
Multi LABELIST Web Component
リファレンスマニュアル
Ver.5.9.11.0

株式会社サトー

2024年2月28日

はじめに

この度は、「Multi LABELIST V5 Web Engine」（以下、MLV5 Web Engine）をご購入頂きまして誠にありがとうございます。

MLV5 Web Engine は、弊社製汎用ラベル・タグ発行ソフトウェア「Multi LABELIST V5」(以下 MLV5) の資産を利用し、お客様のシステムにラベル/タグの発行・プレビュー機能を追加する開発支援ツールです。発行機能として「Multi LABELIST Web Component」（以下、MLWebComponent）、プレビュー機能として「Multi LABELIST Preview Component」（以下、MLPreviewComponent）を提供致します。

本マニュアルでは、**発行機能を提供する MLWebComponent** について説明しております。MLV5 Web Engine の優れた機能がお客様のお役に立つことを開発者一同心より願っております。

ご注意

- 本マニュアルの一部または全部を弊社の許可なく複写・複製することは、その形態を問わず禁じます。
 - 本マニュアルの内容は、訂正・改善のため予告なく変更することがあります。
 - 本マニュアルを運用した結果の影響については責任を負いかねますのでご了承ください。
 - 本マニュアルの内容については万全を期しておりますが、万一ご不審な点やお気づきの点がございましたら、弊社までご連絡ください。
 - 本マニュアルの使用例は JavaScript で記載しております。説明上、変数の宣言やエラー処理など一部の記述を省いておりますのでご了承ください。
-
- SATO、Multi LABELIST は、サトーホールディングス株式会社の登録商標または商標です。
 - Microsoft、Windows、Internet Explorer は、米国マイクロソフト社の登録商標です。
 - その他記載されている会社名、製品名は各社の登録商標または商標です。

目次

はじめに.....	2
ご注意.....	2
第1章 製品情報.....	8
1-1.必要なハードウェア・ソフトウェア.....	9
■サーバ環境.....	9
■クライアント環境.....	9
1-2.MLWebComponent 機能一覧.....	10
■MLWebComponent プロパティ一覧.....	10
■MLWebComponent メソッド一覧.....	11
■FileAccessComponent プロパティ一覧.....	12
■FileAccessComponent メソッド一覧.....	12
第2章 MLWebComponent プロパティ.....	13
2-1.[通信]Setting.....	14
2-2.[通信]Protocol.....	17
2-3.[通信]Timeout.....	19
2-4.[通信]StatusID.....	21
2-5.[通信]JobName.....	22
2-6.[基本]LayoutFile.....	23
2-7.[基本]PrnData.....	24
2-8.[基本]PrnDataType.....	26
2-9.[動作]Darkness.....	27
2-10.[動作]Speed.....	29
2-11.[動作]Offset.....	31
2-12.[動作]MultiCut.....	33
2-13.[動作]SortMark.....	35
2-14.[動作]EjectCut.....	36
2-15.[動作]HeaderTailSetting.....	37
2-16.[動作]HeaderFile.....	38
2-17.[動作]TailFile.....	39
2-18.[特殊]Formoverlay.....	40
2-19.[特殊]LayoutNameCaption.....	42
2-20.[特殊]TotalQtyCaption.....	43
2-21.[特殊]TaxRate.....	44
2-22.[バージョン]Version.....	45
第3章 MLWebComponent メソッド・例外.....	46
3-1.[通信]OpenPort.....	47
3-2.[通信]ClosePort.....	49
3-3.[発行]Output.....	50
3-4.[発行]OutputHeader.....	51
3-5.[発行]OutputTail.....	52
3-6.[発行]SendStringData.....	53
3-7.[発行]GetStatus.....	55
3-8.[制御]Cut.....	57

3-9.[制御]SendCancel.....	59
3-10.[ダウンロード]GetFile	60
3-11.[ダウンロード]GetFileCredential	62
3-12.[ダウンロード]SetProxyCredential	64
3-13.[ダウンロード]SetClientCertificate.....	65
3-14.[ダウンロード]GetClientCertCount.....	67
3-15.[ダウンロード]GetClientCertInfo.....	68
3-16.[情報]GetPrinter.....	69
3-17.[情報]GetLayoutFileProperty	70
3-18.[データ]GetInputCount.....	71
3-19.[データ]GetInputFields	72
3-20.[データ]SetPrnDataField.....	74
3-21.[データ]GetPrnDataRowCount.....	76
3-22.[データ]GetPrnDataRow	77
3-23.[データ]SetPrnDataRow.....	78
3-24.[ドライバ]GetDriverNameList.....	80
3-25.[ドライバ]ClearPrintJob	81
3-26.[デバイス]EnumerateBluetoothDevices.....	82
3-27.[デバイス]AuthenticateBluetoothDevice	84
3-28.[認証]Authenticate.....	86
3-29.[例外]MLWebComponentException クラス.....	88
3-30.[エラー]通信エラー	90
3-31.[エラー]発行エラー	91
■例外エラーについて	95
第4章 FileAccessComponent プロパティ	97
4-1.[基本]FilePath.....	98
4-2.[基本]FileType	99
4-3.[基本]CacheSize	100
4-4.[データ]RecordCount	101
4-5.[データ]RowData	102
4-6.[レコード]RowPosition	103
4-7.[レコード]BOF.....	104
4-8.[レコード]EOF	105
4-9.[バージョン]Version.....	106
第5章 FileAccessComponent メソッド	107
5-1.[ファイル]OpenFile.....	108
5-2.[ファイル]CloseFile.....	109
5-3.[ファイル]RemoveFile.....	110
5-4.[レコード]Move	111
5-5.[レコード]MoveNext.....	112
5-6.[レコード]MovePrevious.....	113
5-7.[レコード]MoveFirst.....	114
5-8.[レコード]MoveLast	115
第6章 ワンポイントテクニック	116
6-1.MLWebComponent を利用する.....	117
■object タグで配置する.....	117

■スクリプト上で宣言する	118
■ライセンス認証を行う	118
■サーバ環境を構築する	118
■クライアント環境を構築する	119
6-2.ライセンス認証を行う	120
■ライセンスファイルと認証キー	120
■ライセンス認証を行う	120
6-3.バージョンを確認する	121
■Version プロパティで取得する	121
■バージョン確認を行ってバージョンアップを通知する	121
■ファイルのプロパティで確認する	121
6-4.バージョンアップを行う	122
■サーバの WebComponent を更新する	122
■クライアントの WebComponent を更新する	122
6-5.データを一括で入力する	124
■入力順位	124
■データ形式	125
6-6.データを変数名で指定して入力する	126
■変数名	126
6-7.動作設定ファイルを利用する	128
■ログファイルを出力する	130
6-8.評価版ライセンスから入れ替える	132
■ライセンスファイルと認証キー	132
■ライセンスファイルの入替	132
第7章 利用方法・注意事項	133
7-1.利用方法について	134
■ライセンスファイルと認証キーについて	134
■ライセンス認証について	134
■IE の保護モードについて	134
■IE10 以降での利用	134
■Windows 7 での利用	134
■.NET Framework のバージョンについて	134
■サーバサイドアプリケーションでの利用	135
■プリンタのスリープモードについて	135
■Windows フォントの全角スペースが.NET Framework のバージョンによって異なる	135
7-2.プログラミングについて	136
■カット動作に関して	136
■初回起動時・レイアウトファイルロード時の遅延	136
■Windows フォント (WPF) の使用について	136
7-3.インターフェースについて	137
■LAN	137
■USB	137

■COM.....	139
■Bluetooth.....	139
■プリンタドライバ.....	140
7-4.レイアウト情報について.....	142
■入力定義.....	142
■Windows フォント.....	142
■グラフィックオブジェクト.....	142
■固定オブジェクト（発行速度の改善）.....	142
■連番変数.....	142
■多面取りラベル.....	143
第8章 付録.....	144
8-1.サポートプリンタ.....	145
■シータシリーズ.....	145
■レスプリシリーズ.....	145
■スキャントロニクスシリーズ.....	145
■SATOC シリーズ、エヴィシリーズ、バートロニクスシリーズ、FLEQV.....	146
■ラパンシリーズ.....	147
■タフアームシリーズ.....	147
■海外プリンタ.....	147
ステータス一覧.....	149
■ステータス文字列.....	149
■ステータス一覧（ステータス3、ステータス4）.....	149
■ステータス一覧（レスプリVシリーズ、PW208NX/PW208シリーズ、PW4NX 追加分）.....	150
■ステータス一覧（ステータスL 追加分）.....	151
■ステータス一覧（PT408e、PT412e、PT200m/e/j用ステータス3）.....	151
■ステータス一覧の送信可否・復旧可否.....	152
■プリンタエラーの解除方法.....	152
8-3.プリンタ本体の通信プロトコル設定方法.....	153
■レスプリシリーズ（レスプリV、レスプリV-ex）.....	154
■エヴィシリーズ.....	155
■スキャントロニクスシリーズ、TR400e/TR410e.....	155
■スキャントロニクス SR400 シリーズ、SG400R シリーズ、SG400R-ex シリーズ、SG600R シリーズ、SG112R/T、HA200R シリーズ、LR4000SR-T シリーズ（ステータスLも選択 可）.....	155
■CL4NX-J、CL6NX-J.....	155
■SATOC ST308R/ST312R.....	156
■バートロニクス RT308R.....	156
■ラパンシリーズ.....	156
■スキャントロニクス GN412T.....	156
■バートロニクス CF408T.....	157

■ラパンシリーズ PW208NX/PW208mNX/PW208/PW208m.....	157
■シータシリーズ、レスプリシリーズ (HC4-LX-J)	157
■FLEQV シリーズ FX3-LX、FX3-LX Plus.....	157
■ラパンシリーズ PW4NX.....	157

第1章

製品情報

1-1

必要なハードウェア・ソフトウェア

■ハードウェア ■ソフトウェア

MLWebComponent をご使用いただくには、次のハードウェアとソフトウェアが必要です。

■サーバ環境

- ハードウェア Web サーバが動作する環境
[サポートプリンタ](#)
- ソフトウェア 対応 OS 限定されません。Windows、Linux など。
Web サーバ 限定されません。IIS、Apache など。

■クライアント環境

- ハードウェア 下記対応 OS が動作する環境
5MB 以上の空き容量を持つハードディスク
- ソフトウェア 対応 OS Windows 11、Windows 10、Windows 8.1
実行環境 .NET Framework^{*1} 4.5/4.6/4.7/4.8
ブラウザ Microsoft Edge (IE モード)、Internet Explorer 11

*1 実行環境に存在しない場合は、別途.NET Framework のインストールが必要です。(Ver.5.7.0.0 以降は必須)

<https://docs.microsoft.com/ja-jp/dotnet/framework/install/guide-for-developers>

古いバージョンの.NET Framework を利用すると、MLWebComponent が正常に動作しない場合があります。

Microsoft でサポートされているバージョンをご利用ください。

<https://docs.microsoft.com/ja-JP/lifecycle/faq/dotnet-framework>

1-2

MLWebComponent 機能一覧

■プロパティ ■メソッド

MLWebComponent は、

ラベル発行を行う「MLWebComponent」と、

ファイルを操作する「FileAccessComponent」の2つのコントロールが利用できます。

それぞれのコントロールで利用できるプロパティ・メソッド・例外の一覧を記載します。

■MLWebComponent プロパティ一覧

プロパティ名	説明	初期値
通信設定		
Setting	出力先の設定	LAN:127.0.0.
Protocol	通信プロトコルの設定	0 (ステータス3)
Timeout	通信タイムアウト値の設定	3
StatusID	ステータスIDの設定	0
JobName	ジョブ名の設定	"" (空文字)
基本設定		
LayoutFile	レイアウトファイルの設定	Default.mllayx
PrnData	印字データの設定	"" (空文字)
PrnDataType	印字データタイプの設定	0
プリンタ動作設定		
Darkness	印字濃度の設定	S (レイアウト依存)
Speed	印字速度の設定	S (レイアウト依存)
Offset	印字補正值の設定	0.0000,0.0000
MultiCut	カット指定の設定	0 (カットしない)
SortMark	仕分けマーク印字の設定	false (印字しない)
EjectCut	排出カットの設定	false (カットしない)
HeaderTailSetting	ヘッダ・テール札の設定	false (印字しない)
HeaderFile	ヘッダ札の取得	"" (空文字)
TailFile	テール札の取得	"" (空文字)
特殊設定		
Formoverlay	フォームオーバーレイの設定	0 (使用しない)
LayoutNameCaption	システム変数「レイアウト名」の設定	"" (空文字)
TotalQtyCaption	システム変数「総発行枚数」の設定	0
TaxRate	税率の設定	"" (空文字)
バージョン情報		
Version	バージョン情報の取得	-

■MLWebComponent メソッド一覧

メソッド名	説明
通信	
OpenPort	通信ポートのオープン
ClosePort	通信ポートのクローズ
発行	
Output	プレビュー表示
OutputHeader	ヘッダ札のプレビュー表示
OutputTail	テール札のプレビュー表示
SendStringData	プリンタコマンドの送信(終了条件指定)
プリンタ制御	
GetStatus	プリンタの状態確認
Cut	発行中のカット
SendCancel	発行のキャンセル
ダウンロード	
GetFile	ファイルのダウンロード
GetFileCredential	ファイルのダウンロード (BASIC 認証)
SetProxyCredential	プロキシサーバの BASIC 認証設定
SetClientCertificate	クライアント証明書の検索・設定
GetClientCertCount	クライアント証明書数の取得
GetClientCertInfo	クライアント証明書情報の取得
レイアウト情報取得	
GetPrinter	プリンタ情報の取得
GetLayoutFileProperty	レイアウト情報の取得
印字データ指定	
GetInputCount	入力項目数の取得
GetInputFields	入力情報の取得
SetPrnDataField	入力項目の印字データ設定
GetPrnDateRowCount	複数データの行数取得
GetPrnDataRow	複数データの取得
SetPrnDataRow	複数データの 1 行分のデータ設定
プリンタドライバ制御	
GetDriverNameList	プリンタドライバ名の取得
ClearPrintJob	プリンタドライバのジョブ削除
デバイス制御	
EnumerateBluetoothDevices	デバイス検索の実行
AuthenticateBluetoothDevice	デバイス認証の実行
ライセンス認証	
Authenticate	ライセンス認証の実行

■FileAccessComponent プロパティ一覧

プロパティ名	説明	初期値
基本設定		
FilePath	ファイルパス名の設定	""
FileType	ファイルデータタイプの設定	0 (TSV)
CacheSize	キャッシュサイズの設定	1
データ取得		
RecordCount	データのレコード数の取得	0
RowData	印字データの取得	""
カレントレコード位置		
RowPosition	カレントレコード位置の設定	-1
BOF	カレントレコードが最初のレコードより前にあるかどうか	False
EOF	カレントレコードが最後のレコードより後にあるかどうか	False
バージョン情報		
Version	バージョン情報の取得	-

■FileAccessComponent メソッド一覧

メソッド名	説明
ファイル操作	
OpenFile	ファイルのオープン
CloseFile	ファイルのクローズ
RemoveFile	ファイルの削除
カレントレコード移動	
Move	カレントレコードを移動
MoveNext	カレントレコードを次のレコードに移動
MovePrevious	カレントレコードを前のレコードに移動
MoveFirst	カレントレコードを先頭のレコードに移動
MoveLast	カレントレコードを末尾のレコードに移動

第2章

MLWebComponent

プロパティ

2-1

[通信]Setting プロパティ

通信パラメータを設定、取得します。

■書式

MLWebComponent.Setting As String

■設定値

出力先	Setting	
	説明	
初期値	LAN:127.0.0.1	
LAN 接続プリンタ	<i>LAN:aaa.aaa.aaa.aaa</i> [,Port1][,Port2]	
	<i>aaa.aaa.aaa.aaa</i>	IP アドレス
	[,Port1]	ポート番号 1 [省略可]
	[,Port2]	ポート番号 2 [省略可]
USB 接続プリンタ	<i>USB:[PrinterModel][,SerialNumber]</i>	
	[,PrinterModel]	プリンタ機種 [省略可]
	[,SerialNumber]	シリアル No. [省略可]
RS-232C 接続プリンタ	<i>COMnnn:bbbb,p,d,s</i>	
	nnn	ポート番号 (1~256)
	bbbb	ボーレート (4800~115200)
	p	パリティビット (n:なし, e:偶数, o:奇数)
	d	データビット (7, 8)
	s	ストップビット (1, 1.5, 2)
Bluetooth 接続プリンタ	<i>BT:BDAddress</i>	
	<i>BDAddress</i>	BD アドレス (12 桁の 16 進数)
プリンタドライバ	DRV:Drivername	
	<i>Drivername</i>	プリンタドライバ名称
ファイル	<i>FILE:Filename,m</i>	
	<i>Filename</i>	ファイル名称
	<i>m</i>	出力モード (0:上書き, 1:追加)

■解説

- [インターフェース毎の利用方法は利用方法・注意事項をご参照ください。](#)
- 1台のプリンタに対してインターフェース出力 (LAN、USB、COM、Bluetooth) とプリンタドライバ出力を切替えて接続することは行わないでください。2重接続が発生し、エラーとなる可能性があります。

- LAN のポート番号を省略した場合、Protocol プロパティとの組み合わせにより、以下の値を使用します。Status 3 は 1 ポート、Status 4 は 2 ポートで通信します。

ポート番号指定	Protocol プロパティ	
	0-Status3	1-Status4
省略	送受信ポート：1024	送信ポート：1024 受信ポート：1025
1 ポート指定	送受信ポート：指定ポート	送受信ポート：指定ポート *1024 指定時 送信ポート：1024 受信ポート：1025
2 ポート指定	エラー	送信ポート：1 番目の指定ポート 受信ポート：2 番目の指定ポート

- USB 指定の[プリンタ機種]、[シリアル No.]を省略した場合、以下のデバイスと接続します。指定された[プリンタ機種]と[レイアウトのプリンタ機種]が不一致でもエラーとなりません。[\[プリンタ機種\]の設定値](#)、[\[シリアル No.\]の確認方法](#)は利用方法・注意事項をご参照ください。

プリンタ機種	シリアル No.	説明
なし	なし	USB デバイスの検索で最初に見つかったサトー製プリンタ
あり	なし	USB デバイスの検索で指定されたプリンタ機種で最初に見つかったサトー製プリンタ
あり	あり	指定されたプリンタ機種、シリアル No.のサトー製プリンタ

■例外

- JavaScript での例外処理は try…catch で実行してください。Error オブジェクトの message プロパティにメッセージが格納されます。

例外	説明 (message)
InvalidOperationException	条件：ポートオープン中にプロパティを変更 The communication parameter cannot be changed when the port is opened.

