

CITIZEN

CITIZEN XML Print サービス JavaScript Label Print SDK プログラムマニュアル

Ver. 1.03 用

シチズン・システムズ株式会社

更新履歴

| 年月日 | バージョン | 履歴 |
|------------|-------|--|
| 2017/12/26 | 1. 00 | 新規 |
| 2018/09/18 | 1. 01 | <ul style="list-style-type: none"> ・drawTextPtrFont に中国モデル、韓国モデル用のロケールを追加 ・ロケールの定義に中国モデル、韓国モデルを追加 |
| 2019/06/26 | | JavaScript Label Print SDK の応答受信コールバック関数のエラーコードに、EConnectNotFound、EConnectOffline を追加 |
| 2019/07/18 | | <ul style="list-style-type: none"> ・SendData タグの解説、使用例の記載を修正 ・drawBitmap メソッドの measurementUnit パラメータの設定内容を修正 |
| 2020/06/02 | | ・新モデル CL-E300EX/303EX、CL-E321EX/331EX を追加 |
| 2020/07/02 | 1. 02 | <ul style="list-style-type: none"> ・対応モデル CL-S700/703、CL-S700II/703II、CL-S620/630、CL-S620II/630II、CL-S520/530、CL-S520II/530II を追加 ・https 通信の URL 記述例を追加 ・PrinterCheck メソッドの説明および使用例を修正 ・LabelPrinter クラスに PrintHeadLowTemp、PrintHeadFailure、PrintHeadOverheat、MechanismOpen、AutoCutterError、FanMotorError、MiscError プロパティを追加 |
| 2020/12/08 | 1. 03 | ・HTTPS に関する説明を修正 |
| 2021/03/03 | | <ul style="list-style-type: none"> ・インターフェース型番を更新（8 頁） ・HTTPS に関する説明を修正（8、9、20 頁） ・サービスバージョンの確認方法を追加（17 頁） |
| 2023/12/22 | | ・対応モデル CL-S700III/703III を追加（8 頁） |

ご注意

1. 本書の内容の一部、または全部を無断で転載することは、固くお断りいたします。
2. 本書の内容については、事前の予告なしに変更することがあります。
3. 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、万一誤り・お気付きの点がございましたら、ご連絡くださいますようお願いいたします。
4. 運用した結果の影響につきましては、3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。
5. 上記に同意いただけない場合は、本SDKをご使用いただけません。

商標

Microsoft、Windows は、米国 Microsoft Corporation (あるいは米国マイクロソフト・コーポレーション) の米国およびその他の国における登録商標です。(Windows の正式名称は Microsoft Windows Operating System です)
その他、記載されている会社名、製品名は、各社の商標または登録商標です。

CITIZEN は、シチズン時計株式会社の登録商標です。

目次

| | |
|--|----|
| 1. はじめに | 7 |
| 1.1. システム概要 | 7 |
| 1.2. ドキュメント対象範囲 | 7 |
| 1.3. システム構成例 | 7 |
| 1.4. 対応機種 | 8 |
| 2. XML Print サービスメッセージ | 9 |
| 2.1. 要求メッセージ | 9 |
| 2.1.1. 送信方法・メッセージ構成 | 9 |
| 2.1.2. POSPrinterRequest タグ | 9 |
| 2.2. 応答メッセージ | 10 |
| 2.2.1. メッセージ構成 | 10 |
| 2.2.2. 要求結果の取得 | 10 |
| 2.2.3. エラーコード | 11 |
| 2.3. デバイスのステータス取得 | 12 |
| 3. デバイス制御タグ | 14 |
| 3.1. デバイス制御タグ一覧 | 14 |
| 3.2. 印刷制御タグ詳細 | 15 |
| 3.2.1. メッセージ ID (MessageID タグ) | 15 |
| 3.2.2. コマンド送信 (SendData タグ) | 16 |
| 4. XML Print サービス設定 | 17 |
| 4.1. Web マネージャ | 17 |
| 4.1.1. Service 設定画面 / XML Print | 17 |
| 4.1.2. Service Status 画面 / XML Print | 17 |
| 5. JavaScript Label Print SDK | 18 |
| 5.1. 動作環境 | 18 |
| 5.2. プログラミングガイド | 18 |
| 5.2.1. SDK ファイルの配置 | 18 |
| 5.2.2. プログラム構成 | 18 |
| 5.2.3. ラベルデザイン処理 | 19 |
| 5.2.4. デバイス制御オブジェクトの作成 | 19 |
| 5.2.5. 応答受信コールバック関数の設定 | 19 |
| 5.2.6. 送信エラーコールバック関数の設定 | 20 |
| 5.2.7. 送信実行 | 20 |
| 5.3. 機能一覧 | 21 |
| 5.3.1. デバイス制御機能 (LabelPrint クラス) | 21 |
| 5.3.2. ラベルデザイン機能 (LabelDesign クラス) | 23 |
| 5.4. 戻り値 | 24 |
| 5.5. LabelPrint クラス詳細 | 25 |
| 5.5.1. コンストラクタ | 25 |
| 5.5.2. messageID メソッド | 26 |
| 5.5.3. printerCheck メソッド | 27 |
| 5.5.4. print メソッド | 29 |
| 5.5.5. storeNVBitmap メソッド | 30 |
| 5.5.6. clearOutput メソッド | 31 |
| 5.5.7. sendData メソッド | 32 |
| 5.5.8. HorizontalMagnification プロパティ | 33 |
| 5.5.9. VerticalMagnification プロパティ | 34 |

| | |
|---|----|
| 5.5.10. FormatAttribute プロパティ..... | 35 |
| 5.5.11. ContinuousMediaLength プロパティ..... | 36 |
| 5.5.12. MeasurementUnit プロパティ..... | 37 |
| 5.5.13. PrintSpeed プロパティ..... | 38 |
| 5.5.14. FeedSpeed プロパティ..... | 39 |
| 5.5.15. SlewSpeed プロパティ..... | 40 |
| 5.5.16. BackupSpeed プロパティ..... | 41 |
| 5.5.17. PrintDarkness プロパティ..... | 42 |
| 5.5.18. DoubleHeat プロパティ..... | 43 |
| 5.5.19. VerticalOffset プロパティ..... | 44 |
| 5.5.20. HorizontalOffset プロパティ..... | 45 |
| 5.5.21. MediaHandling プロパティ..... | 46 |
| 5.5.22. StartOffset プロパティ..... | 47 |
| 5.5.23. StopOffset プロパティ..... | 48 |
| 5.5.24. LabelSensor プロパティ..... | 49 |
| 5.5.25. PrintMethod プロパティ..... | 50 |
| 5.5.26. SensorLocation プロパティ..... | 51 |
| 5.5.27. CommandInterpreterInAction プロパティ..... | 52 |
| 5.5.28. PaperError プロパティ..... | 53 |
| 5.5.29. RibbonEnd プロパティ..... | 54 |
| 5.5.30. BatchProcessing プロパティ..... | 55 |
| 5.5.31. Printing プロパティ..... | 56 |
| 5.5.32. Pause プロパティ..... | 57 |
| 5.5.33. WaitingForPeeling プロパティ..... | 58 |
| 5.5.34. PrintHeadLowTemp プロパティ..... | 59 |
| 5.5.35. PrintHeadFailure プロパティ..... | 60 |
| 5.5.36. PrintHeadOverheat プロパティ..... | 61 |
| 5.5.37. MechanismOpen プロパティ..... | 62 |
| 5.5.38. AutoCutterError プロパティ..... | 63 |
| 5.5.39. FanMotorError プロパティ..... | 64 |
| 5.5.40. MiscError プロパティ..... | 65 |
| 5.6. LabelDesign クラス詳細..... | 66 |
| 5.6.1. コンストラクタ..... | 66 |
| 5.6.2. drawTextPtrFont メソッド..... | 67 |
| 5.6.3. drawTextDLFont メソッド..... | 69 |
| 5.6.4. drawNVBitmap メソッド..... | 71 |
| 5.6.5. drawBitmap メソッド..... | 72 |
| 5.6.6. drawBarCode メソッド..... | 74 |
| 5.6.7. drawMaxiCode メソッド..... | 78 |
| 5.6.8. drawPDF417 メソッド..... | 79 |
| 5.6.9. drawDataMatrix メソッド..... | 80 |
| 5.6.10. drawQRCode メソッド..... | 81 |
| 5.6.11. drawAztec メソッド..... | 82 |
| 5.6.12. drawGS1DataBar メソッド..... | 83 |
| 5.6.13. drawLine メソッド..... | 85 |
| 5.6.14. drawRect メソッド..... | 86 |
| 5.6.15. fillRect メソッド..... | 87 |
| 5.6.16. drawCircle メソッド..... | 88 |
| 5.6.17. fillCircle メソッド..... | 89 |
| 5.6.18. drawPolygon メソッド..... | 90 |
| 5.6.19. fillPolygon メソッド..... | 91 |

| | |
|---|----|
| 5.6.20. <i>embedRawDesignCommand</i> メソッド | 92 |
| 5.7. 補足 | 93 |
| 5.7.1. 印字位置指定 | 93 |
| 5.7.2. パラメータ | 94 |

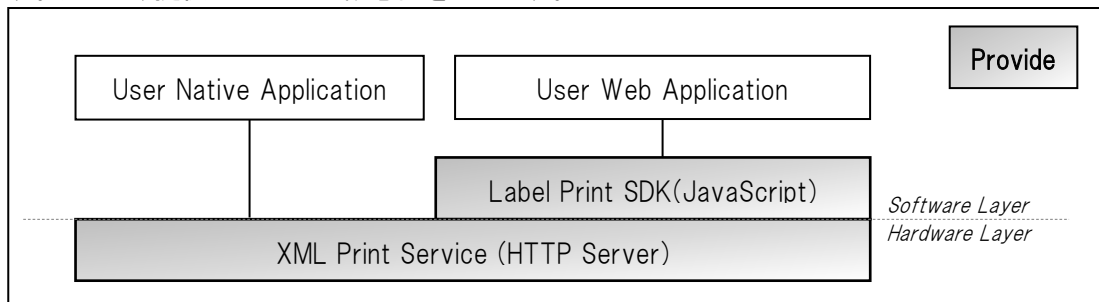
1. はじめに

本ドキュメントは、CITIZEN XML Print サービス、JavaScript Label Print SDK におけるプログラマー様向けのマニュアルです。

1.1. システム概要

CITIZEN XML Print サービスは、OS 非依存のマルチプラットフォーム環境においてデバイスドライバーレスでプリンターを制御する機能を提供します。

HTTP(XML)ベースの制御方式のため、Web サービス環境から、簡単にプリンターを制御できます。また、XML Print サービスをクライアントサイドの JavaScript で制御するためのライブラリとして、Label Print SDK を用意しております。以下に、提供サービスの概念図を示します。



1.2. ドキュメント対象範囲

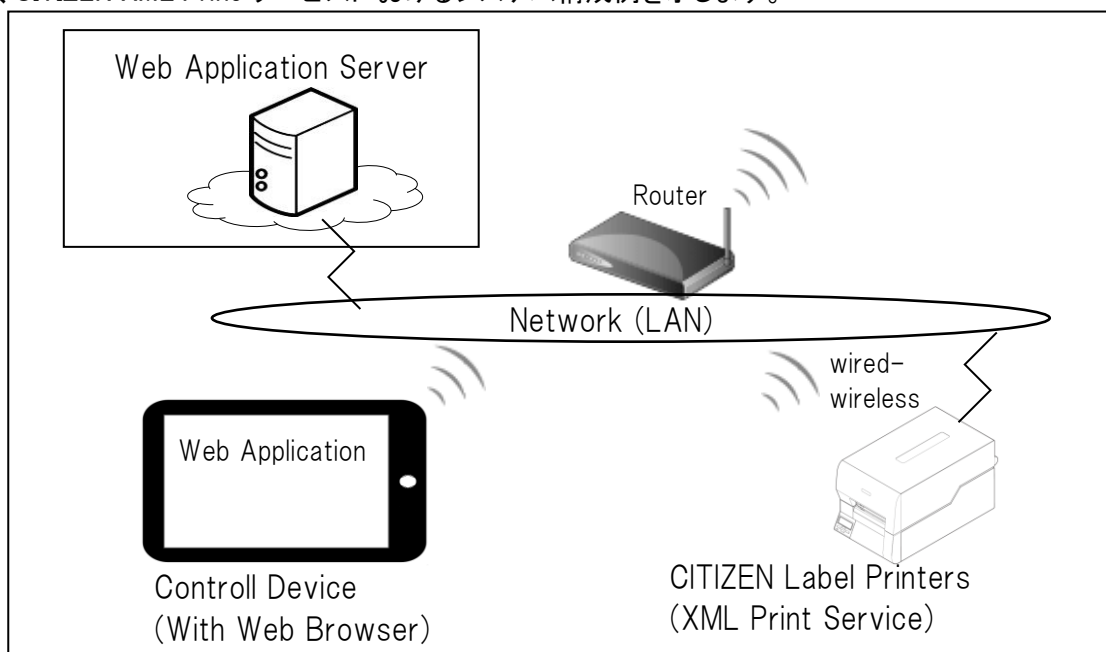
CITIZEN XML Print サービス対応のプリンターを利用するアプリケーション開発者が参照することを目的としています。

HTTP(XML)ベースで印刷するための制御仕様は、本書の「[2. XML Print サービスメッセージ](#)」、「[3. デバイス制御タグ](#)」、「[4. XML Print サービス設定](#)」を参照してください。

Web サービス環境から印刷するための JavaScript SDK 仕様は、「[5. JavaScript Label Print SDK](#)」を参照してください。

1.3. システム構成例

以下に、CITIZEN XML Print サービスにおけるシステム構成例を示します。



1.4. 対応機種

本サービスの対象モデルおよびそのモデルに対応するインターフェースは以下の通りです。
各モデルの機能詳細についてはプリンターの取り扱い説明書をご参照ください。

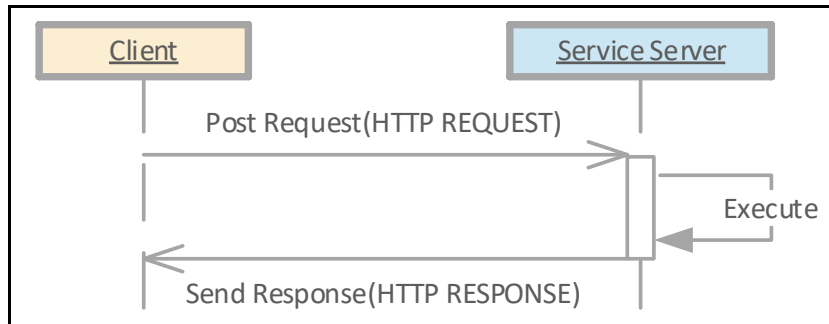
| 対象モデル | インターフェース |
|-------------------|--|
| CL-S400DT | 有線 LAN(型番: IF1-EFX1※, IF1-EFX2) 有線/無線 LAN(型番: IF1-WFx4, IF1-WFx6) |
| CL-S700III/703III | |
| CL-E720/730 | |
| CL-E300EX/303EX | 有線 LAN(型番: IF2-EFX1※, IF2-EFX2) 有線/無線 LAN(型番: IF2-WFx5, IF2-WFx6) |
| CL-E321EX/331EX | |
| CL-S700/703 | 有線 LAN(型番: IF5-EFX1) |
| CL-S700II/703II | |
| CL-S620/630 | |
| CL-S620II/630II | |
| CL-S520/530 | |
| CL-S520II/530II | |

※ HTTPS 環境で利用できません(HTTP 環境のみ利用可能)

本サービスのバージョン番号は、「[4.1.2. Service Status 画面 / XML Print](#)」を確認してください。バージョンにより、利用方法や機能が異なる場合があります。

2. XML Print サービスメッセージ

下図のように、サービス利用者は、要求メッセージを HTTP のリクエストとして発行し、サービスからの応答メッセージは、HTTP のレスポンスとして受信します。要求メッセージおよび応答メッセージは、XML スキーマファイル「CitizenXML-Print.xsd」で定義されています。



2.1. 要求メッセージ

クライアントから発行される要求メッセージに応じて、デバイス(プリンター)の制御を行います。

2.1.1. 送信方法・メッセージ構成

要求メッセージは、次の方法で、SOAP メッセージを送信してください。

- ・ HTTP URL: `http(s):// [本サービスの IP アドレス] /xmlprint`
 ※https は、IF5-EFX1 および、それ以外のサービスバージョン 2.1 以降で対応
 ※IF5-EFX1 以外のサービスバージョン 2.0 以前では、`http:// [本サービスの IP アドレス] :8080`
- ・ HTTP メソッド: `POST`
- ・ HTTP ヘッダー: `Content-Type: text/xml; charset=UTF-8`

要求メッセージの構成は、以下の通りです。

```

<?xml version="1.0" encoding="utf-8"?>
<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/">
  <s:Body>

    <POSPrinterRequest xmlns=http://www.citizen.co.jp/POSPrinter/
      MajorVersion="1">
      <!-- 要求 ID -->
      <MessageID>3ea2110f-8d31-24ef-e2a5-ca98d2af5f8d</MessageID>
      <!-- プリンターコマンドを送信 -->
      <SendData>
        <Data>ASMNCg==</Data>
      </SendData>
    </POSPrinterRequest>

  </s:Body>
</s:Envelope>
  
```

} <POSPrinterRequest>
タグ

2.1.2. POSPrinterRequest タグ

デバイスの制御は、要求メッセージ中の<POSPrinterRequest>タグ内にデバイス制御タグを記述してください。デバイス制御タグの詳細は、本書の[「3. デバイス制御タグ」](#)を参照してください。

2.2. 応答メッセージ

本サービスからの応答メッセージにより、要求の結果を確認できます。

2.2.1. メッセージ構成

応答メッセージの構成は、以下の通りです。

```
<?xml version="1.0" encoding="UTF-8"?>
<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/">
  <s:Body>

    <POSPrinterResponse xmlns=http://www.citizen.co.jp/POSPrinter/
      MajorVersion="1">
      <MessageID>0011e507451e-648e8cd7</MessageID>
      <Response ResponseCode="OK">
        <RequestID>26fc854c-e6f2-9a86-9060-6d76069cf4e2</RequestID>
      </Response>
    </POSPrinterResponse>

  </s:Body>
</s:Envelope>
```

〈POSPrinterResponse〉
タグ

2.2.2. 要求結果の取得

〈POSPrinterResponse〉タグ内の情報でリクエストの結果を確認できます。

| 項目 | 説明 |
|------------------|-----------------------------|
| ResponseCode 属性 | 処理結果が格納されます |
| MessageID 要素 | 応答メッセージを識別のための ID が格納されます |
| RequestID 要素 | 送信時に指定された MessageID が格納されます |
| BusinessError 要素 | エラー発生時にエラー情報が格納されます |

エラーの発生の確認

Response 要素の ResponseCode 属性の値を確認することにより、エラーの発生の有無が確認できます。

| コード | 説明 |
|----------|----------|
| OK | 正常に終了した |
| Rejected | エラーが発生した |

以下は、正常に終了した場合の Response 要素の例です。

```
<Response ResponseCode="OK">
  <RequestID>26fc854c-e6f2-9a86-9060-6d76069cf4e2</RequestID>
</Response>
```

エラー情報の確認

エラー発生時には、Response 要素内の BusinessError 要素内の Code および Description 要素の内容により、結果の要因を確認できます。詳細は、本書の「[2.2.3 エラーコード](#)」を参照してください。

以下は、エラーが発生した場合の Response 要素の例です。

```
<Response ResponseCode="Rejected">
  <RequestID></RequestID>
  <BusinessError Severity="Error">
    <Code>ETimeout</Code>
    <Description>The device is not respond.</Description>
  </BusinessError>
</Response>
```

