	<ul style="list-style-type: none"> <li>• PDF417</li> <li>• QR Code</li> <li>• MaxiCode</li> <li>• 2 次元 GS1 DataBar</li> </ul>
	size	<ul style="list-style-type: none"> <li>• PDF417</li> <li>• QR Code</li> <li>• MaxiCode</li> <li>• 2 次元 GS1 DataBar</li> </ul>
addCut メソッド	type	<ul style="list-style-type: none"> <li>• CUT_NO_FEED</li> <li>• CUT_FEED</li> <li>• undefined</li> </ul>

API	パラメーター	指定可能な設定値
addSound メソッド	pattern	<ul style="list-style-type: none"><li>• PATTERN_NONE</li><li>• PATTERN_A</li><li>• PATTERN_B</li><li>• PATTERN_C</li><li>• PATTERN_D</li><li>• PATTERN_E</li><li>• PATTERN_ERROR</li><li>• PATTERN_PAPER_END</li><li>• undefined</li></ul>

## TM-T88IV

## プリンター仕様

項目		58mm 仕様	80mm 仕様
初期改行量		30 ドット	
ページモード初期領域	モノクロ印字	360 x 831 ドット	512 x 831 ドット
	2色印字	360 x 415 ドット	512 x 415 ドット
ページモード最大領域	モノクロ印字	360 x 1662 ドット	512 x 1662 ドット
	2色印字	360 x 831 ドット	512 x 831 ドット
Font A のベースライン		文字の上端から 21 ドット目	
Font B のベースライン		文字の上端から 16 ドット目	

## パラメーター制限

API	パラメーター	指定可能な設定値
addTextFont メソッド	font	<ul style="list-style-type: none"> <li>• FONT_A (初期値)</li> <li>• FONT_B</li> </ul>
addTextStyle メソッド	color	<ul style="list-style-type: none"> <li>• COLOR_NONE</li> <li>• COLOR_1 (初期値)</li> <li>• COLOR_2</li> <li>• undefined</li> </ul>
addImage メソッド	color	<ul style="list-style-type: none"> <li>• COLOR_NONE</li> <li>• COLOR_1 (初期値)</li> <li>• COLOR_2</li> <li>• undefined</li> </ul>
addBarcode メソッド	type	<ul style="list-style-type: none"> <li>• BARCODE_UPC_A</li> <li>• BARCODE_UPC_E</li> <li>• BARCODE_EAN13</li> <li>• BARCODE_JAN13</li> <li>• BARCODE_EAN8</li> <li>• BARCODE_JAN8</li> <li>• BARCODE_CODE39</li> <li>• BARCODE_ITF</li> <li>• BARCODE_CODABAR</li> <li>• BARCODE_CODE93</li> <li>• BARCODE_CODE128</li> </ul>
	font	<ul style="list-style-type: none"> <li>• FONT_A (初期値)</li> <li>• FONT_B</li> </ul>

API	パラメーター	指定可能な設定値
addSymbol メソッド	type	<ul style="list-style-type: none"><li>• SYMBOL_QRCODE_MODEL_1</li><li>• SYMBOL_QRCODE_MODEL_2</li></ul>
	level	<ul style="list-style-type: none"><li>• LEVEL_L</li><li>• LEVEL_M</li><li>• LEVEL_Q</li><li>• LEVEL_H</li><li>• LEVEL_DEFAULT</li></ul>
	width	<ul style="list-style-type: none"><li>• QR Code</li></ul>
	height	<ul style="list-style-type: none"><li>• QR Code</li></ul>
	size	<ul style="list-style-type: none"><li>• QR Code</li></ul>
addCut メソッド	type	<ul style="list-style-type: none"><li>• CUT_NO_FEED</li><li>• CUT_FEED</li><li>• undefined</li></ul>

## TM-T90

## プリンター仕様

項目		58mm 仕様	80mm 仕様
初期改行量		30 ドット	
ページモード初期領域	モノクロ印字	420 x 738 ドット	576 x 738 ドット
	2色印字	420 x 369 ドット	576 x 369 ドット
ページモード最大領域	モノクロ印字	420 x 1476 ドット	576 x 1476 ドット
	2色印字	420 x 738 ドット	576 x 738 ドット
Font A のベースライン		文字の上端から 21 ドット目	
Font B のベースライン		文字の上端から 21 ドット目	
Font C のベースライン		文字の上端から 15 ドット目	

## パラメーター制限

API	パラメーター	指定可能な設定値
addTextFont メソッド	font	<ul style="list-style-type: none"> <li>• FONT_A (初期値)</li> <li>• FONT_B</li> <li>• FONT_C</li> </ul>
addTextStyle メソッド	color	<ul style="list-style-type: none"> <li>• COLOR_NONE</li> <li>• COLOR_1 (初期値)</li> <li>• COLOR_2</li> <li>• undefined</li> </ul>
addImage メソッド	color	<ul style="list-style-type: none"> <li>• COLOR_NONE</li> <li>• COLOR_1 (初期値)</li> <li>• COLOR_2</li> <li>• undefined</li> </ul>
addBarcode メソッド	type	<ul style="list-style-type: none"> <li>• BARCODE_UPC_A</li> <li>• BARCODE_UPC_E</li> <li>• BARCODE_EAN13</li> <li>• BARCODE_JAN13</li> <li>• BARCODE_EAN8</li> <li>• BARCODE_JAN8</li> <li>• BARCODE_CODE39</li> <li>• BARCODE_ITF</li> <li>• BARCODE_CODABAR</li> <li>• BARCODE_CODE93</li> <li>• BARCODE_CODE128</li> </ul>
	font	<ul style="list-style-type: none"> <li>• FONT_A (初期値)</li> <li>• FONT_B</li> <li>• FONT_C</li> </ul>

API	パラメーター	指定可能な設定値
addSymbol メソッド	type	<ul style="list-style-type: none"> <li>• SYMBOL_PDF417_STANDARD</li> <li>• SYMBOL_PDF417_TRUNCATED</li> </ul>
	level	<ul style="list-style-type: none"> <li>• LEVEL_0</li> <li>• LEVEL_1</li> <li>• LEVEL_2</li> <li>• LEVEL_3</li> <li>• LEVEL_4</li> <li>• LEVEL_5</li> <li>• LEVEL_6</li> <li>• LEVEL_7</li> <li>• LEVEL_8</li> <li>• LEVEL_DEFAULT</li> </ul>
	width	<ul style="list-style-type: none"> <li>• PDF417</li> </ul>
	height	<ul style="list-style-type: none"> <li>• PDF417</li> </ul>
	size	<ul style="list-style-type: none"> <li>• PDF417</li> </ul>
addCut メソッド	type	<ul style="list-style-type: none"> <li>• CUT_NO_FEED</li> <li>• CUT_FEED</li> <li>• undefined</li> </ul>

## TM-T90KP

## プリンター仕様

項目	仕様
初期改行量	30 ドット
ページモード初期領域	576 x 738 ドット
ページモード最大領域	576 x 1476 ドット
Font A のベースライン	文字の上端から 21 ドット目
Font B のベースライン	文字の上端から 21 ドット目
Font C のベースライン	文字の上端から 15 ドット目

## パラメーター制限

API	パラメーター	指定可能な設定値
addTextFont メソッド	font	<ul style="list-style-type: none"> <li>• FONT_A (初期値)</li> <li>• FONT_B</li> <li>• FONT_C</li> </ul>
addTextStyle メソッド	color	<ul style="list-style-type: none"> <li>• COLOR_NONE</li> <li>• COLOR_1 (初期値)</li> <li>• undefined</li> </ul>
addImage メソッド	color	<ul style="list-style-type: none"> <li>• COLOR_NONE</li> <li>• COLOR_1 (初期値)</li> <li>• undefined</li> </ul>
	mode	<ul style="list-style-type: none"> <li>• MODE_MONO</li> <li>• undefined</li> </ul>
addBarcode メソッド	type	<ul style="list-style-type: none"> <li>• BARCODE_UPC_A</li> <li>• BARCODE_UPC_E</li> <li>• BARCODE_EAN13</li> <li>• BARCODE_JAN13</li> <li>• BARCODE_EAN8</li> <li>• BARCODE_JAN8</li> <li>• BARCODE_CODE39</li> <li>• BARCODE_ITF</li> <li>• BARCODE_CODABAR</li> <li>• BARCODE_CODE93</li> <li>• BARCODE_CODE128</li> </ul>
	font	<ul style="list-style-type: none"> <li>• FONT_A (初期値)</li> <li>• FONT_B</li> <li>• FONT_C</li> </ul>

API	パラメーター	指定可能な設定値
addSound メソッド	pattern	<ul style="list-style-type: none"><li>• PATTERN_NONE</li><li>• PATTERN_A</li><li>• PATTERN_B</li><li>• PATTERN_C</li><li>• PATTERN_D</li><li>• PATTERN_E</li><li>• PATTERN_ERROR</li><li>• PATTERN_PAPER_END</li><li>• undefined</li></ul>
	repeat	<ul style="list-style-type: none"><li>• 1 ~ 255</li><li>• undefined</li></ul>
print メソッド	mode	<ul style="list-style-type: none"><li>• MODE_MONO</li><li>• undefined</li></ul>

## TM-L90

## プリンター仕様

項目		レシート仕様	ダイカットラベル仕様
初期改行量		30 ドット	
ページモード初期領域	モノクロ印字	576 x 738 ドット	560 x 738 ドット
	2色印字	576 x 369 ドット	560 x 369 ドット
ページモード最大領域	モノクロ印字	576 x 1476 ドット	560 x 1476 ドット
	2色印字	576 x 738 ドット	560 x 738 ドット
Font A のベースライン		文字の上端から 21 ドット目	
Font B のベースライン		文字の上端から 21 ドット目	
Font C のベースライン		文字の上端から 15 ドット目	

## パラメーター制限

API	パラメーター	指定可能な設定値
addTextFont メソッド	font	<ul style="list-style-type: none"> <li>• FONT_A (初期値)</li> <li>• FONT_B</li> <li>• FONT_C</li> </ul>
addTextStyle メソッド	color	<ul style="list-style-type: none"> <li>• COLOR_NONE</li> <li>• COLOR_1 (初期値)</li> <li>• COLOR_2</li> <li>• undefined</li> </ul>
addImage メソッド	color	<ul style="list-style-type: none"> <li>• COLOR_NONE</li> <li>• COLOR_1 (初期値)</li> <li>• COLOR_2</li> <li>• undefined</li> </ul>
addBarcode メソッド	type	<ul style="list-style-type: none"> <li>• BARCODE_UPC_A</li> <li>• BARCODE_UPC_E</li> <li>• BARCODE_EAN13</li> <li>• BARCODE_JAN13</li> <li>• BARCODE_EAN8</li> <li>• BARCODE_JAN8</li> <li>• BARCODE_CODE39</li> <li>• BARCODE_ITF</li> <li>• BARCODE_CODABAR</li> <li>• BARCODE_CODE93</li> <li>• BARCODE_CODE128</li> </ul>
	font	<ul style="list-style-type: none"> <li>• FONT_A (初期値)</li> <li>• FONT_B</li> <li>• FONT_C</li> </ul>

API	パラメーター	指定可能な設定値
addSymbol メソッド	type	<ul style="list-style-type: none"> <li>• SYMBOL_PDF417_STANDARD</li> <li>• SYMBOL_PDF417_TRUNCATED</li> <li>• SYMBOL_QRCODE_MODEL_1</li> <li>• SYMBOL_QRCODE_MODEL_2</li> <li>• SYMBOL_MAXICODE_MODE_2</li> <li>• SYMBOL_MAXICODE_MODE_3</li> <li>• SYMBOL_MAXICODE_MODE_4</li> <li>• SYMBOL_MAXICODE_MODE_5</li> <li>• SYMBOL_MAXICODE_MODE_6</li> </ul>
	level	<ul style="list-style-type: none"> <li>• LEVEL_0</li> <li>• LEVEL_1</li> <li>• LEVEL_2</li> <li>• LEVEL_3</li> <li>• LEVEL_4</li> <li>• LEVEL_5</li> <li>• LEVEL_6</li> <li>• LEVEL_7</li> <li>• LEVEL_8</li> <li>• LEVEL_L</li> <li>• LEVEL_M</li> <li>• LEVEL_Q</li> <li>• LEVEL_H</li> <li>• LEVEL_DEFAULT</li> </ul>
	width	<ul style="list-style-type: none"> <li>• PDF417</li> <li>• QR Code</li> <li>• MaxiCode</li> </ul>
	height	<ul style="list-style-type: none"> <li>• PDF417</li> <li>• QR Code</li> <li>• MaxiCode</li> </ul>
	size	<ul style="list-style-type: none"> <li>• PDF417</li> <li>• QR Code</li> <li>• MaxiCode</li> </ul>
addCut メソッド	type	<ul style="list-style-type: none"> <li>• CUT_NO_FEED</li> <li>• CUT_FEED</li> <li>• undefined</li> </ul>

## ファームウェア Ver. による使用制限

TM インテリジェントプリンターに搭載されている TM-DT ソフトウェア、TM-i ファームウェアの Ver. によって、使用できる API やパラメーターなどに制限があります。