■使用例

COM ポート 1 に通信設定 9600,n,8,1 の場合

```
MLWebComponent.Setting = "COM1:9600,n,8,1";
```

IP アドレス 192.168.1.1、ポート番号 1024 の場合

```
MLWebComponent.Setting = "LAN:192.168.1.1,1024";
```

プリンタドライバ名称「SATO SG408R-ex」の場合

```
MLWebComponent.Setting = "DRV:SATO SG408R-ex";
```

ファイル「C:¥My Documents¥Printout.prn」に上書きモードで出力する場合

```
MLWebComponent.Setting = "FILE:C:¥¥My Documents¥¥Printout.prn,0";
```

USB の場合

```
MLWebComponent.Setting = "USB:CF408T,0000T123";
```

Bluetooth の場合

```
MLWebComponent.Setting = " BT:000b5d3db4c2"
```

■関連項目

プロパティ [Protocol](#)、[Timeout](#)

メソッド [OpenPort](#)、[Output](#)、[GetStatus](#)、[EnumerateBluetoothDevices](#)

■サポート情報

注意事項 [インターフェースについて](#)

付録 [プリンタ本体の通信プロトコル設定方法](#)

2-2

[通信]Protocol プロパティ

通信プロトコルを設定、取得します。

■書式

MLWebComponent.Protocol *As Integer* / SATO.MLWebComponent.Protocols

■設定値

Protocol	説明
0 - Protocols.Status3	ステータス3プロトコル、ステータスLプロトコル/LR専用（初期値）
1 - Protocols.Status4	ステータス4プロトコル（ドライバ専用プロトコル）

■解説

- 設定値に合わせて、プリンタ本体の通信プロトコルを設定する必要があります。
- OpenPort メソッド成功後（通信ポートがオープン中）に変更すると例外が発生します。
- 通信プロトコルのステータス5には対応していません。
- Setting プロパティが LAN、USB、COM、BT（Bluetooth）の場合に使用します。DRV（プリンタドライバ）、FILE では使用されません。
- Setting プロパティが USB の場合は、「1」を指定してください。
「1」以外を指定した場合は、OpenPort メソッドの戻り値で「12」となります。

■例外

- JavaScript での例外処理は try...catch で実行してください。Error オブジェクトの message プロパティにメッセージが格納されます。

例外	説明 (message)
InvalidOperationException	条件：ポートオープン中にプロパティを変更 The value of Protocol property cannot be changed when the port is opened.
ArgumentOutOfRangeException	条件：範囲外の設定値 The value of Protocol property is invalid.

■使用例

ステータス4プロトコルを設定します。

```
MLWebComponent.Protocol = 1;
```

現在の通信プロトコルを取得します。

```
Protocol = MLWebComponent.Protocol;
```

■関連項目

プロパティ [Setting](#)、[Timeout](#)
メソッド [OpenPort](#)、[Output](#)、[GetStatus](#)、[Cut](#)、[SendCancel](#)、[SendStringData](#)、

■サポート情報

注意事項 [インターフェースについて](#)
付録 [プリンタ本体の通信プロトコル設定方法](#)

2-3

[通信]Timeout プロパティ

通信タイムアウト値を設定、取得します。

■書式

MLWebComponent.Timeout *As Integer*

■設定値

Timeout	説明
3	3秒でタイムアウトが発生（初期値）
任意数値	接続・送信・受信時の通信タイムアウト時間の秒指定（1～60）

■解説

- Setting プロパティが LAN、USB、COM、BT（Bluetooth）の場合に使用します。DRV（プリンタドライバ）、FILE では使用されません。
- OpenPort メソッド成功後（通信ポートがオープン中）に変更すると例外が発生します。
- タイムアウトは以下のメソッドで適用されます。

メソッド	説明
OpenPort	Setting プロパティが LAN、Bluetooth で接続先が見つからない場合にポートオープンエラーとなります。
Output	Protocol プロパティがステータス 3 の場合、プリンタから ACK/NAK 返送がない場合に受信タイムアウトとなります。
GetStatus	プリンタからステータスの返送がない場合に受信タイムアウトとなります。
Cut	プリンタから ACK/NAK 返送がない場合に受信タイムアウトとなります。
SendCancel	
SendStringData	メソッドで指定した終了条件に一致しない場合、受信タイムアウトとなります。

■例外

- JavaScript での例外処理は try…catch で実行してください。Error オブジェクトの message プロパティにメッセージが格納されます。

例外	説明（message）
InvalidOperationException	条件：ポートオープン中プロパティを変更 The value of Timeout property cannot be changed when the port is opened.
ArgumentOutOfRangeException	条件：範囲外の設定値 The value of Timeout property is invalid. This property must be within the range from 1 to 60.

■使用例

通信タイムアウトの設定例を示します。

```
MLWebComponent.Setting = "LAN:192.168.1.123";
MLWebComponent.Timeout = 5;
Result = MLWebComponent.OpenPort(1);
if ( Result != 0 ){
    //エラー処理
}
```

■関連項目

プロパティ

[Setting](#)、[Protocol](#)

メソッド

[OpenPort](#)、[Output](#)、[GetStatus](#)、[SendStringData](#)

2-4

[通信]StatusID プロパティ

ステータス ID を設定・取得します。

■書式

MLWebComponent.StatusID [As Integer](#)

■設定値

StatusID	説明
0	ステータス ID を 0 にする (初期値)
任意数値	ステータス ID に設定する数値 (0~99)

■例外

- JavaScript での例外処理は try…catch で実行してください。Error オブジェクトの message プロパティにメッセージが格納されます。

例外	説明 (message)
ArgumentOutOfRangeException	条件：範囲外の設定値 The value of StatusID property is invalid. This property must be within the range from 0 to 99.

■使用例

ステータス ID の運用例を示します。

```
MLWebComponent.StatusID = 36;
Result = MLWebComponent.Output();
Result = MLWebComponent.GetStatus(Status);
if ( Status.substring( 0 , 2 ) == "36" ) {
    //ステータス ID が 36 の場合
}
```

■関連項目

プロパティ [Protocol](#)
メソッド [Output](#)、[GetStatus](#)

2-5

[通信]JobName プロパティ

ジョブ名を設定・取得します。

■書式

MLWebComponent.JobName *As String*

■設定値

JobName	説明
空文字	ジョブ名を空欄にする（初期値）
任意文字列	ジョブ名に設定する文字列

■解説

- Protocol プロパティが「1」の場合に使用します。

■例外

- JavaScript での例外処理は try…catch で実行してください。Error オブジェクトの message プロパティにメッセージが格納されます。

例外	説明 (message)
ArgumentException	条件：範囲外の設定値 The value of JobName property is invalid. The length of this property must be within 16 bytes.

■使用例

ジョブ名の運用例を示します。

```
MLWebComponent.JobName = "SATO Print";
Result = MLWebComponent.Output();
Result = MLWebComponent.GetStatus(Status);
JobName = Status.substring(8,16);
if ( JobName.replace(/¥s+¥/, "") == "SATO Print" ){
    //ジョブ名が SATO Print の場合
}
```

■関連項目

プロパティ [Protocol](#)、[StatusID](#)
メソッド [Output](#)、[GetStatus](#)

2-6

[基本]LayoutFile プロパティ

MLV5 で作成したレイアウトファイルを設定・取得します。

■書式

MLWebComponent.LayoutFile [As String](#)

■設定値

LayoutFile	説明
任意文字列	レイアウトファイルのローカルファイルパス

■解説

- レイアウトファイル (*.mllayx) のみ使用可能です。
- 値を設定すると、PrnData プロパティ、SetPrnDataField メソッド、SetPrnDataRow メソッドで指定した入力データはすべて初期化されます。

■例外

- JavaScript での例外処理は try…catch で実行してください。Error オブジェクトの message プロパティにメッセージが格納されます。

例外	説明 (message)
ArgumentException	条件：不正な設定値 The value of LayoutFile property is invalid.

■使用例

レイアウト情報ファイルパスを設定します。

```
MLWebComponent.LayoutFile = "C:¥¥SATO¥¥ABC.mllayx";
```

現在のレイアウト情報ファイルを取得します。

```
LayoutFile = MLWebComponent.LayoutFile;
```

■関連項目

- プロパティ [PrnData](#)、[Darkness](#)、[Speed](#)、[HeaderTailSetting](#)、[HeaderFile](#)、[TailFile](#)、[Formoverlay](#)
- メソッド [Output](#)、[GetInputFields](#)、[SetPrnDataField](#)、[SetPrnDataRow](#)、[GetPrinter](#)

2-7

[基本]PrnData プロパティ

印字データを設定、取得します。

■書式

MLWebComponent.PrnData [As String](#)

■設定値

PrnData	説明
任意文字列	PrnDataType プロパティのデータ形式で指定します。文字フォーマットは MLV5 準拠で Unicode(UTF-16)となります。

■解説

- バーコードにチェックデジットを含めたデータを指定する際は、チェックデジットが正しいかお客様のアプリケーションで必ずチェックした上で指定してください。チェックが出来ない場合は、チェックデジットを含まないデータに変更するか、変数の設定でチェックデジット部分を取り除くようにレイアウトファイルを変更してください。
- 発行枚数は、最大 999999（PW208 を除くラパンシリーズは 9999）まで指定可能です。
- データの項目数は、レイアウトファイルの入力項目数に合わせて指定してください。
- LayoutFile プロパティを設定した時点で PrnData プロパティは初期化されます。
- 値を設定すると、SetPrnDataRow メソッドで設定した入力データは初期化されます。
- SetPrnDataField メソッドと併用する場合は、混乱を避けるため PrnData プロパティを先に指定してください。
- PrnDataType プロパティで指定された形式の区切り文字はデータとして設定しないでください。例として、PrnDataType プロパティが「タブ区切り」の場合に、データ内にタブ (0x09) を入力すると、入力項目の入力順がずれて不正な入力データとなります。
- QR コード等のバーコードデータに制御コードを含める場合は、レイアウトの入力変数種類を「16進文字コード」に設定し、上位アプリからは 16 進文字列のデータをセットしてください。（例：12345<TAB>の場合、「313233343509」をセット）
- Setting プロパティが DRV（プリンタドライバ）以外の場合は、連番変数や多面取りを使用しているレイアウトファイルで、1 シート以上にならない発行枚数を指定してください。1 シート以上となる発行枚数を指定した場合は、Output メソッドでエラーNo.804 となります。

■使用例

印字データを設定します。

```
MLWebComponent.PrnDataType = 0;
MLWebComponent.PrnData = "1000" + "¥" + "2000" + "¥" + "1";
```


■関連項目

プロパティ [LayoutFile](#)、[PrnDataType](#)
メソッド [Output](#)、[GetInputFields](#)、[SetPrnDataField](#)、[GetPrnDataRow](#)、[SetPrnDataRow](#)

■サポート情報

ワンポイントテクニック [データを一括で入力する](#)

2-8

[基本]PrnDataType プロパティ

印字データのタイプ(Tsv/Csv/Prn)を設定、取得します。

■書式

MLWebComponent.PrnDataType [As Integer](#)

■設定値

<i>PrnDataType</i>	説明
0 - PrnDataTypes.Tsv	タブ区切り指定/TSV 形式（初期値）
1 - PrnDataTypes.Csv	カンマ区切り指定/CSV 形式
2 - PrnDataTypes.Prn	スペース区切り指定/PRN 形式

■解説

- カンマ区切りを指定した場合は、項目をダブルクォーテーションで括ることにより、区切り文字であるカンマ（0x2C）、タブ（0x09）、改行コード（0x0D、0x0A）を指定することが可能です。印字する際に、括り文字（ダブルクォーテーション）は省かれます。

■例外

- JavaScript での例外処理は try…catch で実行してください。Error オブジェクトの message プロパティにメッセージが格納されます。

例外	説明（message）
ArgumentOutOfRangeException	条件：範囲外の設定値 The value of PrnDataType property is invalid.

■使用例

印字データのタイプに TSV を設定します。

```
MLWebComponent.PrnDataType = 0;
```

■関連項目

プロパティ
メソッド

[PrnData](#)
[SetPrnDataField](#)、[GetPrnDataRow](#)、[SetPrnDataRow](#)

2-9

[動作]Darkness プロパティ

印字濃度を設定・取得します。

■書式

MLWebComponent.Darkness *As String*

■設定値

<i>Darkness</i>	説明
空文字列	プリンタ本体に登録された設定値を使用
S	レイアウト情報の印字濃度を使用（初期値）
印字濃度レベル[,印字濃度レンジ]	印字濃度レベル、印字濃度レンジをカンマ区切りで指定 (印字濃度レンジは省略可能)

■ポイント

- ・ 「S」でレイアウトに設定がない場合、プリンタに登録された設定値が有効になります。
- ・ 印字濃度レンジは通常設定する必要はございません。2 色サーマル用紙など特殊な用紙を使用する際に弊社の担当販売員よりご説明致します。
- ・ GN412T の両面レイアウトは、指定した値が両面の印字濃度になります。

■例外

- ・ JavaScript での例外処理は try…catch で実行してください。Error オブジェクトの message プロパティにメッセージが格納されます。

例外	説明 (message)
ArgumentOutOfRangeException	条件：範囲外の設定値 The value of Darkness property is invalid.
ArgumentException	条件：不正な設定値 The value of Darkness property is invalid. This property must be alphanumeric.

■使用例

印字濃度レベルに 3、印字濃度レンジに A を設定します。

```
MLWebComponent.Darkness = "3,A";
```

レイアウト情報の印字濃度を使用します。

```
MLWebComponent.Darkness = "S";
```

現在の印字濃度プロパティの設定値を取得します。

```
Dark = MLWebComponent.Darkness;
```

■関連項目

プロパティ [Speed](#)、[Offset](#)
メソッド [Output](#)

■サポート情報

付録 [サポートプリンタ](#)

2-10

[動作]Speed プロパティ

印字速度を設定・取得します。

■書式

MLWebComponent.Speed [As String](#)

■設定値

Speed	説明
空文字列	プリンタ本体に登録された設定値を使用
S	レイアウト情報の印字速度を使用（初期値）
任意文字列	設定値を使用

■解説

- 「S」でレイアウトに設定がない場合、プリンタに登録された設定値が有効になります。

■例外

- JavaScript での例外処理は try…catch で実行してください。Error オブジェクトの message プロパティにメッセージが格納されます。

例外	説明 (message)
ArgumentOutOfRangeException	条件：範囲外の設定値 The value of Speed property is invalid. The length of this property must be within 2 bytes.
ArgumentException	条件：不正な設定値 The value of Speed property is invalid. This property must be alphanumeric.

■使用例

印字速度に 5 を設定します。

```
MLWebComponent.Speed = "5";
```

レイアウトファイルの印字速度を使用します。

```
MLWebComponent.Speed = "S";
```

現在の印字速度プロパティの設定値を取得します。

```
Speed = MLWebComponent.Speed;
```

■関連項目

プロパティ [Darkness](#)、[Offset](#)
メソッド [Output](#)

■サポート情報

付録

[サポートプリンタ](#)

2-11

[動作]Offset プロパティ

印字補正を設定・取得します。

■書式

MLWebComponent.Offset [As String](#)]

■設定値

Offset	説明
0,0	補正なし（初期値）
S,S	レイアウト情報の印字補正を使用
横補正值,縦補正值	横・縦の補正値をカンマ区切りで mm 単位指定（-99.9999～99.9999） 小数点以下 4 桁まで有効

■解説

- ・ 「S,S」でレイアウトに設定がない場合、プリンタに登録された設定値が有効になります。
- ・ 印字補正を行うと、プリンタの印字領域をオーバーする、ラベルから文字がはみ出るなどの可能性があります。Output メソッドで以下のエラーが発生する可能性があります。注意が必要です。
 - 610 用紙に印字できない項目が存在します。
 - 611 計算できない無効な項目が存在します。

■例外

- ・ JavaScript での例外処理は try…catch で実行してください。Error オブジェクトの message プロパティにメッセージが格納されます。

例外	説明 (message)
ArgumentOutOfRangeException	条件：範囲外の設定値 The value of Offset property is invalid. This property must be within the range from -99.9 to 99.9.

■使用例

印字位置を横に 3mm、縦に 2mm 補正して発行します。

```
MLWebComponent.Offset = "3.0,2.0";
```

レイアウト情報の印字補正値を使用します。

```
MLWebComponent.Offset = "S,S";
```

■関連項目

プロパティ [Darkness](#)、[Speed](#)
メソッド [Output](#)

■サポート情報

付録

[サポートプリンタ](#)

2-12

[動作]MultiCut プロパティ

カット単位枚数を設定・取得します。

■書式

MLWebComponent.MultiCut As Integer

■設定値

MultiCut	説明
0	カットしない（初期値）
1 以上	指定枚数でカットする(1~9999)
-1	プリンタ本体の動作モードに従う
-2	レイアウトの設定に従う

■解説

- 発行枚数が指定枚数に満たない場合、発行終了時にカットします。
 (例) MultiCut プロパティが「5」、発行枚数が「3」 ⇒ 3 枚目でカット
 MultiCut プロパティが「5」、発行枚数が「7」 ⇒ 5 枚目と 7 枚目でカット
- 「-2」を指定した場合は、レイアウト設定および印字データの指定方法により下記の動作となります。

PrnData でデータを指定する場合

発行時動作設定（発行動作）「仕分けマークの印字、またはカット動作のタイミング」	
なし(プリンタの設定に従う)	プリンタ本体の動作モードに従います。
行わない	カットを行いません。排出カットはレイアウトの発行時動作設定の「排出カットを使用する」に従って行います。
行ごとに行う	最後のラベルのみカットします。排出カットはレイアウトの発行時動作設定の「排出カットを使用する」に従って行います。
ページごとに行う	
該当する項目の値が変わるごとに行う	
発行指示単位ごとに行う	
指定した枚数ごとに行う	MultiCut「1 以上」と同じ動作

PrnDataArray でデータを指定する場合

発行時動作設定（発行動作）「仕分けマークの印字、またはカット動作のタイミング」	
なし(プリンタの設定に従う)	プリンタ本体の動作モードに従います。
行わない	カットを行いません。排出カットはレイアウトの発行時動作設定の「排出カットを使用する」に従って行います。
行ごとに行う	配列要素ごとにカットします。排出カットはレイアウトの発行時動作設定の「排出カットを使用する」に従って行います。
ページごとに行う	
該当する項目の値が変わるごとに行う	
発行指示単位ごとに行う	
指定した枚数ごとに行う	仕分け・カット動作キーに設定されている入力項目が変わるごとにカットします。排出カットはレイアウトの発行時動作

	設定の「排出カットを使用する」に従って行います。
発行指示単位ごとに行う	最後のラベルのみカットします。排出カットはレイアウトの発行時動作設定の「排出カットを使用する」に従って行います。
指定した枚数ごとに行う	MultiCut「1 以上」と同じ動作

■例外

- JavaScript での例外処理は try…catch で実行してください。Error オブジェクトの message プロパティにメッセージが格納されます。

例外	説明 (message)
ArgumentOutOfRangeException	条件：範囲外の設定値 The value of MultiCut property is invalid. This property must be within the range from -2 to 9999.