2.2.3. エラーコード

応答メッセージでは、エラーコードやエラーの説明等の詳細情報を、Response 要素内の BusinessException 要素に設定されます。本サービスで使用するエラーコードを以下に示します。

| コード | 説明 |
|------------------|-----------------|
| RequestInvalid | 要求情報が不正 |
| EConnectNotFound | 対応機種でない |
| EConnectOffline | プリンターの準備ができていない |
| EIllegal | 未サポートまたは無効パラメータ |
| ETimeout | 処理タイムアウト |

2.3. デバイスのステータス取得

デバイスのステータスを取得するには、デバイス情報取得要求を指定する GetDeviceInfo タグを使用します。

パラメータ

| 属性 | 意味 | 設定可能範囲 |
|-------------------------|------------------|----------------------------------|
| BarcodePrinterStatus | BC プリンター情報フラグ | "true"を指定した場合に BC プリンター状態を付加 |
| BarcodePrinterErrorInfo | BC プリンターエラー情報フラグ | "true"を指定した場合に BC プリンターのエラー情報を付加 |

要求メッセージの例は、以下の通りです。

```
<?xml version="1.0" encoding="utf-8"?>
<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/">
  <s:Body>
    <POSPrinterRequest xmlns=http://www.citizen.co.jp/POSPrinter/
      MajorVersion="1">
      <!-- デバイス情報取得要求 -->
      <GetDeviceInfo BarcodePrinterStatus='true'
        BarcodePrinterErrorInfo='true' />
    </POSPrinterRequest>
  </s:Body>
</s:Envelope>
```

応答メッセージの例は、以下の通りです。

```
<?xml version="1.0" encoding="UTF-8"?>
<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/">
  <s:Body>
    <POSPrinterResponse xmlns=http://www.citizen.co.jp/POSPrinter/
      MajorVersion="1">
      <MessageID>0011e507451e-648e8cd8</MessageID>
      <Response ResponseCode="OK">
        <RequestID></RequestID>
      </Response>
      <GetDeviceInfo>
        <Device BarcodePrinterStatus="NNNNNNYNN"
          BarcodePrinterErrorInfo="60404040" />
      </GetDeviceInfo>
    </POSPrinterResponse>
  </s:Body>
</s:Envelope>
```

} <GetDeviceInfo>タグ

デバイス情報の内容は、応答メッセージの Device 要素の BarcodePrinterStatus 属性、BarcodePrinterErrorInfo 属性にステータスが格納されます。

※上記の応答メッセージの例では、プリンターがポーズ中であることを示します。

BarcodePrinterStatus 属性

プリンターのステータスが Yes/No で格納されます。

| Index | 説明 |
|-------|---------------|
| [0] | コマンドインタプリタ動作中 |
| [1] | ペーパーエラー |
| [2] | リボンエンド |
| [3] | バッチ処理(印字中) |
| [4] | 印字動作中 |
| [5] | ポーズ中 |
| [6] | 剥離待ち中 |
| [7] | 予備 ※常に N |

BarcodePrinterErrorInfo 属性

プリンター内部の状態が、4 バイトのデータにて格納されます。(有線 LAN I/F の IF5-EFX1 で対応)

| バイト | ビット | 説明 | エラー状態 |
|-----|-----|-----------------|------------|
| 1 | 0 | 電池切れ(未対応) | Yes:1 No:0 |
| | 1 | ヘッド低温 | |
| | 2 | 基板低温(未対応) | |
| | 3 | ヘッド切れ | |
| | 4 | 予備 | 常に 0 |
| | 5 | ポーズ中 | Yes:1 No:0 |
| | 6 | 固定 | 常に 1 |
| | 7 | | 常に 0 |
| 2 | 0 | 予備 | 常に 0 |
| | 1 | ヘッドオーバーヒート | Yes:1 No:0 |
| | 2 | 予備 | 常に 0 |
| | 3 | | |
| | 4 | メカオープン中 | Yes:1 No:0 |
| | 5 | ペーパーエンド | |
| | 6 | 固定 | 常に 1 |
| | 7 | | 常に 0 |
| 3 | 0 | ペーパーアウト | Yes:1 No:0 |
| | 1 | リボンエンド | |
| | 2 | 基板オーバーヒート(未対応) | |
| | 3 | 予備 | 常に 0 |
| | 4 | オプションボード以上(未対応) | Yes:1 No:0 |
| | 5 | オートカッター異常 | |
| | 6 | 固定 | 常に 1 |
| | 7 | | 常に 0 |
| 4 | 0 | ファンモータストップ(未対応) | Yes:1 No:0 |
| | 1 | 予備 | 常に 0 |
| | 2 | | |
| | 3 | | |
| | 4 | | |
| | 5 | その他のエラー発生中 | Yes:1 No:0 |
| | 6 | 固定 | 常に 1 |
| | 7 | | 常に 0 |

3. デバイス制御タグ

3.1. デバイス制御タグ一覧

本サービスで利用できるデバイス制御タグを以下に示します。

| 機能 | タグ | 説明 |
|----------|-----------------------------------|--|
| メッセージ ID | <MessageID> | 発信者がメッセージを識別するために指定します。 |
| コマンド送信 | <SendData> | プリンターコマンドを指定します。 |
| デバイス情報取得 | <GetDeviceInfo> | 「2.3. デバイスのステータス取得」 を参照。 |

3.2. 印刷制御タグ詳細

3.2.1. メッセージ ID (MessageID タグ)

値

要求メッセージ ID を指定します。

説明

このタグは、発信者がメッセージを識別するために使用します。

指定された要求メッセージ ID が応答メッセージの<RequestID>タグに付加されます。応答メッセージの詳細は本書の「[2.2. 応答メッセージ](#)」を参照してください。

使用例

```
<MessageID>12345678</MessageID>
```

3.2.2. コマンド送信 (SendData タグ)

パラメータ

パラメータの意味と設定可能な値は以下の通りです。

| 要素 | 意味 | 設定可能範囲 |
|------|-------|-----------------|
| Data | 送信データ | Base64 エンコードデータ |

説明

このタグは、プリンター制御コマンドをプリンターに送信するために使用します。
プリンター制御コマンドは、バイナリーデータを Base64 エンコードしている必要があります。

使用例

以下にプリンター制御コマンドを送信する際の、SendData タグの記述例を記載します。

プリンター制御コマンド

| | |
|----------------------------------|------------------------|
| [02] [1B] G0 | コマンドセットを「DMI / DMW」に指定 |
| [02] L | ラベルフォーマットモード開始 |
| D11 | ピクセルサイズを設定 |
| 1X1100000100000b0100010000050005 | 矩形を描画 |
| Q0001 | 印刷枚数「1」を指定 |
| E | ラベルフォーマットモードの終了、印刷 |

```
<?xml version="1.0" encoding="utf-8"?>
<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/">
  <s:Body>
    <POSPrinterRequest xmlns="http://www.citizen.co.jp/POSPrinter/"
      MajorVersion="1">
      <MessageID>00000001</MessageID>
      <SendData>
        <Data>AhtHMA0KAkwNCkQxMQ0KMVgxMTAwMDAwMTAwMDAwYjAxMDAwMTAwMDAw
          NTAwMDUNC1EwMDAxDQpFDQo=</Data>
      </SendData>
    </POSPrinterRequest>
  </s:Body>
</s:Envelope>
```


4. XML Print サービス設定

CITIZEN XML Print サービスの設定方法を説明します。

4.1. Web マネージャ

Web ブラウザーから各プリンターに接続することでプリンターに関する設定を変更することができます。基本的な操作につきましては、プリンターのインターフェイスボード取り扱い説明書をご参照ください。

本ドキュメントでは、XML Print サービスの設定項目を説明します。

4.1.1. Service 設定画面 / XML Print

Service 設定画面では、プリンターが提供するサービスの設定を行います。

LAN board CITIZEN SYSTEMS

HOME | STATUS | CONFIG Logout

General Service SSL/TLS Request Print User Account Maintenance

Media Converter

VCOM Convert ☐ Enable ☒ Disable ☐ Show configuration

HID Scanner Convert ☐ Enable ☒ Disable ☐ Show configuration

XML Print

Port Number

Timeout for connect 5-60[Seconds]

Timeout for print 10-600[Seconds]

XML Device Control

Port Number

| 項目 | 初期値 | 設定範囲 | 説明 |
|---------------------|------|------------|-----------------|
| Port Number | 8080 | 1025～65535 | 接続ポート番号 |
| Timeout for connect | 10 | 5～60 | 印刷開始待ちのタイムアウト時間 |
| Timeout for print | 60 | 10～600 | 未使用の設定項目です。 |

※この画面は、XML Print サービスに未対応のプリンターでは表示されませんので、対応プリンターをご利用ください。

※各設定値は、電源を切断しても値を保持します。工場初期設定(Factory Default)の処理が行われた時には、各設定値を初期値に設定します。

4.1.2. Service Status 画面 / XML Print

Service Status 画面では、サービスのバージョンや設定の情報を確認できます。

LAN board CITIZEN SYSTEMS

HOME | STATUS | CONFIG Logout

System Status Network Status Printer Status **Service Status** Request Print

Port Number: 9210

XML Print

Service Version: 2.0

Port Number: 8080

XML Device Control

Service Version: 1.0

5. JavaScript Label Print SDK

Label Print SDK は、CITIZEN XML Print サービスをクライアントサイドの JavaScript でプリンターを制御するためのライブラリです。JavaScript を使用して、Web アプリケーションから簡単に印刷することが可能です。

5.1. 動作環境

本 SDK が対応する Web ブラウザーは、HTML5 に対応している必要があります。

5.2. プログラミングガイド

5.2.1. SDK ファイルの配置

Label Print SDK は JavaScript で提供しています。SDK を利用するためには、Web サーバーに「cxmip-label-api.js」を配置してください。提供 SDK のソースコードに対して変更を行うと正しく動作しなくなる可能性があるため、変更を行わないでください。

5.2.2. プログラム構成

デバイスを制御するには、Web サーバーに配置した Web ページの HTML<script>タグに記述します。プログラム構成は、以下の通りです。

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="utf-8" />
  <title>Label Print SDK Sample</title>
  <script type="text/javascript" src="cxmip-label-api.js"></script>
  <script type="text/javascript">
    //LabelDesign オブジェクト生成
    var design = new citizen.LabelDesign();
    //ラベルデザイン登録
    design.drawCircle(50, 50, 15);
    design.drawRect(20, 30, 180, 280, 10);
    //デバイス制御オブジェクトの作成
    var print = new citizen.LabelPrint();
    //応答受信コールバック関数の設定
    print.OnReceive = function (res, xml) {
      alert(res.ResponseCode);
    };
    //送信エラーコールバック関数の設定
    print.OnError = function (res) {
      alert(res.status);
    };
    //要求メッセージ ID の設定
    print.MessageID('12345678');
    //送信実行
    print.print('http://192.168.129.178/xmlprint', design, 1);
  </script>
</head>

<body>
  .
  .
</body>
</html>
```

SDK の組み込み

プログラム本体

5.2.3. ラベルデザイン処理

ラベルのデザイン処理は、LabelDesign オブジェクトで行います。「citizen.LabelDesign」インスタンスを生成し、デザインメソッドを呼び出す事により、ラベルデザインを登録する事ができます。

デザインメソッドの詳細は、本書の「[5.6. LabelDesign クラス詳細](#)」を参照してください。

ラベルデザイン処理の登録例は、以下の通りです。

```
//LabelDesign オブジェクト生成
design = new citizen.LabelDesign();
//ラベルデザイン登録
design.drawCircle(50, 50, 15);
design.drawRect(20, 30, 180, 280, 10);
```

5.2.4. デバイス制御オブジェクトの作成

デバイスの制御は LabelPrint オブジェクトで行います。最初に「citizen.LabelPrint」インスタンスを生成してください。

5.2.5. 応答受信コールバック関数の設定

デバイス制御オブジェクトの OnReceive プロパティにコールバック関数を設定する事により、関数の引き数の情報で制御結果を確認できます。

| 項目 | 説明 |
|--------------|-----------------------------|
| ResponseCode | 処理結果が格納されます |
| MessageID | 応答メッセージを識別のための ID が格納されます |
| RequestID | 送信時に指定された MessageID が格納されます |
| ErrorCode | エラー発生時にエラーコードが格納されます |
| Description | エラー発生時の説明が格納されます |

応答受信コールバック関数の設定例は、以下の通りです。

```
//応答受信コールバック関数の設定
print.OnReceive = function (res, xml) {
  var msg;
  if(res.ResponseCode == 'OK'){
    msg = 'Print success!\n\n';
  }
  else{
    msg = 'Print failure!\n\n';
    msg += ' Code: ' + res.ErrorCode + '\n';
    msg += ' Description: ' + res.Description + '\n\n';
  }
  msg += ' RequestID: ' + res.RequestID + '\n';
  alert(msg);
};
```

エラーの発生の確認

ResponseCode の値を確認することにより、エラーの発生の有無が確認できます。

| コード | 説明 |
|----------|----------|
| OK | 正常に終了した |
| Rejected | エラーが発生した |

エラー情報の確認

エラー発生時に格納される、ErrorCode および Description の内容により、結果の要因を確認できます。本サービスで 사용되는エラーコードを以下に示します。

| コード | 説明 |
|------------------|-----------------|
| EConnectNotFound | 対応機種でない |
| EConnectOffline | プリンターの準備ができていない |
| ETimeout | 処理タイムアウト |

5.2.6. 送信エラーコールバック関数の設定

デバイス制御オブジェクトの OnError プロパティにコールバック関数を設定する事により、関数の引き数の情報でエラー内容を確認できます。

エラー発生時のステータスが status に、レスポンスの内容が responseText に格納されます。

送信エラーコールバック関数の設定例は、以下の通りです。

```
//送信エラーコールバック関数の設定
print.OnError = function (res) {
    var msg = 'Send failure!\n\n';
    msg += ' status: ' + res.status + '\n';
    msg += ' responseText: ' + res.responseText + '\n';
    alert(msg);
};
```

5.2.7. 送信実行

デバイス制御オブジェクトで XML Print サービスの URL を指定して LabelPrint オブジェクトのメソッドを呼び出す事により、送信処理が開始されます。処理が終了すると、設定された応答受信コールバック関数が呼び出され制御結果を取得できます。制御結果取得の詳細は、本書の[「5.2.5. 応答受信コールバック関数の設定」](#)を参照してください。

指定する URL の書式は、以下の通りです。

http(s):// [本サービスの IP アドレス] /xmlprint

※https は、IF5-EFX1 および、それ以外のサービスバージョン 2.1 以降で対応

※IF5-EFX1 以外のサービスバージョン 2.0 以前では、http:// [本サービスの IP アドレス] :8080

送信実行の指定例は、以下の通りです。

```
//送信実行 (https の場合)
print.print("https://192.168.129.178/xmlprint", design, 1);

//送信実行 (http の場合)
print.print("http://192.168.129.178/xmlprint", design, 1);

//送信実行 (サービスバージョン 2.0 以前の http の場合)
print.print("http://192.168.129.178:8080/", design, 1);
```

5.3. 機能一覧

5.3.1. デバイス制御機能 (LabelPrint クラス)

本サービスで利用できるデバイス制御を以下に示します。

LabelPrint クラス メソッド

| No | 機能 | 詳細 |
|----|---------------------------------------|--------------------------|
| 1 | クラス生成 (コンストラクタ) | コンストラクタです。 |
| 2 | メッセージ ID 設定 (messageID メソッド) | メッセージ ID を設定します。 |
| 3 | プリンター状態確認処理 (printerCheck メソッド) | プリンターのステータスを取得します。 |
| 4 | 印刷処理 (print メソッド) | デザインしたラベルを印刷します。 |
| 5 | NV ビットマップ登録処理 (storeNVBitmap メソッド) | ビットマップ画像をフラッシュメモリへ登録します。 |
| 6 | プリントバッファクリア (clearOutput メソッド) | プリンターのプリントバッファをクリアします。 |
| 7 | バイトデータ送信処理 (sendData メソッド) | バイトデータをプリンターへ送信します。 |

LabelPrint クラス プロパティ

| No | 機能 | 属性 | 詳細 |
|----|--|-----|-----------------------------------|
| 1 | 水平ピクセルサイズの設定/取得 (HorizontalMagnification プロパティ) | R/W | 水平方向のピクセルサイズを示します。 |
| 2 | 垂直ピクセルサイズの設定/取得 (VerticalMagnification プロパティ) | R/W | 垂直方向のピクセルサイズを示します。 |
| 3 | 展開方法の指定設定/取得 (FormatAttribute プロパティ) | R/W | 描画が重なった部分の重ね書き/白抜きを示します。 |
| 4 | 連続用紙長の設定/取得 (ContinuousMediaLength プロパティ) | R/W | 連続紙を使用した場合のラベル長を示します。 |
| 5 | 単位の設定/取得 (MeasurementUnit プロパティ) | R/W | 距離指定パラメータの単位を示します。 |
| 6 | 速度(印刷部分)の設定/取得 (PrintSpeed プロパティ) | R/W | 印刷速度を示します。 |
| 7 | 速度(非印刷部分)の設定/取得 (FeedSpeed プロパティ) | R/W | 非印刷部分の紙送り速度を示します。 |
| 8 | 速度(紙送り部分)の設定/取得 (SlewSpeed プロパティ) | R/W | 紙送り部分の紙送り速度を示します。 |
| 9 | 速度(バックフィード)の設定/取得 (BackupSpeed プロパティ) | R/W | バックフィード時の速度を示します。 |
| 10 | 印字濃度の設定/取得 (PrintDarkness プロパティ) | R/W | 印字濃度を示します。 |
| 11 | ダブルヒートの設定/取得 (DoubleHeat プロパティ) | R/W | ダブルヒート(同位置に 2 回熱をかけ、より濃く印字)を示します。 |
| 12 | Y 座標オフセットの設定/取得 (VerticalOffset プロパティ) | R/W | 印字開始位置における、Y 座標のオフセットを示します。 |
| 13 | X 座標オフセットの設定/取得 (HorizontalOffset プロパティ) | R/W | 印字開始位置における、X 座標のオフセットを示します。 |

| | | | |
|----|---|-----|--------------------------|
| 14 | 印刷後動作の設定/取得 (MediaHandling プロパティ) | R/W | 印刷後のポーズ、カッター、剥離の動作を示します。 |
| 15 | 開始オフセットの設定/取得 (StartOffset プロパティ) | R/W | 印字開始位置の設定を示します。 |
| 16 | 終了オフセットの設定/取得 (StopOffset プロパティ) | R/W | 印刷後のフィード量を示します。 |
| 17 | センサー選択(透過/反射/なし)の設定/取得 (LabelSensor プロパティ) | R/W | 用紙センサーを示します。 |
| 18 | 印字方法の設定/取得 (PrintMethod プロパティ) | R/W | 印字方法(感熱、熱転写)の選択を示します。 |
| 19 | センサー選択(前方/後方)の設定/取得 (SensorLocation プロパティ) | R/W | 前方センサー、後方センサーの選択を示します。 |
| 20 | ステータス(コマンドインタプリタ動作中)の取得 (CommandInterpreterInAction プロパティ) | R | コマンド処理中のステータスを示します。 |
| 21 | ステータス(ペーパーエラー)の取得 (PaperError プロパティ) | R | ペーパーエラーのステータスを示します。 |
| 22 | ステータス(リボンエンド)の取得 (RibbonEnd プロパティ) | R | リボンエンドのステータスを示します。 |
| 23 | ステータス(バッチ処理印字中)の取得 (BatchProcessing プロパティ) | R | バッチ処理印字中のステータスを示します。 |
| 24 | ステータス(印字中)の取得 (Printing プロパティ) | R | 印字中のステータスを示します。 |
| 25 | ステータス(ポーズ中)の取得 (Pause プロパティ) | R | ポーズ中のステータスを示します。 |
| 26 | ステータス(剥離待ち中)の取得 (WaitingForPeeling プロパティ) | R | 剥離待ち中のステータスを示します。 |
| 27 | ステータス(ヘッド低温)の取得 (PrintHeadLowTemp プロパティ) | R | ヘッド低温のステータスを示します。 |
| 28 | ステータス(ヘッド切れ)の取得 (PrintHeadFailure プロパティ) | R | ヘッド切れのステータスを示します。 |
| 29 | ステータス(ヘッドオーバーヒート)の取得 (PrintHeadOverheat プロパティ) | R | ヘッドオーバーヒートのステータスを示します。 |
| 30 | ステータス(メカオープン中)の取得 (MechanismOpen プロパティ) | R | メカオープン中のステータスを示します。 |
| 31 | ステータス(オートカッター異常)の取得 (AutoCutterError プロパティ) | R | オートカッター異常のステータスを示します。 |
| 32 | ステータス(ファンモーターストップ)の取得 (FanMotorError プロパティ) | R | ファンモーターストップのステータスを示します。 |
| 33 | ステータス(その他エラー発生中)の取得 (MiscError プロパティ) | R | その他エラー発生中のステータスを示します。 |

5.3.2. ラベルデザイン機能(LabelDesign クラス)

本サービスで利用できるラベルデザインを以下に示します。

LabelDesign クラス メソッド

| No | 機能 | 詳細 |
|----|---|-------------------------------------|
| 1 | クラス生成 (コンストラクタ) | コンストラクタです。 |
| 2 | テキスト描画(プリンタフォント) (drawTextPtrFont メソッド) | プリンターのフォントを使用して、テキストを描画します。 |
| 3 | テキスト描画(プリンタダウンロードフォント) (drawTextDLFont メソッド) | プリンターにダウンロードしたフォントを使用して、テキストを描画します。 |
| 4 | 画像描画(プリンタ格納画像) (drawNVBitmap メソッド) | プリンターに格納している画像を描画します。 |
| 5 | 画像描画(ローカル画像ファイル) (drawBitmap メソッド) | PC 上の画像ファイルを描画します。 |
| 6 | バーコード描画(1D) (drawBarCode メソッド) | バーコードを描画します。 |
| 7 | バーコード描画(UPS MaxiCode) (drawMaxiCode メソッド) | |
| 8 | バーコード描画(PDF417) (drawPDF417 メソッド) | |
| 9 | バーコード描画(DataMatrix) (drawDataMatrix メソッド) | |
| 10 | バーコード描画(QRCode) (drawQRCode メソッド) | |
| 11 | バーコード描画(Aztec) (drawAztec メソッド) | |
| 12 | バーコード描画(GS1DataBar(RSS)) (drawGS1DataBar メソッド) | |
| 13 | 線の描画 (drawLine メソッド) | 線を描画します。 |
| 14 | 矩形の描画 (drawRect メソッド) | 矩形を描画します。 |
| 15 | 矩形の描画(塗りつぶし) (fillRect メソッド) | 矩形(塗りつぶし)を描画します。 |
| 16 | 円の描画 (drawCircle メソッド) | 円を描画します。 |
| 17 | 円の描画(塗りつぶし) (fillCircle メソッド) | 円(塗りつぶし)を描画します。 |
| 18 | 多角形の描画 (drawPolygon メソッド) | 多角形を描画します。 |
| 19 | 多角形の描画(塗りつぶし) (fillPolygon メソッド) | 多角形(塗りつぶし)を描画します。 |
| 20 | 生デザインコマンド挿入 (embedRawDesignCommand メソッド) | デザインコマンドを挿入します。 |

5.4. 戻り値

以降に示すメソッドは、下記の値を返します。

| 戻り値 | 説明 |
|----------------------|-----------------------------|
| CLS_SUCCESS (0) | 正常終了。 |
| CLS_E_ILLEGAL (1101) | サポートされていない処理または無効なパラメータ値です。 |

5.5. LabelPrint クラス詳細

5.5.1. コンストラクタ

形式

`citizen.LabelPrint ()`

パラメータ

ありません。

説明

JavaScript ライブラリのコンストラクタです。インスタンスを生成します。

戻り値

ありません。

使用例