TM-DT ソフトウェア、TM-i ファームウェアの Ver. 確認や Ver. Up の方法は、各プリンターの詳細取扱説明書を参照してください。

クラス	API	パラメーター	値	説明
ePOSDevice オブジェクト	createDevice メソッド	options	-	TM-DT ソフトウェア Ver.2.2、TM-i ファームウェア 4.0 以降で対応。
Printer オブジェクト	addTextVPosition メソッド	-	-	
	addLayout メソッド	-	-	
	addRecovery メソッド	-	-	
	addReset メソッド	-	-	
	force プロパティ	-	-	
	onbatterystatuschange イベント	-	-	
	onbatteryok イベント	-	-	
	onbatterylow イベント	-	-	
	onbatterylow イベント	-	-	
	addTextLang メソッド	lang	zh-hans	
		lang	zh-hant	
	addSymbol メソッド	type	SYMBOL_AZTECCODE_FULLRANGE	
			SYMBOL_AZTECCODE_COMPACT	
			SYMBOL_DATAMATRIX_SQUARE	
			SYMBOL_DATAMATRIX_RECTANGLE_8	
	addSymbol メソッド	type	SYMBOL_DATAMATRIX_RECTANGLE_12	

クラス	API	パラメーター	値	説明
Printer オブジェクト	addSymbol メソッド	type	SYMBOL_DATAMATRIX_RECTANGLE_16	TM-DT ソフトウェア Ver.2.2、TM-i ファームウェア 4.0 以降で対応。
		level	整数値 (5 ~ 95)	
	addSound メソッド	pattern	PATERN_1	
			PATERN_2	
			PATERN_3	
			PATERN_4	
			PATERN_5	
			PATERN_6	
			PATERN_7	
			PATERN_8	
			PATERN_9	
			PATERN_10	
		cycle	-	
	recover メソッド	-	-	TM-DT ソフトウェア Ver.2.5、TM-i ファームウェア 4.0 以降で対応。
	reset メソッド	-	-	
	send メソッド	printjobid	-	TM-DT ソフトウェア Ver.2.5、TM-i ファームウェア 4.1 以降で対応。
	print メソッド	printjobid	-	
	getPrintJobStatus メソッド	-	-	
	onreceive イベント	printjobid	-	
		code	EX_SPOOLER	
			JobNotFound	
		status	Printing	
			ASB_SPOOLER_IS_STOPPED	
Printer オブジェクト	drawerOpenLevel プロパティ	-	-	TM-DT ソフトウェア Ver.3.0、TM-i ファームウェア 4.1 以降で対応。
DeviceHubTerminal オブジェクト	restart メソッド	-	-	
CommBox オブジェクト	getCommHistory メソッド	option	-	
ePosDeviceConfiguration オブジェクト	-	-	-	

# サンプルプログラム

Epson ePOS SDK for JavaScript が提供するサンプルプログラムは、Web アプリケーションソフト開発者向けの実装サンプルです。

## 機能

Epson ePOS SDK for JavaScript で提供しているサンプルプログラムには、以下の機能があります。

サンプルプログラム	説明	TM インテリジェントプリンター		TM プリンター
		TM-DT	TM-i	
<a href="#">POS Terminal</a> <ul style="list-style-type: none"> <li>• POS Terminal Sample</li> <li>• POS Terminal Customer Display Sample</li> </ul>	POS システムのサンプルプログラムです。タブレット端末に表示するコミュニケーションボックスの機能を使用することもできます。	○	○	-
<a href="#">Entry Terminal</a> <ul style="list-style-type: none"> <li>• Call Sample</li> <li>• Entry Sample</li> </ul>	受付端末のサンプルプログラムです。コミュニケーションボックスの双方向通信機能を使用して、タブレット等の Web ブラウザー間の通信ができます。	○	○	-
<a href="#">Receipt Designer</a>	レシートプリンターの印刷データを簡単に生成し、印刷できます。	○	○	○
<a href="#">Printer Sample</a>	プリンターの印刷データを生成し、印刷できます。 レシートプリンターの場合は、 <a href="#">Receipt Designer</a> で生成することを推奨します。	○	○	○
<a href="#">Hybrid Printer Sample</a>	ハイブリッドプリンター印刷データの生成と印刷、MICR データの読み取りができます。	○	-	-
<a href="#">Customer Display Sample</a>	カスタマーディスプレイの表示データを生成し、表示できます。	○	○	○
<a href="#">Keyboard Sample</a>	キーボードからの入力データを取得できます。	○	○	-
<a href="#">MSR Sample</a>	MSR データを取得できます。	○	-	-
<a href="#">Barcode Scanner Sample</a>	バーコードスキャナーで、バーコードデータを取得できます。	○	○	-



Internet Explorer で POS Terminal Sample を使用する場合は、Internet Explorer 9 以降を使用してください。

## POS Terminal



<POS Terminal Customer Display Sample>



## Entry Terminal

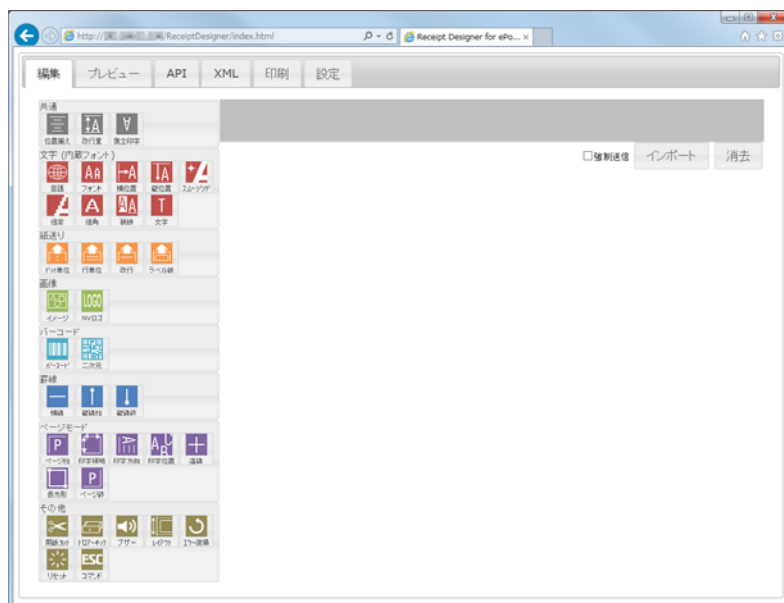
<Entry Sample>



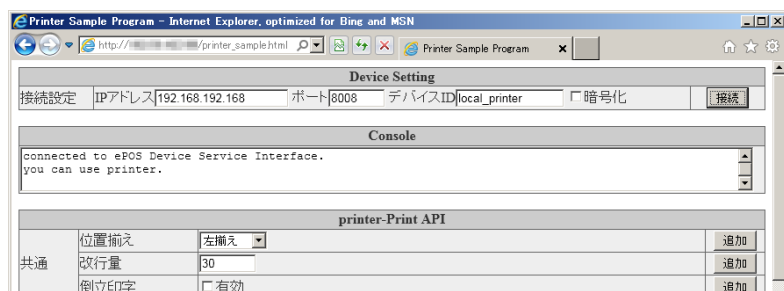
<Call Sample>



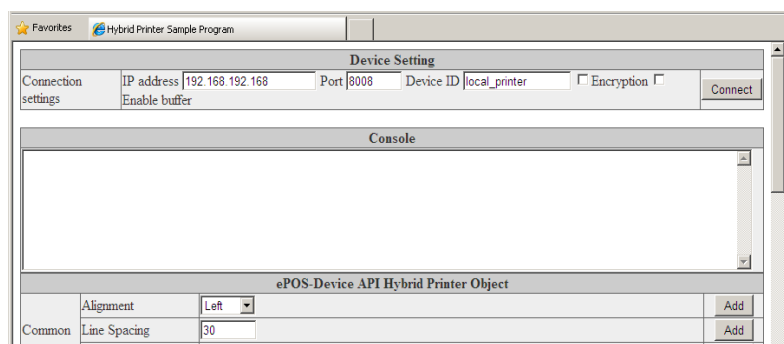
## Receipt Designer



## Printer Sample



## Hybrid Printer Sample



## Customer Display Sample

Customer Display Sample Program - Internet Explorer, optimized for Bing and MSN

http://localhost:8008/display\_sample.html

**Device Setting**

接続設定 IPアドレス 192.168.192.168 ポート 8008 デバイスID local\_display 暗号化 ☐ 接続

**Console**

connected to ePOS Device Service Interface.  
you can use display.

**ePOS Device API Display Object**

ウィンドウ	ウィンドウ設定	ウィンドウ番号 1 X座標 1 Y座標 1 幅 10 高さ 1 スクロールモード	追加
	定義済ウィンドウの破棄	オーバーライト 破棄ウィンドウ番号 1	追加
	カレントウィンドウの変更	移動先ウィンドウ番号 1	追加
	カレントウィンドウ内の表示消去		追加

## Keyboard Sample

POSKeyboard Sample Program - Internet Explorer, optimized for Bing and MSN

http://localhost:8008/keyboard\_sample.html

**Device Setting**

接続設定 IPアドレス 192.168.192.168 ポート 8008 デバイスID local\_keyboard 暗号化 ☐ 接続

onstring用 setPrefix 文字列の始まりと判断するキー F1 登録

**Console**

connected to ePOS Device Service Interface.  
you can use poskeyboard.

**ePOS Device API Keyboard Object**

onkeypress	
------------	--

## MSR Sample

MSR Sample Program - Internet Explorer, optimized for Bing and MSN

http://localhost:8008/msr\_sample.html

**Device Setting**

接続設定 IPアドレス 192.168.192.168 ポート 8008 デバイスID local\_msr 暗号化 ☐ 接続

**Console**

connected to ePOS Device Service Interface.  
you can use msr.

**ePOS Device API MSR Object**

読み取り結果	トラック1のデータ全体	
	トラック2のデータ全体	
	トラック4のデータ全体	
	アカウント番号	
	有効期限 YYMM形式	
	姓	
	名	
	ミドルネームまたはイニシャル	
	タイトル	
	サービスコード	
	トラック1の任意データ	
	トラック2の任意データ	

Barcode Scanner Sample

Barcode Scanner Sample Program - Internet Explorer, optimized for Bing and MSN

http://.../scanner\_sample.htm

Barcode Scanner Sample Pr...

Device Setting

接続設定

IPアドレス192.168.192.168

ポート8008

デバイスIDlocal\_scanner

暗号化

接続

バーコード

体系識別コードの使用

有効

※あらかじめバーコードスキャナで識別コードを使用するための設定をしておいて下さい。

機種設定

タイプ

識別コード長1

識別コード位置バーコードデータの直前

識別コード設定

バーコードタイプ

識別コード

JAN/EAN8

JAN/EAN13

Interleaved 2of5

CODE39

CODE128

NW-7(CODABAR)

## 使用環境

アプリケーションソフトの動作環境を参照してください。

## サンプルプログラムの起動手順

Epson ePOS SDK for JavaScript の提供するサンプルプログラム起動手順を説明します。

### 1. プリンターの環境設定



### 2. サンプルプログラムの起動

#### プリンターの環境設定

プリンターによって環境の設定方法が異なります。

- [TM インテリジェントプリンター](#)
- [TM プリンター](#)

## TM インテリジェントプリンター



各手順の詳細は、各プリンターの詳細取扱説明書を参照してください。

- 1 使用するデバイスをプリンターに接続します。
- 2 プリンターにネットワークを設定します。
- 3 サンプルプログラム (ePOS\_SDK\_Sample\_JavaScript.zip) をプリンターに登録します。  
EPSON TMNet WebConfig の「Web サービス設定」-「Web コンテンツ」-「更新設定」から手動更新機能を使用して登録します。
- 4 デバイスをプリンターに設定します。  
EPSON TMNet WebConfig で、使用するデバイスごとに以下を設定してください。

デバイス	設定項目	設定値
カスタマーディスプレイ	カスタマーディスプレイ	使用する
キーボード	デバイス ID	local_keyboard
	デバイス名	リストから該当するデバイスを選択します。
	制御スクリプト	Keyboard_Generic.js
MSR	デバイス ID	local_msr
	デバイス名	リストから該当するデバイスを選択します。
	制御スクリプト	MSR_V3TU_FK.js
バーコードスキャナー	デバイス ID	local_scanner
	デバイス名	リストから該当するデバイスを選択します。
	制御スクリプト	Scanner_Generic.js

# TM プリンター



各手順の詳細は、各プリンターの詳細取扱説明書を参照してください。

- 1 サンプルプログラム (ePOS\_SDK\_Sample\_JavaScript.zip) を展開し、Web サーバーの以下のフォルダーにコピーします。

システムドライブ :¥lnetpub¥wwwroot (IIS で構築した Web サーバーの場合)



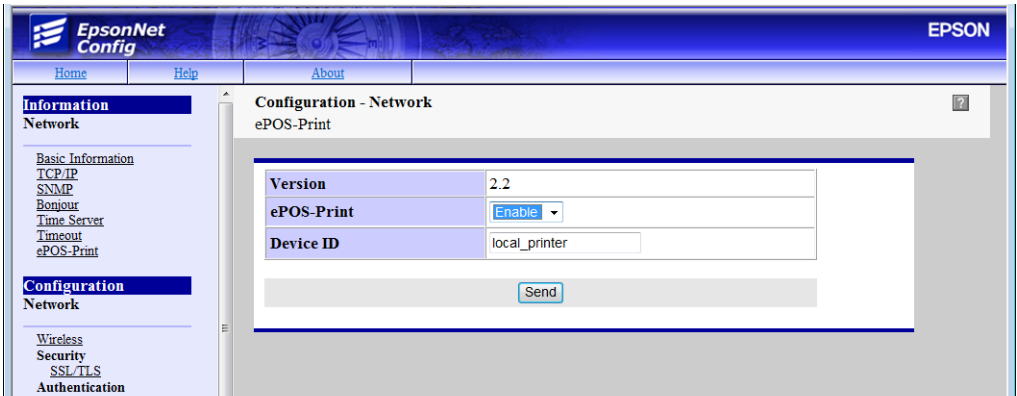
サンプルプログラムは管理者権限のユーザーでコピーしてください。

- 2 プリンターにネットワークを設定します。
- 3 Web ブラウザーを起動し、以下の URL にアクセスします。  
http://[ プリンターの IP アドレス ]/
- 4 [ePOS-Print] をクリックします。“ePOS-Print” 画面が表示されます。  
以下の設定をします。