■使用例

発行時に 3 枚単位でカットします。

```
MLWebComponent.MultiCut = 3;
Result = MLWebComponent.Output();
if ( Result != 0 ) {
    //エラー処理
}
```

■関連項目

プロパティ [LayoutFile](#)、[EjectCut](#)
メソッド [Output](#)、[Cut](#)

■サポート情報

付録 [サポートプリンタ](#)

2-13**SortMark プロパティ**

仕分けマークの印字を設定、取得します。

■書式MLWebComponent.SortMark *As Boolean***■設定値**

SortMark	説明
True	仕分けマークを印字する
False	仕分けマークを印字しない（初期値）

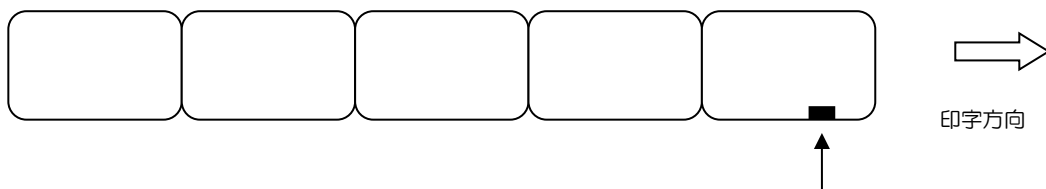
■解説

- レイアウト情報の仕分けマーク印字設定は使用されません。

■使用例

仕分けマークを印字します。

```
MLWebComponent.SortMark = True;
Result = MLWebComponent.Output();
if ( Return != 0 ) {
    //エラー処理
}
```



一発行単位の先頭のタグに印をつけます
スタッカをセットするとより効果的な運用が可能です。

■関連項目

プロパティ [LayoutFile](#)
メソッド [Output](#)

■サポート情報

付録 [サポートプリンタ](#)

2-14

[動作]EjectCut プロパティ

排出カットを設定、取得します。

■書式

MLWebComponent.EjectCut [As Boolean](#)

■設定値

<i>EjectCut</i>	説明
True	排出カットを行う
False	排出カットを行わない（初期値）

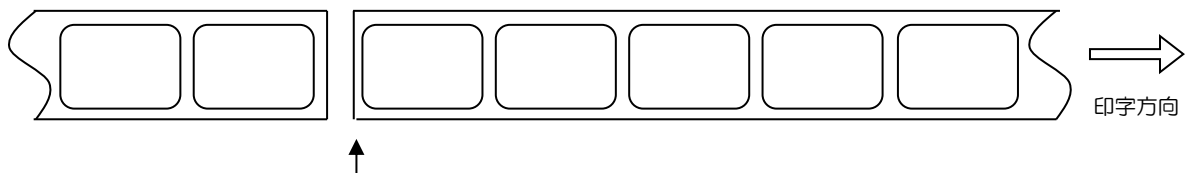
■解説

- MultiCut プロパティが「-1」または「-2」の場合、本プロパティは使用されません。
- レスプリV シリーズ、レスプリV-ex シリーズは、プリンタの動作モードがパーシャルカットモードで、MultiCut プロパティが「0」の場合、排出カットが全カットになります。

■使用例

排出カットを行います。

```
MLWebComponent.EjectCut =True;
Result = MLWebComponent.Output();
if ( Result != 0 ){
    //エラー処理
}
```



EjectCut プロパティで「True」を設定すると一発行単位の末尾でカットします。

「False」を設定すると排出カットコマンドは送信されません。

■関連項目

プロパティ [MultiCut](#)
メソッド [Output](#)、[Cut](#)

■サポート情報

付録 [サポートプリンタ](#)

2-15

[動作]HeaderTailSetting プロパティ

ヘッダ・テール札の印字有無を設定・取得します。

■書式

MLWebComponent.HeaderTailSetting *As Boolean*

■設定値

<i>HeaderTailSetting</i>	説明
False	ヘッダ・テール札は出力しません。（初期値）
True	レイアウトの設定に従いヘッダ・テール札を出力します。

■解説

- ・ 「True」を指定した場合で、レイアウトの設定が「ページごとに出力する」に設定されている場合は、Output ごとにヘッダ・テール札を出力します。
- ・ Setting プロパティが DRV（プリンタドライバ）以外の場合は、本プロパティは無効となります。

■使用例

レイアウトの設定に従いヘッダ・テール札を出力します。

```

MLWebComponent.LayoutFile = "C:¥¥SATO¥¥ABC.mllayx";
MLWebComponent.SetPrnDataRow(0, "商品 A" + "¥t" + "1");
MLWebComponent.SetPrnDataRow(1, "商品 B" + "¥t" + "2");
MLWebComponent.SetPrnDataRow(2, "商品 C" + "¥t" + "1");
MLWebComponent.SetPrnDataRow(3, "商品 D" + "¥t" + "1");
MLWebComponent.HeaderTailSetting = True;
Result = MLWebComponent.Output();
if ( Result != 0 ){
    //発行エラー
}

```

■関連項目

プロパティ [LayoutFile](#)、 [PrnData](#)
メソッド [SetPrnDataRow](#)、 [Output](#)

2-16**[動作]HeaderFile プロパティ**

ヘッダ札ファイルを取得します。設定はできません。

■書式

MLWebComponent.HeaderFile [As String](#)

■設定値

<i>HeaderFile</i>	説明
文字列 (パス名)	ヘッダ札ファイルのローカルファイルパス名
空文字	取得失敗 レイアウトファイルにヘッダ札が設定されていない

■解説

- LayoutFile プロパティで設定されたレイアウトファイルから取得します。

■使用例

ヘッダ札を取得します。

```
HeaderFile = MLWebComponent.HeaderFile;
if ( HeaderFile != "" ){
    window.alert( "HeaderFile = " + HeaderFile );
} else {
    window.alert( "HeaderFile Error !! " );
}
```

■関連項目

プロパティ [LayoutFile](#)、[HeaderTailSetting](#)、[TailFile](#)
メソッド [OutputHeader](#)

2-17**[動作]TailFile プロパティ**

テール札ファイルを取得します。設定はできません。

■書式

MLWebComponent.TailFile [As String](#)

■設定値

<i>TailFile</i>	説明
文字列 (パス名)	テール札ファイルのローカルファイルパス名
空文字	取得失敗 レイアウトファイルにテール札が設定されていない

■解説

- LayoutFile プロパティで設定されたレイアウトファイルから取得します。

■使用例

テール札を取得します。

```
TailFile = MLWebComponent.TailFile;
if ( TailFile != "" ){
    window.alert( "TailFile = " + TailFile );
}else{
    window.alert( "TailFile Error !! " );
}
```

■関連項目

プロパティ [LayoutFile](#)、[HeaderTailSetting](#)、[HeaderFile](#)
メソッド [OutputTail](#)

2-18

[特殊]Formoverlay プロパティ

フォームオーバーレイを設定・取得します。

■書式

MLWebComponent.Formoverlay As Integer / SATO.MLWebComponent.FormoverlayTypes

■設定値

Formoverlay	説明
0 - FormoverlayTypes.None	フォームオーバーレイを使用しない（初期値）
1 - FormoverlayTypes.Save	フォームオーバーレイを登録する
2 - FormoverlayTypes.Load	フォームオーバーレイを呼び出す
3 - FormoverlayTypes.Auto	固定項目を自動登録して複数アイテムを発行する

■解説

- 「1」で登録する対象は Output で出力される全てのデータです。また登録可能数は 1 つです。2 つ以上登録しても、最後に登録したデータが有効になります。
- 多面取りまたは両面レイアウトで「1」または「2」を設定した場合は、発行時にエラーとなります。

■例外

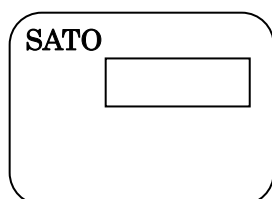
- JavaScript での例外処理は try…catch で実行してください。Error オブジェクトの message プロパティにメッセージが格納されます。

例外	説明 (message)
ArgumentOutOfRangeException	条件：範囲外の設定値 The value of Formoverlay property is invalid.

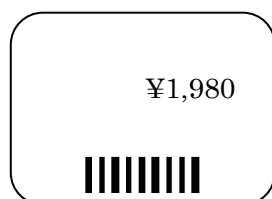
■使用例

フォームオーバーレイ登録後、フォームオーバーレイを呼び出します。

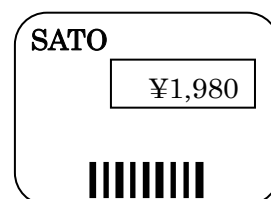
```
MLWebComponent.LayoutFile = "C:¥¥sato¥¥A.mllayx";
MLWebComponent.Formoverlay = 1;
Return = MLWebComponent.Output();
MLWebComponent.LayoutFile = " C:¥¥sato¥¥B.mllayx";
MLWebComponent.PrnData = "1980" + "¥t" + "1";
MLWebComponent.Formoverlay = 2;
Result = MLWebComponent.Output();
```



レイアウト「A.mllayx」



レイアウト「B.mllayx」



合成されたレイアウト

■関連項目

メソッド [Output](#)

■サポート情報

付録 [サポートプリンタ](#)

2-19

[特殊]LayoutNameCaption プロパティ

システム変数「レイアウト名」を設定・取得します。

■書式

MLWebComponent.LayoutNameCaption [As String](#)

■設定値

<i>LayoutNameCaption</i>	説明
空文字	LayoutFile プロパティに設定されているレイアウト名を使用します。(初期値)
任意文字列	システム変数「レイアウト名」に設定する任意文字列か、ローカルファイルパスを指定します。

■解説

- MLV5 のシステム変数「レイアウト名」に値が使用されます。
- 「ローカルファイルパス」指定の場合、ファイル名のみが「レイアウト名」になります。

■使用例

「レイアウト名」を設定します。

```
MLWebComponent.LayoutNameCaption = "C:¥¥SATO¥¥ABC.mllayx";
```

「レイアウト名」を取得します。

```
LayoutNameCaption = MLWebComponent.LayoutNameCaption;
```

■関連項目

プロパティ [LayoutFile](#)、 [TotalQtyCaption](#)
メソッド [Output](#)

2-20

[特殊]TotalQtyCaption プロパティ

システム変数「総発行枚数」を設定・取得します。

■書式

MLWebComponent.TotalQtyCaption [As String](#)

■設定値

TotalQtyCaption	説明
0	Output メソッド実行時に、総発行枚数を自動計算します。(初期値)
任意数値(0 以上)	システム変数「総発行枚数」を指定します。

■解説

- MLV5 のシステム変数「総発行枚数」に値が使用されます。

■例外

- JavaScript での例外処理は try…catch で実行してください。Error オブジェクトの message プロパティにメッセージが格納されます。

例外	説明 (message)
ArgumentOutOfRangeException	条件：範囲外の設定値 メッセージ：The value of TotalQtyCaption property is invalid.

■使用例

「総発行枚数」を設定します。

```
MLWebComponent.TotalQtyCaption = 24;
```

「総発行枚数」を取得します。

```
TotalQtyCaption = MLWebComponent.TotalQtyCaption;
```

■関連項目

プロパティ [LayoutFile](#)
メソッド [Output](#)

2-21

[特殊]TaxRate プロパティ

税率を設定・取得します。

■書式

MLWebComponent.TaxRate [As String](#)

■設定値

<i>TaxRate</i>	説明
空文字	税率を設定しない（初期値）
任意文字列	税率 1～20 をカンマ区切りで指定（範囲:0～99.9）

■解説

- 条件印字のデータ種類を「税率」に設定した場合に設定値が使用されます。
- レイアウトファイルで変数の編集パラメータで設定した税編集で「固定値を使用する」が有効な場合は使用されません。
- 値が設定されていない場合は、Output メソッドの戻り値が「413」となります。

■例外

- JavaScript での例外処理は try…catch で実行してください。Error オブジェクトの message プロパティにメッセージが格納されます。

例外	説明 (message)
ArgumentOutOfRangeException	条件：範囲外の設定値(税率が 20 以上) メッセージ：The value of TaxRate property is invalid.
	条件：範囲外の設定値(0～99.9 の範囲外) メッセージ：The value of TaxRate property is invalid. This property must be within the range from 0 to 99.9.

■使用例

税率 No.1 に「5.0」、税率 No.2 に「8.0」を設定します。

```
MLWebComponent.TaxRate =" 5.0,8.0";
```

現在の税率を取得します。

```
TaxRate = MLWebComponent.TaxRate;
```

■関連項目

プロパティ [LayoutFile](#)
メソッド [Output](#)

2-22**[バージョン]Version プロパティ**

バージョン情報を取得します。設定はできません。

■書式MLWebComponent.Version *As String***■設定値**

<i>Version</i>	説明
Multi LABELIST Web Component, Version x.x.x.x (x : バージョン番号)	MLWebComponent のバージョン情報を取得します。

■使用例

バージョン情報を取得します。

```
Version= MLWebComponent.Version;
```

取得結果 : " Multi LABELIST Web Component, Version 5.2.3.2"

第3章

MLWebComponent

メソッド・例外

3-1

[通信]OpenPort メソッド

通信ポートをオープンします。

■書式

MLWebComponent.OpenPort(ByVal *SyncMode* As Integer) As Integer

■パラメータ

パラメータ	設定値	説明
<i>SyncMode</i>	1 (固定)	同期通信モード

■戻り値

戻り値	説明
0	正常終了です。
1	Setting プロパティの値が不正です。
3	既にオープンされています。
4	ポートオープン時にエラーが発生しました。
12	通信設定と通信プロトコルの組み合わせが未サポートです。
31	ライセンス認証が実行されていません。
36	OpenPort実行時の日付が、認証ファイルの有効期限を超えています。

■解説

- WebComponent を利用するには、はじめに Authenticate メソッドで **ライセンス認証が必要**です。正しくライセンスが認証されていないと戻り値「31」、「36」となります。
- Setting プロパティで設定されたデバイスをオープンします。
- OpenPort 成功後は、Output メソッドで連続して発行することが可能です。Setting プロパティが LAN、USB、COM、Bluetooth の場合は、Output 実行前に毎回 GetStatus でプリンタの状態を確認してください。
- 戻り値「4」は、Setting プロパティで存在しないポート番号や IP アドレス、プリンタドライバ名を指定している場合や指定した USB デバイスや Bluetooth デバイスが見つからない場合、プリンタが他のプログラムで使用されている場合などに発生します。[Setting プロパティの解説もご参照ください。](#)
- Setting プロパティで Bluetooth を使用する場合は、Microsoft 標準の Bluetooth スタックのみ使用可能です。その他の Bluetooth スタックでは動作保証されていません。
- Setting プロパティで Bluetooth が設定されている場合は、指定した Bluetooth デバイスがペアリングされている必要があります。未ペアリングの場合には、PC 側に表示される Windows 標準の設定画面でペアリングを行ってください。設定画面表示中は OpenPort の処理が返送されません。設定画面は操作しないと 30~40 秒程でエラーとなり、OpenPort が戻り値「4」となります。
- [注意事項「インターフェースについて」を必ずご確認ください。](#)

■使用例

通信ポートのオープンからクローズまでのプログラム例を示します。

```
MLWebComponent.Setting = "COM1:9600,n,8,1";
Result = MLWebComponent.OpenPort(1);
if ( Result = 0 ){
    //一連の出力処理
    MLWebComponent.ClosePort();
}
```

■関連項目

プロパティ

[Setting](#)、[Protocol](#)、[Timeout](#)

メソッド

[Output](#)、[GetStatus](#)、[Cut](#)、[SendCancel](#)、[SendStringData](#)、[AuthenticateBluetoothDevice](#)

3-2**[通信]ClosePort メソッド**

通信ポートをクローズします。

■書式MLWebComponent.ClosePort() [As Integer](#)**■戻り値**

戻り値	説明
0	正常終了です。
5	ポートがオープンされていません。
6	ポートクローズ時にエラーが発生しました。

■使用例

通信ポートのオープンからクローズまでのプログラム例を示します。

```
MLWebComponent.Setting = "USB:";
Result = MLWebComponent.OpenPort(1);
if ( Result = 0 ){
    //一連の出力処理
    MLWebComponent.ClosePort();
}
```

■関連項目

プロパティ

[Setting](#)

メソッド

[OpenPort](#)

3-3**[発行]Output メソッド**

ラベル・タグを発行します。

■書式MLWebComponent.Output () [As Integer](#)**■戻り値**

戻り値	説明
0	正常終了です。
5	ポートがオープンされていません。
11	プリンタでエラーが発生しています（プリンタから NAK を受信しました）。 印字データはプリンタに送信されているため、エラー復旧後に発行される場合があります。 エラーの詳細は GetStatus メソッド で確認してください。 また、プリンタがスリープモードになっていると本エラーが発生する場合があります。 詳しくは「 プリンタのスリープについて 」を確認してください。
0 以外	「 通信エラー 」 「 発行エラー 」をご参照ください。

■解説

- OpenPort 成功後は、Output メソッドで連続して発行することが可能です。
- Setting プロパティで USB、LAN、COM、Bluetooth を指定する場合、出力メソッド（Output、SendStringData）を実行する前に GetStatus メソッドで必ずプリンタの状態をご確認下さい。
- [注意事項「インターフェースについて」](#)を必ずご確認ください。

■使用例

連番印字を行います。

```
MLWebComponent.PrnData = "10";
Result = MLWebComponent.Output();

if ( Result != 0 ){
    //エラー処理
}
```