```
var print = new citizen.LabelPrint();
```

5.5.2. messageID メソッド

形式

messageID (messageID)

パラメータ

パラメータの意味と設定可能な値は以下の通りです。

| 値 | [IN/OUT] | 意味 | 設定可能範囲 |
|-----------|----------|------------|--------|
| messageID | [IN] | 要求メッセージ ID | |

説明

このメソッドは、発信者がメッセージを識別するために使用します。

指定された要求メッセージ ID が制御結果の RequestID パラメータに付加されます。制御結果の詳細は本書の[「5.2.5. 応答受信コールバック関数の設定」](#)を参照してください。

戻り値

ありません。

使用例

```
print.messageID("00000001");
```

5.5.3. printerCheck メソッド

形式

printerCheck (address)

パラメータ

パラメータの意味と設定可能な値は以下の通りです。

| 値 | [IN/OUT] | 意味 | 設定可能範囲 |
|---------|----------|----------|--------|
| address | [IN] | HTTP URL | |

説明

このメソッドは、プリンターのステータスを取得し、ステータス情報を [CommandInterpreterInAction](#)、[PaperError](#)、[RibbonEnd](#)、[BatchProcessing](#)、[Printing](#)、[Pause](#)、[WaitingForPeeling](#)、および [PrintHeadLowTemp](#)、[PrintHeadFailure](#)、[PrintHeadOverheat](#)、[MechanismOpen](#)、[AutoCutterError](#)、[FanMotorError](#)、[MiscError](#) プロパティに格納します (PrintHeadLowTemp、PrintHeadFailure、PrintHeadOverheat、MechanismOpen、AutoCutterError、FanMotorError、MiscError プロパティは、IF5-EFX1 および、それ以外のサービスバージョン 2.1 以降で対応)。

本メソッドの実行結果が失敗の場合は、通信異常やデバイスの異常が発生した可能性があります。この場合は、指定した address の内容に間違いが無いこと、または、プリンターがネットワーク接続されていることを確認してください。

CL-S5xx/6xx/70x 系のプリンターは、予め「Parallel Error Output」の設定を OFF に変更しておいてください。設定の変更は、ラベルプリンターユーティリティの「拡張」タブから行うことができます。

戻り値

成功時は CLS_SUCCESS(0) を返します。それ以外のエラーコードは「[5.4 戻り値](#)」を参照してください。

使用例

// 以下、受信コールバック関数内の処理

```
function myOnReceive(res, xml)
{
    if (res.ResponseCode == 'OK')
    {
        // CommandInterpreterInActionプロパティ
        if (print.getCommandInterpreterInAction() == 1) {
            // Command interpreter in action
        }
        // PaperErrorプロパティ
        if (print.getPaperError() == 1) {
            // Paper error
        }
        // RibbonEndプロパティ
        if (print.getRibbonEnd() == 1) {
            // Ribbon end
        }
        // BatchProcessingプロパティ
        if (print.getBatchProcessing() == 1) {
            // Batch processing
        }
        // Printingプロパティ
        if (print.getPrinting() == 1) {
            // Printing
        }
    }
}
```

```
// Pauseプロパティ
if (print.getPause() == 1) {
    // Pause
}
// WaitingForPeelingプロパティ
if (print.getWaitingForPeeling() == 1) {
    // Waitiong for peeling
}

// PrintHeadLowTempプロパティ
if (print.getPrintHeadLowTemp() == 1) {
    // Print head low temp
}
// PrintHeadFailureプロパティ
if (print.getPrintHeadFailure() == 1) {
    // Print head failure
}
// PrintHeadOverheatプロパティ
if (print.getPrintHeadOverheat() == 1) {
    // Print head overheat
}
// MechanismOpenプロパティ
if (print.getMechanismOpen() == 1) {
    // Mechanism open
}
// AutoCutterErrorプロパティ
if (print.getAutoCutterError() == 1) {
    // Auto cutter error
}
// FanMotorErrorプロパティ
if (print.getFanMotorError() == 1) {
    // Fan motor error
}
// MiscErrorプロパティ
if (print.getMiscError() == 1) {
    // Misc error
}
}
else
{
    // Fail
}
}
```

```
// 受信コールバック関数をセット
print.OnReceive = myOnReceive;
```

```
// ステータス取得コマンドを送信
print.messageID("00000001");
print.printerCheck("http://192.168.129.178/xmlprint");
```

```
// プリンターからステータス取得コマンドの応答を受信したとき、
// 受信コールバック関数myOnReceive()が呼び出される
```

5.5.4. print メソッド

形式

print (address, design, quantity)

パラメータ

パラメータの意味と設定可能な値は以下の通りです。

| 値 | [IN/OUT] | 意味 | 設定可能範囲 |
|----------|----------|-----------------------|--------|
| address | [IN] | HTTP URL | |
| design | [IN] | LabelDesign クラスインスタンス | |
| quantity | [IN] | 印刷枚数 | 1～9999 |

説明

このメソッドは、デザインしたラベルを印刷する際に使用します。プリンターへ印刷要求を送信します。印刷要求を行う際、LabelDesign クラスで指定されたラベルのフォーマットや、プロパティの設定をコマンドに反映します。

戻り値

成功時は CLS_SUCCESS(0) を返します。それ以外のエラーコードは[「5.4 戻り値」](#)を参照してください。

使用例

```
design.fillCircle(50, 50, 50, citizen.LabelConst.CLS_SHADED_PTN_11);  
print.messageID("00000001");  
print.print("http://192.168.129.178/xmlprint", design, 1);
```

5.5.5. storeNVBitmap メソッド

形式

storeNVBitmap (image, address, name, rotation, width, height)

パラメータ

パラメータの意味と設定可能な値は以下の通りです。

| 値 | [IN/OUT] | 意味 | 設定可能範囲 |
|----------|----------|--------------|---|
| image | [IN] | イメージオブジェクト | |
| address | [IN] | HTTP URL | |
| name | [IN] | 格納名称 | 以下の条件を除いた ASCII コードが使用可能 ・先頭がアンダースコアは不可 ・以下の文字は使用不可(禁則文字) ¥ / : * ? " < > |
| rotation | [IN] | 回転方向(定数) | CLS_RT_NORMAL: 回転なし CLS_RT_RIGHT90: 右 90 度回転 CLS_RT_ROTATE180: 右 180 度回転 CLS_RT_LEFT90: 左 90 度回転 |
| width | [IN] | 横サイズ(ピクセル単位) | |
| height | [IN] | 縦サイズ(ピクセル単位) | |

説明

このメソッドは、ビットマップのファイル名、印刷幅、回転を指定して、プリンターのフラッシュメモリにビットマップ画像(ロゴ)を登録するために使用します。登録したロゴは、[drawNVBitmap メソッド](#)を使用して描画できます。登録可能なビットマップ形式は、BMP/GIF/EXIF/JPG/PNG/TIFF です。

登録されるビットマップ画像サイズは、指定された縦横サイズに合わせて、アスペクト比を維持した状態でリサイズします。

例: 以下の場合、リサイズ後の画像サイズは「200x50」ピクセルとなる。

画像サイズ: 400x100 ピクセル
横サイズ: 200
縦サイズ: 200

指定のサイズが 0 の場合、対象の画像サイズを基準とする。

例: 以下の場合、リサイズ後の画像サイズは「800x200」ピクセルとなる。

画像サイズ: 400x100 ピクセル
横サイズ: 0
縦サイズ: 200

戻り値

成功時は CLS_SUCCESS(0) を返します。それ以外のエラーコードは[「5.4 戻り値」](#)を参照してください。

使用例

```
var image = new Image();

// 画像読み込み処理は省略

print.messageID("00000001");
print.storeNVBitmap(image, "http://192.168.129.178/xmlprint",
    "Dice_1.bmp", citizen.LabelConst.CLS_RT_NORMAL, 0, 0);
```

5.5.6. clearOutput メソッド

形式

clearOutput (address)

パラメータ

パラメータの意味と設定可能な値は以下の通りです。

| 値 | [IN/OUT] | 意味 | 設定可能範囲 |
|---------|----------|----------|--------|
| address | [IN] | HTTP URL | |

説明

プリンターをリセットします。プリンター側では、電源 ON 時と同等の初期化処理が行われるため、未印字データもクリアされます。

戻り値

成功時は CLS_SUCCESS(0) を返します。それ以外のエラーコードは「[5.4 戻り値](#)」を参照してください。

使用例

```
print.messageID("00000001");  
print.clearOutput();
```

5.5.7. sendData メソッド

形式

sendData (address, data)

パラメータ

パラメータの意味と設定可能な値は以下の通りです。

| 値 | [IN/OUT] | 意味 | 設定可能範囲 |
|---------|----------|----------|--------|
| address | [IN] | HTTP URL | |
| data | [IN] | 送信データ | |

説明

このメソッドは、バイトデータをそのままプリンターに送信するために使用します。
通常は必要ありませんが、プリンターのコマンドを直接送信したい場合に使用します。
ご使用の際は、他のメソッドに影響を与えない様に注意する必要があります。

戻り値

成功時は CLS_SUCCESS(0) を返します。それ以外のエラーコードは「[5.4 戻り値](#)」を参照してください。

使用例

```
var data = new Array(0x01, 0x23, 0x0D, 0x0A); // [01]# (リセット)

print.messageID("00000001");
print.sendData("http://192.168.129.178/xmlprint", data);
```


5.5.8. HorizontalMagnification プロパティ

形式

HorizontalMagnification

属性

Read/Write

説明

このプロパティは、水平方向のピクセルサイズ(ドット構成単位)を保持します。値は、1(dot)もしくは、2(dot)で指定します。

設定方法

setHorizontalMagnification (horizontalMagnification)

パラメータに、設定したいプロパティ値を指定してください。

デフォルトを「1」とし、未設定時はデフォルトでコマンドを送信します。

成功時は CLS_SUCCESS(0) を返します。それ以外のエラーコードは[「5.4 戻り値」](#)を参照してください。

取得方法

getHorizontalMagnification ()

戻り値として、設定されている水平ピクセルサイズを返します。

デフォルトを「1」とし、未設定時はデフォルトが返ります。

使用例

```
var pixelsize;  
  
pixelsize = print.getHorizontalMagnification();  
print.setHorizontalMagnification(2);
```

5.5.9. VerticalMagnification プロパティ

形式

VerticalMagnification

属性

Read/Write

説明

このプロパティは、垂直方向のピクセルサイズ(dot 構成単位)を保持します。値は、1(dot)から 3(dot)の間で指定します。

設定方法

setVerticalMagnification (verticalMagnification)

パラメータに、設定したいプロパティ値を指定してください。

デフォルトを「1」とし、未設定時はデフォルトでコマンドを送信します。

成功時は CLS_SUCCESS(0) を返します。それ以外のエラーコードは[「5.4 戻り値」](#)を参照してください。

取得方法

getVerticalMagnification ()

戻り値として、設定されている垂直ピクセルサイズを返します。

デフォルトを「1」とし、未設定時はデフォルトが返ります。

使用例

```
var pixelsize;  
  
pixelsize = print.getVerticalMagnification();  
print.setVerticalMagnification(3);
```

5.5.10. FormatAttribute プロパティ

形式

FormatAttribute

属性

Read/Write

説明

このプロパティは、印刷領域が重なっている箇所の展開方法を保持します。値は、以下の通りです。

0: XOR 展開指定となり、文字やバーコードの重なった部分が白抜きとなります。(初期値)

1: OR 展開指定となり、文字やバーコードの重ね書きを行います。

設定方法

setFormatAttribute (formatAttribute)

パラメータに、設定したいプロパティ値を指定してください。

未設定時は、コマンドを送信しません。

プリンターのデフォルト値は「0」です。

成功時は CLS_SUCCESS(0) を返します。それ以外のエラーコードは「[5.4 戻り値](#)」を参照してください。

取得方法

getFormatAttribute ()

戻り値として、設定されている展開方法を返します。

未設定時は、デフォルトとして CLS_PROPERTY_DEFAULT(999999) を返します。

使用例

```
var attr;  
  
print.setFormatAttribute(1);  
attr = print.getFormatAttribute();
```

印刷結果



5.5.11. ContinuousMediaLength プロパティ

形式

ContinuousMediaLength

属性

Read/Write

説明

連続紙を使用した場合のラベル長の設定です。
ラベルフォーマットの長さは、このコマンドで設定した長さになります。
オートカッター使用時は、この設定の長さで、ラベルカットを行います。

| | |
|-------|----------------------------------|
| インチ設定 | 0001 ~ 9999 (0.01 インチ~99.99 インチ) |
| ミリ設定 | 0001 ~ 9999 (0.1mm~999.9mm) |

設定方法

setContinuousMediaLength (continuousMediaLength)

パラメータに、設定したいプロパティ値を指定してください。
未設定時は、コマンドを送信しません。
プリンターのデフォルト値は「0000」です。
成功時は CLS_SUCCESS(0) を返します。それ以外のエラーコードは[「5.4 戻り値」](#)を参照してください。

取得方法

getContinuousMediaLength ()

戻り値として、設定されている連続用紙長を返します。
未設定時は、デフォルトとして CLS_PROPERTY_DEFAULT(999999) を返します。

使用例

```
var medialength;  
  
print.setContinuousMediaLength(2000);  
medialength = print.getContinuousMediaLength();
```

5.5.12. MeasurementUnit プロパティ

形式

MeasurementUnit

属性

Read/Write

説明

このプロパティは、全ての全距離指定コマンドのパラメータの単位を保持します。値は以下の通りです。

| 値 | 意味 |
|--------------------|-----------------|
| CLS_UNIT_MILLI (0) | ミリ(0.1mm 単位) |
| CLS_UNIT_INCH (1) | インチ(0.01 インチ単位) |

設定方法

setMeasurementUnit (measurementUnit)

パラメータに、設定したいプロパティ値を指定してください。

未設定時は、コマンドを送信しません。

プリンターのデフォルト値は「1」です。

成功時は CLS_SUCCESS(0) を返します。それ以外のエラーコードは[「5.4 戻り値」](#)を参照してください。

取得方法

getMeasurementUnit ()

戻り値として、設定されている距離指定パラメータの単位を返します。

未設定時は、デフォルトとして CLS_PROPERTY_DEFAULT(999999) を返します。

使用例

```
var unit;  
  
print.setMeasurementUnit(citizen.LabelConst.CLS_UNIT_MILLI);  
unit = print.getMeasurementUnit();
```

5.5.13. PrintSpeed プロパティ

形式

PrintSpeed

属性

Read/Write

説明

このプロパティは、印字部分の速度を保持します。値は、1 文字のアルファベットか数字で設定します。(「[5.7.2. パラメータ](#)」の No18 を参照)

※指定可能範囲、プリンターの初期値は、機種により異なります。

使用機種の取扱い説明書をご参照ください。

設定方法

setPrintSpeed (printSpeed)

パラメータに、設定したいプロパティ値を指定してください。

未設定時はコマンドを送信しません。

成功時は CLS_SUCCESS(0) を返します。それ以外のエラーコードは「[5.4 戻り値](#)」を参照してください。

取得方法

getPrintSpeed ()

戻り値として、設定されている印刷速度を返します。

未設定時は、デフォルトとして CLS_PROPERTY_DEFAULT(999999) を返します。

使用例

```
var printspeed;  
  
print.setPrintSpeed(citizen.LabelConst.CLS_SPEEDSETTING_X);  
printspeed = print.getPrintSpeed();
```

5.5.14. FeedSpeed プロパティ

形式

FeedSpeed

属性

Read/Write

説明

このプロパティは、非印字部分の速度を保持します。値は、1 文字のアルファベットか数字で設定します。
([「5.7.2. パラメータ」](#)の No18 を参照)

※指定可能範囲、プリンターの初期値は、機種により異なります。

使用機種の取扱い説明書をご参照ください。

設定方法

setFeedSpeed (feedSpeed)

パラメータに、設定したいプロパティ値を指定してください。

未設定時はコマンドを送信しません。

成功時は CLS_SUCCESS(0) を返します。それ以外のエラーコードは[「5.4 戻り値」](#)を参照してください。

取得方法

getFeedSpeed ()

戻り値として、設定されている非印刷部分の紙送り速度を返します。

未設定時は、デフォルトとして CLS_PROPERTY_DEFAULT(999999) を返します。

使用例