- 無線 LAN インターフェイス (UB-R04) の場合

項目	設定
ePOS-Print	Enable に設定する。(初期値 : Disabled)
Device ID	システムの設定に合わせる。(初期値 : local_printer)
Printing Method	接続されている TM プリンターに合わせて設定する。
Character Code Table	接続されている TM プリンターに合わせて、TM プリンターのコードページを設定する。

- モバイルプリンターの場合



項目	設定
ePOS-Print	Enable に設定する。(初期値 : Disabled)
Device ID	システムの設定に合わせる。(初期値 : local_printer)

- 5 [Send] をクリックします。
- 6 TM プリンターの電源を切り、再度電源を入れます。

## サンプルプログラムの起動

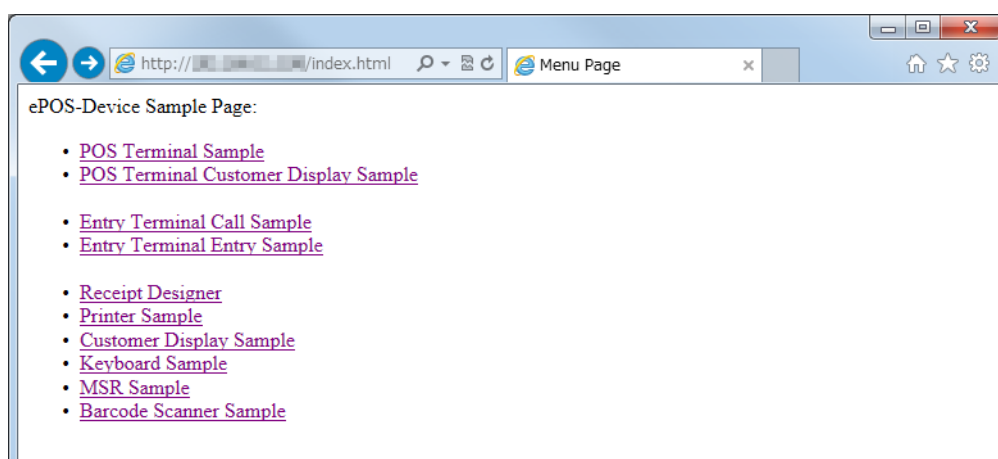
- 1 Web ブラウザーを起動し、以下の URL にアクセスします。

プリンター		URL
TM インテリジェント プリンター	TM-DT シリーズ	http://( プリンターの IP アドレス )/index.html
	TM-i シリーズ	http://( プリンターの IP アドレス )/webapp/index.html
TM プリンター		http://( サーバーの IP アドレス )/index.html

- 2 Web ブラウザーでサンプルプログラムが表示されます。



表示されない場合は、プリンターのネットワーク接続状態を確認してください。



# サンプルプログラムの使い方

- ❑ [POS Terminal Sample](#)
- ❑ [Entry Terminal](#)
- ❑ [Receipt Designer](#)
- ❑ [Printer Sample](#)
- ❑ [Hybrid Printer Sample](#)
- ❑ [Customer Display Sample](#)
- ❑ [Keyboard Sample](#)
- ❑ [MSR Sample](#)
- ❑ [Barcode Scanner Sample](#)

## POS Terminal Sample

POS システムのような動作をさせることができます。

### 準備

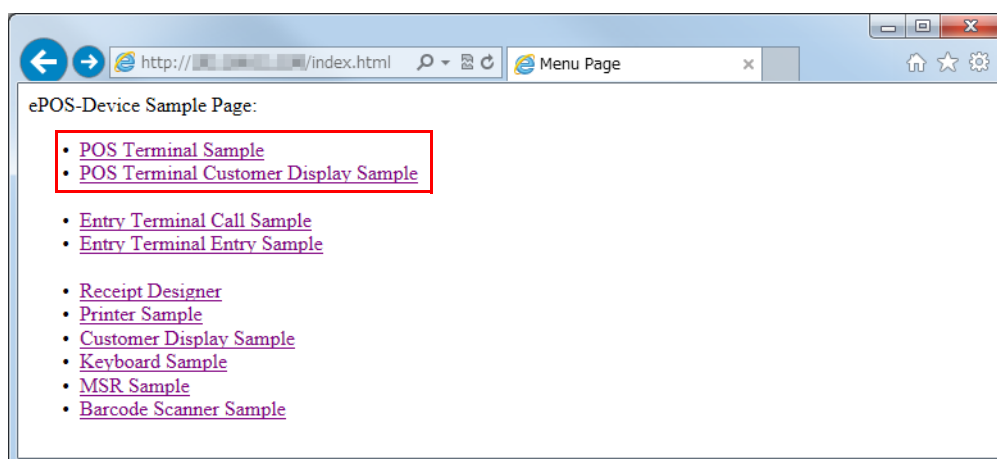
サンプルプログラムを起動し、各デバイスを設定します。

以下のサンプルプログラムを、別々の Web ブラウザーで起動します。

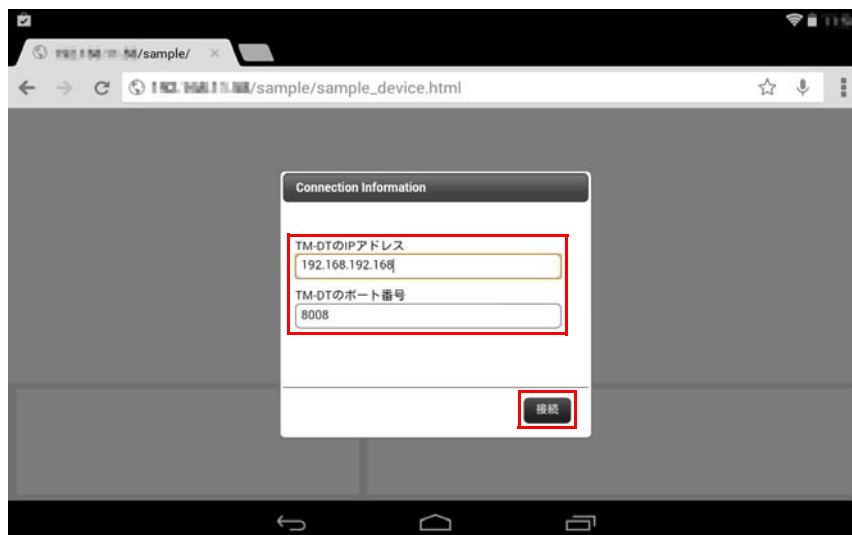
- ❑ [POS Terminal Customer Display Sample](#)
- ❑ [POS Terminal Sample](#)



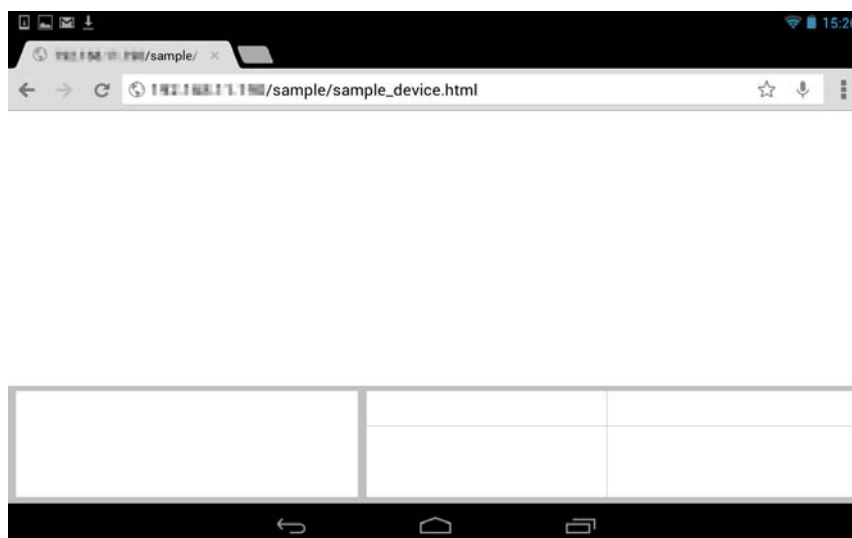
以降の手順に従い、[POS Terminal Customer Display Sample]–[POS Terminal Sample] の順番でサンプルプログラムと接続してください。  
順序を誤ると、POS Terminal Customer Display Sample を正常に使用できません。



- 1 POS Terminal Customer Display Sample 画面を表示します。
- 2 “Connection Information” 画面が表示されます。  
TM インテリジェントプリンターの IP アドレス / ポート番号を設定し、[ 接続 ] ボタンを押します。



- 3 以下の画面が表示されます。



以上で、POS Terminal Customer Display Sample の設定は終了です。

- 4 “POS Terminal Sample” 画面を表示します。[ 設定 ] ボタンを押します。

- 5 “設定” 画面が表示されます。以下の設定をし、[ 接続 ] ボタンを押します。  
ステータスが [OK] になることを確認後、画面を閉じます。

- [ 接続先 ] を設定します。
- 使用するデバイスにチェックし、[ デバイス ID ] を入力します。

使用デバイス	デバイスID	暗号化	その他	ステータス
<input checked="" type="checkbox"/> プリンター	local_printer	<input type="checkbox"/>	紙幅 58mm	OK
<input checked="" type="checkbox"/> ディスプレイ	local_display	<input type="checkbox"/>		OK
<input checked="" type="checkbox"/> キーボード	local_keyboard	<input checked="" type="checkbox"/>		OK
<input checked="" type="checkbox"/> スキャナー	local_scanner	<input checked="" type="checkbox"/>		OK
<input checked="" type="checkbox"/> MSR	local_msr	<input checked="" type="checkbox"/>		OK

以上で、POS Terminal Sample の設定は終了です。

- 6 しばらくすると、POS Terminal Customer Display Sample 画面に、デジタルサイネージが表示されます。



## 操作

商品の読み込みから、レシート発行までの手順を説明します。



商品のバーコードサンプルは、サンプルプログラムに PDF で付属されます。印刷してお使いください。

### 1 バーコードスキャナーまたは、POS キーボードから商品入力します。

商品入力後、[ 小計 ] ボタンを押します。

[ 商品取消 ] ボタンをクリックすると、選択中の商品が削除されます。また、[ 商品全取消 ] ボタンを押すとすべての商品を削除します。

数量を変更する場合、商品入力後に [ 数量 ] ボタンを押し、数字を入力します。数量を確定する時は [ 現計 ] ボタンを押します。

商品名	単価	数量	金額	合計	¥10,980
パーカー	6,500	1	6,500	預り	¥0
スカート	3,400	1	3,400	釣り	¥0
ソックス	540	2	1,080		
				c 数量	小計
				7 8 9	
				4 5 6	
				1 2 3	現計
				0 00 万円	
小計			¥10,980		

商品取消 商品全取消 商品コード

設定 印刷 画面 商品 商品 商品



POS Terminal Customer Display Sample の画面に、スキャンしたデータ、および [ 合計 ] が表示されます。

**2** 預かり金額を入力します。金額入力後、[ 現計 ] ボタンを押します。

[illegible]

### 3 釣りが表示され、プリンターからレシートが印刷されます。

商品名	単価	数量	金額
パーカー	6,500	1	6,500
スカート	3,400	1	3,400
ソックス	540	2	1,080
小計			¥10,980

合計	¥10,980	
預り	¥11,000	
釣り	¥20	

c	数量	小計
7	8	9
4	5	6
1	2	3
0	00	万円

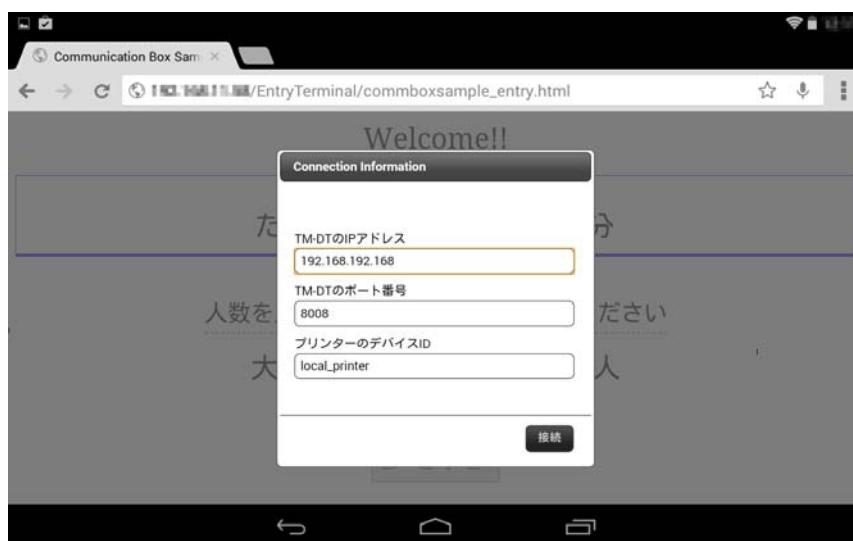
現計

## Entry Terminal

受付端末を想定したサンプルプログラムです。

コミュニケーションボックスの機能を使用し、アプリケーション間でデータの通信を行います。

- 1 受付側のサンプルプログラムを起動します。[Entry Terminal Entry Sample] を選択します。
- 2 “Connection Information” 画面が表示されます。  
TM インテリジェントプリンターの IP アドレス / ポート番号 / プリンターのデバイス ID を設定し、[接続] ボタンを押します。