■関連項目

プロパティ [Setting](#)、[Protocol](#)、[Timeout](#)、[LayoutFile](#)、[PrnData](#)
メソッド [GetStatus](#)、[Cut](#)、[SendCancel](#)

■サポート情報

メソッド・例外 [通信エラー](#)
メソッド・例外 [発行エラー](#)

3-4

[発行]OutputHeader メソッド

ヘッダ札を発行します。

■書式

MLWebComponent.OutputHeader() *As Integer*

■戻り値

戻り値	説明
0	正常終了です。
5	ポートがオープンされていません。
11	プリンタでエラーが発生しています（プリンタから NAK を受信しました）。 印字データはプリンタに送信されているため、エラー復旧後に発行される場合があります。 エラーの詳細は GetStatus メソッド で確認してください。 また、プリンタがスリープモードになっていると本エラーが発生する場合があります。 詳しくは「 プリンタのスリープについて 」を確認してください。
0 以外	「 通信エラー 」 「 発行エラー 」 をご参照ください。

■解説

- レイアウトファイルに設定されているヘッダ札を発行枚数の指定に関わらず 1 枚のみ発行します。
- ヘッダ札を発行する場合は、Formoverlay プロパティの設定値は適用されません。

■使用例

ヘッダ札を発行します。

```
MLWebComponent.PrnData = "5";
Result = MLWebComponent.OutputHeader();
if ( Result != 0 ){
    //エラー処理
}
```

■関連項目

プロパティ [Setting](#)、[Protocol](#)、[Timeout](#)、[HeaderTailSetting](#)、[HeaderFile](#)、[TailFile](#)
メソッド [GetStatus](#)、[Cut](#)、[SendCancel](#)

■サポート情報

メソッド・例外 [通信エラー](#)
メソッド・例外 [発行エラー](#)

3-5

[発行]OutputTail メソッド

テール札を発行します。

■書式

MLWebComponent.OutputTail () *As Integer*

■戻り値

戻り値	説明
0	正常終了です。
5	ポートがオープンされていません。
11	プリンタでエラーが発生しています（プリンタから NAK を受信しました）。 印字データはプリンタに送信されているため、エラー復旧後に発行される場合があります。 エラーの詳細は GetStatus メソッド で確認してください。 また、プリンタがスリープモードになっていると本エラーが発生する場合があります。 詳しくは「 プリンタのスリープについて 」を確認してください。
0 以外	「 通信エラー 」「 発行エラー 」をご参照ください。

■解説

- レイアウトファイルに設定されているヘッダ札を発行枚数の指定に関わらず 1 枚のみ発行します。
- ヘッダ札を発行する場合は、Formoverlay プロパティの設定値は適用されません。

■使用例

テール札を発行します。

```
MLWebComponent.PrnData = "5";
Result = MLWebComponent.OutputTail();
if ( Result != 0 ){
    //エラー処理
}
```

■関連項目

プロパティ [Setting](#)、[Protocol](#)、[Timeout](#)、[HeaderTailSetting](#)、[HeaderFile](#)、[TailFile](#)
メソッド [GetStatus](#)、[Cut](#)、[SendCancel](#)

■サポート情報

メソッド・例外 [通信エラー](#)
メソッド・例外 [発行エラー](#)

3-6

[発行]SendStringData メソッド

終了条件を指定して、プリンタコマンド（SBPL）を送信します。

■書式

MLWebComponent.SendStringData (

ByVal *Type* As Integer
 ByVal *Command* As String,
 ByVal *Length* As Integer,
 ByVal *EndChar* As String) As Variant

■パラメータ

パラメータ	設定値	説明
<i>Type</i>	0	受信データをテキスト(String 型)で取得
	1	受信データをバイト配列で取得
	2	受信データを 16 進文字コード(String 型)で取得
<i>Command</i>	任意文字列	送信データ（プリンタコマンド）を文字列で指定
<i>Length</i>	任意数値	受信処理を終了するバイト数を指定
<i>EndChar</i>	任意キャラクタ	受信処理を終了するキャラクタを指定

■戻り値

取得データ 設定値に応じたデータを返します。

■例外

詳細は「[MLWebComponentException クラス](#)」をご参照ください。

例外	説明
5	ポートがオープンされていません。
7	コマンド送信中にエラーが発生しました。
8	コマンド送信中にタイムアウトが発生しました。
9	応答受信中にエラーが発生しました。
10	応答受信中にタイムアウトが発生しました。
54	コマンド文字列が空です。
55	カラープリンタドライバへの出力はできません。

■解説

- Type* の「1」（バイト配列）は、使用言語や Internet Explorer のバージョンによって使用できない場合があります。バイナリデータを取得する場合は、3（16 進文字コード）で取得することを推奨します。
- 終了条件の組み合わせ（*Length*、*EndChar*）による本メソッドの動作は以下となります。

Length	EndChar	説明
--------	---------	----

0	指定なし	コマンド送信後の受信は行いません。 戻り値は空文字(バイト配列の場合は要素数が0)となります。
0	指定あり	EndChar で指定したキャラクタを受信するまで待機します。
1 以上	指定なし	Length で指定したバイト数を受信するまで待機します。
1 以上	指定あり	Length で指定したバイト数を受信するか、指定されたキャラクタを受信するまで待機します。

- Setting プロパティが USB で Length に 1 以上を指定した場合は、1 バイト以上の受信がされるまで待機し、Length 以上または EndChar で指定したキャラクタが含まれるメッセージまで受信して戻り値に設定します。
- Setting プロパティが DRV (プリンタドライバ) または FILE の場合、Length および EndChar の終了条件は使用されません。
- 正常に送信できた場合エラーとはなりません、プリンタがどのような動作を行うかは送信したプリンタコマンドに依存します。不正なコマンドを送信した場合の動作については責任を負いかねますのでご了承ください。
- 終了条件が一致しない場合は、Timeout プロパティの指定時間で受信タイムアウトになります。
- 印字コマンドを送信する場合、1 アイテムとしてください。1 度に複数アイテムを送信した場合、2 つ目以降のアイテムに関してはデータ送信が保証されません。

■使用例

任意の制御コマンドをプリンタに送信します。

```
var RecvData;
var Command;
//中略(コマンド生成) JavaScript では「%x@@」(@は 16 進数)でバイナリデータも指定可能です。
try{
    RecvData = MLWebComponent.SendStringData(0,Command,10, "%x06");
    //データ解析
} catch(e){
    if ( e.Number == 5 ){
        //ポートオープンエラー
    } else if ( e.Number == 54){
        //コマンド文字列エラー
    }
}
```

■関連項目

プロパティ [Timeout](#)

■サポート情報

メソッド・例外 [通信エラー](#)
メソッド・例外 [発行エラー](#)

3-7

[発行]GetStatus メソッド

プリンタの状態確認を行います。

■書式

MLWebComponent.GetStatus() [As String](#)

■戻り値

戻り値	説明
ステータス文字列	プリンタの状態を表すステータス文字列
空文字	状態確認に失敗

■例外

詳細は「[MLWebComponentException クラス](#)」をご参照ください。

例外	説明
5	ポートがオープンされていません。
7	コマンド送信中にエラーが発生しました。
8	コマンド送信中にタイムアウトが発生しました。
9	応答受信中にエラーが発生しました。
10	応答受信中にタイムアウトが発生しました。
55	プリンタドライバ出力、ファイル出力では使用できません。

■使用例

プリンタのステータスを取得します。

```
try{
    Status = MLWebComponent.GetStatus();
    if ( Status.substring(3,4) == "A"){
        Result = MLWebComponent.Output();
        if ( Result != 0 ){
            //エラー処理
        }
    }
} catch(e){
    ErrorNumber = ( e.number & 0xffff ) - 512;
    //取得エラー
}
```

■関連項目

プロパティ [Setting](#)、[Protocol](#)、[Timeout](#)
メソッド [OpenPort](#)

■サポート情報

メソッド・例外

[通信エラー](#)

3-8

[制御]Cut メソッド

用紙の排出カットを行います。

■書式

MLWebComponent.Cut() [As Integer](#)

■戻り値

戻り値	説明
0	正常終了です。
5	ポートがオープンされていません。
11	プリンタでエラーが発生しています（プリンタから NAK を受信しました）。 印字データはプリンタに送信されているため、エラー復旧後に発行される場合があります。エラーの詳細は GetStatus メソッド で確認してください。 また、プリンタがスリープモードになっていると本エラーが発生する場合があります。 詳しくは「 プリンタのスリープについて 」を確認してください。
53	カットコマンドがサポートされていないプリンタ機種です。
0 以外	「 通信エラー 」をご参照ください。

■解説

- ・ 印字動作終了後、用紙がカットされずに停止している状態の時に有効です。
- ・ LayoutFile プロパティの指定後に本メソッドを実行してください。

■使用例

発行毎にカットを行います。

```
for( n = 0 ; n < 5 ; n++ ){
    Result = MLWebComponent.Output();
    if ( Result != 0 ){
        break;
    }
    Result = MLWebComponent.Cut();
    if ( Result != 0 ){
        break;
    }
}
```

■関連項目

プロパティ [LayoutFile](#)
メソッド [Output](#)

■ サポート情報

メソッド・例外

[通信エラー](#)

3-9

【制御】SendCancel メソッド

発行のキャンセルを行います。

■書式

MLWebComponent.SendCancel() *As Integer*

■戻り値

戻り値	説明
0	正常終了です。
5	ポートがオープンされていません。
11	プリンタでエラーが発生しています（プリンタから NAK を受信しました）。 印字データはプリンタに送信されているため、エラー復旧後に発行される場合があります。エラーの詳細は GetStatus メソッド で確認してください。 また、プリンタがスリープモードになっていると本エラーが発生する場合があります。 詳しくは「 プリンタのスリープについて 」を確認してください。
0 以外	「 通信エラー 」をご参照ください。

■解説

- 発行動作を終了させ、プリンタが受信しているデータをすべてクリアします。
SendCancel 実行後は、500ms 以上経過してから次の Output メソッドを実行してください。

■使用例

発行エラーが発生した場合にキャンセルするか促します。

```
Result1 = MLWebComponent.Output();
if ( Result1 != 0 ){
    if (window.confirm("発行時にエラーが発生しました。キャンセルしますか?")){
        Result2 = MLWebComponent.SendCancel();
    }
}
```

■関連項目

プロパティ [Setting](#)
メソッド [Output](#)、[SendStringData](#)、[GetStatus](#)、[Cut](#)

■サポート情報

メソッド・例外 [通信エラー](#)

3-10

[ダウンロード]GetFile メソッド

サーバから HTTP 通信でファイルをダウンロードします。

■書式

MLWebComponent.GetFile(

ByVal *URL* As String,

ByVal *LocalPath* As String,

ByVal *Param* As Integer) As Integer

■パラメータ

パラメータ	設定値	説明
<i>URL</i>	任意文字列	ファイルを公開している URL を指定します。
<i>LocalPath</i>	任意文字列	ダウンロード先のローカルフォルダパスを指定します。
<i>Param</i>	0	形式を変えずにダウンロードします。
	1	CAB 形式ファイルの場合、ダウンロード後にファイルを展開します。
	2	サーバにあるファイルの更新日付が、ローカルの同一ファイルより新しい場合にダウンロードします。
	3	サーバにあるファイルの更新日付が、ローカルの同一ファイルより新しい場合にダウンロードします。CAB 形式ファイルの場合、ファイルを展開します。

■戻り値

戻り値	説明
0	正常終了です。
-1	レイアウトファイル名が不正です。
-2	プロトコルが HTTP および HTTPS ではありません。
-3	指定した URL の記述が不正です。(URL の書式ではありません。)
-5	ダウンロード実行中に予期せぬエラーが発生しました。 ①HTTP のステータスを取得することができません。 ②HTTPS でサーバ証明書が不正です。(認証エラー) ③指定した URL にアクセスできません。 ④ファイルの最終更新日時を示すヘッダ情報「Last-Modified」が取得できません。 ⑤.NET Framework を 4.5 以降にバージョンアップしてください。 ⑥HTTP から HTTPS にリダイレクトされました。URL を HTTPS にしてください。
-6	受信ファイルがオープンできません。 セキュリティ設定を確認してください。
300 以上	HTTP 要求エラー (HTTP ステータスコード)。

■解説

- 接続する URL がプロキシ経由(インターネット)の場合、インターネットオプションで設定を行ってください。設定がない場合(IE での閲覧ができない場合)は、接続できませんのでご注意ください。

ローカル(イントラネット)の場合は必要ありません。

- WebサイトにBASIC認証が設定されている場合は、GetFileCredential メソッドをご利用ください。GetFile メソッドでは戻り値が「401」となります。
- Webサイトにクライアント証明書認証が設定されている場合は、GetFile メソッドの前に SetClientCertificate メソッドを実行してください。GetFile メソッドでは戻り値が「403」となります。
- クライアント側のプロキシにBASIC認証を設定している場合は、GetFile メソッドの前に SetProxyCredential メソッドを実行してください。GetFile メソッドでは戻り値が「407」となります。
- Ver.5.9.1.0以降は、TLS 1.1、TLS 1.2への対応により、実行環境に.NET Framework 4.5以降が必要です。.NET Framework 4.0の環境では戻り値が「-5」となります。

■使用例

レイアウトファイルをダウンロードします。

```
Result = MLWebComponent.GetFile("http://example.com/label.mllayx", "C:¥¥TEMP¥¥SATO¥¥", 3);
if ( Result != 0 ){
    //ダウンロードエラー
}
```

■関連項目

プロパティ
メソッド

[LayoutFile](#)
[Output](#)、[GetFileCredential](#)、[SetProxyCredential](#)、[SetClientCertificate](#)、
[GetClientCertCount](#)、[GetClientCertInfo](#)

3-11

[ダウンロード]GetFileCredential メソッド

BASIC 認証が必要な Web サイトから HTTP 通信でファイルをダウンロードします。

■書式

MLWebComponent.GetFileCredential(

ByVal *URL* As String,
 ByVal *UserName* As String,
 ByVal *Password* As String,
 ByVal *Domain* As String,
 ByVal *LocalPath* As String,
 ByVal *Param* As Integer) As Integer

■パラメータ

パラメータ	設定値	説明
<i>URL</i>	任意文字列	ファイルを公開している URL を指定します。
<i>UserName</i>	任意文字列	Web サイトの資格情報のユーザー名を指定します。
<i>Password</i>	任意文字列	Web サイトの資格情報のパスワードを指定します。
<i>Domain</i>	任意文字列	Web サイトの資格情報のドメイン名を指定します。
<i>LocalPath</i>	任意文字列	ダウンロード先のローカルフォルダパスを指定します。
<i>Param</i>	0	形式を変えずにダウンロードします。
	1	CAB 形式ファイルの場合、ダウンロード後にファイルを展開します。
	2	サーバにあるファイルの更新日付が、ローカルの同一ファイルより新しい場合にダウンロードします。
	3	サーバにあるファイルの更新日付が、ローカルの同一ファイルより新しい場合にダウンロードします。CAB 形式ファイルの場合、ファイルを展開します。

■戻り値

戻り値	説明
0	正常終了です。
-1	レイアウトファイル名が不正です。
-2	プロトコルが HTTP および HTTPS ではありません。
-3	指定した URL の記述が不正です。(URL の書式ではありません。)
-5	①HTTP のステータスを取得することができません。 ②HTTPS でサーバ証明書が不正です。(認証エラー) ③指定した URL にアクセスできません。 ④ファイルの最終更新日時を示すヘッダ情報「Last-Modified」が取得できません。 ⑤.NET Framework を 4.5 以降にバージョンアップしてください。
-6	受信ファイルがオープンできません。 セキュリティ設定を確認してください。
300 以上	HTTP 要求エラー (HTTP ステータスコード)。

■解説

- 接続する URL がプロキシ経由(インターネット)の場合、インターネットオプションで設定を行ってください。設定がない場合(IE での閲覧ができない場合)は、接続できませんのでご注意ください。ローカル(イントラネット)の場合は必要ありません。
- Web サイトに BASIC 認証を設定している場合に使用します。NTLM 認証、ネゴシエート認証、および Kerberos ベースの認証では Windows 資格情報を使用するため GetFile メソッドをご使用ください。
- Web サイトにクライアント証明書認証が設定されている場合は、GetFileCredential メソッドの前に SetClientCertificate メソッドを実行してください。GetFileCredential メソッドでは戻り値が「403」となります。
- クライアント側のプロキシに BASIC 認証を設定している場合は、GetFileCredential メソッドの前に SetProxyCredential メソッドを実行してください。GetFileCredential メソッドでは戻り値が「407」となります。
- Ver.5.9.1.0 以降は、TLS 1.1、TLS 1.2 への対応により、実行環境に .NET Framework 4.5 以降が必要です。 .NET Framework 4.0 の環境では戻り値が「-5」となります。

■使用例

レイアウトファイルをダウンロードします。

```
Result = MLWebComponent.GetFileCredential("http://example.com/label.mllayx", "user1234",
"pw5678", "", "C:¥¥TEMP¥¥SATO¥¥", 3);
if ( Result != 0 ){
    //ダウンロードエラー
}
```

■関連項目

プロパティ [LayoutFile](#)
メソッド [Output](#)、 [GetFile](#)、 [SetProxyCredential](#)、 [SetClientCertificate](#)、
 [GetClientCertCount](#)、 [GetClientCertInfo](#)

3-12**[ダウンロード]SetProxyCredential** メソッド

クライアント側でプロキシ認証が必要な場合にあらかじめ資格情報を設定します。

■書式

```
MLWebComponent.SetProxyCredential(
    ByVal UserName As String,
    ByVal Password As String,
    ByVal Domain As String) As Integer
```

■パラメータ

パラメータ	設定値	説明
<i>UserName</i>	任意文字列	Web サイトの資格情報のユーザー名を指定します。
<i>Password</i>	任意文字列	Web サイトの資格情報のパスワードを指定します。
<i>Domain</i>	任意文字列	Web サイトの資格情報のドメイン名を指定します。

■戻り値

戻り値	説明
0	正常終了です。
その他	設定値が不正です。

■解説

- クライアント側のプロキシサーバに BASIC 認証を設定している場合に、GetFile メソッドまたは GetFileCredential メソッドよりも先に実行してください。
- 設定値は MLWebComponent のインスタンスごとに有効で、インスタンスが破棄されると設定値も破棄されます。

■使用例

プロキシ認証の資格情報を設定します。

```
Result = MLWebComponent.SetProxyCredential("user0123", "pw4567", "");
if ( Result != 0 ){
    //資格情報設定エラー
}
```