```
var feedspeed;  
  
print.setFeedSpeed(citizen.LabelConst.CLS_SPEEDSETTING_W);  
feedspeed = print.getFeedSpeed();
```

5.5.15. SlewSpeed プロパティ

形式

SlewSpeed

属性

Read/Write

説明

このプロパティは、紙送りのフィード速度を保持します。値は、1 文字のアルファベットか数字で設定します。
([「5.7.2.パラメータ」](#)の No18 を参照)

※指定可能範囲、プリンターの初期値は、機種により異なります。

使用機種の取扱い説明書をご参照ください。

設定方法

setSlewSpeed (slewSpeed)

パラメータに、設定したいプロパティ値を指定してください。

未設定時はコマンドを送信しません。

成功時は CLS_SUCCESS(0) を返します。それ以外のエラーコードは[「5.4 戻り値」](#)を参照してください。

取得方法

getSlewSpeed ()

戻り値として、設定されている紙送り部分の紙送り速度を返します。

未設定時は、デフォルトとして CLS_PROPERTY_DEFAULT(999999) を返します。

使用例

```
var slewspeed;  
  
print.setSlewSpeed(citizen.LabelConst.CLS_SPEEDSETTING_V);  
slewspeed = print.getSlewSpeed();
```


5.5.16. BackupSpeed プロパティ

形式

BackupSpeed

属性

Read/Write

説明

このプロパティは、バックフィード時の速度を保持します。値は、1 文字のアルファベットか数字で設定します。

| 値 | 意味 |
|---|--|
| CLS_SPEEDSETTING_A or CLS_SPEEDSETTING_B | 1.0 インチ (25.4mm) / 秒 |
| CLS_SPEEDSETTING_C or CLS_SPEEDSETTING_D | 2.0 インチ (50.8mm) / 秒 |
| CLS_SPEEDSETTING_E or CLS_SPEEDSETTING_F | 3.0 インチ (76.2mm) / 秒 |
| CLS_SPEEDSETTING_G or CLS_SPEEDSETTING_H | 4.0 インチ (101.6mm) / 秒 |
| CLS_SPEEDSETTING_I or CLS_SPEEDSETTING_J | 5.0 インチ (127.0mm) / 秒 |
| CLS_SPEEDSETTING_K or CLS_SPEEDSETTING_L | 6.0 インチ (152.4mm) / 秒 |
| CLS_SPEEDSETTING_M or CLS_SPEEDSETTING_N | 7.0 インチ (177.8mm) / 秒 |
| CLS_SPEEDSETTING_O | 8.0 インチ (203.2mm) / 秒 |
| CLS_SPEEDSETTING_1 ～ CLS_SPEEDSETTING_8 | 1.0 インチ (25.4mm) / 秒 ～ 8.0 インチ (203.2mm) / 秒 |

※指定可能範囲、プリンターの初期値は、機種により異なります。

使用機種の取扱い説明書をご参照ください。

設定方法

setBackupSpeed (backupSpeed)

パラメータに、設定したいプロパティ値を指定してください。

未設定時はコマンドを送信しません。

成功時は CLS_SUCCESS(0) を返します。それ以外のエラーコードは「[5.4 戻り値](#)」を参照してください。

取得方法

getBackupSpeed ()

戻り値として、設定されているバックフィード設定を返します。

未設定時は、デフォルトとして CLS_PROPERTY_DEFAULT(999999) を返します。

使用例

```
var backupspeed;

print.setBackupSpeed(citizen.LabelConst.CLS_SPEEDSETTING_O);
backspeed = print.getBackupSpeed();
```

5.5.17. PrintDarkness プロパティ

形式

PrintDarkness

属性

Read/Write

説明

このプロパティは、印刷濃度の設定を保持します。設定範囲と初期値は以下の通りです。

| | |
|------|--------|
| 設定範囲 | 0 ~ 30 |
| 初期値 | 10 |

設定方法

setPrintDarkness (printDarkness)

パラメータに、設定したいプロパティ値を指定してください。

未設定時はコマンドを送信しません。

成功時は CLS_SUCCESS(0) を返します。それ以外のエラーコードは[「5.4 戻り値」](#)を参照してください。

取得方法

getPrintDarkness ()

戻り値として、設定されている印字濃度を返します。

未設定時は、デフォルトとして CLS_PROPERTY_DEFAULT(999999) を返します。

使用例

```
var printdarkness;  
  
print.setPrintDarkness(30);  
printdarkness = print.getPrintDarkness();
```

5.5.18. DoubleHeat プロパティ

形式

DoubleHeat

属性

Read/Write

説明

このプロパティは、ダブルヒート設定を保持します。ダブルヒートを設定することで、同じ位置に 2 回熱をかけて印字します。そのため、印字速度は半分になりますが、より濃く印字することができます。

0: ダブルヒート OFF (初期値)

1: ダブルヒート ON

設定方法

setDoubleHeat (doubleHeat)

パラメータに、設定したいプロパティ値を指定してください。

未設定時はコマンドを送信しません。

成功時は CLS_SUCCESS(0) を返します。それ以外のエラーコードは「[5.4 戻り値](#)」を参照してください。

取得方法

getDoubleHeat ()

戻り値として、設定されているダブルヒート設定を返します。

未設定時は、デフォルトとして CLS_PROPERTY_DEFAULT(999999) を返します。

使用例

```
var doubleheat;
```

```
print.setDoubleHeat(1);  
doubleheat = print.getDoubleHeat();
```

5.5.19. VerticalOffset プロパティ

形式

VerticalOffset

属性

Read/Write

説明

印字内容全体の位置を調整する為に、紙の上下(行)方向の印字開始位置のオフセット値を設定します。

| | |
|-------|----------------------------------|
| インチ設定 | 0000 ~ 9999 (0.00 インチ~99.99 インチ) |
| ミリ設定 | 0000 ~ 9999 (0.0mm~999.9mm) |
| 初期値 | 0000 |

設定方法

setVerticalOffset (verticalOffset)

パラメータに、設定したいプロパティ値を指定してください。

未設定時はコマンドを送信しません。

成功時は CLS_SUCCESS(0) を返します。それ以外のエラーコードは[「5.4 戻り値」](#)を参照してください。

取得方法

getVerticalOffset ()

戻り値として、設定されている上下オフセットを返します。

未設定時は、デフォルトとして CLS_PROPERTY_DEFAULT(999999) を返します。

使用例

```
var verticaloffset;  
  
print.setVerticalOffset(10);  
verticaloffset = print.getVerticalOffset();
```

5.5.20. HorizontalOffset プロパティ

形式

HorizontalOffset

属性

Read/Write

説明

印字内容全体の位置を調整する為、用紙の左右オフセット値(列方向の印字開始位置)を設定します。

| | |
|-------|----------------------------------|
| インチ設定 | 0000 ~ 9999 (0.00 インチ~99.99 インチ) |
| ミリ設定 | 0000 ~ 9999 (0.0mm~999.9mm) |
| 初期値 | 0000 |

設定方法

setHorizontalOffset (horizontalOffset)

パラメータに、設定したいプロパティ値を指定してください。

未設定時はコマンドを送信しません。

成功時は CLS_SUCCESS(0) を返します。それ以外のエラーコードは[「5.4 戻り値」](#)を参照してください。

取得方法

getHorizontalOffset ()

戻り値として、設定されている左右オフセットを返します。

未設定時は、デフォルトとして CLS_PROPERTY_DEFAULT(999999) を返します。

使用例

```
var horizontaloffset;  
  
print.setHorizontalOffset(20);  
horizontaloffset = print.getHorizontalOffset();
```

5.5.21. MediaHandling プロパティ

形式

MediaHandling

属性

Read/Write

説明

このプロパティは、印刷後の動作設定を保持します。設定可能な動作は、以下の通りです。

| 値 | 意味 |
|-----------------------------------|--------------------------------|
| CLS_MEDIAHANDLING_NONE (0) | 印刷後の動作を OFF にします。 |
| CLS_MEDIAHANDLING_TEAROFF (1) | 排出動作のみ ON にします。 |
| CLS_MEDIAHANDLING_DISPENSES (2) | 剥離センサーと排出動作を ON にします。 |
| CLS_MEDIAHANDLING_PAUSE (3) | 排出動作を ON にし、排出後にポーズ状態にします。 |
| CLS_MEDIAHANDLING_CUT (4) | オートカッターのみ ON にします。 |
| CLS_MEDIAHANDLING_CUTANDPAUSE (5) | オートカッターを ON にし、カット後にポーズ状態にします。 |
| CLS_MEDIAHANDLING_PEELOFF (6) | 剥離センサーのみ ON にします。 |
| CLS_MEDIAHANDLING_REWIND (7) | リワインダー(巻き取り)モードを ON にします。 |

設定方法

setMediaHandling (mediaHandling)

パラメータに、設定したいプロパティ値を指定してください。

未設定時はコマンドを送信しません。

成功時は CLS_SUCCESS(0) を返します。それ以外のエラーコードは「[5.4 戻り値](#)」を参照してください。

取得方法

getMediaHandling ()

戻り値として、設定されている印刷後動作を返します。

未設定時は、デフォルトとして CLS_PROPERTY_DEFAULT(999999) を返します。

使用例

```
var mediahandling;  
  
print.setMediaHandling(citizen.LabelConst.CLS_MEDIAHANDLING_PAUSE);  
mediahandling = print.getMediaHandling();
```

5.5.22. StartOffset プロパティ

形式

StartOffset

属性

Read/Write

説明

基準点から印字ヘッドまでの距離を指定することができます。この値を変える事により、物理的な印字開始位置を変更することが出来ます。

| インチ指定 | | | ミリ指定 | | |
|-------|------|------|------|------|------|
| 初期値 | 最小値 | 最大値 | 初期値 | 最小値 | 最大値 |
| 0220 | 0120 | 0320 | 0559 | 0305 | 0813 |

※単位 0.01 インチ又は 0.1mm

設定方法

setStartOffset(startOffset)

パラメータに、設定したいプロパティ値を指定してください。

未設定時はコマンドを送信しません。

インチ/ミリ指定の対応のチェックは行いません。

成功時は CLS_SUCCESS(0) を返します。それ以外のエラーコードは「[5.4 戻り値](#)」を参照してください。

取得方法

getStartOffset ()

戻り値として、設定されている開始オフセットを返します。

未設定時は、デフォルトとして CLS_PROPERTY_DEFAULT(999999) を返します。

使用例

```
var startoffset;

print.setStartOffset(220);
startoffset = print.getStartOffset();
```

5.5.23. StopOffset プロパティ

形式

StopOffset

属性

Read/Write

説明

下記範囲の値にて、基準点(基準線)からカット位置又は、剥離位置までの距離を指定できます。

指定値が小さい場合、フィード量が小さいので印刷したラベルをカットしてしまいます。

指定値が適切な場合、必要量フィード後、紙間でカットします。

指定値が大きい場合、フィード量が大きいので、次のラベルをカットしてしまいます。

| 動作 | インチ指定 | | | ミリ指定 | | |
|-------|-------|------|------|------|------|------|
| | 初期値 | 最小値 | 最大値 | 初期値 | 最小値 | 最大値 |
| 通常印刷 | 0000 | 0000 | 9999 | 0000 | 0000 | 9999 |
| カッター | 0100 | | | 0254 | | |
| 剥離 | 0050 | | | 0127 | | |
| ティアオフ | 0070 | | | 0178 | | |

※単位 0.01 インチ又は 0.1mm

設定方法

setStopOffset (stopOffset)

パラメータに、設定したいプロパティ値を指定してください。

未設定時はコマンドを送信しません。

成功時は CLS_SUCCESS(0) を返します。それ以外のエラーコードは[「5.4 戻り値」](#)を参照してください。

取得方法

getStopOffset ()

戻り値として、設定されている終了オフセットを返します。

未設定時は、デフォルトとして CLS_PROPERTY_DEFAULT(999999) を返します。

使用例

```
var stopoffset;  
  
print.setStopOffset(240);  
stopoffset = print.getStopOffset();
```


5.5.24. LabelSensor プロパティ

形式

LabelSensor

属性

Read/Write

説明

このプロパティは、用紙センサーの設定を保持します。設定値は、以下の通りです。

| 値 | 意味 |
|------------------------------|---|
| CLS_SELSENSOR_NONE (0) | なし。 なしの場合、連続用紙長設定の値(ContinuousMediaLength プロパティ)をチェックし、未設定、または 0000 の場合は引数エラーとします。 |
| CLS_SELSENSOR_SEETHROUGH (1) | エッジセンサー。 ラベル紙の紙間、ダイカット紙、タグ紙のノッチ穴検出などに使用します。 |
| CLS_SELSENSOR_REFLECT (2) | 反射型用紙センサー。 ラベル裏面に印刷された黒線を検出して、ラベル位置を認識します。 |

設定方法

setLabelSensor (labelSensor)

パラメータに、設定したいプロパティ値を指定してください。

未設定時はコマンドを送信しません。

成功時は CLS_SUCCESS(0) を返します。それ以外のエラーコードは[「5.4 戻り値」](#)を参照してください。

取得方法

getLabelSensor ()

戻り値として、設定されている用紙センサーを返します。

未設定時は、デフォルトとして CLS_PROPERTY_DEFAULT(999999) を返します。

使用例

```
// センサー「なし」以外のとき  
var labelsensor;
```

```
print.setLabelSensor(citizen.LabelConst.CLS_SELSENSOR_SEETHROUGH);  
labelsensor = print.getLabelSensor();
```

```
// センサー「なし」のとき  
var labelsensor;
```

```
print.setContinuousMediaLength(100);  
print.setLabelSensor(citizen.LabelConst.CLS_SELSENSOR_NONE);  
labelsensor = print.getLabelSensor();
```

5.5.25. PrintMethod プロパティ

形式

PrintMethod

属性

Read/Write

説明

リボンを使用する熱転写モードと感熱紙を使用する感熱モードの印刷方法の指定を行います。

| 値 | 意味 |
|----------------------|--------|
| CLS_PRTMETHOD_TT (0) | 熱転写モード |
| CLS_PRTMETHOD_DT (1) | 感熱モード |

設定方法

setPrintMethod (printMethod)

パラメータに、設定したいプロパティ値を指定してください。

未設定時はコマンドを送信しません。

成功時は CLS_SUCCESS(0) を返します。それ以外のエラーコードは[「5.4 戻り値」](#)を参照してください。

取得方法

getPrintMethod ()

戻り値として、設定されている印字方法を返します。

未設定時は、デフォルトとして CLS_PROPERTY_DEFAULT(999999) を返します。

使用例

```
var printmethod;  
  
print.setPrintMethod(citizen.LabelConst.CLS_PRTMETHOD_DT);  
printmethod = print.getPrintMethod();
```

5.5.26. SensorLocation プロパティ

形式

SensorLocation

属性

Read/Write

説明

前方センサーと後方センサーの 2 種類の紙検出センサーを搭載している機種において、使用する紙検出センサーを切り替えます。設定した内容はバックアップメモリに記憶され電源を切っても設定は有効です。

| 値 | 意味 |
|----------------------------------|--------|
| CLS_SENS_LOCATION_FRONT (0) | 前方センサー |
| CLS_SENS_LOCATION_ADJUSTABLE (1) | 後方センサー |

設定方法

setSensorLocation (sensorLocation)

パラメータに、設定したいプロパティ値を指定してください。

未設定時はコマンドを送信しません。

成功時は CLS_SUCCESS(0) を返します。それ以外のエラーコードは「[5.4 戻り値](#)」を参照してください。

取得方法

getSensorLocation ()

戻り値として、設定されているセンサー設定(前方/後方)を返します。

未設定時は、デフォルトとして CLS_PROPERTY_DEFAULT(999999) を返します。

使用例

```
var location;  
  