- 3 Entry Terminal Entry Sample 画面が表示されます。



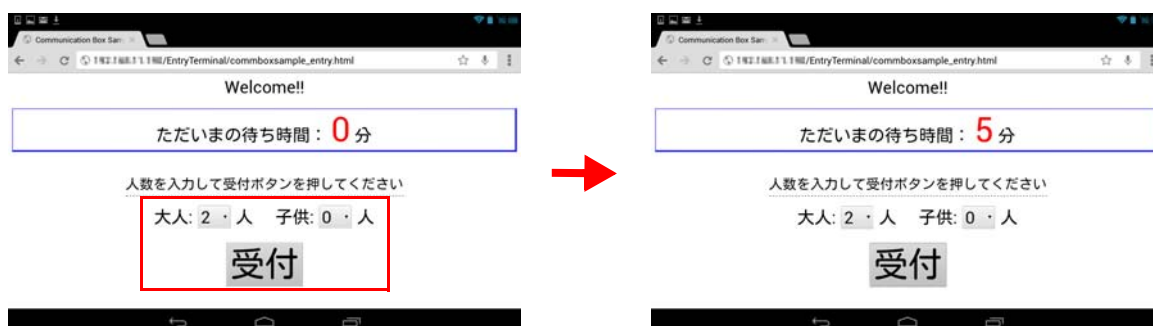
- 4 呼び出し側のサンプルプログラムを起動します。[Entry Terminal Call Sample] を選択します。
- 5 “Connection Information” 画面が表示されます。  
TM インテリジェントプリンターの IP アドレス / ポート番号を設定し、[接続] ボタンを押します。



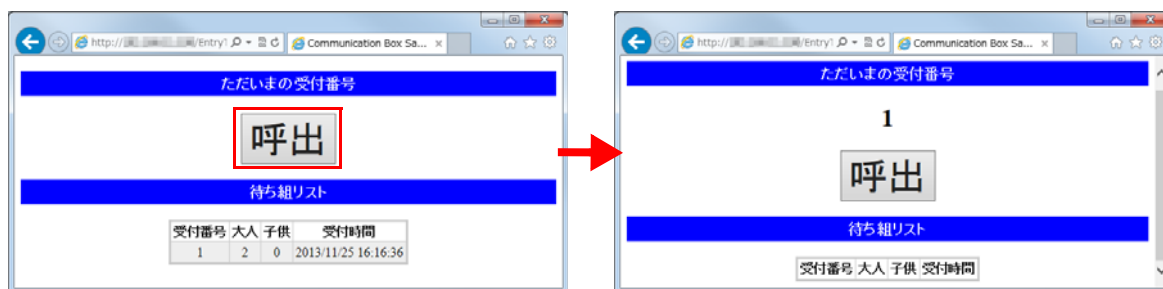
- 6 Entry Terminal Call Sample 画面が表示されます。



- 7 Entry Terminal Entry Sample 画面にしたがって操作します。  
受付番号が、TM インテリジェントプリンターから印刷され、待ち時間が表示されます。



- 8 手順7の操作をすると、Entry Terminal Call Sample 画面に受付番号が表示され、[ 呼出 ] が有効になります。[ 呼出 ] を押します。



- 9 Entry Terminal Entry Sample 画面の待ち時間が更新されます。



## Receipt Designer

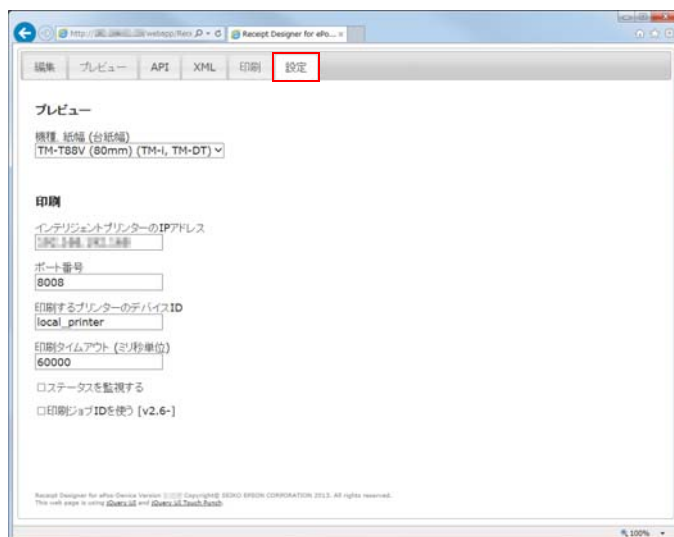
レシートプリンターの印刷データのサンプルコードを簡単に生成し、印刷できます。



- Google Chrome では、画像を含むプレビューを表示すると「SECURITY\_ERR: DOM Exception 18」エラーが発生します。
- Windows Internet Explorer 9 では、印刷を実行すると「SCRIPT5: Access is denied.」エラーが発生します。

## 準備

- 1 サンプルプログラムを起動します。[Receipt Designer] を選択します。
- 2 “Receipt Designer for ePOS-Device” 画面が表示されます。[設定] タブを押します。



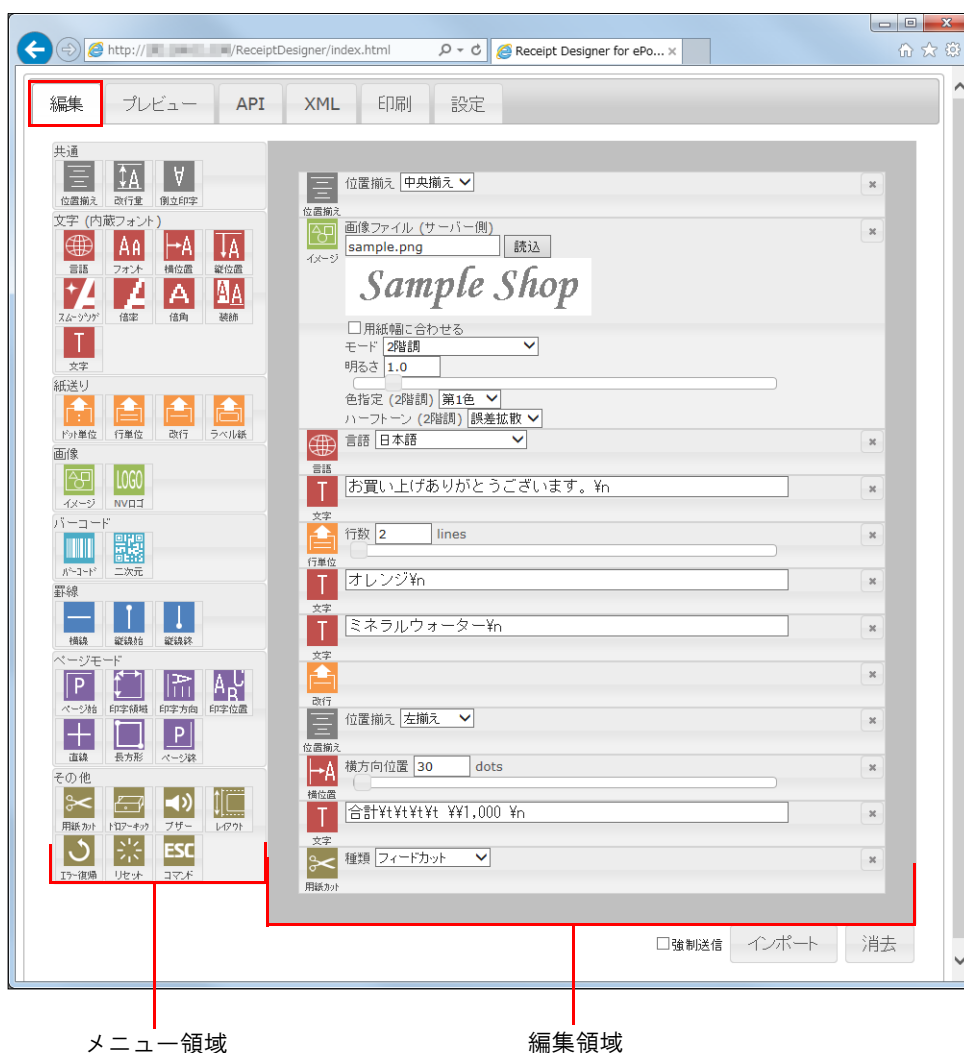
以下を設定します。

機能名	説明
機種、紙幅（台紙幅）	使用するプリンターの機種と紙幅を選択します。 機種の用紙幅にあわせてプレビュー画面が変化します。
インテリジェントプリンターの IP アドレス	プリンターの IP アドレスを指定します。 本項目は必ず指定してください。
ポート番号	TM-DT ソフトウェアのポート番号を指定します。 通常は、8008 を指定します。SSL 通信の場合、8043 を指定します。
印刷するプリンターのデバイス ID	プリンターのデバイス ID を指定します。 本項目は必ず指定してください。
印刷タイムアウト（msec 単位）	印刷のタイムアウト時間を msec 単位で指定します。 最大値は 60000（60 秒）です。
ステータスを監視する	チェックすると、プリンターのステータスを監視します。
印刷ジョブ ID を使う *	チェックすると、印刷ジョブ ID を付与して印刷します。

\* TM-DT ソフトウェア Ver.3.0 以降、TM-i ファームウェア Ver.4.1 以降対応。

## サンプルコードの作成

[ 編集 ] タブを選択すると、編集画面が表示されます。編集画面で Epson ePOS SDK for JavaScript のサンプルコードを作成します。



項目	説明
メニュー領域	使用できる機能が表示されます。 クリックで編集領域の最後に追加、ドラッグで編集領域の任意の位置に挿入できます。
編集領域	メニュー領域で選択された機能が表示されます。 要素の順序は、ドラッグによる入れ替えが可能です。要素右の × ボタンで要素を削除できます。
インポート	Receipt Designer は、ePOS-Device XML を使用して、過去に保存した XML データをインポートできます。詳細は <a href="#">インポート</a> を参照してください。
消去	編集内容を消去します。
強制送信	強制送信モードの設定をします。

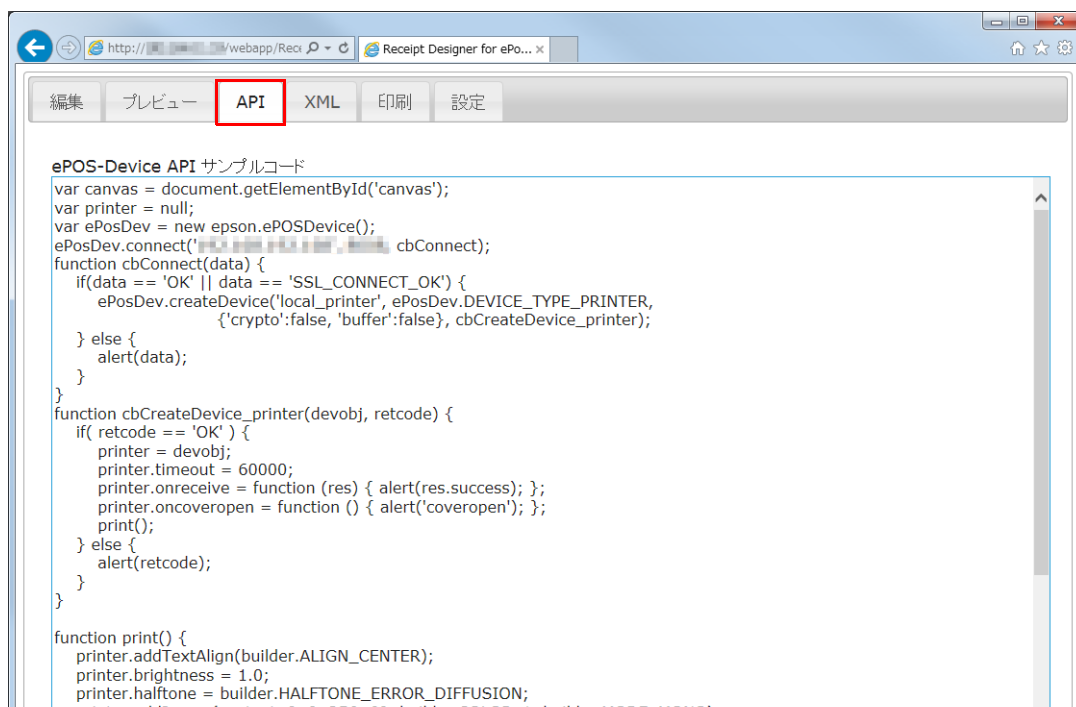
サンプルコードを以下の手順で作成します。

- 1 メニュー領域のアイコンを押し、編集領域に要素を追加します。  
追加した機能は、ドラッグによる順番の入れ替えが可能です。
- 2 追加した要素を設定します。  
例 :NV ロゴを追加した場合はキーコードを設定します。
- 3 [プレビュー] タブを選択し、プレビューを確認します。  
プリンターが接続されている場合は、印刷して確認することもできます。  
(詳細は[印刷](#)を参照してください。)