■関連項目

プロパティ [LayoutFile](#)
メソッド [Output](#)、[GetFile](#)、[GetFileCredential](#)

3-13

[ダウンロード]SetClientCertificate メソッド

ダウンロード時に使用するクライアント証明書を設定します。

■書式

```
MLWebComponent.SetClientCertificate(
    ByVal SubjectName As String,
    ByVal IssuerName As String,
    ByVal IsMultiErr As Boolean) As Integer
```

■パラメータ

パラメータ	設定値	説明
<i>SubjectName</i>	任意文字列	検索条件の発行先を指定します
<i>IssuerName</i>	任意文字列	検索条件の発行者を指定します。
<i>IsMultiErr</i>	true	該当する証明書が複数ある場合にエラーとします。
	false	該当する証明書が複数ある場合は、有効期限が最も遅い証明書を使用します。

■戻り値

戻り値	説明
0	正常終了です。
80	該当する証明書が存在しません。
81	該当する証明書が複数あります。(IsMultiErr が true の場合)

■解説

- Web サイトにクライアント証明書認証が設定されている場合に、GetFile メソッドまたは GetFileCredential メソッドよりも先に実行してください。
- 設定値は MLWebComponent のインスタンスごとに有効で、インスタンスが破棄されると設定値も破棄されます。
- 現在のログインユーザで個人証明書として有効な証明書から、検索条件で指定した発行先、発行者に部分一致する証明書を検索して設定します。検索した結果、複数の証明書が該当する場合は、パラメータ IsMultiErr の設定値に従ってエラーまたは設定完了とします。
- Windows 標準の証明書マネージャー ツール(Certmgr.exe)で[個人]-[証明書]から参照できる証明書と同じ情報から検索します。有効期限が切れている証明書は除外します。

■使用例

クライアント証明書を取得します。

```
// クライアント証明書設定
Result = MLWebComponent.SetClientCertificate("subject", "issuer", false);
If (Result != 0 ){
    //設定エラー
}
```

■関連項目

メソッド [GetFile](#)、[GetFileCredential](#)、[GetClientCertCount](#)、[GetClientCertInfo](#)

3-14

[ダウンロード]GetClientCertCount メソッド

現在のログインユーザで有効なクライアント証明書の数を取得します。

■書式

MLWebComponent.GetClientCertCount() *As Integer*

■戻り値

戻り値	説明
1 以上の整数	有効なクライアント証明書の数返します。
0	有効なクライアント証明書が見つかりません。

■解説

- 現在のログインユーザで個人証明書として有効な証明書の数を取得します。有効期限が切れている証明書は除外します。
- Windows 標準の証明書マネージャー ツール(Certmgr.exe)で[個人]-[証明書]から参照できる証明書と同じ情報が取得できます。
- GetClientCertInfo メソッドと組み合わせて使用します。

■使用例

クライアント証明書の情報を取得します。

```
CertCount = MLWebComponent.GetClientCertCount;           //クライアント証明書数取得
for(var i = 0; i < CertCount - 1; i++){                  //クライアント証明書数ループ
    CertInfo = MLWebComponent.GetClientCertInfo(n);      //クライアント証明書情報の取得
    //取得したクライアント証明書情報を使用する
}
```

■関連項目

メソッド

[GetFile](#)、[GetFileCredential](#)、[SetClientCertificate](#)、[GetClientCertInfo](#)

3-15

[ダウンロード]GetClientCertInfo メソッド

現在のログインユーザで有効なクライアント証明書を取得します。

■書式

MLWebComponent.GetClientCertInfo(ByVal Index As Integer) As String

■パラメータ

パラメータ	設定値	説明
<i>Index</i>	任意数値	取得するクライアント証明書のインデックス

■戻り値

戻り値	説明
(入力情報)	指定したインデックスのクライアント証明書情報 (カンマ区切り)
空文字	取得に失敗

- 入力情報のフォーマット

- ①発行先 文字列
- ②発行者 文字列
- ③有効期限 文字列
- ④フレンドリ名 文字列
- ⑤秘密鍵の有無 true (あり) / false (なし)

■解説

- 現在のログインユーザで個人証明書として有効な証明書の情報を取得します。有効期限が切れている証明書は除外します。
- Windows 標準の証明書マネージャー ツール(Certmgr.exe)で[個人]-[証明書]から参照できる証明書と同じ情報が取得できます。
- GetClientCertCount メソッドと組み合わせて使用します。

■使用例

クライアント証明書の情報を取得します。

```

CertCount = MLWebComponent.GetClientCertCount;           //クライアント証明書数取得
for(var i = 0; i < CertCount - 1; i++){                  //クライアント証明書数ループ
    CertInfo = MLWebComponent.GetClientCertInfo(i);      //クライアント証明書情報の取得
    //取得したクライアント証明書情報を使用する
}

```

■関連項目

メソッド

[GetFile](#)、[GetFileCredential](#)、[SetClientCertificate](#)、[GetClientCertCount](#)

3-16**[情報]GetPrinter メソッド**

レイアウトファイルからプリンタ情報を取得します。

■書式MLWebComponent.GetPrinter() [As String](#)**■戻り値**

戻り値	説明
文字列	レイアウトファイルに設定されているプリンタ機種名
空文字	取得に失敗

■解説

- LayoutFile プロパティで設定されたレイアウトファイルから取得します。

■使用例

レイアウトファイル"C:¥SATO¥ABC.mllayx"のプリンタ機種を取得します。

```
MLWebComponent.LayoutFile = "C:¥SATO¥ABC.mllayx";
PrinterModel = MLWebComponent.GetPrinter();
if ( PrinterModel != "" ){
    //正常に取得
} else {
    //取得エラー
}
```

■関連項目プロパティ [LayoutFile](#)

3-17

[情報] GetLayoutFileProperty メソッド

レイアウトファイルからプリンタ情報を取得します。

■書式

MLWebComponent.GetLayoutFileProperty() [As String](#)

■戻り値

戻り値	説明
(レイアウト情報)	カンマ区切りのレイアウト情報
空文字	取得に失敗

- レイアウト情報のフォーマット

①センサタイプ

「0」 アイマークラベル、「1」ギャップラベル、
「2」センターホールタグ、「3」横ホールタグ、
「4」アイマークタグ、「5」角Rタグ、
「-1」その他（カラープリンタなど）

②用紙サイズ

横サイズ(タブ)縦サイズ 単位:mm(小数点以下 1 桁固定)

③1 シートの枚数

多面取り設定の枚数

④グローバル情報ファイルパス

フルパスのローカルファイルパス

■解説

- LayoutFile プロパティで設定されたレイアウトファイルから取得します。

■使用例

レイアウトファイルの情報を取得します。

```
MLWebComponent.LayoutFile = "C:¥¥SATO¥¥ABC.mllayx";
LayoutInfo = MLWebComponent.GetLayoutFileProperty();
if (LayoutInfo != "") {
    //正常に取得
} else {
    //取得エラー
}
```

■関連項目

プロパティ

[LayoutFile](#)

3-18**[データ]GetInputCount メソッド**

レイアウトファイルの入力項目の数を取得します。

■書式MLWebComponent.GetInputCount() [As Integer](#)**■戻り値**

戻り値	説明
1 以上の整数	入力項目の数
0	取得に失敗

■解説

- LayoutFile プロパティで設定されたレイアウトファイルから取得します。
- GetInputFields メソッドと組み合わせて使用します。

■使用例

入力情報を取得します。

```
InputCount = MLWebComponent.GetInputCount();
For( n = 0 ; InputCount - 1 ; i++){
    InputStatus = MLWebComponent.GetInputFields(n);
    //取得した入力情報を使用する。
}
```

■関連項目

プロパティ [LayoutFile](#)
メソッド [GetInputFields](#)

3-19

[データ]GetInputFields メソッド

レイアウトファイルの入力情報を取得します。

■書式

MLWebComponent.GetInputFields(ByVal *Index* As Integer) As String

■パラメータ

パラメータ	設定値	説明
<i>Index</i>	任意数値	取得する入力情報のインデックス

■戻り値

戻り値	説明
(入力情報)	指定したインデックスのカンマ区切りの入力情報
空文字	取得に失敗

• 入力情報のフォーマット

- ①項目名 入力項目名称
- ②ヘッダ・行 0 (ヘッダ) / 1 (行)
- ③表示位置 Y(縦)X(横)H(高さ)W(幅) 単位:ピクセル
※行項目の場合、Y および X は 0 になります。
- ④桁数 入力桁数
- ⑤入力許可 0 (入力可) / 1 (入力不可) / 2 (非表示)
- ⑥発行後削除 0 (OFF) / 1 (ON)
- ⑦漢字入力 0 (OFF) / 1 (ON)
- ⑧仕分・カット 「0」OFF / 「1」ON
- ⑨入力チェック 0 (なし) / 1 (数値) / 2 (英字) / 3 (英数字) / 4 (半角)
5 (全角) / 10 (日付) / 11 (16進) / 12 (CODE39) / 13 (NW-7)
- ⑩入力画面文字種 P(ポイント数)N(フォント名)
- ⑪その他チェック 0 (なし) / 1 (CD) / 2 (テーブル) / 3 (丸め) / 4 (日付範囲)
5 (特殊) / 6 (条件)
- ⑫初期値 初期値文字列

■解説

- LayoutFile プロパティで設定されたレイアウトファイルから取得します。
- パラメータ InputStatus で取得できる入力情報の順番がレイアウトファイルの入力順です。PrnData で指定する項目順になります。
- GetInputCount メソッドと組み合わせて使用します。
- 入力画面文字種は、ML デザインの入力定義で設定した入力項目の表示用文字種です。ラベル上の文字フォントではありません。

■使用例

入力情報を取得します。

```
InputCount = MLWebComponent.GetInputCount();
for( n = 0 ; InputCount - 1 ; i++){
    InputStatus = MLWebComponent.GetInputFields(n);
    //取得した入力情報を使用する。
}
```

- 取得結果

品番,0,Y100X150H10W70,7,0,0,0,0,1, P12NMS Pゴシック,0,12345,
商品コード,1,YOX0H10W80,8,0,0,0,0,1,,P12NMS Pゴシック, 0,12345678
発行枚数,1,YOX0H10W50,6,0,0,0,0,0,,P12NMS Pゴシック,0,1

■関連項目

プロパティ

[LayoutFile](#)

メソッド

[GetInputCount](#)、[SetPrnDataField](#)

3-20

[データ]SetPrnDataField メソッド

指定した入力項目の印字データを設定します。

■書式

MLWebComponent.SetPrnDataField

(ByVal Name As String, ByVal Data As String) As Integer

■パラメータ

パラメータ	設定値	説明
Name	任意文字列	入力項目名称
Data	任意文字列	設定するデータ 文字フォーマットは MLV5 準拠で Unicode(UTF-16)

■戻り値

戻り値	説明
0	設定完了しました。
1	レイアウト情報ファイルのロード時にエラーが発生しました。 <ul style="list-style-type: none"> 指定されたレイアウトファイルが存在しません。 指定されたレイアウトファイルが開けません。 該当する入力項目が存在しません。
61	レイアウトファイルが MLWebComponent より新しいバージョンの MLV5 で作成されています。MLWebComponent のバージョンを更新してください。

■解説

- 同一名称が複数存在する場合は、入力順で最初の入力項目のみに印字データを設定します。
- LayoutFile プロパティを設定した時点で、設定したデータはすべてクリアされます。
- バーコードにチェックデジットを含めたデータを指定する際は、チェックデジットが正しいかお客様のアプリケーションで必ずチェックした上で指定してください。チェックが出来ない場合は、チェックデジットを含まないデータに変更するか、変数の設定でチェックデジット部分を取り除くようにレイアウトファイルを変更してください。
- 発行枚数は、最大 999999 (PW208 を除くラパンシリーズは 9999) まで指定可能です。
- PrnData プロパティと併用する場合は、混乱を避けるため PrnData プロパティを先に指定してください。
- PrnDataType プロパティで指定された形式の区切り文字はデータとして設定しないでください。例として、PrnDataType プロパティが「タブ区切り」の場合に、データ内にタブ (0x09) を入力すると、入力項目の入力順がずれて不正な入力データとなります。
- QR コード等のバーコードデータに制御コードを含める場合は、レイアウトの入力変数種類を「16 進文字コード」に設定し、上位アプリからは 16 進文字列のデータをセットしてください。(例：12345<TAB>の場合、「313233343509」をセット)

■使用例

各入力名の位置に、印字データをセットします。

```
Result = MLWebComponent.SetPrnDataField("品番", "12-34");           //品番セット
if ( Result != 0 ){
//設定エラー
}
Result = MLWebComponent.SetPrnDataField("商品 No", "012345");       //商品 No セット
if ( Result != 0 ){
//設定エラー
}
Result = MLWebComponent.SetPrnDataField("取引先名", "サトー");     //取引先名セット
if ( Result != 0 ){
//設定エラー
}

Result = MLWebComponent.SetPrnDataField("発行枚数", "1");          //発行枚数セット
if ( Result != 0 ){
//設定エラー
}
```

■関連項目

プロパティ [LayoutFile](#)、[PrnData](#)、[PrnDataType](#)
メソッド [Output](#)、[GetInputFields](#)

3-21**[データ]GetPrnDataRowCount メソッド**

設定されている印字データの行数を取得します。

■書式

MLWebComponent.GetPrnDataRowCount() [As Integer](#)

■戻り値

戻り値	説明
0以上の整数	設定されている印字データの行数

■使用例

データの行数を取得します。

```
RowCount = MLWebComponent.GetPrnDataRowCount();
```

■関連項目

プロパティ

[PrnData](#)

メソッド

[SetPrnDataField](#)、[GetPrnDataRow](#)、[SetPrnDataRow](#)

3-22**[データ]GetPrnDataRow メソッド**

指定された行に設定されている印字データを取得します。

■書式

MLWebComponent.GetPrnDataRow(ByVal *Index* As Integer) As String

■パラメータ

パラメータ	設定値	説明
<i>Index</i>	任意数値	データの行番号

■戻り値

戻り値	説明
(印字データ)	指定された行に設定されている印字データ
空文字	データが存在しません。

■解説

- LayoutFile プロパティを設定した時点で、入力データはすべてクリアされます。

■使用例

印字データを取得します。

```
InputData = MLWebComponent.GetPrnDataRow(2);
```

■関連項目

プロパティ [PrnData](#)、[PrnDataType](#)
メソッド [SetPrnDataField](#)、[GetPrnDataRowCount](#)、[SetPrnDataRow](#)

3-23

[データ]SetPrnDataRow メソッド

1 行分のデータを設定します。

■書式

MLWebComponent.SetPrnDataRow(ByVal *Data* As String) As Integer

■パラメータ

パラメータ	設定値	説明
<i>Data</i>	任意文字列	各項目を PrnDataType プロパティで指定に対応した文字の区切りで指定します。文字フォーマットは MLV5 準拠で Unicode(UTF-16)となります。

■戻り値

戻り値	説明
0 以上の数値	印字データを設定した行番号です。
-1	設定失敗しました。

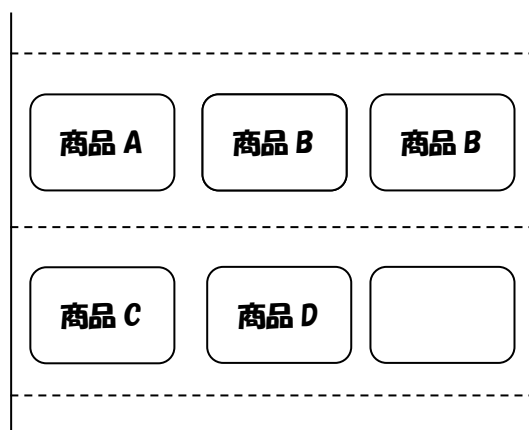
■解説

- LayoutFile プロパティを設定した時点で、設定したデータは初期化されます。
- 値を設定すると PrnData プロパティ、SetPrnDataField メソッドで設定したデータは初期化されます。
- バーコードにチェックデジットを含めたデータを指定する際は、チェックデジットが正しいかお客様のアプリケーションで必ずチェックした上で指定してください。チェックが出来ない場合は、チェックデジットを含まないデータに変更するか、変数の設定でチェックデジット部分を取り除くようにレイアウトファイルを変更してください。
- 発行枚数は、最大 999999 (PW208 を除くラパンシリーズは 9999) まで指定可能です。
- PrnDataType プロパティで指定された形式の区切り文字はデータとして設定しないでください。例として、PrnDataType プロパティが「タブ区切り」の場合に、データ内にタブ (0x09) を入力すると、入力項目の入力順がずれて不正な入力データとなります。
- QR コード等のバーコードデータに制御コードを含める場合は、レイアウトの入力変数種類を「16 進文字コード」に設定し、上位アプリからは 16 進文字列のデータをセットしてください。(例：12345<TAB>の場合、「313233343509」をセット)
- Setting プロパティが DRV (プリンタドライバ) 以外の場合は、1 シート以上とならない発行枚数を指定してください。
1 シート以上となる発行枚数を指定した場合は Output メソッドでエラーとなります。

■使用例

多面取りレイアウト(横 3 面取り)の複数データをセットします。

```
Result = MLWebComponent.SetPrnDataRow( "商品 A" & Chr$(9) & "1");
if ( Result == -1 ){
    //設定エラー
}
Result = MLWebComponent.SetPrnDataRow( "商品 B" & Chr$(9) & "2");
if ( Result == -1 ){
    //設定エラー
}
Result = MLWebComponent.SetPrnDataRow( "商品 C" & Chr$(9) & "1");
if ( Result == -1 ){
    //設定エラー
}
Result = MLWebComponent.SetPrnDataRow( "商品 D" & Chr$(9) & "1");
if ( Result == -1 ){
    //設定エラー
}
Result = MLWebComponent.Output();
if ( Result != 0 ){
    //発行エラー
}
```



■関連項目

プロパティ

[PrnData](#)、[PrnDataType](#)

メソッド

[Output](#)、[SetPrnDataField](#)、[GetPrnDataRowCount](#)、[GetPrnDataRow](#)

3-24**[データ]GetDriverNameList メソッド**

インストールされているプリンタドライバ名を取得します。

■書式MLWebComponent.GetDriverNameList(ByVal *Param* As Integer) As String**■パラメータ**