print.setSensorLocation(citizen.LabelConst.CLS_SENS_LOCATION_ADJUSTABLE);  
location = print.getSensorLocation();
```

5.5.27. CommandInterpreterInAction プロパティ

形式

CommandInterpreterInAction

属性

Read

説明

このプロパティは、プリンターがコマンド処理中であることを示します。

このプロパティを参照する際は、あらかじめ [printerCheck メソッド](#)により、プリンターからステータスを取得しておく必要があります。

設定方法

なし。

取得方法

getCommandInterpreterInAction ()

戻り値として、以下の値を返します。

| 値 | 意味 |
|------------------------------|----------|
| 0 | No |
| 1 | Yes |
| CLS_PROPERTY_DEFAULT(999999) | ステータス未取得 |

使用例

[printerCheck メソッド](#)を参照

5.5.28. PaperError プロパティ

形式

PaperError

属性

Read

説明

このプロパティは、プリンターのペーパーエラー状態を示します。
このプロパティを参照する際は、あらかじめ [printerCheck メソッド](#)により、プリンターからステータスを取得しておく必要があります。

設定方法

なし。

取得方法

getPaperError ()

戻り値として、以下の値を返します。

| 値 | 意味 |
|------------------------------|----------|
| 0 | No |
| 1 | Yes |
| CLS_PROPERTY_DEFAULT(999999) | ステータス未取得 |

使用例

[printerCheck メソッド](#)を参照

5.5.29. RibbonEnd プロパティ

形式

RibbonEnd

属性

Read

説明

このプロパティは、プリンターのリボンエンド状態を示します。

このプロパティを参照する際は、あらかじめ [printerCheck メソッド](#)により、プリンターからステータスを取得しておく必要があります。

設定方法

なし。

取得方法

getRibbonEnd ()

戻り値として、以下の値を返します。

| 値 | 意味 |
|------------------------------|----------|
| 0 | No |
| 1 | Yes |
| CLS_PROPERTY_DEFAULT(999999) | ステータス未取得 |

使用例

[printerCheck メソッド](#)を参照

5.5.30. BatchProcessing プロパティ

形式

BatchProcessing

属性

Read

説明

このプロパティは、プリンターがバッチ処理印字中であることを示します。

このプロパティを参照する際は、あらかじめ [printerCheck メソッド](#)により、プリンターからステータスを取得しておく必要があります。

設定方法

なし。

取得方法

getBatchProcessing ()

戻り値として、以下の値を返します。

| 値 | 意味 |
|------------------------------|----------|
| 0 | No |
| 1 | Yes |
| CLS_PROPERTY_DEFAULT(999999) | ステータス未取得 |

使用例

[printerCheck メソッド](#)を参照

5.5.31. Printing プロパティ

形式

Printing

属性

Read

説明

このプロパティは、プリンターが印字中であることを示します。

このプロパティを参照する際は、あらかじめ [printerCheck メソッド](#)により、プリンターからステータスを取得しておく必要があります。

設定方法

なし。

取得方法

getPrinting ()

戻り値として、以下の値を返します。

| 値 | 意味 |
|------------------------------|----------|
| 0 | No |
| 1 | Yes |
| CLS_PROPERTY_DEFAULT(999999) | ステータス未取得 |

使用例

[printerCheck メソッド](#)を参照

5.5.32. Pause プロパティ

形式

Pause

属性

Read

説明

このプロパティは、プリンターがポーズ中であることを示します。

このプロパティを参照する際は、あらかじめ [printerCheck メソッド](#)により、プリンターからステータスを取得しておく必要があります。

設定方法

なし。

取得方法

getPause ()

戻り値として、以下の値を返します。

| 値 | 意味 |
|------------------------------|----------|
| 0 | No |
| 1 | Yes |
| CLS_PROPERTY_DEFAULT(999999) | ステータス未取得 |

使用例

[printerCheck メソッド](#)を参照

5.5.33. WaitingForPeeling プロパティ

形式

WaitingForPeeling

属性

Read

説明

このプロパティは、プリンターが剥離待ち中であることを示します。

このプロパティを参照する際は、あらかじめ [printerCheck メソッド](#)により、プリンターからステータスを取得しておく必要があります。

設定方法

なし。

取得方法

getWaitingForPeeling ()

戻り値として、以下の値を返します。

| 値 | 意味 |
|------------------------------|----------|
| 0 | No |
| 1 | Yes |
| CLS_PROPERTY_DEFAULT(999999) | ステータス未取得 |

使用例

[printerCheck メソッド](#)を参照

5.5.34. PrintHeadLowTemp プロパティ

形式

PrintHeadLowTemp

属性

Read

説明

このプロパティは、プリンターがプリントヘッド低温エラーであることを示します。(有線 LAN I/F の IF5-EFX1 に対応)

このプロパティを参照する際は、あらかじめ [printerCheck メソッド](#)により、プリンターからステータスを取得しておく必要があります。

設定方法

なし。

取得方法

getPrintHeadLowTemp ()

戻り値として、以下の値を返します。

| 値 | 意味 |
|------------------------------|----------|
| 0 | No |
| 1 | Yes |
| CLS_PROPERTY_DEFAULT(999999) | ステータス未取得 |

使用例

[printerCheck メソッド](#)を参照

5.5.35. PrintHeadFailure プロパティ

形式

PrintHeadFailure

属性

Read

説明

このプロパティは、プリンターがプリントヘッド切れエラーであることを示します。(有線 LAN I/F の IF5-EFX1 に対応)

このプロパティを参照する際は、あらかじめ [printerCheck メソッド](#)により、プリンターからステータスを取得しておく必要があります。

設定方法

なし。

取得方法

getPrintHeadFailure ()

戻り値として、以下の値を返します。

| 値 | 意味 |
|------------------------------|----------|
| 0 | No |
| 1 | Yes |
| CLS_PROPERTY_DEFAULT(999999) | ステータス未取得 |

使用例

[printerCheck メソッド](#)を参照

5.5.36. PrintHeadOverheat プロパティ

形式

PrintHeadOverheat

属性

Read

説明

このプロパティは、プリンターがプリントヘッドオーバーヒートエラーであることを示します。（有線 LAN I/F の IF5-EFX1 で対応）

このプロパティを参照する際は、あらかじめ [printerCheck メソッド](#)により、プリンターからステータスを取得しておく必要があります。

設定方法

なし。

取得方法

getPrintHeadOverheat ()

戻り値として、以下の値を返します。

| 値 | 意味 |
|------------------------------|----------|
| 0 | No |
| 1 | Yes |
| CLS_PROPERTY_DEFAULT(999999) | ステータス未取得 |

使用例

[printerCheck メソッド](#)を参照

5.5.37. MechanismOpen プロパティ

形式

MechanismOpen

属性

Read

説明

このプロパティは、プリンターのプリントヘッドオープンエラーであることを示します。(有線 LAN I/F の IF5-EFX1 で対応)

このプロパティを参照する際は、あらかじめ [printerCheck メソッド](#)により、プリンターからステータスを取得しておく必要があります。

設定方法

なし。

取得方法

getMechanismOpen ()

戻り値として、以下の値を返します。

| 値 | 意味 |
|------------------------------|----------|
| 0 | No |
| 1 | Yes |
| CLS_PROPERTY_DEFAULT(999999) | ステータス未取得 |

使用例

[printerCheck メソッド](#)を参照

5.5.38. AutoCutterError プロパティ

形式

AutoCutterError

属性

Read

説明

このプロパティは、プリンターのオートカッター異常を示します。(有線 LAN I/F の IF5-EFX1 で対応)

このプロパティを参照する際は、あらかじめ [printerCheck メソッド](#)により、プリンターからステータスを取得しておく必要があります。

設定方法

なし。

取得方法

getAutoCutterError ()

戻り値として、以下の値を返します。

| 値 | 意味 |
|------------------------------|----------|
| 0 | No |
| 1 | Yes |
| CLS_PROPERTY_DEFAULT(999999) | ステータス未取得 |

使用例

[printerCheck メソッド](#)を参照

5.5.39. FanMotorError プロパティ

形式

FanMotorError

属性

Read

説明

このプロパティは、プリンターのファンモーターエラーを示します。(有線 LAN I/F の IF5-EFX1 に対応)

このプロパティを参照する際は、あらかじめ [printerCheck メソッド](#)により、プリンターからステータスを取得しておく必要があります。

設定方法

なし。

取得方法

getFanMotorError ()

戻り値として、以下の値を返します。

| 値 | 意味 |
|------------------------------|----------|
| 0 | No |
| 1 | Yes |
| CLS_PROPERTY_DEFAULT(999999) | ステータス未取得 |

使用例

[printerCheck メソッド](#)を参照

5.5.40. MiscError プロパティ

形式

MiscError

属性

Read

説明

このプロパティは、本 SDK がサポートしているステータスプロパティ以外のエラーが、プリンターで発生していることを示します。(有線 LAN I/F の IF5-EFX1 に対応)

このプロパティを参照する際は、あらかじめ [printerCheck メソッド](#)により、プリンターからステータスを取得しておく必要があります。

設定方法

なし。

取得方法

getMiscError ()

戻り値として、以下の値を返します。

| 値 | 意味 |
|------------------------------|----------|
| 0 | No |
| 1 | Yes |
| CLS_PROPERTY_DEFAULT(999999) | ステータス未取得 |

使用例

[printerCheck メソッド](#)を参照

5.6. LabelDesign クラス詳細

5.6.1. コンストラクタ

形式

citizen.LabelDesign ()

パラメータ

ありません。

説明

JavaScript ライブラリのコンストラクタです。インスタンスを生成します。

戻り値

ありません。

使用例

```
var design = new citizen.LabelDesign();
```

5.6.2. drawTextPtrFont メソッド

形式

drawTextPtrFont (data, locale, font, rotation, hexp, vexp, size, x, y)

パラメータ

| 値 | [IN/OUT] | 意味 | 設定可能範囲 |
|----------|----------|------------|--|
| data | [IN] | 印字文字列 | ※文字コード変換できない文字は"?"に変換する。 |
| locale | [IN] | ロケール(定数) | CLS_LOCALE_JP: 日本モデル用ロケール CLS_LOCALE_OTHER: 海外モデル用ロケール CLS_LOCALE_CN: 中国モデル用ロケール CLS_LOCALE_KR: 韓国モデル用ロケール |
| font | [IN] | 文字種(定数) | CLS_PRT_FNT_0: システムフォント 0 CLS_PRT_FNT_1: システムフォント 1 CLS_PRT_FNT_2: システムフォント 2 CLS_PRT_FNT_3: システムフォント 3 CLS_PRT_FNT_4: システムフォント 4 CLS_PRT_FNT_5: システムフォント 5 CLS_PRT_FNT_6: システムフォント 6 CLS_PRT_FNT_7: システムフォント 7 CLS_PRT_FNT_8: システムフォント 8 CLS_PRT_FNT_TRIUMVIRATE: スムースフォント CLS_PRT_FNT_TRIUMVIRATE_B: スムースフォント(Bold) CLS_PRT_FNT_KANJI: 漢字(横書き) CLS_PRT_FNT_KANJIT: 漢字(縦書き) |
| rotation | [IN] | 回転方向(定数) | CLS_RT_NORMAL: 回転なし CLS_RT_RIGHT90: 右 90 度回転 CLS_RT_ROTATE180: 右 180 度回転 CLS_RT_LEFT90: 左 90 度回転 |
| hexp | [IN] | 水平拡大率 | 1～24 |
| vexp | [IN] | 垂直拡大率 | 1～24 |
| size | [IN] | 文字サイズ(定数) | CLS_PRT_FNT_SIZE_4: 文字種ポイント数(4pt) CLS_PRT_FNT_SIZE_5: 文字種ポイント数(5pt) CLS_PRT_FNT_SIZE_6: 文字種ポイント数(6pt) CLS_PRT_FNT_SIZE_8: 文字種ポイント数(8pt) CLS_PRT_FNT_SIZE_10: 文字種ポイント数(10pt) CLS_PRT_FNT_SIZE_12: 文字種ポイント数(12pt) CLS_PRT_FNT_SIZE_14: 文字種ポイント数(14pt) CLS_PRT_FNT_SIZE_18: 文字種ポイント数(18pt) CLS_PRT_FNT_SIZE_24: 文字種ポイント数(24pt) CLS_PRT_FNT_SIZE_30: 文字種ポイント数(30pt) CLS_PRT_FNT_SIZE_36: 文字種ポイント数(36pt) CLS_PRT_FNT_SIZE_48: 文字種ポイント数(48pt) CLS_PRT_FNT_KANJI_SIZE_16: 漢字文字種(16ドット) CLS_PRT_FNT_KANJI_SIZE_24: 漢字文字種(24ドット) CLS_PRT_FNT_KANJI_SIZE_32: 漢字文字種(32ドット) ※ CLS_PRT_FNT_KANJI_SIZE_48: 漢字文字種(48ドット) ※ ※中国モデル用ロケールでは、非サポート。 |
| x | [IN] | 印字位置(X 座標) | 0000～9999 |
| y | [IN] | 印字位置(Y 座標) | ※x,y は、左下を基準(0,0)として位置とする。 |

説明

プリンターのシステムフォントを用いて、回転・縦横拡大率・フォント種・印字位置等・指定条件で、入力された内容の文字を描画します。

文字サイズは、文字種ごとに以下のパターンを指定可能です。

漢字:16,24,32,48 ※ドット単位

スムースフォント:4,5,6,8,10,12,14,18,24,30,36,48

上記以外:文字種ごとにサイズ固定

戻り値

成功時は CLS_SUCCESS(0) を返します。それ以外のエラーコードは[「5.4 戻り値」](#)を参照してください。

使用例

```
// Locale:OTHER
design.drawTextPtrFont("drawTextPtrFont",
    citizen.LabelConst.CLS_LOCALE_OTHER,
    citizen.LabelConst.CLS_PRT_FNT_TRIUMVIRATE,
    citizen.LabelConst.CLS_RT_NORMAL, 1, 1,
    citizen.LabelConst.CLS_PRT_FNT_SIZE_10,
    100, 100);

// Locale:JP
design.drawTextPtrFont("テキスト印刷(プリンタフォント)",
    citizen.LabelConst.CLS_LOCALE_JP,
    citizen.LabelConst.CLS_PRT_FNT_KANJI,
    citizen.LabelConst.CLS_RT_NORMAL, 1, 1,
    citizen.LabelConst.CLS_PRT_FNT_KANJI_SIZE_16, 100, 300);

// Locale:CN
design.drawTextPtrFont("测试打印",
    citizen.LabelConst.CLS_LOCALE_CN,
    citizen.LabelConst.CLS_PRT_FNT_KANJI,
    citizen.LabelConst.CLS_RT_NORMAL, 1, 1,
    citizen.LabelConst.CLS_PRT_FNT_KANJI_SIZE_16, 100, 300);

// Locale:KR
design.drawTextPtrFont("테스트 인쇄",
    citizen.LabelConst.CLS_LOCALE_KR,
    citizen.LabelConst.CLS_PRT_FNT_KANJI,
    citizen.LabelConst.CLS_RT_NORMAL, 1, 1,
    citizen.LabelConst.CLS_PRT_FNT_KANJI_SIZE_16, 100, 300);
```

5.6.3. drawTextDLFont メソッド

形式

drawTextDLFont (data, encoding, fontID, rotation, hexp, vexp, point, x, y)

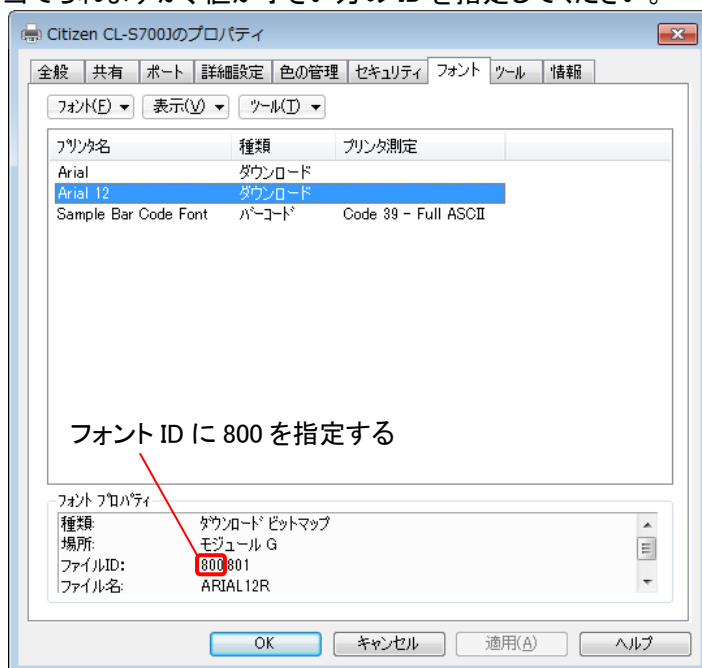
パラメータ

| 値 | [IN/OUT] | 意味 | 設定可能範囲 |
|----------|----------|------------------------|---|
| data | [IN] | 印字文字列 | ※文字コード変換できない文字は"?"に変換する。 |
| encoding | [IN] | 印字文字列の文字コード(定数) | 「5.7.2.パラメータ」の No9 を参照 |
| fontID | [IN] | プリンターにダウンロードしたフォントの ID | ・fontID が全て数字の場合 ビットマップダウンロードフォント ・fontID に数字以外が含まれる場合 TrueType ダウンロードフォント |
| rotation | [IN] | 回転方向(定数) | CLS_RT_NORMAL: 回転なし CLS_RT_RIGHT90: 右 90 度回転 CLS_RT_ROTATE180: 右 180 度回転 CLS_RT_LEFT90: 左 90 度回転 |
| hexp | [IN] | 水平拡大率 | 1～24 |
| vexp | [IN] | 垂直拡大率 | 1～24 |
| point | [IN] | 文字サイズ | 001～999 |
| x | [IN] | 印字位置(X 座標) | 0000～9999 |
| y | [IN] | 印字位置(Y 座標) | ※x,y は、左下を基準(0,0)として位置とする。 |

説明

プリンターにダウンロードしたフォントを使用して、回転・縦横拡大率・印字位置・文字サイズ等・指定条件で、入力された内容の文字を描画します。

ビットマップダウンロードフォントにつきましては、フォントをプリンターへダウンロードした際に 2 つの ID が割り当てられますが、値が小さい方の ID を指定してください。



戻り値

成功時は CLS_SUCCESS(0) を返します。それ以外のエラーコードは[「5.4 戻り値」](#)を参照してください。

使用例

```
// TrueTypeダウンロードフォント
design.drawTextDLFont("TrueType",
    citizen.LabelConst.CLS_ENC_CDPG_IBM850, "S50",
    citizen.LabelConst.CLS_RT_NORMAL, 1, 1, 12, 100, 100);

// ビットマップダウンロードフォント
design.drawTextDLFont("Bitmap",
    citizen.LabelConst.CLS_ENC_CDPG_IBM850, "800",
    citizen.LabelConst.CLS_RT_NORMAL, 1, 1, 12, 100, 200);
```

5.6.4. drawNVBitmap メソッド

形式

drawNVBitmap (name, hexp, vexp, x, y)

パラメータ

| 値 | [IN/OUT] | 意味 | 設定可能範囲 |
|------|----------|------------|--|
| name | [IN] | 画像ファイル名 | 以下の条件を除いた ASCII コードが使用可能 ・先頭がアンダースコアは不可 ・以下の文字は使用不可(禁則文字) ¥ / : * ? " < > |
| hexp | [IN] | 水平拡大率 | 1～24 |
| vexp | [IN] | 垂直拡大率 | 1～24 |
| x | [IN] | 印字位置(X 座標) | 0000～9999 |
| y | [IN] | 印字位置(Y 座標) | ※x,y は、左下を基準(0,0)として位置とする。 |

説明

縦横拡大率・印字位置等・指定条件で、プリンターに登録されている画像データを描画します。
指定された画像ファイルが、プリンターに登録されているかのチェックは行いません。

戻り値

成功時は CLS_SUCCESS(0) を返します。それ以外のエラーコードは[「5.4 戻り値」](#)を参照してください。

使用例

```
design.drawNVBitmap("Dice_1.bmp", 1, 1, 10, 10);
```

5.6.5. drawBitmap メソッド

形式

drawBitmap (image, rotation, width, height, x, y, resolution, measurementUnit)

パラメータ

| 値 | [IN/OUT] | 意味 | 設定可能範囲 |
|-----------------|----------|--------------|---|
| image | [IN] | イメージオブジェクト | |
| rotation | [IN] | 回転方向(定数) | CLS_RT_NORMAL: 回転なし CLS_RT_RIGHT90: 右 90 度回転 CLS_RT_ROTATE180: 右 180 度回転 CLS_RT_LEFT90: 左 90 度回転 |
| width | [IN] | 横サイズ(ピクセル単位) | |
| height | [IN] | 縦サイズ(ピクセル単位) | |
| x | [IN] | 印字位置(X 座標) | 0000～9999 |
| y | [IN] | 印字位置(Y 座標) | ※x,y は、左下を基準(0,0)として位置とする。 |
| resolution | [IN] | 解像度(dpi) | CLS_PRT_RES_203(203dpi) CLS_PRT_RES_300(300dpi) ※省略時は、CLS_PRT_RES_203 となる。 |
| measurementUnit | [IN] | プリンターの単位選択設定 | CLS_UNIT_MILLI CLS_UNIT_INCH ※省略時は、CLS_UNIT_INCH となる。 |

説明

回転・縦横拡大率・印字位置等の指定条件で、指定された画像ファイルを描画します。

本メソッドは内部的に、指定された画像ファイルをプリンターへ登録してから描画を行っていますが、このとき、プリンターのメモリ容量の問題等で、画像をプリンターへ登録可能であるか・登録されたかのチェックは行いません。

イメージの回転指定をする場合は、基準位置補正のため解像度と単位選択設定をプリンターに合わせて指定する必要があります。

描画可能なビットマップ形式は、BMP/GIF/EXIF/JPG/PNG/TIFF です。

描画される画像サイズは、指定された縦横サイズに合わせて、アスペクト比を維持した状態でリサイズします。

例: 以下の場合、リサイズ後の画像サイズは「200x50」ピクセルとなる。

画像サイズ: 400x100 ピクセル

横サイズ: 200

縦サイズ: 200

指定のサイズが 0 の場合、対象の画像サイズを基準とする。

例: 以下の場合、リサイズ後の画像サイズは「800x200」ピクセルとなる。

画像サイズ: 400x100 ピクセル

横サイズ: 0

縦サイズ: 200

戻り値

成功時は CLS_SUCCESS(0) を返します。それ以外のエラーコードは「[5.4 戻り値](#)」を参照してください。

使用例

```
var image = new Image();