- ロゴ印字、バーコード印字、2D コード印字、ESC コマンド、ブザーの鳴動、ドロアーキック、用紙カットはアイコンで表示されます。
- プレビュー設定によって、レイアウトが変わる場合があります。  
(詳細は、[準備](#)を参照してください)

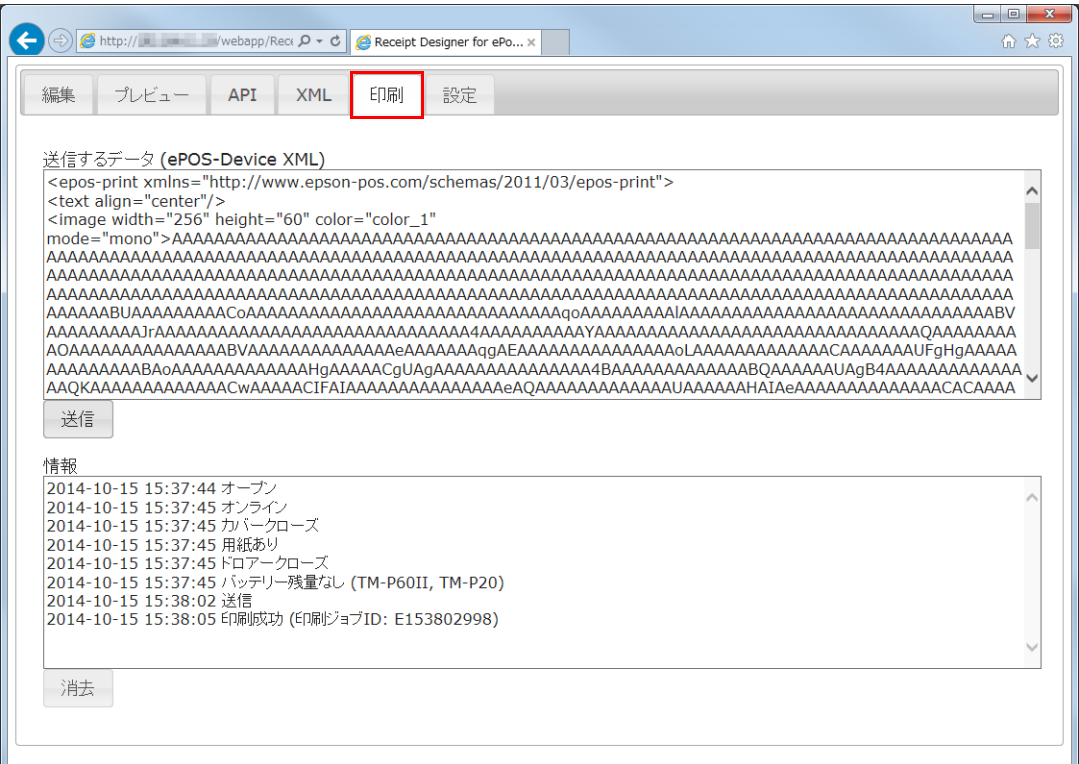
- 4 [API] タブを選択します。Epson ePOS SDK for JavaScript サンプルコードが表示されます。コピーしてお使いください。



インポートで ePOS-Device XML の印刷ドキュメントを使用します。  
必要な方は、XML タブを選択し ePOS-Device XML の印刷ドキュメントの内容をコピーして保存しておいてください。

印刷

プリンター設定に基づいて、印刷ドキュメントをプリンターでテスト印字します。  
(プリンター設定に関しては準備を参照してください)



項目	説明
送信するデータ (ePOS-Device XML):	ePOS-Device XML の印刷ドキュメントが表示されます。
送信	データをプリンターに送信し、印刷します。
情報	印刷ステータスが表示されます。
消去	(情報) ウィンドウの中身を消去します。

以下の手順で印刷を行います。

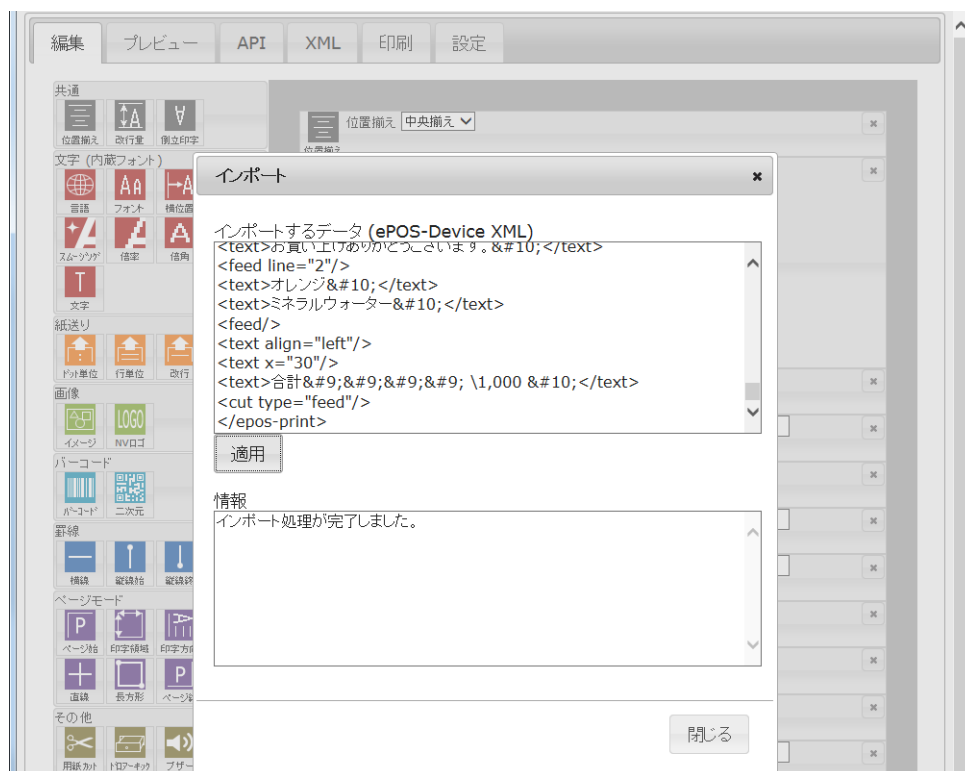
- 1 [印刷] タブを選択します。
- 2 [送信するデータ (ePOS-Device XML)] の内容を確認して、[送信] ボタンを押します。  
[編集] タブで作成した ePOS-Device XML の印刷ドキュメントが“送信するデータ (ePOS-Device XML)”に表示されます。
- 3 インテリジェントプリンターに印刷ドキュメントが印刷されます。  
情報ウィンドウに取得したステータスが表示されます。

## インポート

Receipt Designer は、インポートして一度作成した ePOS-Device XML の印刷ドキュメントを再編集することができます。



Epson ePOS SDK for JavaScript のソースコードを使用したインポートはできません。  
ePOS-Device XML の印字データを使用してインポートを行ってください。



項目	説明
インポートするデータ (ePOS-Device XML):	インポートする ePOS-Device XML の印刷ドキュメントの貼り付け、確認をします。
適用	ePOS-Device XML の印刷ドキュメントをインポートします。
情報	インポート情報が表示されます。
閉じる	インポートウィンドウを閉じます。

Receipt Designer は、以下の手順で ePOS-Device XML の印刷ドキュメントをインポートできます。

- 1 [編集] タブを選択し、[インポート] ボタンを押します。
- 2 [インポート] ウィンドウが表示されます。[インポートするデータ (ePOS-Device XML)] に ePOS-Device XML の印刷ドキュメントを貼り付けます。
- 3 [適用] ボタンを押します。
- 4 [確認] 画面が表示されます。[はい] ボタンを押します。

## Printer Sample

プリンターを動作させたり、プリンターのサンプルコードを生成したりすることができます。

- 1 サンプルプログラムを起動します。[Printer Sample] を選択します。
- 2 “Printer Sample” 画面が表示されます。[接続] ボタンを押します。接続に成功すると、[Console] に以下のメッセージが表示されます。



[Device Setting] の設定を変更する必要はありません。ただし、IP アドレスなどを変更している場合、変更した値に合わせてください。

Printer Sample Program - Internet Explorer, optimized for Bing and MSN

http://.../printer\_sample.html

Printer Sample Program

Device Setting

接続設定 IPアドレス 192.168.192.168 ポート 8008 デバイスID local\_printer 暗号化 [接続]

Console

connected to ePOS Device Service Interface.  
you can use printer.

printer-Print API

共通	位置揃え	左揃え	追加
	改行量	30	追加
	倒立印字	<input type="checkbox"/> 有効	追加
文字	印字文字列	Hello, World! 水平タブ(HT): '\t', 改行(LF): '\n', 復帰(CR): '\r', バックスラッシュ: '\'	追加
	対象言語	日本語	追加
	フォント	フォントA	追加
	スムージング	<input type="checkbox"/> 有効	追加
	倍角	<input type="checkbox"/> 横倍角 <input type="checkbox"/> 縦倍角	追加
	倍率	横方向 [標準] 縦方向 [標準]	追加
装飾	<input type="checkbox"/> 白黒反転 <input type="checkbox"/> アンダーライン <input type="checkbox"/> 強調 色指定 第1色	追加	

- 3 動作させたい機能を設定し、[追加] ボタンを押します。

printer-Print API

共通	位置揃え	左揃え	追加
	改行量	30	追加
	倒立印字	<input type="checkbox"/> 有効	追加
文字	印字文字列	Hello, World! 水平タブ(HT): '\t', 改行(LF): '\n', 復帰(CR): '\r', バックスラッシュ: '\'	追加
	対象言語	日本語	追加
	フォント	フォントA	追加
	スムージング	<input type="checkbox"/> 有効	追加
	倍角	<input type="checkbox"/> 横倍角 <input type="checkbox"/> 縦倍角	追加
	倍率	横方向 [標準] 縦方向 [標準]	追加
装飾	<input type="checkbox"/> 白黒反転 <input type="checkbox"/> アンダーライン <input type="checkbox"/> 強調 色指定 第1色	追加	

- 4 [printer-Print API サンプルコード] に、手順3で追加した機能のソースが表示されます。コピーして使用することができます。

printer-Print API サンプルコード

```

var canvas = document.getElementById('canvas');
var printer = null;
var ePosDev = new Epson.ePOSDevice();
ePosDev.connect('192.168.192.168', 8008, cbConnect);
function cbConnect(data) {
    if (data == 'OK') {
        ePosDev.createDevice('local_printer', ePosDev.DEVICE_TYPE_PRINTER, false,
        cbCreateDevice_printer);
    } else {
        alert(data);
    }
}

```

テスト印刷

送信

開始

終了

- 5 [送信] ボタンをクリックすると、手順3で追加した機能に合わせてテスト印字されます。

## Hybrid Printer Sample

ハイブリッドプリンターを動作させたり、ハイブリッドプリンターのサンプルコードを生成したりすることができます。

### 準備

- 1 サンプルプログラムを起動します。[Hybrid Printer Sample] を選択します。
- 2 “Hybrid Printer Sample” 画面が表示されます。[Connect] ボタンを押します。接続に成功すると、[Console] に以下のメッセージが表示されます。



[Device Setting] の設定を変更する必要はありません。ただし、IP アドレスなどを変更している場合、変更した値に合わせてください。

ePOS-Device API Hybrid Printer Object			
Common	Alignment	Left	Add
	Line Spacing	30	Add
	Upside-down	<input type="checkbox"/> Upside-down	Add

## 印刷機能の使い方

- 1 [ePOS-Device API Hybrid Printer Object] から使用したい機能を設定し、[Add] ボタンを押します。

ePOS-Device API Hybrid Printer Object			
Common	Alignment	Left	Add
	Line Spacing	30	Add
	Upside-down	<input type="checkbox"/> Upside-down	Add
Text	Print Characters	Hello, World!	Add
	Language	ANK	Add
	Font	Font A	Add
	Smoothing	<input type="checkbox"/> Enabled	Add
	Double	<input type="checkbox"/> Double-width <input type="checkbox"/> Double-height	Add

- 2 [target] から印刷方法を選択します。

Hybrid Printer Object Sample Code — target: Receipt

```

var canvas = document.getElementById('canvas');
var printer = null;
var ePosDev = new Epson.ePOSDevice();
ePosDev.connect('192.168.192.168', 8008, cbConnect);
function cbConnect(data) {
    if(data == 'OK') {
        ePosDev.createDevice('local_printer', ePosDev.DEVICE_TYPE_HYBRIDPRINTER,
        cbCreateDevice_hybridprinter);
    } else {
        alert(data);
    }
}

```

Clear

- 3 [Hybrid Printer Object Sample Code] に、手順 1 で追加した機能のソースが表示されます。コピーして使用することができます。

Hybrid Printer Object Sample Code — target: Receipt

```

var canvas = document.getElementById('canvas');
var printer = null;
var ePosDev = new Epson.ePOSDevice();
ePosDev.connect('192.168.192.168', 8008, cbConnect);
function cbConnect(data) {
    if(data == 'OK') {
        ePosDev.createDevice('local_printer', ePosDev.DEVICE_TYPE_HYBRIDPRINTER,
        cbCreateDevice_hybridprinter);
    } else {
        alert(data);
    }
}