パラメータ	設定値	説明
<i>Param</i>	0	サトープリンタのみ
	1	すべてのプリンタ

■戻り値

戻り値	説明
(プリンタドライバ名)	カンマ区切りのプリンタドライバ名
空文字	取得に失敗

■解説

- *Param*が「0」(サトープリンタ) の場合、プリンタドライバのモデル名が「SATO」で始まる機種のみ取得します。

■使用例

プリンタドライバ名を取得します。

```
DriverName = MLWebComponent.GetDriverNameList(0);
```

■関連項目

- プロパティ [Setting](#)
- メソッド [OpenPort](#)、[ClearPrintJob](#)

3-25**[データ]ClearPrintJob メソッド**

プリンタドライバの印刷ジョブを削除します。

■書式MLWebComponent.ClearPrintJob(ByVal *DriverName* As String) As Integer**■パラメータ**

パラメータ	設定値	説明
<i>DriverName</i>	任意文字列	印刷ジョブを削除するプリンタドライバ名

■戻り値

戻り値	説明
0	印刷ジョブを削除しました。
21	プリンタドライバが指定されていません。 指定されたプリンタドライバがありません。
22	印刷ジョブの削除に失敗しました。 プリンタドライバを操作する権限がありません。 ユーザ権限で実行されている場合、管理者権限の印刷ジョブは削除できません。

■使用例

プリンタドライバの印刷ジョブを削除します。

```
Result = MLWebComponent.ClearPrintJob("SATO SG412R-ex");
if ( Result != 0 ){
    //削除エラー
}
```

■関連項目

メソッド

[GetDriverNameList](#)

3-26

[デバイス]EnumerateBluetoothDevices メソッド

Bluetooth デバイスの検索を実行します。

■書式

MLWebComponent.EnumerateBluetoothDevices(
ByVal SearchTime As Integer) As Variant

■パラメータ

パラメータ	設定値	説明
SearchTime	任意数値	検索時間の秒指定 (1~61)

■戻り値

戻り値	配列	説明
デバイス情報 (2次元配列)	行数	検出したデバイス数
	列 1	BD アドレス (12 桁の 16 進数)
	列 2	デバイス名
	列 3	ペアリングの有無 (True : ペアリング済み、False : 未ペアリング)

■解説

- PC 側の Bluetooth デバイスは、Microsoft 標準の Bluetooth スタック (Microsoft Bluetooth Enumerator) のみ使用可能です。その他の Bluetooth スタックでは動作保証されていません。
- 検索結果が 0 件、SearchTime に不正な値が指定されている、Bluetooth デバイスが存在しない、Bluetooth スタックが Microsoft 標準ではないなどの場合に、戻り値の要素数が「0」となります。
- ペアリング済みのデバイスは、実際にデバイスが見つからない状況 (電源が OFF または電波が届かない) でも検索結果に反映されます。

■使用例

デバイスを検索します。

```
var oMLCmp = document.getElementById("ctrlPrintLayout").object;
var arrayVar = oMLCmp.EnumerateBluetoothDevices(10);           // 10 秒間デバイスを検索
var dvArray = new VBArray(arrayVar);                             // 配列に変換
for(i=dvArray.lbound(); i <= dvArray.ubound(); i++){
    var address = dvArray.getItem(i,0);                          // BD アドレス取得
    var name = dvArray.getItem(i,1);                             // デバイス名取得
    var isauth = dvArray.getItem(i,2);                           // ペアリング有無取得
    alert(address + ", " + name + ", " + isauth);
}
```

- 出力結果
OOaabbccdde, SATO MOBILE PRINTER, False

■関連項目

メソッド [AuthenticateBluetoothDevice](#)

3-27

[デバイス]AuthenticateBluetoothDevice メソッド

Bluetooth デバイス認証（ペアリング）を実行します。

■書式

```
MLWebComponent.AuthenticateBluetoothDevice(
    ByVal Address As String,
    ByVal PIN As String,
    ByVal SearchTime As Integer) As Integer
```

■パラメータ

パラメータ	設定値	説明
<i>Address</i>	任意文字列	BD アドレス（12 桁の 16 進数）
<i>PIN</i>	任意文字列	Bluetooth デバイスの PIN コード
<i>SearchTime</i>	任意数値	検索時間の秒指定（1～48）

■戻り値

戻り値	説明
0	ペアリングが完了しました。
3	既にオープンされています。
700	PC に Bluetooth デバイスがありません（Bluetooth デバイスが無効です）。 Bluetooth スタックが Microsoft 標準ではありません。
701	不正な BD アドレスが指定されています。 指定された BD アドレスのデバイスが存在しません（検索タイムアウト）。
702	不正な PIN コードが指定されています。
703	指定された PIN コードで認証できません。（プリンタから認証を拒否されました。）

■解説

- PC 側の Bluetooth デバイスは、Microsoft 標準の Bluetooth スタック（Microsoft Bluetooth Enumerator）のみ使用可能です。その他の Bluetooth スタックでは動作保証されていません。
- 本メソッドはペアリングのみ行います。仮想 COM ポートへのサービス登録は行いません。
- Bluetooth Ver.3.0 のプリンタで認証レベルをレベル 1（認証なし）に設定している場合、本メソッドは必要ありません。
- 指定された時間だけ検索を行った後にペアリングを行います。デバイスが見つからない場合は、検索時間を長くして再実行してください。
- 指定された BD アドレスのデバイスがペアリング済みの場合は、一旦デバイス情報を削除してから再度ペアリングを行います。

■使用例

指定アドレスのデバイスをペアリングします。

```
var Address = "00aabbccdde";           //BD アドレスを設定
var PIN = "1234";                       //PIN を設定
var SearchTime = 3;                     //検索時間を設定
var result = MLComponent.AuthenticateBluetoothDevice (Address, PIN, Timeout);
if ( result == 0 ){
    //ペアリング成功
}
```

■関連項目

メソッド

[EnumerateBluetoothDevices](#)

3-28**[認証]Authenticate メソッド**

ライセンス認証を実行します。

■書式

MLWebComponent.Authenticate(

ByVal *Key* As String , ByVal *LocalPath* As String) As Integer**■パラメータ**

パラメータ	設定値	説明
<i>Key</i>	任意文字列	ライセンスファイルに対応した認証キーを指定します。
<i>LocalPath</i>	任意文字列	ライセンスファイルのローカルファイルパス（フルパス）を指定します。
	空文字	MLWebComponent と同じディレクトリを参照します。

■戻り値

戻り値	説明
0	認証成功
32	ライセンスファイルが存在しません。
33	ライセンスファイルにアクセスできません。
34	ライセンスファイルのデータが読み取れません。
35	ライセンスファイルが改ざんされています。
36	ライセンスファイルの有効期限を超えています。
37	認証キーがライセンスファイルと一致しません。
38	ライセンスファイルの対象アプリケーションが一致しません。

■解説

- MLWebComponent を使用する際に必ず指定してください。ライセンス認証がされていない場合は、OpenPort でエラーNo.31 となります。
- ライセンスファイルと認証キーの入手方法はワンポイントテクニック「[ライセンス認証を行う](#)」をご参照ください。

■使用例

ライセンス認証を実行します。

```
Key = "d6aa8272-16f8-4c83-9b14-cf4b8fd35f1c";
Result = MLWebComponent.Authenticate(Key, "");
if ( Result != 0 ){
    //認証エラー
}
```

■関連項目

プロパティ

[Setting](#)

メソッド

[OpenPort](#)

3-29**[例外] MLWebComponentException クラス**

MLWebComponent の実行エラーで通知される例外です。

■例外が発生するメソッド[GetStatus](#)、[SendStringData](#)**■名前空間**

SATO.MLWebComponent

■継承階層

System.Exception

↳SATO.MLWebComponentException

■Public プロパティ

プロパティ名	型	get	set	説明
Number	int	○	×	エラー番号を格納します。

■解説

- JavaScript での例外処理は try…catch で実行してください。Error オブジェクトの number プロパティにエラーコード、message プロパティにメッセージが格納されます。

■使用例

```
try{
  PrinterStatus = MLWebComponent.GetStatus();
  //データ解析
}catch(e){
  ErrorNumber = ( e.number & 0xffff ) - 512;
  if (ErrorNumber == 5 ){
    //ポートオープンエラー
  }
  //エラーコードに応じた処理
}
```

■エラー内容

エラー番号	Error オブジェクト エラーコード	説明 メッセージ
5	&H80040205	ポートがオープンされていません Port is not open.

7	&H80040207	コマンド送信中にエラーが発生しました。 An error occurred while sending command.
8	&H80040208	コマンド送信中にタイムアウトが発生しました。 An timeout occurred while sending command.
9	&H80040209	応答受信中にエラーが発生しました。 An error occurred while receiving response.
10	&H8004020A	応答受信中にタイムアウトが発生しました。 An timeout occurred while receiving response.
54	&H80040236	コマンド文字列が空です。 Command is empty.

3-30

[エラー]通信エラー

通信エラーについて説明します。

■通信エラー発生メソッド

[Output](#)、[OutputHeader](#)、[OutputTail](#)、[SendStringData](#)、[GetStatus](#)、[Cut](#)、[SendCancel](#)

■通信エラー一覧

通信エラー発生時は ClosePort メソッドで切断し、下記対策を行った上で再度 OpenPort メソッドを実行してください。

No.	内容	原因
7	コマンド送信中（ステータス要求中）にエラーが発生しました。	①②③④
8	コマンド送信中（ステータス要求中）にタイムアウトが発生しました。	⑤
9	応答受信中（ステータス受信中）にエラーが発生しました。	①②③④⑥⑦
10	応答受信中（ステータス受信中）にタイムアウトが発生しました。	

原因① プリンタの電源が入っていない。

⇒プリンタの電源を確認してください。

原因② ケーブルが差さっていない。

⇒PC とプリンタがしっかりとケーブルで接続されているか、ケーブルが断線していないか確認してください。

原因③ プリンタが既に使用されている。（LAN の場合のみ）

⇒他の PC やソフト、プリンタドライバ等がプリンタを使用していないか確認してください。

原因④ プリンタドライバが起動していない。

⇒コントロールパネルの管理ツールでサービスの「Print Spooler」が正常に起動しているかご確認ください。

原因⑤ ボーレートが異なる。（RS-232C の場合のみ）

⇒プリンタのボーレートと Setting プロパティの設定を合わせてください。

原因⑥ 通信プロトコルが異なる。

⇒プリンタの通信プロトコルと Protocol プロパティの設定を合わせてください。

LAN、ステータス4の場合は、必ず ENQ 応答モードに設定してください。

設定方法は付録「[プリンタ本体の通信プロトコル設定方法](#)」をご参照ください。

原因⑦ ネットワーク環境によって、通信の遅延が発生している。（LAN の場合のみ）

⇒Timeout プロパティの値を調整してください。（初期値 3 秒）

3-31

[エラー]発行エラー

発行エラーについて説明します。

■発行エラー発生メソッド

[Output](#)、[OutputHeader](#)、[OutputTail](#)

■発行エラー一覧

発行エラーはレイアウトファイルなど発行に必要な情報がない場合に発生します。エラー発生時は内容に応じた対処を行ってください。

No.	内容
	詳細
ファイルロードエラー：必要なファイルを用意してください。	
51	レイアウト情報ファイルのロード時にエラーが発生しました。
	レイアウトファイル（.mllayx）が存在しないか、読取りが出来ません。リクエストで指定したファイルパスをご確認ください。他のアプリケーションで利用されている場合、アクセス権限がない場合は読取りできません。 Windows アップデートの不具合により PC のフォント情報が読み取れません。更新プログラムを適用してください。 https://support.microsoft.com/ja-jp/help/4074906/
52	プリンタ情報のロード時にエラーが発生しました。
	レイアウトファイルで使用されているプリンタ機種が現在のバージョンでサポートされていません。MLWebComponent をバージョンアップしてください。
57	印字データが無効です。
	PrnDataType プロパティに設定したデータタイプと異なるデータが設定されています。指定しているデータをご確認ください。
61	レイアウトファイルのロード時にエラーが発生しました。
	レイアウトファイルが MLWebComponent より新しいバージョンの MLV5 で作成されています。MLWebComponent をバージョンアップしてください。
62	グローバル情報ファイルのロード時にエラーが発生しました。
	グローバル情報ファイルが MLWebComponent より新しいバージョンの MLV5 で作成されています。MLWebComponent をバージョンアップしてください。
63	ヘッダ札ファイルのロード時にエラーが発生しました。
	ヘッダ札ファイルが MLWebComponent より新しいバージョンの MLV5 で作成されています。MLWebComponent をバージョンアップしてください。
64	テール札ファイルのロード時にエラーが発生しました。
	テール札ファイルが MLWebComponent より新しいバージョンの MLV5 で作成されています。MLWebComponent をバージョンアップしてください。
65	グローバルテーブルファイルのロード時にエラーが発生しました。
	グローバルテーブルファイルが MLWebComponent より新しいバージョンの MLV5 で作成

	されています。MLWebComponent をバージョンアップしてください。
66	グローバルチェックテーブルファイルのロード時にエラーが発生しました。
	グローバルチェックテーブルファイルが MLWebComponent より新しいバージョンの MLV5 で作成されています。MLWebComponent をバージョンアップしてください。
100	グローバル情報ファイルのロード時にエラーが発生しました。
	レイアウトに設定されているグローバル情報ファイル (.mlglex) が存在しないか、読取りが出来ません。レイアウトに設定されているファイルパスをご確認ください。他のアプリケーションで利用されている場合、アクセス権限がない場合は読取りできません。
101	ヘッダ札ファイルのロード時にエラーが発生しました。
	レイアウトに設定されているヘッダ札レイアウトファイル (.mlhtlx) が存在しないか、読取りが出来ません。レイアウトに設定されているファイルパスをご確認ください。他のアプリケーションで利用されている場合、アクセス権限がない場合は読取りできません。
102	テール札ファイルのロード時にエラーが発生しました。
	レイアウトに設定されているテール札レイアウトファイル (.mlhtlx) が存在しないか、読取りが出来ません。レイアウトに設定されているファイルパスをご確認ください。他のアプリケーションで利用されている場合、アクセス権限がない場合は読取りできません。
103	グローバル情報ファイルの設定がヘッダ・テール札と一致しません。
	レイアウトとヘッダ・テール札レイアウトに設定されているグローバル情報ファイル (.mlgle) の設定をご確認ください。
110	レイアウトファイルの保存時にエラーが発生しました。
	レイアウトファイルの保存ができません。書き込みが禁止されていないかご確認ください。
111	グローバル情報ファイルの保存時にエラーが発生しました。
	グローバル情報ファイルの保存ができません。書き込みが禁止されていないかご確認ください。
152	ヘッダ札のプリンタタイプが一致していません。
	レイアウトとヘッダ札ファイル (.mlhtlx) のプリンタタイプ (サトープリンタ/カラープリンタ) を合せてください。
153	テール札のプリンタタイプが一致していません。
	レイアウトとテール札ファイル (.mlhtlx) のプリンタタイプ (サトープリンタ/カラープリンタ) を合せてください。
例外エラー	
201	プリンタコマンド生成以外の処理で予期しないエラーが発生しました。
	例外エラーについてをご参照ください。
202	プリンタコマンド生成処理中に予期しないエラーが発生しました。
	例外エラーについてをご参照ください。
対象変数エラー：対象となる変数を再度設定し直してください。	
300	日時変数の対象変数検索時にエラーが発生しました。
	日時変数の経過値に設定した変数がレイアウトに存在しません。
301	編集前複写変数の対象変数検索時にエラーが発生しました。
	編集前複写変数の複写対象に設定した変数がレイアウトに存在しません。
302	入力日時変数の対象変数検索時にエラーが発生しました。

	入力日時変数の経過値か位上対象に設定した変数がレイアウトに存在しません。
303	テーブル変換の対象変数検索時にエラーが発生しました。
	テーブル変換の変換対象キーに設定した変数がレイアウトに存在しません。
304	プライス丸め変数の対象変数検索時にエラーが発生しました。
	計算変数のプライス丸めで丸め対象に設定した変数がレイアウトに存在しません。
306	プライスチェックCD変数の対象変数検索時にエラーが発生しました。
	計算変数のプライスチェックCDで計算対象に設定した変数がレイアウトに存在しません。
307	クロスチェック変数の対象変数検索時にエラーが発生しました。
	分割変数のクロスチェックで比較対象に設定した変数がレイアウトに存在しません。
308	編集後複写変数の対象変数検索時にエラーが発生しました。
	編集後複写変数の複写対象に設定した変数がレイアウトに存在しません。
309	結合変数の結合子変数検索時にエラーが発生しました。
	結合変数の子項目に設定した変数がレイアウトに存在しません。
310	結合項目の親変数検索時にエラーが発生しました。
	結合変数の親項目となる変数がレイアウトに存在しません。
320	レイアウトに割り付いたローカル変数検索時にエラーが発生しました。
	デザインオブジェクトで設定したローカル変数がレイアウトに存在しません。
321	レイアウトに割り付いたグローバル変数検索時にエラーが発生しました。
	デザインオブジェクトで設定したグローバル変数がレイアウトに存在しません。
編集エラー：発行時動作設定のエラー処理で「グラフィック変換エラー」が有効な場合に発生します。	
330	貼付グラフィックで該当するグラフィックファイルがありません。
	指定されたパスにグラフィックがありません。設定したパスをご確認ください。
332	呼出グラフィックで該当する登録グラフィックがありません。
	指定した No.のグラフィックがグラフィックテーブルに登録されているかご確認ください。
333	変数グラフィックで指定したグラフィックがありません。
	指定先したパスやグラフィック No.に該当するグラフィックがあるかご確認ください。
編集エラー：発行時動作設定のエラー処理で括弧内に記載した設定を有効にした場合に発生します。	
400	日時変数編集時にエラーが発生しました。(日時項目編集エラー)
	日時の計算が出来ません。経過値を確認してください。
403	テーブル変換時にエラーが発生しました。(テーブル変換エラー)
	テーブル一覧に入力値に合致する値がありません。
404	ODBCテーブル変換時にエラーが発生しました。(ODBC テーブル変換エラー)
	テーブル一覧 (ODBC) に入力値に合致する値がありません。
405	プライス丸め計算時にエラーが発生しました。(プライス丸め変換エラー)
	プライス丸め計算が出来ません。丸め対象変数の入力値を確認してください。
406	四則演算計算時にエラーが発生しました。(数式計算エラー)
	四則演算が出来ません。入力値を確認してください。
407	プライスチェックCD計算時にエラーが発生しました。(プライスC/D 計算エラー)
	プライスチェックCD 計算が出来ません。対象変数の入力値を確認してください。
408	¥・カンマ編集時にエラーが発生しました。(通貨カンマ編集エラー)