// 画像読み込み処理は省略

design.drawBitmap(image,
    citizen.LabelConst.CLS_RT_RIGHT90, 0, 0, 10, 10,
    citizen.LabelConst.CLS_PRT_RES_203,
    citizen.LabelConst.CLS_UNIT_INCH);
```

5.6.6. drawBarcode メソッド

形式

drawBarcode (data, symbology, rotation, thick, narrow, height, x, y, showText)

パラメータ

| 値 | [IN/OUT] | 意味 | 設定可能範囲 |
|-----------|----------|------------------|---|
| data | [IN] | バーコードデータ文字列 | ASCII コード文字列 |
| symbology | [IN] | バーコードの種類(定数) | CLS_BCS_CODE39: Code 3 of 9 CLS_BCS_UPCA: UPC-A CLS_BCS_UPCE: UPC-E CLS_BCS_INTERLEAVED25: Interleaved 2 of 5 CLS_BCS_CODE128: Code 128 CLS_BCS_EAN13: EAN-13(JAN-13) CLS_BCS_EAN8: EAN-8(JAN-8) CLS_BCS_HIBC: HIBC CLS_BCS_CODABAR: CODABAR(NW-7) CLS_BCS_INT25: Int 2 of 5 CLS_BCS_PLESSEY: Plessey CLS_BCS_CASECODE: CASE CODE CLS_BCS_UPC2DIG: UPC 2DIG ADD CLS_BCS_UPC5DIG: UPC 5DIG ADD CLS_BCS_CODE93: Code93 CLS_BCS_ITF14: ITF-14 CLS_BCS_ZIP: ZIP CLS_BCS_ITF16: IFT-16 CLS_BCS_UCCEAN128: UCC/EAN-128 CLS_BCS_INDUSTRIAL25: Industrial 2 of 5 CLS_BCS_UCCEAN128KMART: UCC/EAN-128(for K-MART) CLS_BCS_COOP25: COOP 2 of 5 CLS_BCS_UCCEAN128RANDOMWEIGHT: UCC/EAN-128 Random Weight CLS_BCS_TELEPEN: Telepen |
| rotation | [IN] | 回転方向(定数) | CLS_RT_NORMAL: 回転なし CLS_RT_RIGHT90: 右 90 度回転 CLS_RT_ROTATE180: 右 180 度回転 CLS_RT_LEFT90: 左 90 度回転 |
| thick | [IN] | 太バーの幅 | 1～24 |
| narrow | [IN] | 細バーの幅 | 1～24 |
| height | [IN] | バーコードデータの高さ | 001～999 |
| x | [IN] | 印字位置(X 座標) | 0000～9999 ※x,y は、左下を基準(0,0)として位置とする。 |
| y | [IN] | 印字位置(Y 座標) | |
| showText | [IN] | バーコード文字の表示有無(定数) | CLS_BCS_TEXT_HIDE: 非表示 CLS_BCS_TEXT_SHOW: 表示 |

説明

このメソッドは、一次元のバーコードを描画します。また、バーコードに対して、バーコード種類・太/細バーの幅・バーコードデータの高さ・印字位置・バーコード文字の表示有無等の条件を指定することができます。

※バーコードの種類によって、thick/narrow 比率や、有効なバーコードデータ文字列が異なります。詳細は、ラベルプリンターのコマンドリファレンス「バーコードフィールドの定義」、「各バーコードの説明」の解説をご確認ください。

戻り値

成功時は CLS_SUCCESS(0) を返します。それ以外のエラーコードは[「5.4 戻り値」](#)を参照してください。

使用例

```
// Code39of9
var code39 = "ABC123456789";
design.drawBarCode(code39, citizen.LabelConst.CLS_BCS_CODE39,
    citizen.LabelConst.CLS_RT_NORMAL, 6, 2, 50, 10, 10,
    citizen.LabelConst.CLS_BCS_TEXT_SHOW);

// UPC-A
var upca = "01234567890";
design.drawBarCode(upca, citizen.LabelConst.CLS_BCS_UPCA,
    citizen.LabelConst.CLS_RT_NORMAL, 2, 2, 50, 10, 100,
    citizen.LabelConst.CLS_BCS_TEXT_SHOW);

// UPC-E
var upce = "123456";
design.drawBarCode(upce, citizen.LabelConst.CLS_BCS_UPCE,
    citizen.LabelConst.CLS_RT_NORMAL, 2, 2, 50, 10, 10,
    citizen.LabelConst.CLS_BCS_TEXT_SHOW);

// Interleaved2of5
var interleaved25 = "1234567890";
design.drawBarCode(interleaved25, citizen.LabelConst.CLS_BCS_INTERLEAVED25,
    citizen.LabelConst.CLS_RT_NORMAL, 5, 2, 50, 10, 100,
    citizen.LabelConst.CLS_BCS_TEXT_SHOW);

// Code128
var code128 = "1234567890";
design.drawBarCode(code128, citizen.LabelConst.CLS_BCS_CODE128,
    citizen.LabelConst.CLS_RT_NORMAL, 2, 4, 50, 10, 10,
    citizen.LabelConst.CLS_BCS_TEXT_SHOW);

// EAN-13
var ean13 = "123456789012";
design.drawBarCode(ean13, citizen.LabelConst.CLS_BCS_EAN13,
    citizen.LabelConst.CLS_RT_NORMAL, 3, 3, 50, 10, 100,
    citizen.LabelConst.CLS_BCS_TEXT_SHOW);

// EAN-8
var ean8 = "1234567";
design.drawBarCode(ean8, citizen.LabelConst.CLS_BCS_EAN8,
    citizen.LabelConst.CLS_RT_NORMAL, 4, 4, 50, 10, 10,
    citizen.LabelConst.CLS_BCS_TEXT_SHOW);

// HIBC
var hibi = "1234567890";
design.drawBarCode(hibi, citizen.LabelConst.CLS_BCS_HIBC,
    citizen.LabelConst.CLS_RT_NORMAL, 6, 2, 50, 10, 100,
    citizen.LabelConst.CLS_BCS_TEXT_SHOW);
```

```
// CODABAR
var codabar = "a1234567890b";
design.drawBarCode(codabar, citizen.LabelConst.CLS_BCS_CODABAR,
    citizen.LabelConst.CLS_RT_NORMAL, 6, 2, 40, 10, 10,
    citizen.LabelConst.CLS_BCS_TEXT_SHOW);

// Interleaved2of5 W/BARS
var int25 = "1234567890";
design.drawBarCode(int25, citizen.LabelConst.CLS_BCS_INT25,
    citizen.LabelConst.CLS_RT_NORMAL, 5, 2, 50, 10, 100,
    citizen.LabelConst.CLS_BCS_TEXT_SHOW);

// PLESSEY
var plessey = "1234567890";
design.drawBarCode(plessey, citizen.LabelConst.CLS_BCS_PLESSEY,
    citizen.LabelConst.CLS_RT_NORMAL, 4, 2, 50, 10, 10,
    citizen.LabelConst.CLS_BCS_TEXT_SHOW);

// CASE CODE
var cascode = "1234567890123";
design.drawBarCode(cascode, citizen.LabelConst.CLS_BCS_CASECODE,
    citizen.LabelConst.CLS_RT_NORMAL, 5, 2, 50, 10, 100,
    citizen.LabelConst.CLS_BCS_TEXT_SHOW);

// UPC 2DIG ADD
var upca = "01234567890";
var upc2dig = "12";

design.drawBarCode(upca, citizen.LabelConst.CLS_BCS_UPCA,
    citizen.LabelConst.CLS_RT_NORMAL, 2, 2, 50, 10, 100,
    citizen.LabelConst.CLS_BCS_TEXT_SHOW);

design.drawBarCode(upc2dig, citizen.LabelConst.CLS_BCS_UPC2DIG,
    citizen.LabelConst.CLS_RT_NORMAL, 4, 2, 50, 130, 100,
    citizen.LabelConst.CLS_BCS_TEXT_SHOW);

// UPC 5DIG ADD
var upca = "01234567890";
var upc5dig = "12345";

design.drawBarCode(upca, citizen.LabelConst.CLS_BCS_UPCA,
    citizen.LabelConst.CLS_RT_NORMAL, 2, 2, 50, 10, 100,
    citizen.LabelConst.CLS_BCS_TEXT_SHOW);

design.drawBarCode(upc5dig, citizen.LabelConst.CLS_BCS_UPC5DIG,
    citizen.LabelConst.CLS_RT_NORMAL, 4, 2, 50, 130, 100,
    citizen.LabelConst.CLS_BCS_TEXT_SHOW);

// Code93
var code93 = "1234ABCD";
design.drawBarCode(code93, citizen.LabelConst.CLS_BCS_CODE93,
    citizen.LabelConst.CLS_RT_NORMAL, 6, 6, 50, 10, 10,
    citizen.LabelConst.CLS_BCS_TEXT_SHOW);

// ITF-14
var itf14 = "1234567890123";
design.drawBarCode(itf14, citizen.LabelConst.CLS_BCS_ITF14,
    citizen.LabelConst.CLS_RT_NORMAL, 5, 2, 50, 10, 100,
    citizen.LabelConst.CLS_BCS_TEXT_SHOW);
```

```
// ZIP
var zip = "123456789";
design.drawBarCode(zip, citizen.LabelConst.CLS_BCS_ZIP,
    citizen.LabelConst.CLS_RT_NORMAL, 1, 1, 10, 10, 10,
    citizen.LabelConst.CLS_BCS_TEXT_SHOW);

// ITF-16
var itf16 = "123456789012345";
design.drawBarCode(itf16, citizen.LabelConst.CLS_BCS_ITF16,
    citizen.LabelConst.CLS_RT_NORMAL, 5, 2, 50, 10, 100,
    citizen.LabelConst.CLS_BCS_TEXT_SHOW);

// UCC/EAN-128
var uccean128 = "1234567890123456789";
design.drawBarCode(uccean128, citizen.LabelConst.CLS_BCS_UCCEAN128,
    citizen.LabelConst.CLS_RT_NORMAL, 4, 4, 50, 10, 10,
    citizen.LabelConst.CLS_BCS_TEXT_SHOW);

// Industrial2of5
var industrial25 = "1234567890";
design.drawBarCode(industrial25, citizen.LabelConst.CLS_BCS_INDUSTRIAL25,
    citizen.LabelConst.CLS_RT_NORMAL, 5, 2, 50, 10, 100,
    citizen.LabelConst.CLS_BCS_TEXT_SHOW);

// UCC/EAN-128(for K-MART)
var uccean128kmart = "123456789012345678";
design.drawBarCode(uccean128kmart,
    citizen.LabelConst.CLS_BCS_UCCEAN128KMART,
    citizen.LabelConst.CLS_RT_NORMAL, 3, 3, 50, 10, 10,
    citizen.LabelConst.CLS_BCS_TEXT_SHOW);

// COOP 2 of 5
var coop25 = "0123456789";
design.drawBarCode(coop25, citizen.LabelConst.CLS_BCS_COOP25,
    citizen.LabelConst.CLS_RT_NORMAL, 10, 4, 50, 10, 100,
    citizen.LabelConst.CLS_BCS_TEXT_SHOW);

// UCC/EAN-128 Random Weight
var uccean128randomweight = "1234567890123456789012345678909999";
design.drawBarCode(uccean128randomweight,
    citizen.LabelConst.CLS_BCS_UCCEAN128RANDOMWEIGHT,
    citizen.LabelConst.CLS_RT_NORMAL, 2, 2, 50, 10, 10,
    citizen.LabelConst.CLS_BCS_TEXT_SHOW);

// Telepen
var telepen = "1234567890";
design.drawBarCode(telepen, citizen.LabelConst.CLS_BCS_TELEPEN,
    citizen.LabelConst.CLS_RT_NORMAL, 2, 2, 50, 10, 100,
    citizen.LabelConst.CLS_BCS_TEXT_SHOW);
```

5.6.7. drawMaxiCode メソッド

形式

drawMaxiCode (arryData, rotation, x, y)

パラメータ

| 値 | [IN/OUT] | 意味 | 設定可能範囲 |
|----------|----------|-------------|---|
| arryData | [IN] | バーコードデータ文字列 | ASCII コード文字列 |
| rotation | [IN] | 回転方向(定数) | CLS_RT_NORMAL: 回転なし CLS_RT_RIGHT90: 右 90 度回転 CLS_RT_ROTATE180: 右 180 度回転 CLS_RT_LEFT90: 左 90 度回転 |
| x | [IN] | 印字位置(X 座標) | 0000～9999 ※x,y は、左下を基準(0,0)として位置とする。 |
| y | [IN] | 印字位置(Y 座標) | |

説明

このメソッドは、UPS MaxiCode 形式のバーコードを描画します。また、バーコードに対して、回転・印字位置の条件を指定することができます。

バーコードデータ文字列には、以下 5 つの要素を、①から順にセットしてください。

- ① 5 桁の Zip コード
- ② 4 桁の+4Zip コード
- ③ 3 桁の国別コード
- ④ 3 桁の class of service code
- ⑤ 84 桁以内のデータ文字列

戻り値

成功時は CLS_SUCCESS(0) を返します。それ以外のエラーコードは[「5.4 戻り値」](#)を参照してください。

使用例

```
var data = new Array(5);  
data[0] = "90501";  
data[1] = "6282";  
data[2] = "840";  
data[3] = "001";  
data[4] = "1Z00004951";  
design.drawMaxiCode(data, citizen.LabelConst.CLS_RT_NORMAL, 100, 0);
```

5.6.8. drawPDF417 メソッド

形式

drawPDF417 (data, encoding, rotation, exp, ecLevel, x, y)

パラメータ

| 値 | [IN/OUT] | 意味 | 設定可能範囲 |
|----------|----------|--------------------|--|
| data | [IN] | バーコードデータ文字列 | ※指定の文字コードで使用可能な文字 |
| encoding | [IN] | バーコードデータの文字コード(定数) | 「5.7.2.パラメータ」 の No9 を参照 |
| rotation | [IN] | 回転方向(定数) | CLS_RT_NORMAL: 回転なし CLS_RT_RIGHT90: 右 90 度回転 CLS_RT_ROTATE180: 右 180 度回転 CLS_RT_LEFT90: 左 90 度回転 |
| exp | [IN] | 拡大率 | 1～5 |
| ecLevel | [IN] | エラー修正レベル(定数) | CLS_PDF417_EC_LEVEL_0: レベル 0 CLS_PDF417_EC_LEVEL_1: レベル 1 CLS_PDF417_EC_LEVEL_2: レベル 2 CLS_PDF417_EC_LEVEL_3: レベル 3 CLS_PDF417_EC_LEVEL_4: レベル 4 CLS_PDF417_EC_LEVEL_5: レベル 5 CLS_PDF417_EC_LEVEL_6: レベル 6 CLS_PDF417_EC_LEVEL_7: レベル 7 CLS_PDF417_EC_LEVEL_8: レベル 8 |
| x | [IN] | 印字位置(X 座標) | 0000～9999 ※x,y は、左下を基準(0,0)として位置とする。 |
| y | [IN] | 印字位置(Y 座標) | |

説明

このメソッドは、PDF-417 形式のバーコードを描画します。また、バーコードに対して、エンコード・回転・拡大率・エラー修正レベル・印字位置の条件を指定することができます。

戻り値

成功時は CLS_SUCCESS(0) を返します。それ以外のエラーコードは[「5.4 戻り値」](#)を参照してください。

使用例

```
design.drawPDF417("0123456789",
    citizen.LabelConst.CLS_ENC_CDPG_IBM850,
    citizen.LabelConst.CLS_RT_NORMAL, 3,
    citizen.LabelConst.CLS_PDF417_EC_LEVEL_0, 0, 0);
```

5.6.9. drawDataMatrix メソッド

形式

drawDataMatrix (data, encoding, rotation, exp, ecLevel, x, y)

パラメータ

| 値 | [IN/OUT] | 意味 | 設定可能範囲 |
|----------|----------|--------------------|---|
| data | [IN] | バーコードデータ文字列 | ・数字のみの場合、3116 文字以内 ・英数字混在の場合、2335 文字以内 ※指定の文字コードで使用可能な文字 |
| encoding | [IN] | バーコードデータの文字コード(定数) | 「5.7.2.パラメータ」 の No9 を参照 |
| rotation | [IN] | 回転方向(定数) | CLS_RT_NORMAL: 回転なし CLS_RT_RIGHT90: 右 90 度回転 CLS_RT_ROTATE180: 右 180 度回転 CLS_RT_LEFT90: 左 90 度回転 |
| exp | [IN] | 拡大率 | 1～15 |
| ecLevel | [IN] | エラー修正レベル(定数) | CLS_DATAMATRIX_EC_LEVEL_200: 200 |
| x | [IN] | 印字位置(X 座標) | 0000～9999 ※x,y は、左下を基準(0,0)として位置とする。 |
| y | [IN] | 印字位置(Y 座標) | |

説明

このメソッドは、DataMatrix 形式のバーコードを描画します。また、バーコードに対して、エンコード・回転・拡大率・印字位置の条件を指定することができます。

戻り値

成功時は CLS_SUCCESS(0) を返します。それ以外のエラーコードは[「5.4 戻り値」](#)を参照してください。

使用例

```
design.drawDataMatrix("0123456789",
    citizen.LabelConst.CLS_ENC_CDPG_IBM850,
    citizen.LabelConst.CLS_RT_NORMAL, 15,
    citizen.LabelConst.CLS_DATAMATRIX_EC_LEVEL_200, 0, 0);
```


5.6.10. drawQRCode メソッド

形式

drawQRCode (data, encoding, rotation, exp, ecLevel, x, y)

パラメータ

| 値 | [IN/OUT] | 意味 | 設定可能範囲 |
|----------|----------|--------------------|---|
| data | [IN] | バーコードデータ文字列 | ※指定の文字コードで使用可能な文字 |
| encoding | [IN] | バーコードデータの文字コード(定数) | 「5.7.2.パラメータ」 の No9 を参照 |
| rotation | [IN] | 回転方向(定数) | CLS_RT_NORMAL: 回転なし CLS_RT_RIGHT90: 右 90 度回転 CLS_RT_ROTATE180: 右 180 度回転 CLS_RT_LEFT90: 左 90 度回転 |
| exp | [IN] | 拡大率 | 1～15 |
| ecLevel | [IN] | エラー修正レベル(定数) | CLS_QRCODE_EC_LEVEL_L: レベル L(7%) CLS_QRCODE_EC_LEVEL_M: レベル M(15%) CLS_QRCODE_EC_LEVEL_Q: レベル Q(25%) CLS_QRCODE_EC_LEVEL_H: レベル H(30%) |
| x | [IN] | 印字位置(X 座標) | 0000～9999 ※x,y は、左下を基準(0,0)として位置とする。 |
| y | [IN] | 印字位置(Y 座標) | |

説明

このメソッドは、QR コードを描画します。また、QR コードに対して、エンコード・回転・拡大率・エラー修正レベル・印字位置の条件を指定することができます。

戻り値

成功時は CLS_SUCCESS(0) を返します。それ以外のエラーコードは[「5.4 戻り値」](#)を参照してください。

使用例

```
design.drawQRCode("アイエオ12345",  
    citizen.LabelConst.CLS_ENC_CDPG_SHIFT_JIS,  
    citizen.LabelConst.CLS_RT_NORMAL, 10,  
    citizen.LabelConst.CLS_QRCODE_EC_LEVEL_H, 100, 100);
```

5.6.11. drawAztec メソッド

形式

drawAztec (data, encoding, rotation, exp, ecLevel, x, y)

パラメータ

| 値 | [IN/OUT] | 意味 | 設定可能範囲 |
|----------|----------|--------------------|---|
| data | [IN] | バーコードデータ文字列 | ※指定の文字コードで使用可能な文字 |
| encoding | [IN] | バーコードデータの文字コード(定数) | 「5.7.2.パラメータ」 の No9 を参照 |
| rotation | [IN] | 回転方向(定数) | CLS_RT_NORMAL: 回転なし CLS_RT_RIGHT90: 右 90 度回転 CLS_RT_ROTATE180: 右 180 度回転 CLS_RT_LEFT90: 左 90 度回転 |
| exp | [IN] | 拡大率 | 1～15 |
| ecLevel | [IN] | エラー修正レベル(定数) | CLS_AXTEC_EC_LEVEL_000: レベル 0(誤り訂正率 23%) |
| x | [IN] | 印字位置(X 座標) | 0000～9999 ※x,y は、左下を基準(0,0)として位置とする。 |
| y | [IN] | 印字位置(Y 座標) | |

説明

このメソッドは、Aztec 形式のバーコードを描画します。また、バーコードに対して、エンコード・回転・拡大率・印字位置の条件を指定することができます。

戻り値

成功時は CLS_SUCCESS(0) を返します。それ以外のエラーコードは[「5.4 戻り値」](#)を参照してください。

使用例