```

Clear

- 4 [Test print] の各印刷方法を設定します。[Print] ボタンを押すと、手順 3 で追加した機能と合わせてテスト印刷を実行します。

Hybrid Printer Object Sample Code — target: Receipt

```

var canvas = document.getElementById('canvas');
var printer = null;
var ePosDev = new Epson.ePOSDevice();
ePosDev.connect('192.168.192.168', 8008, cbConnect);
function cbConnect(data) {
    if(data == 'OK') {
        ePosDev.createDevice('local_printer', ePosDev.DEVICE_TYPE_HYBRIDPRINTER,
        cbCreateDevice_hybridprinter);
    } else {
        alert(data);
    }
}

```

Clear

Test print

lock/unlock: lock, unlock

MICR: ignoreerror: ☒ true ☐ false, font: E13B, timeout: 1 (min), Read, Eject, Cancel, Clean

Test print: Receipt print: ☐ Forced Transmission, Print; Slip print: timeout 1 (min), Print, Cancel; Endorse print: timeout 1 (min), Print, Cancel

## MICR 機能の使い方

- 1 [Test print] から [lock] ボタンを押します。

Test print

lock/unlock

lock  
unlock

- 2 MICR の設定後、[Read] ボタンを押します。

Hybrid Printer Object Sample Code -- target Receipt

```

var canvas = document.getElementById('canvas');
var printer = null;
var ePosDev = new Epson.EPOSDevice();
ePosDev.connect('192.168.192.168', 8008, cbConnect);
function cbConnect(data) {
  if(data == 'OK') {
    ePosDev.createDevice('local_printer', ePosDev.DEVICE_TYPE_HYBRIDPRINTER,
    cbCreateDevice_hybridprinter);
  } else {
    alert(data);
  }
}
  
```

Test print

lock/unlock

MICR

ignoreerror ☒ true ☐ false

font E13B

timeout 1 (min)

Read  
Eject  
Cancel  
Clean

Test print

Receipt print ☐ Forced Transmission

Print

Slip print

timeout 1 (min)

Print  
Cancel

Endorse print

timeout 1 (min)

Print  
Cancel

- 3 ハイブリッドプリンターにチェック紙をセットします。  
手順2の設定で MICR データをスキャンします。

- 4 処理の完了後、[Eject] ボタンを押します。  
チェック紙が排出されます。

Test print

lock/unlock

lock  
unlock

MICR

ignoreerror ☒ true ☐ false

font E13B

timeout 1 (min)

Read  
Eject  
Cancel  
Clean

- 5 [unlock] ボタンを押します。

Test print

lock/unlock

lock  
unlock

## Customer Display Sample

カスタマーディスプレイに表示させるサンプルコードを生成させたり、生成させたコードを送信してカスタマーディスプレイを動作させたりできます。

- 1 サンプルプログラムを起動します。[Customer Display Sample] を選択します。
- 2 “Customer Display Sample” 画面が表示されます。[ 接続 ] ボタンを押します。接続に成功すると、[Console] に以下のメッセージが表示されます。また、カスタマーディスプレイの表示が変わります。



[Device Setting] の設定を変更する必要はありません。ただし、IP アドレスなどを変更している場合、変更した値に合わせてください。

The screenshot shows the 'Customer Display Sample Program' interface. The 'Device Setting' section includes fields for '接続設定' (Connection Setting), 'IPアドレス' (IP Address: 192.168.192.168), 'ポート' (Port: 8008), 'デバイスID' (Device ID: local\_display), and a checkbox for '暗号化' (Encryption). A red box highlights the '接続' (Connect) button. The 'Console' section shows a message: 'connected to ePOS Device Service Interface. you can use display.' Below this is the 'ePOS Device API Display Object' section, which contains various settings for windows and cursors, each with an '追加' (Add) button.

- 3 動作させたい機能を設定し、[ 追加 ] ボタンを押します。

The screenshot shows the 'Customer Display Sample Program' interface with the 'Text' and 'Display Properties' sections expanded. The 'Text' section includes 'テキスト表示' (Text Display) and '反転テキスト表示' (Inverted Text Display), both with 'Hello World!' text and '追加' (Add) buttons. The 'Display Properties' section includes '画面の点滅' (Screen Blinking) and '輝度' (Brightness), both with '追加' (Add) buttons. The 'その他' (Others) section includes '時計表示' (Clock Display) and 'コマンド' (Command), both with '追加' (Add) buttons. A red box highlights the '追加' (Add) button for the 'テキスト表示' (Text Display) section.

- 4 [Display Object サンプルコード] に、手順3で追加した機能のソースが表示されます。コピーして使用することができます。

テキスト	反転テキスト表示	<input type="checkbox"/> Hello World! 言語 [指定なし] X座標 [指定なし] Y座標 [2]	追加
	マーキー表示	文字列 Hello World! 表示方法 [右端から表示] 表示間隔(100~2,000)(ms) [100] 繰り返し間隔(100~2,000)(ms) [100] 繰り返し回数(0~127) [0] 言語 [指定なし]	追加
表示属性	画面の点滅	点滅間隔(0~12,700)(ms) [300]	追加
	輝度	[20%]	追加
その他	時計表示		追加
	コマンド	[12345]	追加
	初期化		追加
<div style="border: 1px solid gray; padding: 5px; margin-top: 5px;">           Display Object サンプルコード         </div> <div style="border: 1px solid red; height: 100px; width: 100%; margin-top: 5px;"></div> <div style="text-align: right; margin-top: 5px;"> <input type="button" value="消去"/> </div>			
テスト表示			送信

- 5 [送信] ボタンをクリックすると、手順3で追加した機能に合わせてカスタマーディスプレイの表示が変わります。

## Keyboard Sample

POS キーボードからの入力を取得することができます。

- 1 サンプルプログラムを起動します。[Keyboard Sample] を選択します。
- 2 “Keyboard Sample” 画面が表示されます。[ 接続 ] ボタンを押します。接続に成功すると、[Console] に以下のメッセージが表示されます。



[Device Setting] の設定を変更する必要はありません。ただし、IP アドレスなどを変更している場合、変更した値に合わせてください。

Device Setting	
接続設定	IPアドレス 192.168.192.168    ポート 8008    デバイスID local_keyboard <input type="checkbox"/> 暗号化 <input type="button" value="接続"/> <input type="button" value="登録"/>
onstring用 setPrefix	文字列の始まりと判断するキー F1

Console	
connected to ePOS Device Service Interface. you can use poskeyboard.	

ePOS Device API Keyboard Object	
	onkeypress

- 3 文字列の始まりと判断するキーを選択し、[ 登録 ] ボタンを押します。  
あらかじめ POS キーボード側の登録が必要です。  
例：F1+ 指定文字列 +Enter
- 4 POS キーボードから入力すると、入力結果が [ePOS Device API Keyboard Object] に表示されます。  
文字列を直接入力： onkeypress に表示されます。  
POS キーに設定された文字列を入力：文字列が指定されたキーを押すと、onstring に表示されます。

Device Setting	
接続設定	IPアドレス 192.168.192.168    ポート 8008    デバイスID local_keyboard <input type="checkbox"/> 暗号化 <input type="button" value="接続"/> <input type="button" value="登録"/>
onstring用 setPrefix	文字列の始まりと判断するキー F1

Console	
connected to ePOS Device Service Interface. you can use poskeyboard.	

ePOS Device API Keyboard Object	
読み取り結果	onkeypress
	onstring

## MSR Sample

MSR からの入力を取得することができます。



TM インテリジェントプリンターの TM-DT シリーズのみ使用できます。

- 1 サンプルプログラムを起動します。[MSR Sample] を選択します。
- 2 “MSR Sample” 画面が表示されます。[ 接続 ] ボタンを押します。接続に成功すると、[Console] に以下のメッセージが表示されます。



[Device Setting] の設定を変更する必要はありません。ただし、IP アドレスなどを変更している場合、変更した値に合わせてください。

Device Setting	
接続設定	IPアドレス 192.168.192.168    ポート 8008    デバイスID local_msr <input type="checkbox"/> 暗号化 <input type="button" value="接続"/>

Console	
connected to ePOS Device Service Interface. you can use msr.	

ePOS Device API MSR Object	
読み取り結果	トラック1のデータ全体
	トラック2のデータ全体
	トラック4のデータ全体
	アカウント番号
	有効期限"YYMM"形式
	姓
	名
	ミドルネームまたはイニシャル
	タイトル
	サービスコード
トラック1の任意データ	
トラック2の任意データ	

- 3 カードを MSR に読み込ませます、読み込ませた情報が [ePOS Device API MSR Object] に表示されます。

ePOS Device API MSR Object	
読み取り結果	トラック1のデータ全体
	トラック2のデータ全体
	トラック4のデータ全体
	アカウント番号
	有効期限"YYMM"形式
	姓
	名
	ミドルネームまたはイニシャル
	タイトル
	サービスコード
トラック1の任意データ	
トラック2の任意データ	

## Barcode Scanner Sample

バーコードスキャナーからの入力を取得することができます。

- 1 サンプルプログラムを起動します。[Barcode Scanner Sample] を選択します。
- 2 “Barcode Scanner Sample” 画面が表示されます。[ 接続 ] ボタンを押します。接続に成功すると、[Console] に以下のメッセージが表示されます。



[Device Setting] の設定を変更する必要はありません。ただし、IP アドレスなどを変更している場合、変更した値に合わせてください。

The screenshot shows the 'Barcode Scanner Sample Program' interface in Internet Explorer. The 'Device Setting' tab is active, displaying fields for connection settings (IP address: 192.168.192.168, Port: 8008, Device ID: local\_scanner) and a '接続' (Connect) button. Below this, there are sections for 'Barcode System Identification Code Usage' (with a checkbox for '有効' - Valid), 'Device Setting' (with a dropdown for 'タイプ' and a text input for '識別コード長'), and 'Barcode Identification Code Setting' (with a dropdown for '識別コード位置' and a table for selecting barcode types and their corresponding identification codes). The 'Console' section at the bottom shows the message: 'connected to ePOS Device Service Interface. you can use scanner.'

- 3 バーコードを読み込ませます、読み込ませた情報が [ePOS Device API Scanner Object] に表示されます。

This screenshot shows the same interface as the previous one, but with the 'ePOS Device API Scanner Object' section expanded at the bottom. It contains a table with the following data:

読み取り結果	識別用データ	識別コード	バーコードタイプ
	バーコードデータ	1000000001044	

# 活用ガイド

## 常時監視するには

```

var ePosDev = new epos.ePOSDevice();
var printer = null;

function connect(){
    // デバイスと接続
    ePosDev.connect('192.168.192.168', '8008', callback_connect);
}

function callback_connect(resultConnect){
    if ((resultConnect == 'OK') || (resultConnect == 'SSL_CONNECT_OK')) {
        //Printer オブジェクトを取得する
        ePosDev.createDevice('local_printer', ePosDev.DEVICE_TYPE_PRINTER, {'crypto' :
false, 'buffer' : false}, callback_createDevice);
    }
    else {
        // エラーメッセージ表示
    }
}

function callback_createDevice(deviceObj, retcode){
    printer = deviceObj;
    if( retcode == 'OK' ) {
        printer = devobj;
        printer.timeout = 60000;
        // イベントを登録
        printer.onstatuschange = function (res) { alert(res.success); };
        printer.onbatterystatuschange = function (res) { alert(res.success); };
        print();
    } else {
        alert(retcode);
    }
}

function startMonitor(){
    // ステータス監視の開始
    printer.startMonitor();
}

// プリンターのカバーを開ける

function startMonitor(){
    // ステータス監視の終了
    printer.stopMonitor();
}

function disconnect(){
    //Printer オブジェクトを破棄する
    ePosDev.deleteDevice(printer, callback_deleteDevice);
}

function callback_deleteDevice(errorCode){
    // デバイスと切断
    ePosDev.disconnect();
}

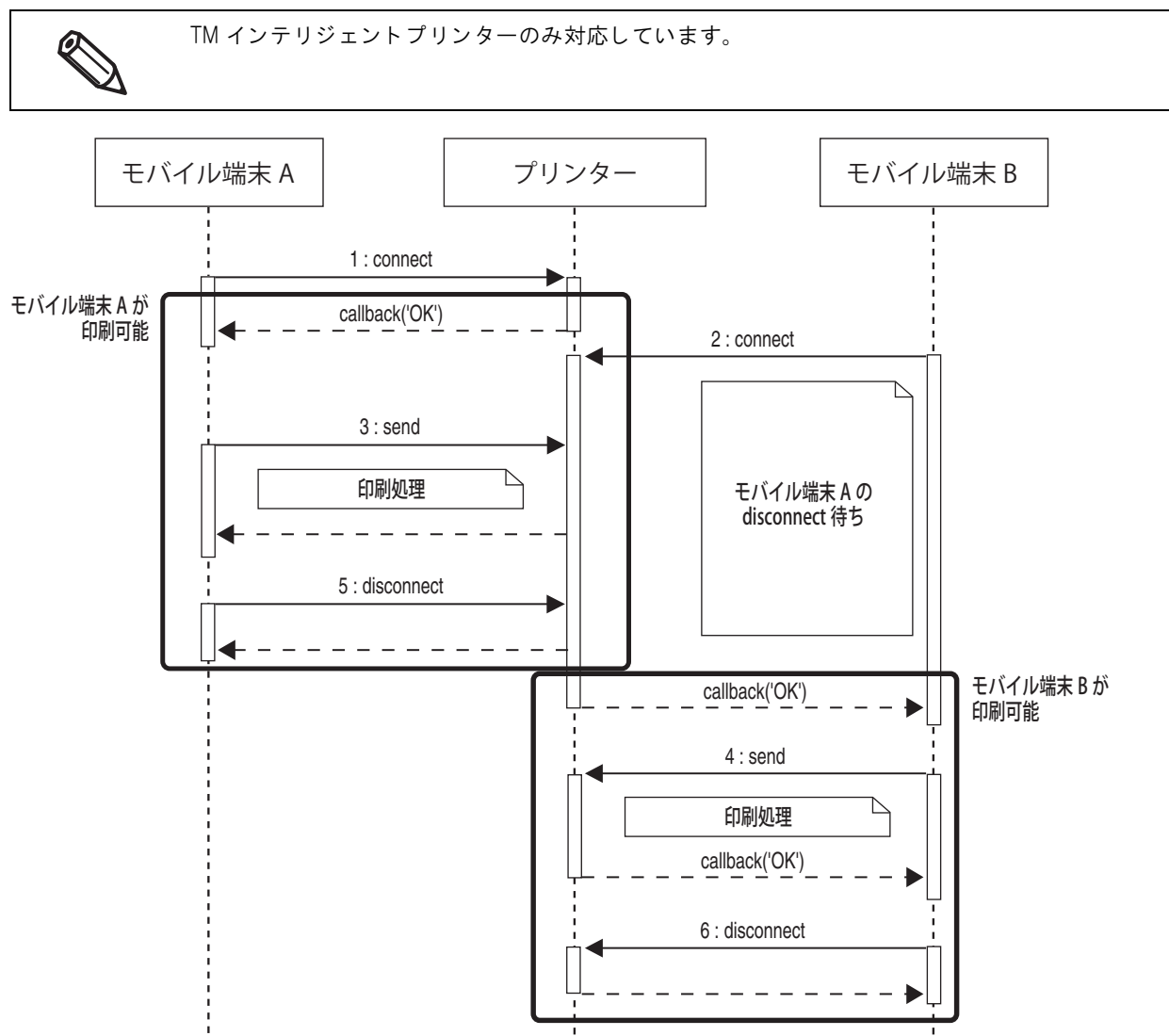
```