	テーブル変換や計算などで¥・カンマを付加するための桁数が足りません。入力値や変数の桁数を確認してください。
409	1文字補填編集時にエラーが発生しました。(1文字補填編集エラー)
	テーブル変換や計算などで1文字補填するための桁数が足りません。入力値や変数の桁数を確認してください。
410	CD計算時にエラーが発生しました。(C/D計算エラー)
	モジュラス11でCD置換をエラー判定にした場合、置換できないCD値の場合にエラーが発生します。入力値を確認してください。
411	特殊編集時にエラーが発生しました。
	特殊編集で利用するDLLや関数が存在しないなど編集情報が足りません。再度設定し直してください。
413	税編集時にエラーが発生しました。(税編集エラー)
	消費税率が設定されていません。TaxRateプロパティを設定してください。
430	条件判定でエラーが発生しました。(条件判定エラー)
	条件変数の条件式で判定結果が計算できません。条件式や対象の変数に正しい値が設定されているかご確認ください。
431	条件判定結果で使用する変数が見つかりません。(条件判定エラー)
	条件変数の判定結果に指定した変数が存在するかご確認ください。
610	用紙に印字できない項目が存在します。(用紙外に指定した項目が存在する場合)
	オブジェクトが用紙外に配置されています。MLデザインでエラーのオブジェクトがないかご確認ください。印字に必要なオブジェクトは属性設定の印字で「印字しない」に設定してください。
描画エラー：発行時動作設定のその他で「画像描画をチェックして黒塗りの場合はエラーメッセージを表示する」を有効にした場合に発生します。	
130	画像描画で黒塗りが発生しました。
	<ul style="list-style-type: none"> Windows Updateの不具合が発生している可能性があります。最新の更新プログラムを適用して改善されるかご確認ください。 発行中に「ユーザーの切り替え」やリモートデスクトップ接続により操作するPCが切り替わる場合に発生している可能性があります。Windowsの制限事項になるため、発行中はログインユーザーや操作PCが切り替わるような運用は避けてください。
描画エラー：オブジェクトの描画に必要な情報がありません。	
450	条件印字の判定でエラーが発生しました。
	条件印字で指定している入力チェックテーブルのファイルが存在しないか、ファイル内に該当のテーブルが見つかりません。
470	文字の書式タグが不正です。
	開始・終了タグの数が一致しているかなど書式タグの記述方法を確認してください。
471	文字の書式タグとして利用できない変数が設定されています。
	シンボル変数で書式タグ指定は利用できません。文字オブジェクトの変数を変更してください。
472	文字の書式タグが有効の場合はC/D自動付加は利用できません。
	C/D自動付加を無効にしてください。

473	文字の書式タグが有効の場合はフォントサイズの幅は任意に設定できません。
	フォントサイズの幅を自動にしてください。
600	レイアウトで使用している Windows フォントがありません。
	発行環境に該当する Windows フォントをインストールしてください。開発環境で一時的に発行テストを行う場合は、動作設定ファイルをご利用ください。動作設定ファイルの詳細は「第 6 章ワンポイントテクニック」をご参照ください。
601	レイアウトで使用しているバーコードの情報がありません。
	MLWebComponent をバージョンアップしてください。
サポート外エラー：プロパティが利用できない組み合わせで設定されています。	
800	プリンタドライバ出力、ファイル出力以外で、連番変数[数値 (レイアウト)]、[文字]を使用しているレイアウトを発行することはできません。
	プリンタドライバ出力に変更するか、入力変数に変更してアプリケーションから連番を入力する形に変更してください。
801	プリンタドライバ、ファイル出力以外で、多面取りレイアウトに対して 1 シート以上の枚数を指定することはできません。
	プリンタドライバ出力に変更するか、1 シート以内の発行枚数に変更してください。
802	発行枚数が指定されていません。
	PrnData プロパティ、SetPrnDataField メソッド、SetPrnDataRow メソッドで入力するデータを確認してください。
803	多面取りまたは両面レイアウトでフォームオーバーレイは指定できません。
	フォームオーバーレイを使用せずに登録と発行を合せたレイアウトを利用してください。
804	プリンタドライバ出力、ファイル出力以外で、多面取りでないレイアウトに対して SetPrnDataRow メソッドで複数データを指定することはできません。
	インターフェース出力 (USB、LAN、COM、Bluetooth) は 1 枚毎にデータを指定してプリンタの状態を確認しながら発行してください。
805	多面取りレイアウトで指定枚数カットが使用できません。
	指定枚数カットを使用する場合は Ver.5.1.1.0 以降へバージョンアップしてください。
820	レイアウトにヘッダ札が設定されていません。
	ヘッダ札を出力する場合は、発行時動作設定のヘッダ・テール札出力を設定してください。
821	レイアウトにテール札が設定されていません。
	テール札を出力する場合は、発行時動作設定のヘッダ・テール札出力を設定してください。
上記以外	レイアウトファイルがサトープリンタレイアウトであるかご確認ください。カラープリンタレイアウトの利用には、「 カラープリンタ発行機能追加 」をご購入ください。

■例外エラーについて

MLWebComponent が発行処理中に予期せぬタイミングで OS から例外をキャッチした場合には発生します。例外の詳細は、エラー発生後にイベントビューアに出力されるイベント「SATO MLV5 MLWebComponent」でご確認ください。

- IE の再起動や発行間隔を空けることでエラーが復旧する場合

メモリ不足が発生している可能性があります。[動作設定ファイルで「メモリ自動解放」](#)を有効にして、メモリ使用量が安定するかご確認ください。メモリの自動解放で改善されない場合は、タスクマネージャーでアプリケーションのハンドル、GDI オブジェクトなどメモリ使用量を確認して、アプリケーションのロジックを改善してください。

第4章

FileAccessComponent

プロパティ

4-1

[基本]FilePath プロパティ

データを読み出すファイルを設定、取得します。

■書式

FileAccessComponent.FilePath [As String](#)

■設定値

FilePath	説明
任意文字列	データファイルのローカルファイルパス

■解説

- ファイルが開かれていない場合のみ設定することができます。すでにファイルが開かれている場合には、設定値を変更できません。

■例外

例外	説明
ArgumentException	条件：不正な設定値 The value of FilePath property is invalid.
InvalidOperationException	条件：ファイルオープン中 The value of FilePath property cannot be changed when the file is opened.

■使用例

ファイルを開きます。

```
FileAccessComponent.FilePath = "C:¥¥Data.CSV";
FileAccessComponent.FileDataType = 1;
Result = FileAccessComponent.OpenFile();
```

■関連項目

プロパティ [FileDataType](#)
メソッド [OpenFile](#)

4-2

[基本]FileDataType プロパティ

ファイルから読み出すデータのタイプを設定、取得します。

■書式

FileAccessComponent.FileDatatype [As Integer](#) / [SATO.MLWebComponent.FileDataType](#)

■設定値

<i>FileDatatype</i>	説明
0 - PrnDataTypes.Tsv	タブ区切り指定/TSV 形式(初期値)
1 - PrnDataTypes.Csv	カンマ区切り指定/CSV 形式
2 - PrnDataTypes.Prn	スペース区切り指定/PRN 形式

■解説

- ファイルが開かれていない場合のみ設定することができます。すでにファイルが開かれている場合には、設定値を変更できません。

■例外

例外	説明
ArgumentOutOfRangeException	条件：範囲外の設定値 The value of FilePath property is invalid.
InvalidOperationException	条件：ファイルオープン中 The value of FilePath property cannot be changed when the file is opened.

■使用例

ファイルを開きます。

```
FileAccessComponent.FilePath = "C:¥¥Data.CSV";
FileAccessComponent.FileDataType = 1;
Result = FileAccessComponent.OpenFile();
```

■関連項目

プロパティ [FilePath](#)
メソッド [OpenFile](#)

4-3

[基本]CacheSize プロパティ

メモリーにキャッシュするデータのレコード数を設定、取得します。

■書式

FileAccessComponent.CacheSize *As Integer*

■設定値

CacheSize	説明
1	1レコードをキャッシュする（初期値）
任意数値	キャッシュするレコード数（1～65536）

■解説

- メモリーの使用量とパフォーマンスのバランスを調整することができます。
- Move・MovePrevious・MoveLast メソッド、RecordCount プロパティ、RowPosition プロパティを使用する場合、本設定値を大きくするとパフォーマンスが向上します。
- CSV やタブ区切りのテキストデータはレコード長が可変のため、レコード数を取得するときやカレントレコードを移動するときには、改行を検索してレコードの開始位置を決定する必要があり時間がかかります。
- キャッシュを大きく設定することにより、すでに検索済みのレコードはキャッシュ内でレコード番号に関連付けて保持されるため、すばやくレコードを移動することができます。

■例外

例外	説明
ArgumentOutOfRangeException	条件：範囲外の設定値 The value of FilePath property is invalid.

■使用例

キャッシュサイズに 10 を設定します。

```
FileAccessComponent.CacheSize = 10;
```

現在のキャッシュサイズの設定値を取得します。

```
Size = FileAccessComponent.CacheSize;
```

■関連項目

プロパティ [RecordCount](#)、[RowPosition](#)
メソッド [Move](#)、[MovePrevious](#)、[MoveLast](#)

4-4**[データ]RecordCount プロパティ**

ファイルがオープンされている場合に、データのレコード数を取得します。

■書式

FileAccessComponent.RecordCount [As Integer](#)

■設定値

<i>RecordCount</i>	説明
1 以上	データのレコード数
0	<ul style="list-style-type: none">・ファイルが開かれていない。(初期値)・データがない。

■解説

- ・ファイルのデータ量が大きい場合、ファイルをオープンした後の最初のRecordCount取得に時間がかかることがあります。

■使用例

データのレコード数を調べます。

```
FileAccessComponent.FilePath = "C:¥¥Data.CSV";
FileAccessComponent.FileDataType = 1;
FileAccessComponent.OpenFile();
RecordCount = FileAccessComponent.RecordCount;
FileAccessComponent.CloseFile();
```

■関連項目

プロパティ [FilePath](#)
メソッド [OpenFile](#)

4-5

[データ]RowData プロパティ

カレントレコードのデータを MLV5 用印字データに変換した結果を取得します。

■書式

FileAccessComponent.RowData [As String](#)

■設定値

RowData	説明
文字列	データのレコード数
空文字	<ul style="list-style-type: none"> ファイルが開かれていない。(初期値) データがない。 カレントレコードが先頭レコードより前、もしくは末尾レコードより後に設定されている。

■解説

- RowData プロパティの値を MLWebComponent の PrnData プロパティに指定して利用します。

■使用例

データの先頭レコードから末尾に向かって発行します。

```
FileAccessComponent.FilePath = "C:¥¥Data.CSV";
FileAccessComponent.FileDataType = 1;
FileAccessComponent.OpenFile();
while( !FileAccessComponent.EOF){
    MLWebComponent.PrnData = FileAccessComponent.RowData;
    MLWebComponent.Output();
    FileAccessComponent.MoveNext();
}
FileAccessComponent.CloseFile();
```

■関連項目

プロパティ [FilePath](#)、[FileDataType](#)、[RowPosition](#)
メソッド [OpenFile](#)

4-6

[レコード]RowPosition プロパティ

カレントレコードの位置を 0 から始まるインデックスで設定、取得します。

■書式

FileAccessControl.RowPosition [As Integer](#)

■設定値

RowPosition	説明
整数値(0~RecordCount-1)	カレントレコードのインデックス
-1	<ul style="list-style-type: none"> ファイルが開かれていない。(初期値) カレントレコードが先頭より前、もしくは末尾より後にある。

■解説

- ファイルがオープンされていない状態で設定した場合、範囲外の値を設定した場合にはエラーが発生します。

■例外

例外	説明
ArgumentOutOfRangeException	条件：範囲外の設定値 The value of FilePath property is invalid.
InvalidOperationException	条件：ファイルオープン中 The value of FilePath property cannot be changed when the file is opened.

■使用例

ファイルを開いてカレントレコードを設定、取得します。

```
FileAccessComponent.FilePath = "C:¥¥Data.CSV";
FileAccessComponent.FileDataType = 1;
FileAccessComponent.OpenFile();
FileAccessComponent.RowPosition = 10;
FileAccessComponent.MoveNext();
Position = FileAccessComponent.RowPosition;
FileAccessComponent.CloseFile();
```

■関連項目

プロパティ

[RowData](#)

メソッド

[Move](#)、[MoveNext](#)、[MovePrevious](#)、[MoveFirst](#)、[MoveLast](#)

4-7

[レコード]BOF プロパティ

カレントレコードの位置が最初のレコードより前にあるかどうか取得します。

■書式

FileAccessComponent.BOF As Boolean

■設定値

BOF	説明
True	<ul style="list-style-type: none"> カレントレコードの位置が最初のレコードより前にあります。 ファイルを開いていない。
False	カレントレコードの位置が最初のレコード以降にあります。(初期値)

■解説

- ファイルを開いた場合、またはカレントレコードを移動した場合に、カレントレコードの位置が最初のレコードより前にあるかどうかを調べるために使用します。

■使用例

最後のレコードから先頭に向かって発行します。

```
FileAccessComponent.FilePath = "C:¥¥Data.CSV";
FileAccessComponent.FileDataType = 1;
FileAccessComponent.OpenFile();
FileAccessComponent.MoveLast();
while( !FileAccessComponent.BOF){
    MLWebComponent.PrnData = FileAccessComponent.RowData;
    MLWebComponent.Output();
    FileAccessComponent.MovePrevious();
}
FileAccessComponent.CloseFile();
```

■関連項目

プロパティ [EOF](#)、[FilePath](#)
メソッド [OpenFile](#)

4-8

[レコード]EOF プロパティ

カレントのレコード位置が最後のレコードより後にあるかどうか取得します。

■書式

FileAccessComponent.EOF [As Boolean](#)

■設定値

EOF	説明
True	・ カレントレコードの位置が最後のレコードより後にあります。 ・ ファイルを開いていない。
False	カレントレコードの位置が最後のレコード以前にあります。（初期値）

■解説

- ・ ファイルをオープンした場合、またはカレントレコードを移動した場合に、カレントレコードの位置が最後のレコードより後にあるかどうかを調べるために使用します。

■使用例

データの先頭レコードから末尾に向かって発行します。

```
FileAccessComponent.FilePath = "C:¥¥Data.CSV";
FileAccessComponent.FileDataType = 1;
FileAccessComponent.OpenFile();
while( !FileAccessComponent.EOF){
    MLWebComponent.PrnData = FileAccessComponent.RowData;
    MLWebComponent.Output();
    FileAccessComponent.MoveNext();
}
FileAccessComponent.CloseFile();
```

■関連項目

- プロパティ [BOF](#)、[FilePath](#)
メソッド [OpenFile](#)

4-9**[バージョン]Version プロパティ**

バージョン情報を取得します。設定はできません。

■書式

FileAccessComponent.Version *As String*

■設定値

<i>Version</i>	説明
Version	FileAccessComponent のバージョンを取得します。

■使用例

バージョン情報を取得します。

```
Version= FileAccessComponent.Version;
```

取得結果 : "Multi LABELIST FileAccessComponent, Version 5.2.3.1"

第5章

FileAccessComponent

メソッド

5-1

[ファイル]OpenFile メソッド

FilePath プロパティで指定されているファイルを開きます。

■書式

FileAccessComponent.OpenFile() [As Integer](#)

■戻り値

戻り値	説明
0	正常終了です。
-1	予期しないエラー
1001	ファイルはすでに開かれています。
1002	指定されたファイルは存在しません。
1003	指定されたファイルをオープンできません。
1004	最大ファイルサイズ(2GB)を超えているためオープンできません。

■解説

- ファイルが開かれてレコードが存在する場合は、カレントレコードを先頭の行に移動します。
- ファイル内にデータがない場合は、BOF プロパティと EOF プロパティが両方 true になり、カレントレコードは-1 となります。

■使用例

ファイルをオープンします。

```
FileAccessComponent.FilePath = "C:¥¥Data.CSV";
FileAccessComponent.FileDataType = 1;
FileAccessComponent.OpenFile();
if ( Result == 0 ){
    //一連の出力処理
    FileAccessComponent.CloseFile();
}
```

■関連項目

プロパティ [FilePath](#)、 [FileDataType](#)
メソッド [CloseFile](#)

5-2**[ファイル]CloseFile メソッド**

現在使用しているファイルをクローズします。

■書式FileAccessComponent.CloseFile() [As Integer](#)**■戻り値**

戻り値	説明
0	正常終了です。
1005	ファイルがオープンされていません。
1006	ファイルクローズ時にエラーが発生しました。

■使用例

ファイルをクローズします。

```
FileAccessComponent.FilePath = "C:¥¥Data.CSV";
FileAccessComponent.FileDataType = 1;
FileAccessComponent.OpenFile();
if ( Result == 0 ){
    //一連の出力処理
    FileAccessComponent.CloseFile();
}
```

■関連項目

プロパティ [FilePath](#)
メソッド [OpenFile](#)

5-3

[ファイル]RemoveFile メソッド

FilePath プロパティで指定されているファイルを削除します。

■書式

FileAccessComponent.RemoveFile() *As Integer*

■戻り値

戻り値	説明
0	正常終了です。
-1	予期しないエラー
1001	ファイルはすでに開かれています。
1002	指定されたファイルは存在しません。
1007	指定されたファイルはオープン・クローズを実行していません。
1008	指定されたファイルはシステムファイル、または読み込み専用のため削除できません。
1009	指定されたファイルを削除できません。

■解説

- OpenFile メソッド、CloseFile メソッドが正常に完了しているファイルのみ削除することが可能です。FilePath プロパティで指定されたファイルで、一度もオープン、クローズされていないファイルは削除できません。

■使用例

ファイルを削除します。

```
FileAccessComponent.FilePath = "C:¥¥Data.CSV";
FileAccessComponent.FileDataType = 1;
FileAccessComponent.OpenFile();
if ( Result == 0 ){
    //一連の出力処理
    FileAccessComponent.CloseFile();
    FileAccessComponent.RemoveFile();
}
```

■関連項目

プロパティ [FilePath](#)
メソッド [OpenFile](#)、[CloseFile](#)