```
design.drawAztec("0123456789",
    citizen.LabelConst.CLS_ENC_CDPG_IBM850,
    citizen.LabelConst.CLS_RT_NORMAL, 10,
    citizen.LabelConst.CLS_AXTEC_EC_LEVEL_000, 0, 0);
```

5.6.12. drawGS1DataBar メソッド

形式

drawGS1DataBar (arrayData, type, rotation, exp, x, y)

パラメータ

| 値 | [IN/OUT] | 意味 | 設定可能範囲 |
|-----------|----------|--------------|--|
| arrayData | [IN] | バーコードデータ文字列 | ASCII コード文字列 |
| type | [IN] | バーコードタイプ(定数) | CLS_GS1_DATABAR_OMNI_DIRECTIONAL: Omni-directional CLS_GS1_DATABAR_COMPOSITE: Composite CLS_GS1_DATABAR_TRUNCATION: Truncation CLS_GS1_DATABAR_STACKED: Stacked CLS_GS1_DATABAR_STACKED_OMNI_DIRECTIONAL: Stacked Omni-directional CLS_GS1_DATABAR_LIMITED: Limited CLS_GS1_DATABAR_EXPANDED: Expanded |
| rotation | [IN] | 回転方向(定数) | CLS_RT_NORMAL: 回転なし CLS_RT_RIGHT90: 右 90 度回転 CLS_RT_ROTATE180: 右 180 度回転 CLS_RT_LEFT90: 左 90 度回転 |
| exp | [IN] | 拡大率 | 1～15 |
| x | [IN] | 印字位置(X 座標) | 0000～9999 |
| y | [IN] | 印字位置(Y 座標) | ※x,y は、左下を基準(0,0)として位置とする。 |

説明

このメソッドは、GS1 DataBar 形式のバーコードを描画します。また、バーコードに対して、バーコードタイプ・回転・拡大率・印字位置等の条件を指定することができます。

※バーコードの種類によって、指定可能なバーコードデータ文字列などが異なります。詳細は、ラベルプリンターのコマンドリファレンス「バーコード W1k(国内モデル、海外モデル共通): GS1 DataBar (RSS)」の解説をご確認ください。

戻り値

成功時は CLS_SUCCESS(0) を返します。それ以外のエラーコードは「[5.4 戻り値](#)」を参照してください。

使用例

```
// GS1 DataBar Omni-Directional
var omnidirectional = new Array("1234567890123");
design.drawGS1DataBar(omnidirectional,
    citizen.LabelConst.CLS_GS1_DATABAR_OMNI_DIRECTIONAL,
    citizen.LabelConst.CLS_RT_NORMAL, 3, 10, 10);

// GS1 DataBar Composite
var composit = new Array(2);
composit[0] = "1234567890123";
composit[1] = "1234567890-07/07/07";
design.drawGS1DataBar(composit,
    citizen.LabelConst.CLS_GS1_DATABAR_COMPOSITE,
    citizen.LabelConst.CLS_RT_NORMAL, 3, 10, 10);
```

```
// GS1 DataBar Truncation
var truncation = new Array("1234567890123");
design.drawGS1DataBar(truncation,
    citizen.LabelConst.CLS_GS1_DATABAR_TRUNCATION,
    citizen.LabelConst.CLS_RT_NORMAL, 3, 10, 10);

// GS1 DataBar Stacked
var stacked = new Array("1234567890123");
design.drawGS1DataBar(stacked,
    citizen.LabelConst.CLS_GS1_DATABAR_STACKED,
    citizen.LabelConst.CLS_RT_NORMAL, 3, 10, 10);

// GS1 DataBar Stacked-Omni-Directional
var stackedOD = new Array("1234567890123");
design.drawGS1DataBar(stackedOD,
    citizen.LabelConst.CLS_GS1_DATABAR_STACKED_OMNI_DIRECTIONAL,
    citizen.LabelConst.CLS_RT_NORMAL, 3, 10, 10);

// GS1 DataBar Limited
var limited = new Array("1234567890123");
design.drawGS1DataBar(limited,
    citizen.LabelConst.CLS_GS1_DATABAR_LIMITED,
    citizen.LabelConst.CLS_RT_NORMAL, 3, 10, 10);

// GS1 DataBar Expanded
var expanded = new Array("041234567890123");
design.drawGS1DataBar(expanded,
    citizen.LabelConst.CLS_GS1_DATABAR_EXPANDED,
    citizen.LabelConst.CLS_RT_NORMAL, 3, 10, 10);
```

5.6.13. drawLine メソッド

形式

drawLine (x1, y1, x2, y2, thickness)

パラメータ

| 値 | [IN/OUT] | 意味 | 設定可能範囲 |
|-----------|----------|------------------|---|
| x1 | [IN] | 印字開始位置(X 座標、中心点) | 0000～9999 ※x,y は、左下を基準(0,0)として位置とする。 |
| y1 | [IN] | 印字開始位置(Y 座標、中心点) | |
| x2 | [IN] | 印字終了位置(X 座標、中心点) | |
| y2 | [IN] | 印字終了位置(Y 座標、中心点) | |
| thickness | [IN] | 線幅(中心点を基準) | 0000～9999 |

説明

設定された幅の線を描画します。

印字開始・終了位置は、中心線の座標を示します。線幅を考慮した線の座標が、0～9999 の範囲外になる場合、CLS_E_ILLEGAL(1101)エラーになります。

戻り値

成功時は CLS_SUCCESS(0) を返します。それ以外のエラーコードは「[5.4 戻り値](#)」を参照してください。

使用例

```
// 印字領域が重なる部分を上書きに指定
print.setFormatAttribute(1);

// 縦線
design.drawLine(20, 30, 20, 300, 10);

// 横線
design.drawLine(16, 34, 200, 34, 10);
```

5.6.14. drawRect メソッド

形式

drawRect (x, y, width, height, thickness)

パラメータ

| 値 | [IN/OUT] | 意味 | 設定可能範囲 |
|-----------|----------|--------------|---|
| x | [IN] | 印字開始位置(X 座標) | 0000～9999 ※x,y は、左下を基準(0,0)として位置とする。 |
| y | [IN] | 印字開始位置(Y 座標) | |
| width | [IN] | 水平幅 | 0000～9999 |
| height | [IN] | 垂直幅 | |
| thickness | [IN] | 線幅 | 0000～9999 |

説明

設定された寸法の矩形を描画します。

印字開始位置は、線の外周の左下を示します。線幅を太くした際は、印字開始位置から内側に線が太くなります。

戻り値

成功時は CLS_SUCCESS(0) を返します。それ以外のエラーコードは「[5.4 戻り値](#)」を参照してください。

使用例

```
design.drawRect(20, 30, 180, 280, 10);
```

5.6.15. fillRect メソッド

形式

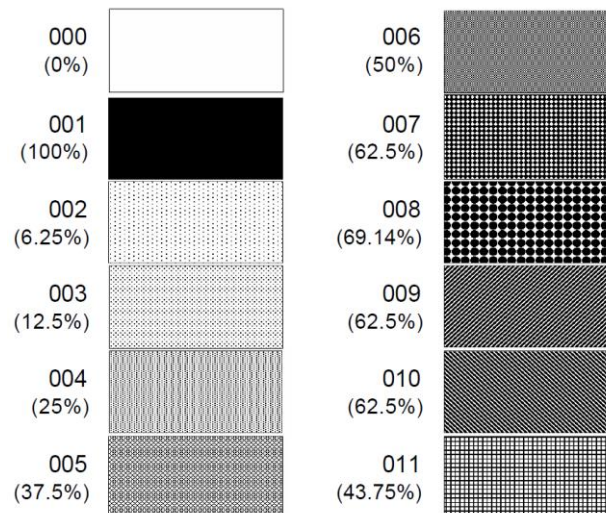
fillRect (x, y, width, height, pattern)

パラメータ

| 値 | [IN/OUT] | 意味 | 設定可能範囲 |
|---------|----------|--------------|--|
| x | [IN] | 印字開始位置(X 座標) | 0000～9999 ※x,y は、左下を基準(0,0)として位置とする。 |
| y | [IN] | 印字開始位置(Y 座標) | |
| width | [IN] | 水平幅 | 0000～9999 |
| height | [IN] | 垂直幅 | |
| pattern | [IN] | 網掛けパターン(定数) | CLS_SHADED_PTN_0: 網掛けパターン 000 CLS_SHADED_PTN_1: 網掛けパターン 001 CLS_SHADED_PTN_2: 網掛けパターン 002 CLS_SHADED_PTN_3: 網掛けパターン 003 CLS_SHADED_PTN_4: 網掛けパターン 004 CLS_SHADED_PTN_5: 網掛けパターン 005 CLS_SHADED_PTN_6: 網掛けパターン 006 CLS_SHADED_PTN_7: 網掛けパターン 007 CLS_SHADED_PTN_8: 網掛けパターン 008 CLS_SHADED_PTN_9: 網掛けパターン 009 CLS_SHADED_PTN_10: 網掛けパターン 010 CLS_SHADED_PTN_11: 網掛けパターン 011 |

説明

設定された寸法の矩形を描画し、設定されたパターンで内部を網掛けします。



戻り値

成功時は CLS_SUCCESS(0) を返します。それ以外のエラーコードは「[5.4 戻り値](#)」を参照してください。

使用例

```
design.fillRect(20, 30, 180, 280, citizen.LabelConst.CLS_SHADED_PTN_10);
```

5.6.16. drawCircle メソッド

形式

drawCircle (x, y, radius)

パラメータ

| 値 | [IN/OUT] | 意味 | 設定可能範囲 |
|--------|----------|------------------|---|
| x | [IN] | 印字開始位置(X 座標、中心点) | 0000～9999 ※x,y は、左下を基準(0,0)として位置とする。 |
| y | [IN] | 印字開始位置(Y 座標、中心点) | |
| radius | [IN] | 円の半径 | 0000～0398 |

説明

設定された中心と半径で円を描画します。

戻り値

成功時は CLS_SUCCESS(0) を返します。それ以外のエラーコードは[「5.4 戻り値」](#)を参照してください。

使用例

```
design.drawCircle(50, 50, 15);
```


5.6.17. fillCircle メソッド

形式

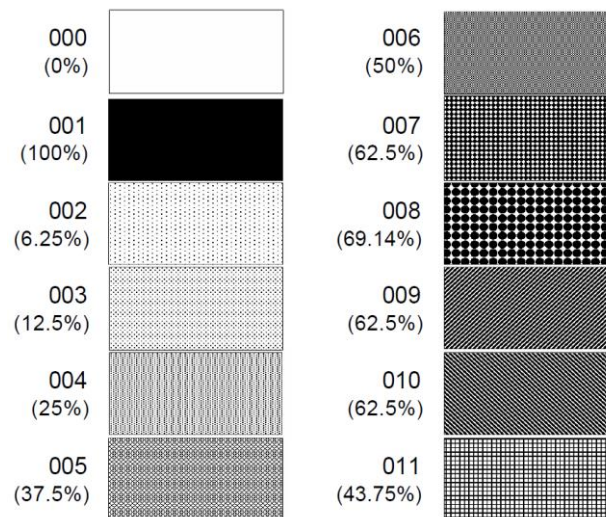
fillCircle (x, y, radius, pattern)

パラメータ

| 値 | [IN/OUT] | 意味 | 設定可能範囲 |
|---------|----------|------------------|--|
| x | [IN] | 印字開始位置(X 座標、中心点) | 0000～9999 ※x,y は、左下を基準(0,0)として位置とする。 |
| y | [IN] | 印字開始位置(Y 座標、中心点) | |
| radius | [IN] | 円の半径 | 0000～0398 |
| pattern | [IN] | 網掛けパターン(定数) | CLS_SHADED_PTN_0: 網掛けパターン 000 CLS_SHADED_PTN_1: 網掛けパターン 001 CLS_SHADED_PTN_2: 網掛けパターン 002 CLS_SHADED_PTN_3: 網掛けパターン 003 CLS_SHADED_PTN_4: 網掛けパターン 004 CLS_SHADED_PTN_5: 網掛けパターン 005 CLS_SHADED_PTN_6: 網掛けパターン 006 CLS_SHADED_PTN_7: 網掛けパターン 007 CLS_SHADED_PTN_8: 網掛けパターン 008 CLS_SHADED_PTN_9: 網掛けパターン 009 CLS_SHADED_PTN_10: 網掛けパターン 010 CLS_SHADED_PTN_11: 網掛けパターン 011 |

説明

設定された中心と半径で円を描画し、設定されたパターンで内部を網掛けします。



戻り値

成功時は CLS_SUCCESS(0) を返します。それ以外のエラーコードは[「5.4 戻り値」](#)を参照してください。

使用例

```
design.fillCircle(100, 100, 40, citizen.LabelConst.CLS_SHADED_PTN_2);
```

5.6.18. drawPolygon メソッド

形式

drawPolygon (arrayX, arrayY)

パラメータ

| 値 | [IN/OUT] | 意味 | 設定可能範囲 |
|--------|----------|------------|---|
| arrayX | [IN] | 印字位置(X 座標) | 0000～9999 ※x,y は、左下を基準(0,0)として位置とする。 |
| arrayY | [IN] | 印字位置(Y 座標) | |

説明

設定されたポイントで多角形を描画します。多角形の頂点の数だけ、x と y の座標を指定する。x と y の組み合わせは、最低 3 つ必要となり、3 つ未満のときは、CLS_E_ILLEGAL(1101)エラーとなる。

戻り値

成功時は CLS_SUCCESS(0) を返します。それ以外のエラーコードは「[5.4 戻り値](#)」を参照してください。

使用例

```
var arrayX = new Array(100, 200, 250, 150);
var arrayY = new Array(100, 100, 200, 200);

// 平行四辺形を描画
design.drawPolygon(arrayX, arrayY);
```

5.6.19. fillPolygon メソッド

形式

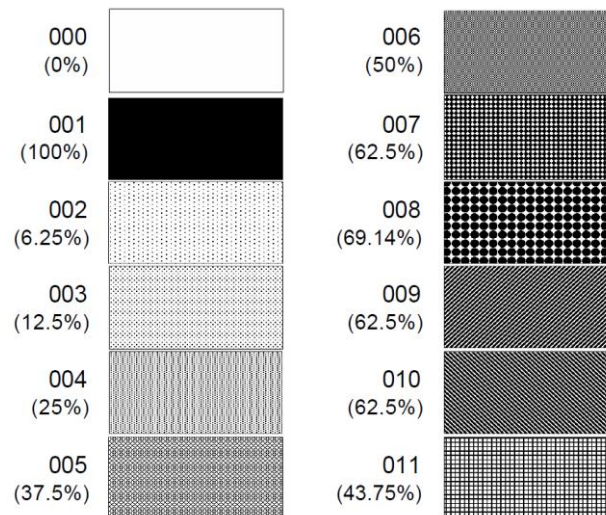
fillPolygon (arrayX, arrayY, pattern)

パラメータ

| 値 | [IN/OUT] | 意味 | 設定可能範囲 |
|---------|----------|-------------|--|
| arrayX | [IN] | 印字位置(X 座標) | 0000～9999 ※x,y は、左下を基準(0,0)として位置とする。 |
| arrayY | [IN] | 印字位置(Y 座標) | |
| pattern | [IN] | 網掛けパターン(定数) | CLS_SHADED_PTN_0: 網掛けパターン 000 CLS_SHADED_PTN_1: 網掛けパターン 001 CLS_SHADED_PTN_2: 網掛けパターン 002 CLS_SHADED_PTN_3: 網掛けパターン 003 CLS_SHADED_PTN_4: 網掛けパターン 004 CLS_SHADED_PTN_5: 網掛けパターン 005 CLS_SHADED_PTN_6: 網掛けパターン 006 CLS_SHADED_PTN_7: 網掛けパターン 007 CLS_SHADED_PTN_8: 網掛けパターン 008 CLS_SHADED_PTN_9: 網掛けパターン 009 CLS_SHADED_PTN_10: 網掛けパターン 010 CLS_SHADED_PTN_11: 網掛けパターン 011 |

説明

設定されたポイントで多角形を描画し、設定されたパターンで内部を網掛けします。多角形の頂点の数だけ、x と y の座標を指定する。x と y の組み合わせは、最低 3 つ必要となり、3 つ未満のときは、CLS_E_ILLEGAL(1101)エラーとなる。



戻り値

成功時は CLS_SUCCESS(0) を返します。それ以外のエラーコードは[「5.4 戻り値」](#)を参照してください。

使用例

```
var arrayX = new Array(100, 200, 250, 150);  
var arrayY = new Array(100, 100, 200, 200);  
  
// 平行四辺形を描画  
design.fillPolygon(arrayX, arrayY, citizen.LabelConst.CLS_SHADED_PTN_3);
```

5.6.20. embedRawDesignCommand メソッド

形式

embedRawDesignCommand (data)

パラメータ

| 値 | [IN/OUT] | 意味 | 設定可能範囲 |
|------|----------|-------------|--------|
| data | [IN] | ラベルデザインコマンド | |

説明

このメソッドは、ラベルデザインコマンドとして、バイトデータを挿入することができます。
ご使用の際は、他のメソッドに影響を与えない様に注意する必要があります。

戻り値

成功時は CLS_SUCCESS(0) を返します。それ以外のエラーコードは[「5.4 戻り値」](#)を参照してください。

使用例

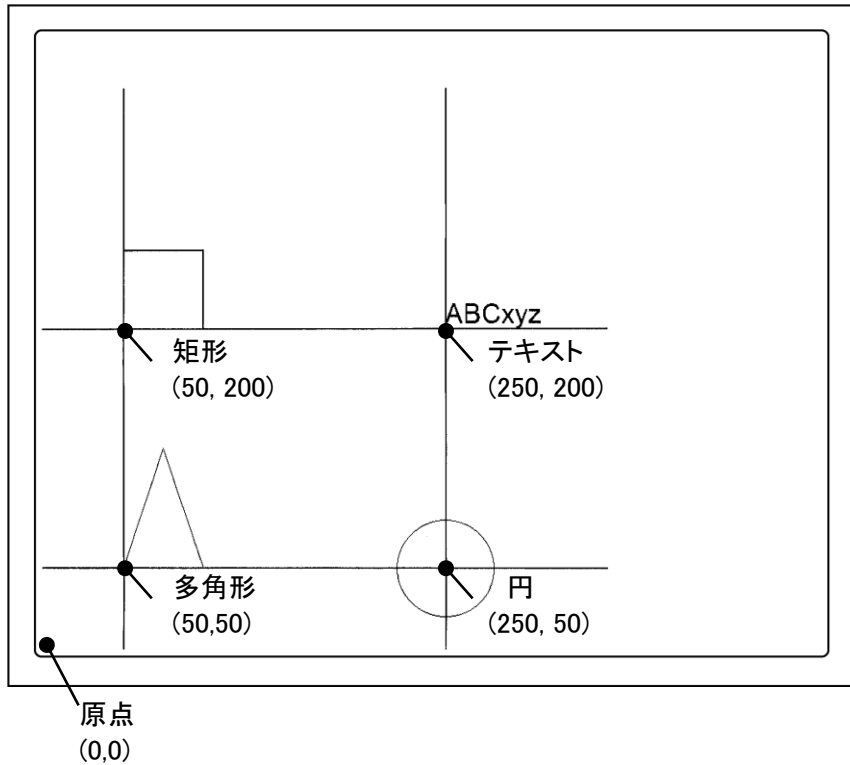
「drawRect(100, 200, 300, 250, 100)」と同様のコマンドを、本メソッドを使用して挿入するとき。