「対処メッセージ表示」に表示するメッセージは、`onreceive` イベントで取得するエラーコードと対処方法を参照してください。

## 一台のプリンターを複数のモバイル端末から使用するには

モバイル端末 A とモバイル端末 B から、1 台のプリンターを使用する場合の処理の流れを以下に示します。

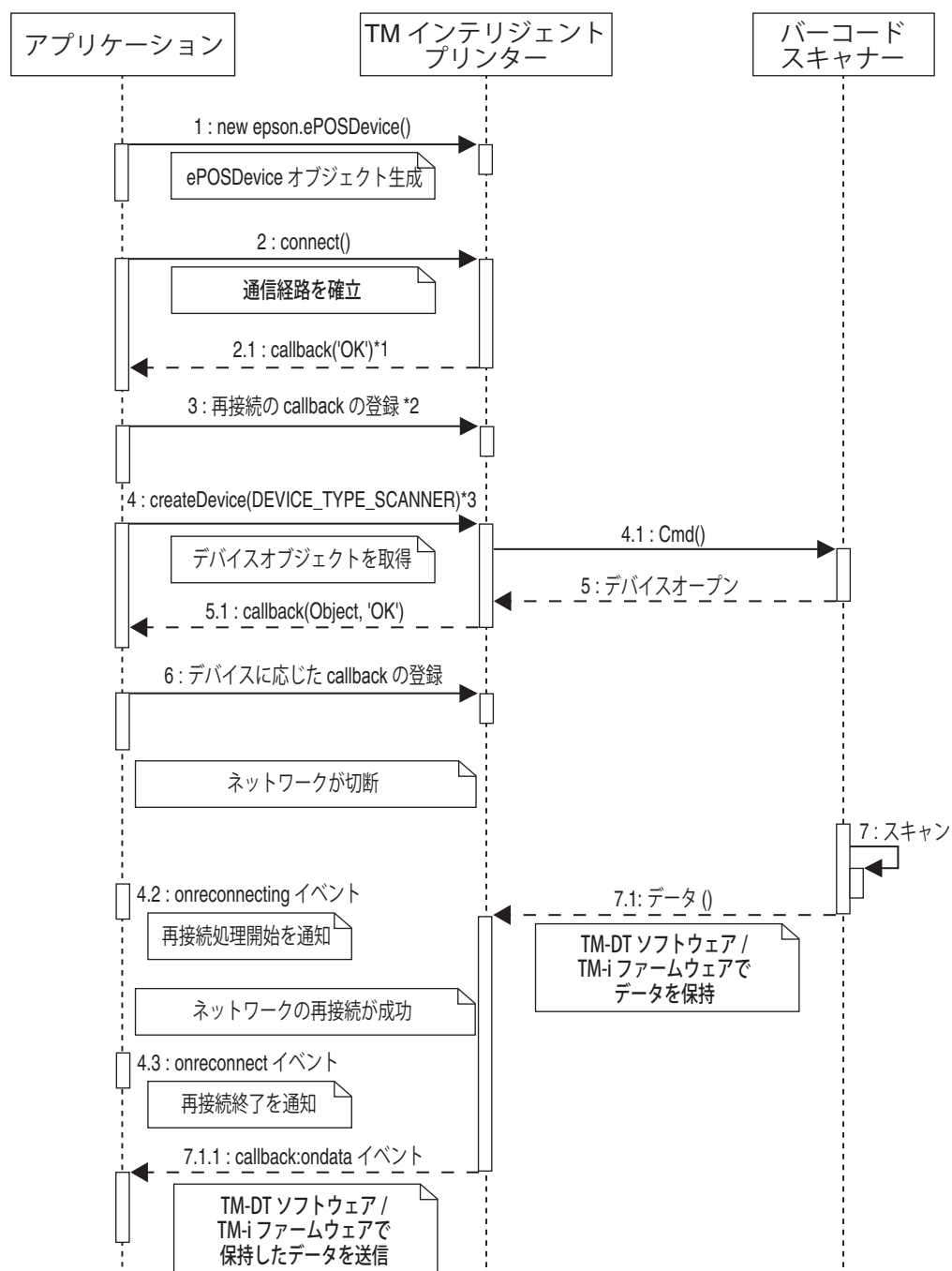


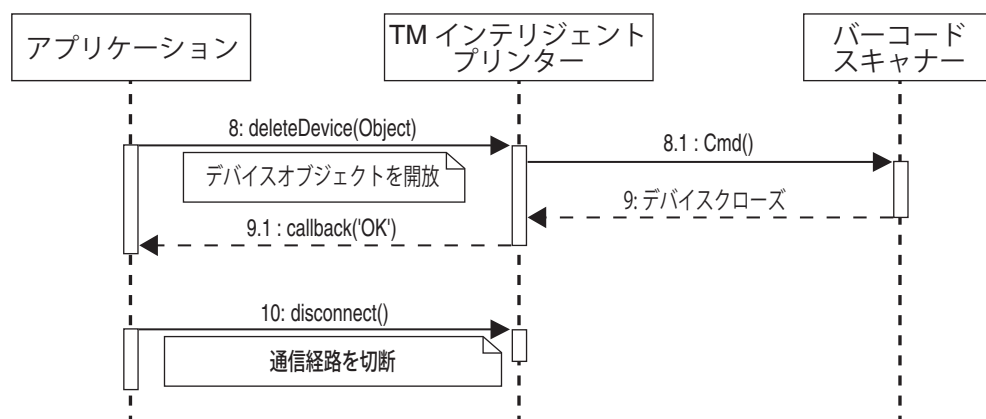
## ネットワークを自動で再接続するには

TM インテリジェントプリンターとのネットワーク通信が切断された場合、自動で再接続するシーケンスは、以下の通りです。



以下はバーコードスキャナー使用時を例にしています。





\*1:“SSL\_CONNECT\_OK” (SSL 通信)

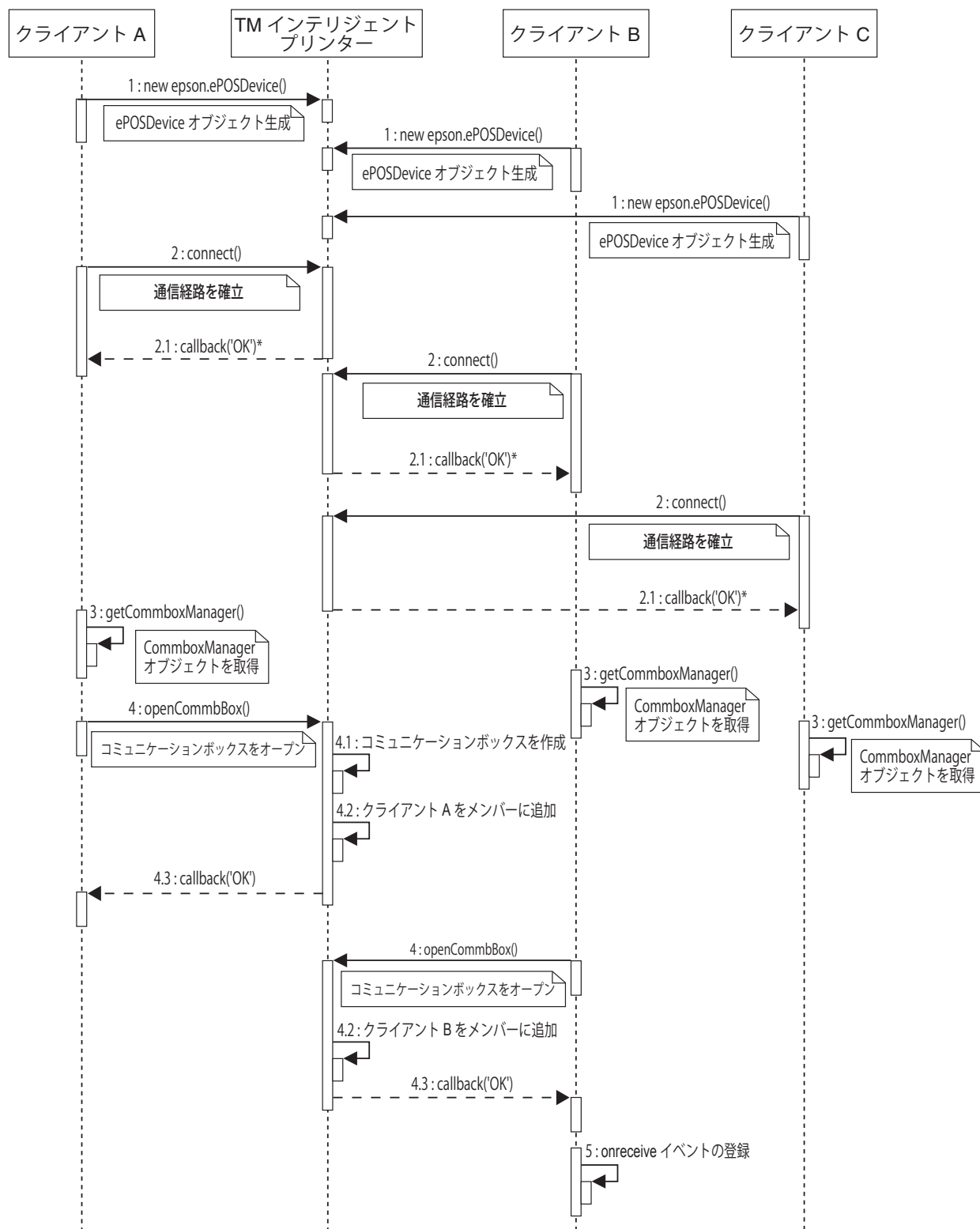
\*2: 再接続処理開始イベント / 再接続終了イベント / 再接続失敗イベントが含まれます。