```
var hexdata = new Array(  
    0x31, 0x58, 0x31, 0x31, 0x30, 0x30, 0x30, 0x30, 0x32, 0x30,  
    0x30, 0x30, 0x31, 0x30, 0x30, 0x62, 0x30, 0x33, 0x30, 0x30,  
    0x30, 0x32, 0x35, 0x30, 0x30, 0x31, 0x30, 0x30, 0x30, 0x31,  
    0x30, 0x30, 0x0D, 0x0A );  
  
design.embedRawDesignCommand(hexdata);
```

5.7. 補足

5.7.1. 印字位置指定

ラベルにバーコードや文字を印字する場合の位置はラベルの左下が原点となり、原点からの距離を用いて印字位置指定を行います。原点より上方向の距離を Y 座標、右方向の距離を X 座標といいます。



使用例

```
// 多角形
arrayX = new Array(50, 75, 100);
arrayY = new Array(50, 125, 50);
design.drawPolygon(arrayX, arrayY);

// 矩形
design.drawRect(50, 200, 50, 50, 1);

// 円
design.drawCircle(250, 50, 30);

// テキスト
design.drawTextPtrFont("ABCxyz", citizen.LabelConst.CLS_LOCALE_JP,
    citizen.LabelConst.CLS_PRT_FNT_5, citizen.LabelConst.CLS_RT_NORMAL,
    1, 1, citizen.LabelConst.CLS_PRT_FNT_SIZE_6, 250, 200);

// 罫線
design.drawLine(0, 200, 350, 200, 1); // 罫線1
design.drawLine(50, 0, 50, 350, 1); // 罫線2
design.drawLine(0, 50, 350, 50, 1); // 罫線3
design.drawLine(250, 0, 250, 350, 1); // 罫線4
```

5.7.2. パラメータ

| No | 項目 | 定数名 | 型 | 値 | 説明 |
|----|-----------------|---------------------------|-----|--------|----------------------------|
| 1 | 処理結果 | CLS_SUCCESS | int | 0 | 正常終了 |
| | | CLS_E_ILLEGAL | int | 1101 | 未対応処理または無効パラメータ |
| 2 | プロパティ デフォルト値 | CLS_PROPERTY_DEFAULT | int | 999999 | プロパティのデフォルト値 |
| 3 | プリンタのステータス | CLS_STS_NO | int | 0 | ステータス:NO |
| | | CLS_STS_YES | int | 1 | ステータス:YES |
| 4 | ロケール | CLS_LOCALE_JP | int | 0 | 日本モデル用ロケール |
| | | CLS_LOCALE_OTHER | int | 1 | 海外モデル用ロケール |
| | | CLS_LOCALE_CN | int | 2 | 中国モデル用ロケール |
| | | CLS_LOCALE_KR | int | 3 | 韓国モデル用ロケール |
| 5 | 文字種 | CLS_PRT_FNT_0 | int | 0 | システムフォント:0 |
| | | CLS_PRT_FNT_1 | int | 1 | システムフォント:1 |
| | | CLS_PRT_FNT_2 | int | 2 | システムフォント:2 |
| | | CLS_PRT_FNT_3 | int | 3 | システムフォント:3 |
| | | CLS_PRT_FNT_4 | int | 4 | システムフォント:4 |
| | | CLS_PRT_FNT_5 | int | 5 | システムフォント:5 |
| | | CLS_PRT_FNT_6 | int | 6 | システムフォント:6 |
| | | CLS_PRT_FNT_7 | int | 7 | システムフォント:7 |
| | | CLS_PRT_FNT_8 | int | 8 | システムフォント:8 |
| | | CLS_PRT_FNT_TRIUMVIRATE | int | 9 | スムーズフォント(Triumvirate) |
| | | CLS_PRT_FNT_TRIUMVIRATE_B | int | 10 | スムーズフォント(Triumvirate Bold) |
| | | CLS_PRT_FNT_KANJI | int | 11 | 漢字(横書き) |
| | | CLS_PRT_FNT_KANJIT | int | 12 | 漢字(縦書き) |
| 6 | 文字サイズ | CLS_PRT_FNT_SIZE_4 | int | 0 | 文字種ポイント数(4pt) |
| | | CLS_PRT_FNT_SIZE_5 | int | 1 | 文字種ポイント数(5pt) |
| | | CLS_PRT_FNT_SIZE_6 | int | 2 | 文字種ポイント数(6pt) |
| | | CLS_PRT_FNT_SIZE_8 | int | 3 | 文字種ポイント数(8pt) |
| | | CLS_PRT_FNT_SIZE_10 | int | 4 | 文字種ポイント数(10pt) |
| | | CLS_PRT_FNT_SIZE_12 | int | 5 | 文字種ポイント数(12pt) |
| | | CLS_PRT_FNT_SIZE_14 | int | 6 | 文字種ポイント数(14pt) |
| | | CLS_PRT_FNT_SIZE_18 | int | 7 | 文字種ポイント数(18pt) |
| | | CLS_PRT_FNT_SIZE_24 | int | 8 | 文字種ポイント数(24pt) |
| | | CLS_PRT_FNT_SIZE_30 | int | 9 | 文字種ポイント数(30pt) |
| | | CLS_PRT_FNT_SIZE_36 | int | 10 | 文字種ポイント数(36pt) |
| | | CLS_PRT_FNT_SIZE_48 | int | 11 | 文字種ポイント数(48pt) |
| | | CLS_PRT_FNT_KANJI_SIZE_16 | int | 100 | 漢字文字種(16ドット) |
| | | CLS_PRT_FNT_KANJI_SIZE_24 | int | 101 | 漢字文字種(24ドット) |
| | | CLS_PRT_FNT_KANJI_SIZE_32 | int | 102 | 漢字文字種(32ドット) |
| | | CLS_PRT_FNT_KANJI_SIZE_48 | int | 103 | 漢字文字種(48ドット) |
| 7 | エンコード | CLS_ENC_CDPG_DEFAULT | int | 0 | デフォルト |
| | | CLS_ENC_CDPG_IBM437 | int | 437 | IBM437 OEM 米国 |
| | | CLS_ENC_CDPG_IBM850 | int | 850 | ibm850 西ヨーロッパ語 (DOS) |
| | | CLS_ENC_CDPG_IBM852 | int | 852 | ibm852 中央ヨーロッパ言語 (DOS) |
| | | CLS_ENC_CDPG_IBM857 | int | 857 | ibm857 トルコ語 (DOS) |
| | | CLS_ENC_CDPG_IBM860 | int | 860 | IBM860 ポルトガル語 (DOS) |

| | | | | | |
|----|-------------------|-------------------------------|-----|-------|--------------------------------------|
| | | CLS_ENC_CDPG_IBM863 | int | 863 | IBM863 フランス語 (カナダ)(DOS) |
| | | CLS_ENC_CDPG_IBM864 | int | 864 | IBM864 アラビア語 |
| | | CLS_ENC_CDPG_IBM865 | int | 865 | IBM865 ノルウェー語 (DOS) |
| | | CLS_ENC_CDPG_CP866 | int | 866 | cp866 キリル語 (DOS) |
| | | CLS_ENC_CDPG_SHIFT_JIS | int | 932 | shift_jis 日本語 (シフト JIS) |
| | | CLS_ENC_CDPG_BIG5 | int | 950 | big5 繁体字中国語 (Big5) |
| | | CLS_ENC_CDPG_WINDOWS_1252 | int | 1252 | Windows-1252 西ヨーロッパ言語 (Windows) |
| | | CLS_ENC_CDPG_EUC_KR | int | 51949 | euc-kr 韓国語 (EUC) |
| | | CLS_ENC_CDPG_GB18030 | int | 54936 | GB18030 簡体字中国語 (GB18030) |
| 8 | 回転方向 | CLS_RT_NORMAL | int | 1 | 回転無し(0°) |
| | | CLS_RT_RIGHT90 | int | 2 | 右回転(90°) |
| | | CLS_RT_ROTATE180 | int | 3 | 反転(180°) |
| | | CLS_RT_LEFT90 | int | 4 | 左回転(270°) |
| 9 | バーコードの種類 | CLS_BCS_CODE39 | int | 100 | Code 3 of 9 |
| | | CLS_BCS_UPCA | int | 101 | UPC-A |
| | | CLS_BCS_UPCE | int | 102 | UPC-E |
| | | CLS_BCS_INTERLEAVED25 | int | 103 | Interleaved 2 of 5 |
| | | CLS_BCS_CODE128 | int | 104 | Code 128 |
| | | CLS_BCS_EAN13 | int | 105 | EAN-13 (JAN-13) |
| | | CLS_BCS_EAN8 | int | 106 | EAN-8 (JAN-8) |
| | | CLS_BCS_HIBC | int | 107 | HIBC |
| | | CLS_BCS_CODABAR | int | 108 | CODABAR (NW-7) |
| | | CLS_BCS_INT25 | int | 109 | Int 2 of 5 |
| | | CLS_BCS_PLESSEY | int | 110 | Plessey |
| | | CLS_BCS_CASECODE | int | 111 | CASE CODE |
| | | CLS_BCS_UPC2DIG | int | 112 | UPC 2DIG ADD (UPC 用の 2 桁の補足コード) |
| | | CLS_BCS_UPC5DIG | int | 113 | UPC 5DIG ADD (UPC 用の 5 桁の補足コード) |
| | | CLS_BCS_CODE93 | int | 114 | Code93 |
| | | CLS_BCS_ITF14 | int | 115 | (国内モデル)ITF-14 |
| | | CLS_BCS_ZIP | int | 116 | (海外モデル)ZIP |
| | | CLS_BCS_ITF16 | int | 117 | (国内モデル)ITF-16 |
| | | CLS_BCS_UCCEAN128 | int | 118 | (海外モデル)UCC/EAN-128 |
| | | CLS_BCS_INDUSTRIAL25 | int | 119 | (国内モデル)Industrial 2 of 5 |
| | | CLS_BCS_UCCEAN128KMART | int | 120 | (海外モデル) UCC/EAN-128(for K-MART) |
| | | CLS_BCS_COOP25 | int | 121 | (国内モデル)COOP 2 of 5 |
| | | CLS_BCS_UCCEAN128RANDOMWEIGHT | int | 122 | (海外モデル) UCC/EAN-128 Random Weight |
| | | CLS_BCS_TELEPEN | int | 123 | Telepen |
| 10 | バーコード文字の表示有無 | CLS_BCS_TEXT_HIDE | int | 0 | 非表示 |
| | | CLS_BCS_TEXT_SHOW | int | 1 | 表示 |
| 11 | エラー修正レベル (PDF417) | CLS_PDF417_EC_LEVEL_0 | int | 0 | レベル 0 |
| | | CLS_PDF417_EC_LEVEL_1 | int | 1 | レベル 1 |
| | | CLS_PDF417_EC_LEVEL_2 | int | 2 | レベル 2 |
| | | CLS_PDF417_EC_LEVEL_3 | int | 3 | レベル 3 |

| | | | | | |
|----|-----------------------|--|-----|-----|-------------------------------------|
| | | CLS_PDF417_EC_LEVEL_4 | int | 4 | レベル 4 |
| | | CLS_PDF417_EC_LEVEL_5 | int | 5 | レベル 5 |
| | | CLS_PDF417_EC_LEVEL_6 | int | 6 | レベル 6 |
| | | CLS_PDF417_EC_LEVEL_7 | int | 7 | レベル 7 |
| | | CLS_PDF417_EC_LEVEL_8 | int | 8 | レベル 8 |
| 12 | エラー修正レベル(Data Matrix) | CLS_DATAMATRIX_EC_LEVEL_200 | int | 200 | |
| 13 | エラー修正レベル(QR Code) | CLS_QRCODE_EC_LEVEL_L | int | 0 | エラー修正レベル L(7%) |
| | | CLS_QRCODE_EC_LEVEL_M | int | 1 | エラー修正レベル M(15%) |
| | | CLS_QRCODE_EC_LEVEL_Q | int | 2 | エラー修正レベル Q(25%) |
| | | CLS_QRCODE_EC_LEVEL_H | int | 3 | エラー修正レベル H(30%) |
| 14 | エラー修正レベル(Aztec) | CLS_AXTEC_EC_LEVEL_000 | int | 0 | 誤り訂正率 23% |
| 15 | バーコードタイプ(GS1 DataBar) | CLS_GS1_DATABAR_OMNI_DIRECTIONAL | int | 0 | GS1DataBar Omni-directional |
| | | CLS_GS1_DATABAR_COMPOSITE | int | 1 | GS1DataBar Composite |
| | | CLS_GS1_DATABAR_TRUNCATION | int | 2 | GS1DataBar Truncation |
| | | CLS_GS1_DATABAR_STACKED | int | 3 | GS1DataBar Stacked |
| | | CLS_GS1_DATABAR_STACKED_OMNI_DIRECTIONAL | int | 4 | GS1DataBar Stacked Omni-directional |
| | | CLS_GS1_DATABAR_LIMITED | int | 5 | GS1DataBar Limited |
| | | CLS_GS1_DATABAR_EXPANDED | int | 6 | GS1DataBar Expanded |
| 16 | 網掛けパターン | CLS_SHADED_PTN_0 | int | 0 | 網掛けパターン:000 |
| | | CLS_SHADED_PTN_1 | int | 1 | 網掛けパターン:001 |
| | | CLS_SHADED_PTN_2 | int | 2 | 網掛けパターン:002 |
| | | CLS_SHADED_PTN_3 | int | 3 | 網掛けパターン:003 |
| | | CLS_SHADED_PTN_4 | int | 4 | 網掛けパターン:004 |
| | | CLS_SHADED_PTN_5 | int | 5 | 網掛けパターン:005 |
| | | CLS_SHADED_PTN_6 | int | 6 | 網掛けパターン:006 |
| | | CLS_SHADED_PTN_7 | int | 7 | 網掛けパターン:007 |
| | | CLS_SHADED_PTN_8 | int | 8 | 網掛けパターン:008 |
| | | CLS_SHADED_PTN_9 | int | 9 | 網掛けパターン:009 |
| | | CLS_SHADED_PTN_10 | int | 10 | 網掛けパターン:010 |
| | | CLS_SHADED_PTN_11 | int | 11 | 網掛けパターン:011 |
| 17 | 単位 | CLS_UNIT_MILLI | int | 0 | ミリ |
| | | CLS_UNIT_INCH | int | 1 | インチ |
| 18 | 速度設定 | CLS_SPEEDSETTING_1 | int | 1 | 1:1.0 インチ(25.4mm)/秒 |
| | | CLS_SPEEDSETTING_2 | int | 2 | 2:2.0 インチ(50.8mm)/秒 |
| | | CLS_SPEEDSETTING_3 | int | 3 | 3:3.0 インチ(76.2mm)/秒 |
| | | CLS_SPEEDSETTING_4 | int | 4 | 4:4.0 インチ(101.6mm)/秒 |
| | | CLS_SPEEDSETTING_5 | int | 5 | 5:5.0 インチ(127.0mm)/秒 |
| | | CLS_SPEEDSETTING_6 | int | 6 | 6:6.0 インチ(152.4mm)/秒 |
| | | CLS_SPEEDSETTING_7 | int | 7 | 7:7.0 インチ(177.8mm)/秒 |
| | | CLS_SPEEDSETTING_8 | int | 8 | 8:8.0 インチ(203.2mm)/秒 |
| | | CLS_SPEEDSETTING_9 | int | 9 | 9:9.0 インチ(228.6mm)/秒 |
| | | CLS_SPEEDSETTING_A | int | 10 | A:1.0 インチ(25.4mm)/秒 |
| | | CLS_SPEEDSETTING_B | int | 11 | B:1.0 インチ(25.4mm)/秒 |
| | | CLS_SPEEDSETTING_C | int | 12 | C:2.0 インチ(50.8mm)/秒 |
| | | CLS_SPEEDSETTING_D | int | 13 | D:2.0 インチ(50.8mm)/秒 |

| | | | | | |
|----|--------------------------|-------------------------------|-----|----|------------------------|
| | | CLS_SPEEDSETTING_E | int | 14 | E: 3.0 インチ(76.2mm)／秒 |
| | | CLS_SPEEDSETTING_F | int | 15 | F: 3.0 インチ(76.2mm)／秒 |
| | | CLS_SPEEDSETTING_G | int | 16 | G: 4.0 インチ(101.6mm)／秒 |
| | | CLS_SPEEDSETTING_H | int | 17 | H: 4.0 インチ(101.6mm)／秒 |
| | | CLS_SPEEDSETTING_I | int | 18 | I: 5.0 インチ(127.0mm)／秒 |
| | | CLS_SPEEDSETTING_J | int | 19 | J: 5.0 インチ(127.0mm)／秒 |
| | | CLS_SPEEDSETTING_K | int | 20 | K: 6.0 インチ(152.4mm)／秒 |
| | | CLS_SPEEDSETTING_L | int | 21 | L: 6.0 インチ(152.4mm)／秒 |
| | | CLS_SPEEDSETTING_M | int | 22 | M: 7.0 インチ(177.8mm)／秒 |
| | | CLS_SPEEDSETTING_N | int | 23 | N: 7.0 インチ(177.8mm)／秒 |
| | | CLS_SPEEDSETTING_O | int | 24 | O: 8.0 インチ(203.2mm)／秒 |
| | | CLS_SPEEDSETTING_P | int | 25 | P: 8.0 インチ(203.2mm)／秒 |
| | | CLS_SPEEDSETTING_Q | int | 26 | Q: 9.0 インチ(228.6mm)／秒 |
| | | CLS_SPEEDSETTING_R | int | 27 | R: 9.0 インチ(228.6mm)／秒 |
| | | CLS_SPEEDSETTING_S | int | 28 | S: 10.0 インチ(254.0mm)／秒 |
| | | CLS_SPEEDSETTING_T | int | 29 | T: 10.0 インチ(254.0mm)／秒 |
| | | CLS_SPEEDSETTING_U | int | 30 | U: 11.0 インチ(279.4mm)／秒 |
| | | CLS_SPEEDSETTING_V | int | 31 | V: 11.0 インチ(279.4mm)／秒 |
| | | CLS_SPEEDSETTING_W | int | 32 | W: 12.0 インチ(304.8mm)／秒 |
| | | CLS_SPEEDSETTING_X | int | 33 | X: 12.0 インチ(304.8mm)／秒 |
| 19 | 印刷後動作 設定 | CLS_MEDIAHANDLING_NONE | int | 0 | なし |
| | | CLS_MEDIAHANDLING_TEAROFF | int | 1 | ティアオフ |
| | | CLS_MEDIAHANDLING_DISPENSES | int | 2 | ディスペンス |
| | | CLS_MEDIAHANDLING_PAUSE | int | 3 | ポーズ |
| | | CLS_MEDIAHANDLING_CUT | int | 4 | カット |
| | | CLS_MEDIAHANDLING_CUTANDPAUSE | int | 5 | カット、ポーズ |
| | | CLS_MEDIAHANDLING_PEELOFF | int | 6 | 剥離 |
| | | CLS_MEDIAHANDLING_REWIND | int | 7 | リワインダー |
| 20 | センサー選 択(透過/反 射/なし) | CLS_SELSENSOR_NONE | int | 0 | なし |
| | | CLS_SELSENSOR_SEETHROUGH | int | 1 | 透過 |
| | | CLS_SELSENSOR_REFLECT | int | 2 | 反射 |
| 21 | 印字方法 (TT/DT) | CLS_PRTMETHOD_TT | int | 0 | 熱転写 |
| | | CLS_PRTMETHOD_DT | int | 1 | 感熱 |
| 22 | センサー選 択(前方/後 方) | CLS_SENS_LOCATION_FRONT | int | 0 | フロントセンサ |
| | | CLS_SENS_LOCATION_ADJUSTABLE | int | 1 | アジャスタブルセンサ |