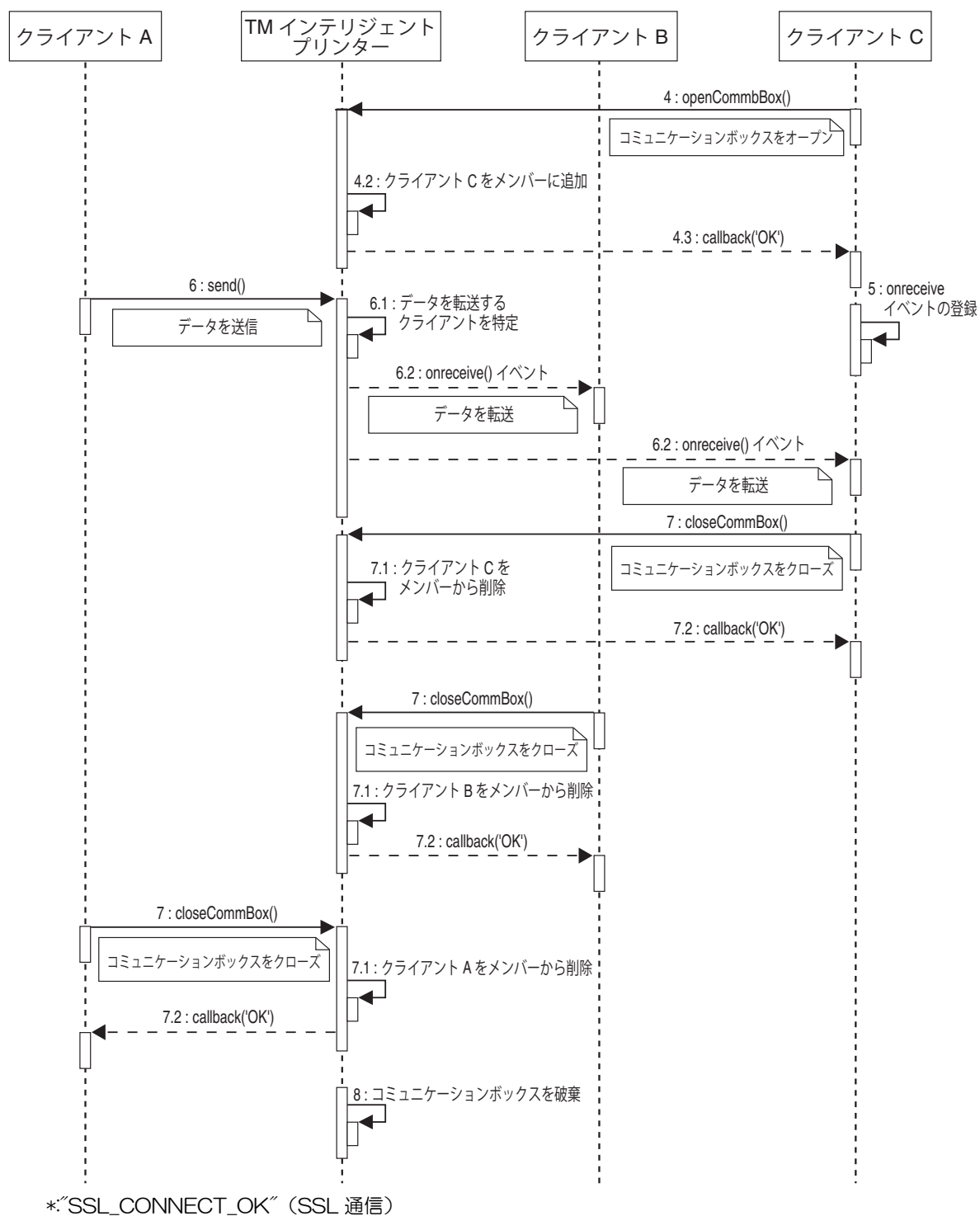
\*3:options パラメーターを、ネットワークの再接続時にデータを再送する設定にします。

## アプリケーション間でデータを送受信するには

TM インテリジェントプリンターのコミュニケーションボックス機能を使用します。

コミュニケーションボックス使用した基本的なプログラミングのシーケンスは、以下の通りです。シーケンス図中の「クライアント」は、アプリケーションを指しています。

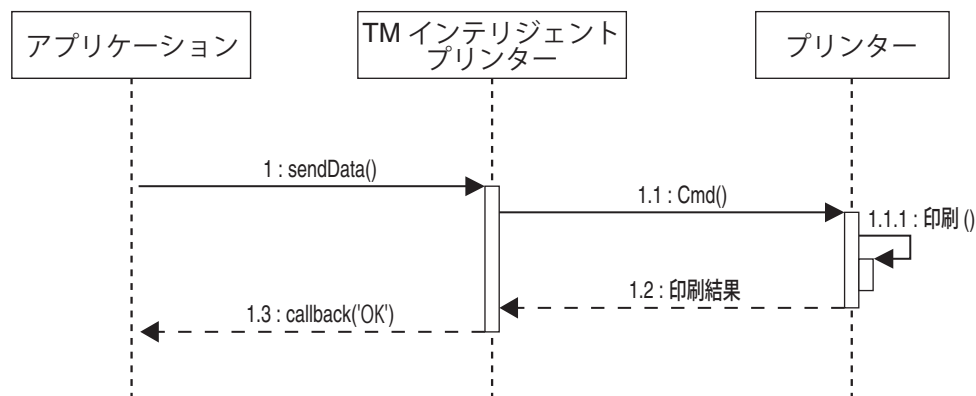




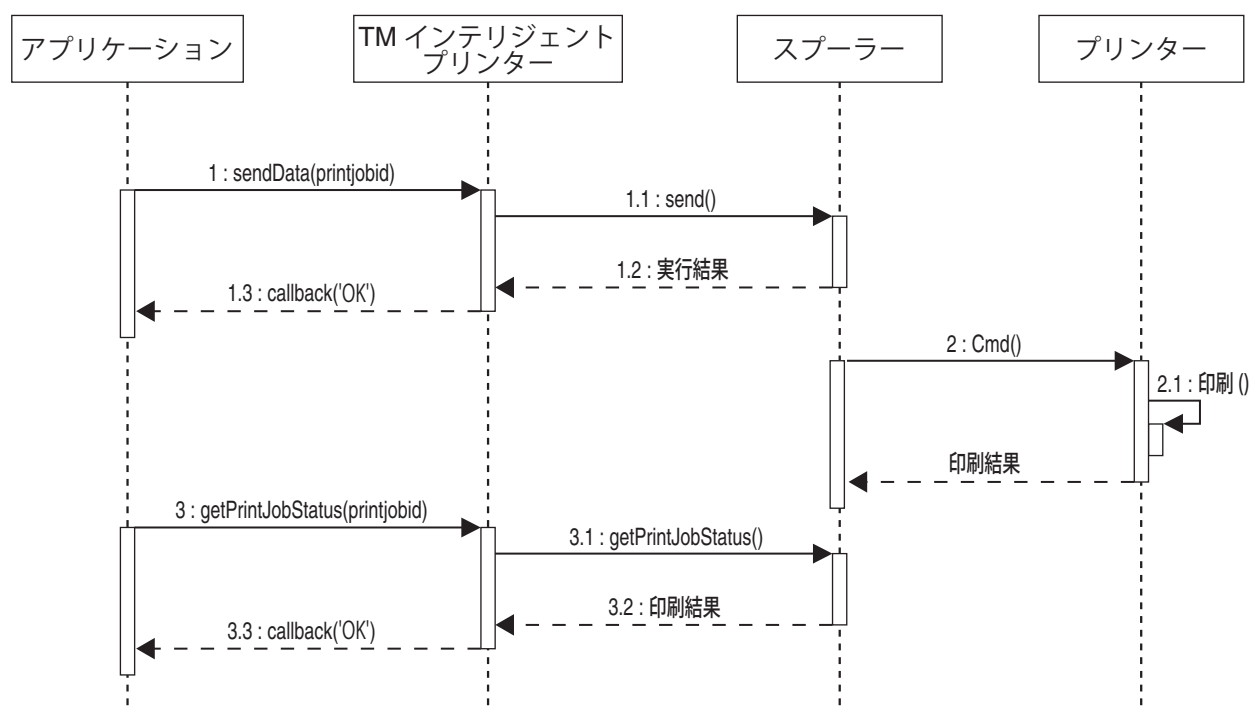
## 迂回印刷するには

TM インテリジェントプリンターのスプーラー機能を使用します。スプーラー機能、迂回印刷は TM-DT ソフトウェア Ver.3.0 以降、TM-i ファームウェア Ver.4.1 以降で対応しています。

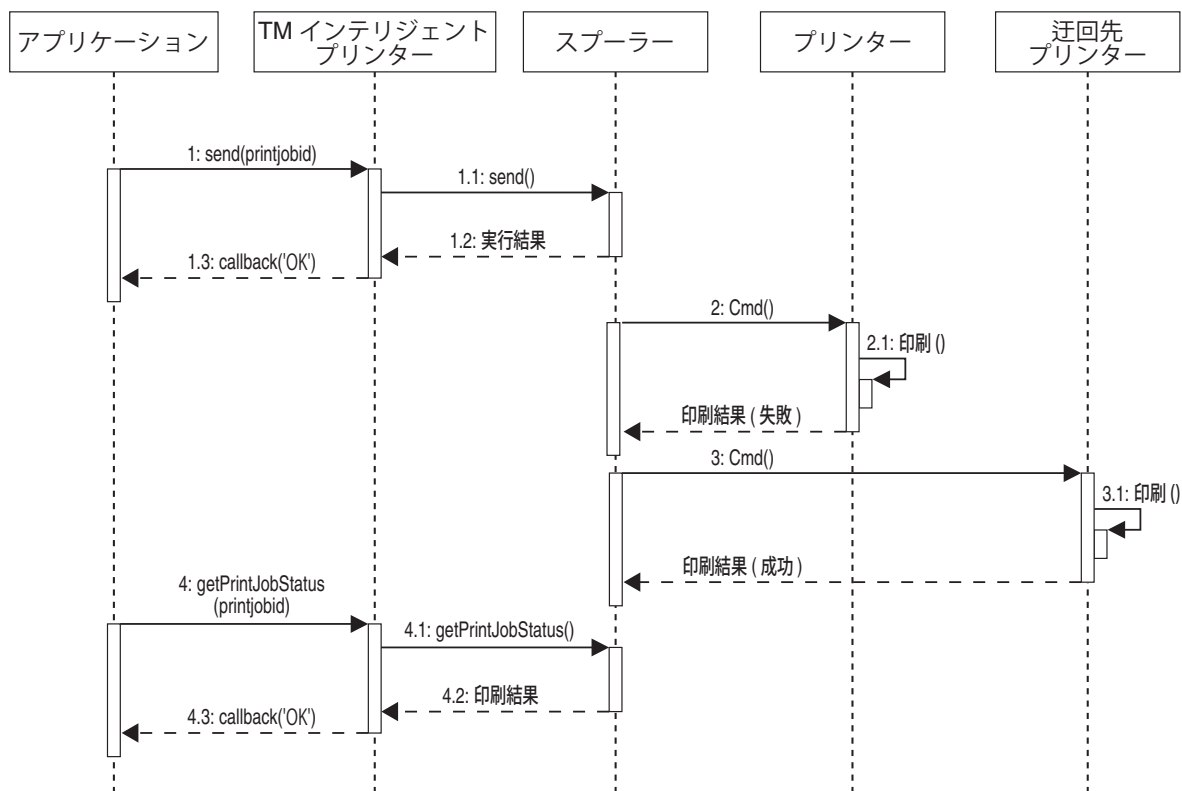
スプーラー機能を無効にした場合、アプリケーションから TM-DT ソフトウェア /TM-i ファームウェアにリクエストを送信すると、印刷を直ちに実行し、印刷完了後にレスポンスをアプリケーションに返します。



スプーラー機能を有効にした場合、アプリケーションから TM-DT ソフトウェア /TM-i ファームウェアにリクエストを送信すると、印刷データをキューに入れ、印刷完了を待たずにレスポンスをアプリケーションに返します。



この時、出力先のプリンターが印刷できない場合、TM-DT ソフトウェア /TM-i ファームウェアは、アプリケーションにエラーを返しません。迂回印刷を有効にすると、代替プリンターで印刷でき、アプリケーションは、印刷結果も後で取得できます。以下のシーケンス図を参照してください。



#### □ プログラミング例

スプーラー機能を有効にすると、印刷ジョブ ID を持つレスポンスを返します。

アプリケーションが印刷ジョブ ID を指定しない場合は、TM-DT ソフトウェア /TM-i ファームウェアが印刷ジョブ ID を発行します。

印刷ジョブの状態は、レスポンスに含まれる印刷ジョブ ID で取得します。

```

var printjobid = '';

function sendJob() {
    printer.addText('Hello, World!\n');
    printer.addCut();
    printer.onreceive = function (res) {
        if (res.success) {
            printjobid = res.printjobid;
        }
    }
    printer.onerror = function (err) { alert(err.status); };
    printer.send();
}

function getJobStatus() {
    if (printjobid.length > 0) {
        printer.onreceive = function (res) { alert(res.success); };
        printer.onerror = function (err) { alert(err.status); };
        printer.getPrintJobStatus(printjobid);
    }
}
  
```

## Web ページを頻繁に更新 / 遷移するアプリケーションを開発するには

Epson ePOS SDK for JavaScript で開発したアプリケーションは、Web ページを更新 / 遷移する度に接続デバイスとの通信を切断し、再接続する処理が発生します。このため TM インテリジェントプリンター制御時は、印字開始やディスプレイ表示が遅くなることがあります。

ePOS-Print オプションを有効にすることで、接続処理にかかる時間を短縮できます。

### ePOS-Print オプションを有効にするには

ePOS-Print オプションは、[connect メソッド](#)の option パラメーターで設定します。

例) ePOS-Print オプションを有効にする。

```
connect("192.168.192.168", "8008", callback, {"eposprint" : true});
```

### 制限事項

ePOS-Print オプションを有効にすると、TM インテリジェントプリンターの以下の機能が使用できなくなります。

- ❑ TM インテリジェントプリンターで利用できる周辺機器
- ❑ TM インテリジェントプリンター特有の機能



- ePOS-Print オプションは、TM プリンターの制御に影響しません。
- 使用できるオブジェクトは、以下の通りです。
  - \* [ePOSDevice オブジェクト](#) (TM プリンターで利用できる API のみ)
  - \* [Display オブジェクト](#)
  - \* [Printer オブジェクト](#)

# 付録

## オープンソースライセンス

This product contains the following open-source libraries:

### MIT Licensed

Socket.IO.js build:0.8.7, development.

Copyright(c) 2011 LearnBoost dev@learnboost.com

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.