5-4

[レコード]Move メソッド

カレントレコードを指定したレコード数分移動します。

■書式

FileAccessComponent.Move(ByVal *NumRecords* As Integer) As Integer

■パラメータ

パラメータ	設定値	説明
<i>NumRecords</i>	正数	末尾方向に移動するレコード数
	負数	先頭方向に移動するレコード数

■戻り値

戻り値	説明
0	正常終了です。
1005	ファイルがオープンされていません。
1010	レコードがありません。
1011	カレントレコードはすでにファイルの先頭を越えています。
1012	カレントレコードはすでにファイルの末尾を越えています。

■使用例

Move メソッドを使用して、データを 1 レコードおきに発行する例を示します。

```
FileAccessComponent.FilePath = "C:¥¥Data.CSV";
FileAccessComponent.FileDataType = 1;
FileAccessComponent.OpenFile();
//末尾に向かって発行する
while( !FileAccessComponent.EOF){
    MLWebComponent.PrnData = FileAccessComponent.RowData;
    MLWebComponent.Output();
    FileAccessComponent.Move(2);
}
FileAccessComponent.CloseFile();
```

■関連項目

プロパティ [BOF](#)、[EOF](#)、[RowPosition](#)
メソッド [OpenFile](#)

5-5

[レコード]MoveNext メソッド

カレントレコードを次のレコードに移動します。

■書式

FileAccessComponent.MoveNext() [As Integer](#)

■戻り値

戻り値	説明
0	正常終了です。
-1	予期しないエラー
1005	ファイルがオープンされていません。
1010	レコードがありません。
1012	カレントレコードはすでにファイルの末尾を越えています。

■使用例

MoveNext メソッドを使用して、全てのデータを発行する例を示します。

```
FileAccessComponent.FilePath = "C:¥¥Data.CSV";
FileAccessComponent.FileDataType = 1;
FileAccessComponent.OpenFile();
while( !FileAccessComponent.EOF){
    MLWebComponent.PrnData = FileAccessComponent.RowData;
    MLWebComponent.Output();
    FileAccessComponent.MoveNext();
}
FileAccessComponent.CloseFile();
```

■関連項目

プロパティ [EOF](#)、 [RowPosition](#)
メソッド [OpenFile](#)、 [MovePrevious](#)

5-6

[レコード]MovePrevious メソッド

カレントレコードを前のレコードに移動します。

■書式

FileAccessComponent.MovePrevious() *As Integer*

■戻り値

戻り値	説明
0	正常終了です。
-1	予期しないエラー
1005	ファイルがオープンされていません。
1010	レコードがありません。
1011	カレントレコードはすでにファイルの先頭を越えています。

■使用例

MovePrevious メソッドを使用して、全てのデータを発行する例を示します。

```
FileAccessComponent.FilePath = "C:¥¥Data.CSV";
FileAccessComponent.FileDataType = 1;
FileAccessComponent.OpenFile();
FileAccessComponent.MoveLast();
while( !FileAccessComponent.BOF){
    MLWebComponent.PrnData = FileAccessComponent.RowData;
    MLWebComponent.Output();
    FileAccessComponent.MovePrevious();
}
FileAccessComponent.CloseFile();
```

■関連項目

プロパティ [BOF](#)、[RowPosition](#)
メソッド [OpenFile](#)、[MoveNext](#)

5-7

[レコード]MoveFirst メソッド

カレントレコードを先頭のレコードに移動します。

■書式

FileAccessComponent.MoveFirst() [As Integer](#)

■戻り値

戻り値	説明
0	正常終了です。
-1	予期しないエラー
1005	ファイルがオープンされていません。
1010	レコードがありません。

■使用例

MoveFirst メソッドを使用して、ラベルを 2 組発行する例を示します。

```
FileAccessComponent.FilePath = "C:¥¥Data.CSV";
FileAccessComponent.OpenFile();
while( !FileAccessComponent.BOF){
    MLWebComponent.PrnData = FileAccessComponent.RowData;
    MLWebComponent.Output();
    FileAccessComponent.MoveNext();
}
FileAccessComponent.MoveFirst();
while( !FileAccessComponent.BOF){
    MLWebComponent.PrnData = FileAccessComponent.RowData;
    MLWebComponent.Output();
    FileAccessComponent.MoveNext();
}
FileAccessComponent.CloseFile();
```

■関連項目

プロパティ [RowPosition](#)
メソッド [OpenFile](#)、 [MoveLast](#)

5-8**[レコード]MoveLast メソッド**

カレントレコードを末尾のレコードに移動します。

■書式FileAccessComponent.MoveLast() [As Integer](#)**■戻り値**

戻り値	説明
0	正常終了です。
-1	予期しないエラー
1005	ファイルがオープンされていません。
1010	レコードがありません。

■使用例

MoveLast メソッドを使用して、ラベルを末尾から先頭に向かって発行します。

```
FileAccessComponent.FilePath = "C:¥¥Data.CSV";
FileAccessComponent.FileDataType = 1;
FileAccessComponent.OpenFile();
FileAccessComponent.MoveLast();
while( !FileAccessComponent.BOF){
    MLWebComponent.PrnData = FileAccessComponent.RowData;
    MLWebComponent.Output();
    FileAccessComponent.MovePrevious();
}
FileAccessComponent.CloseFile();
```

■関連項目

プロパティ [RowPosition](#)
メソッド [OpenFile](#)、 [MoveFirst](#)

第6章

ワンポイント テクニック

6-1

MLWebComponent を利用する

■宣言 ■サーバ ■クライアント

■object タグで配置する

Web ページで利用するには、body 内に object タグを記述します。

属性	説明	変更の可否
id	ソース上で宣言する際に利用する名前	可能
classid	コントロールのクラス ID	不可
codebase	MLWebComponent インストーラの相対パス	可能
type	ブラウザに認識させる MIME の種類	不可
style="display:none"	Web ページ上にコントロールを表示させないスタイル	可能

(補足) object タグ内に文字列など記述すると、MLPreviewComponent の読み込みに失敗した際に表示させることができます。

```
<body>
  <object
    id="objMLWebComponent"
    type="application/x-oleobject"
    classid="clsid:C137E319-41FE-4F0F-BD1F-190424FD7E2B"
    codebase="WebComponent-Installer-ja.exe"
    style="display:none"
  >WebComponent が使用できません。 </object>
  <object
    id="objFileAccessComponent"
    type="application/x-oleobject"
    classid="clsid:A3F14F83-0717-444B-9DE5-BFC3AF5C32E8"
    style="display:none"
  ></object>

  //ページの内容

</body>
```

※<object>タグ内の改行は必要ありません。説明のために改行しています。

③ Web サーバに MIME タイプを登録します。

ファイル名の拡張子	MIME の種類	設定の要否
.lic	application/octet-stream	必須
.mllayx	application/octet-stream	必須
.mlglex	application/octet-stream	任意
.mltblx	application/octet-stream	任意
.mlchkx	application/octet-stream	任意

■クライアント環境を構築する

① Internet Explorer のセキュリティを設定します。

下記のいずれかの方法で設定してください。

- 信頼済みサイトに登録する（推奨）

MLWebComponent を使用したページを信頼済みサイトに登録してください。

- 「インターネット オプション」を開く。
- セキュリティタブで「信頼済みサイト」を選択して、「サイト」をクリック。
- 「この Web サイトをゾーンに追加する」に、ページの URL を入力し、「追加」をクリック。
- 「閉じる」をクリック

- 個別に変更する

信頼済みサイトに登録せずに、セキュリティを個別に設定する場合は、下記の項目を有効に設定してください。設定項目の名称は IE のバージョンにより若干異なります。

○ActiveX コントロールとプラグイン

- ActiveX コントロールとプラグインの実行
- スクリプトを実行しても安全だとマークされている ActiveX コントロールの実行
- 署名された ActiveX コントロールのダウンロード

○スクリプト

- アクティブ スクリプト

*IE10 以降の場合、詳細設定の拡張保護モードが有効になっていると、ファイルのダウンロード（GetFile）に失敗します。拡張保護モードを無効にするか、セキュリティ設定の保護モードを無効にしてご利用ください。

② .NET Framework をインストールする。（Windows 7 の場合）

Windows 7 で .NET Framework 4.5.2 以降がインストールされていない場合は、別途 .NET Framework 4.5.2 以降のインストールをお願いいたします。JavaScript を利用して自動的にチェックしてインストールを案内することも可能です。利用方法はサンプルプログラムをご参照ください。

<https://docs.microsoft.com/ja-jp/dotnet/framework/install/guide-for-developers>

6-2

ライセンス認証を行う

■ライセンスファイル ■認証する

■ライセンスファイルと認証キー

MLWebComponent を利用するには**ライセンス認証**が必要です。認証に利用する「ライセンスファイル」「認証キー」は、テクニカルサポートサイトでダウンロードしてください。

ログインに必要な「ライセンスID」と「パスワード」は、製品に同梱されている「はじめに、ライセンス情報」に記載されております。(「[仮ライセンス](#)」と記載されているお客様は、[仮ライセンスの「ライセンスID」、「パスワード」](#)をご利用ください。)

① テクニカルサポートサイトにログインする。

<http://dcs.sato.co.jp/mlwebcmp/support/index.asp>

② ライセンスファイルをダウンロードする。

■ライセンス認証を行う

ライセンスファイルをサーバに配置し、GetFile メソッドでクライアントにダウンロード、Authenticate メソッドで認証キーとパスを指定し、ライセンス認証を行います。

ライセンスファイルは、お客様毎に個別のシリアル番号で管理しております。認証キーおよびライセンスファイルは第三者に悪用されないように管理し、セキュリティ対策を行ってください。ライセンスファイルの流用が発覚した場合は、MLWebComponent の使用を停止させて頂く場合もございます。

セキュリティ対策

- ・ライセンスファイルのファイル名、拡張子を変更する（ライセンスファイルの流用防止）
- ・ソースコードをHTML と分離する（ソースの直接参照防止） …など

```
var result;
var LicUrl = "(ライセンスファイルの URL)";
var LicDir = "(ライセンスファイルの保存先ディレクトリ)";

//ライセンスファイルをダウンロードする
result = MLWebComponent.GetFile( LicUrl , LicDir , 2 );
if (result != 0){
    window.alert("ライセンスファイルダウンロードエラー No." + result);
    return;
}
// ライセンス認証
result = MLWebComponent.Authenticate("(認証キー)" , LicDir + "¥¥MLWebComponent.lic");
if (result != 0) {
    window.alert("ライセンス認証エラー No." + result);
    return;
}
```


6-3

バージョンを確認する

■プロパティ ■ファイルバージョン

■Version プロパティで取得する

Version プロパティで取得できる情報からバージョン番号を確認できます。

Multi LABELIST Web Component, Version **5.x.x.x**

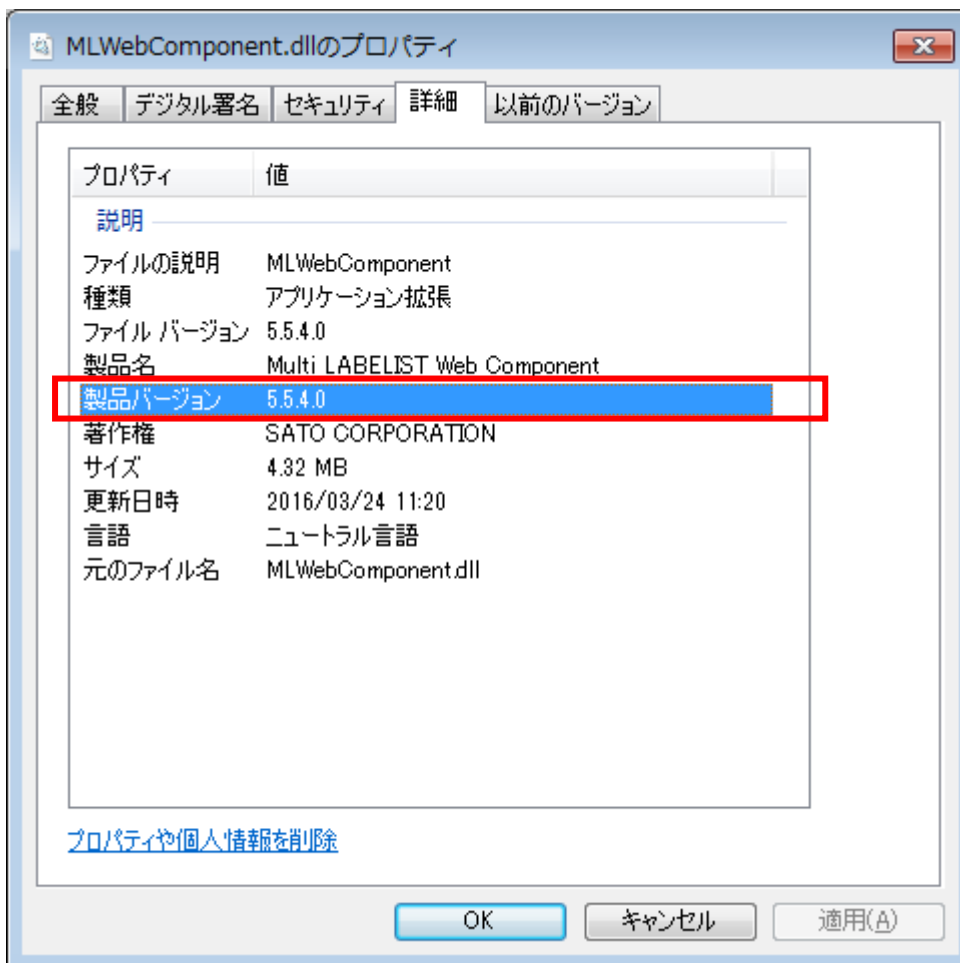
Multi LABELIST FileAccessComponent, Version **5.x.x.x**

■バージョン確認を行ってバージョンアップを通知する

クライアントにインストールされているバージョンを Version プロパティで確認することで、指定したバージョンより古い場合にバージョンアップを通知することができます。利用方法はサンプルプログラムをご参照ください。

■ファイルのプロパティで確認する

WebComponentがインストールされている環境で、インストール先にある「MLWebComponent.dll」のファイルプロパティで確認できます。



6-4

バージョンアップを行う

■サーバ環境 ■クライアント環境

■サーバの WebComponent を更新する

- ①MLWebComponent のセットアップ「WebComponent-Installer-ja.exe」を入替える
- ②[Javascript でバージョン確認を行っている場合](#)、バージョン番号を書換える

※Ver.5.6.1.0 以前のバージョンから Ver.5.7.0.0 以降へ入替える場合

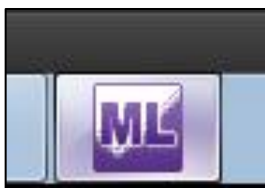
Ver.5.7.0.0 からインストーラに包含していた.NET Framework 4.0 のインストーラを削除しました。WebComponent を既にご利用のお客様には影響ございませんが、**Windows 7 で.NET Framework 4.5.2 以降がインストールされていない環境**に新規導入する場合には、別途.NET Framework 4.5.2 以降のインストールをお願いいたします。JavaScript を利用して自動的にチェックしてインストールを案内することも可能です。利用方法はサンプルプログラムをご参照ください。

■クライアントの WebComponent を更新する

- ①MLWebComponent を利用したページを開く
(既にページを開いている場合は、更新ボタンまたは F5 でページを更新します)
- ②「インストール」をクリック



- ③タスクバーの ML アイコンをクリック



④ 「インストール」をクリック



⑤ 「完了」をクリック



⑫ ページが表示されれば完了です。

6-5

データを一括で入力する

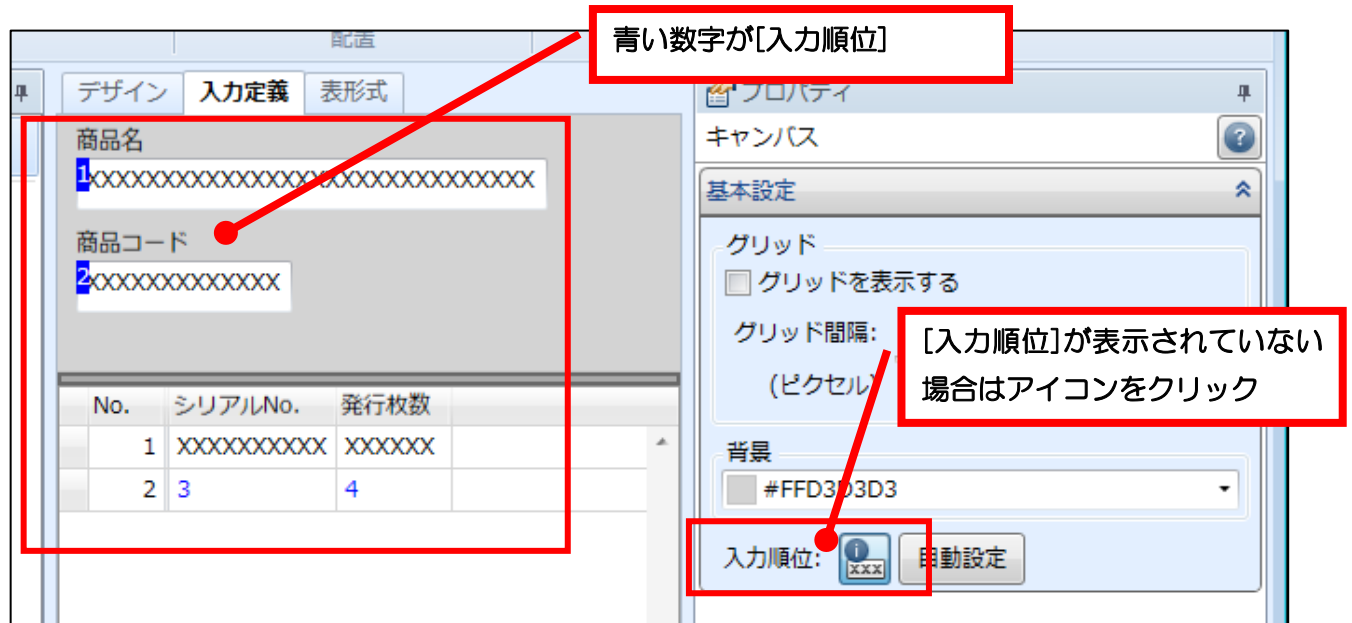
■入力順位 ■データ形式

指定したデータ形式に従って、一括でデータを入力します。プリンタドライバを利用する場合は、1つのデータだけでなく、複数のデータを一括して入力することもできます。

順番を意識せず変数名で入力したい場合は「[データを変数名で指定して入力する](#)」をご確認ください。

■入力順位

まずデータを入力する順番を、MLデザインの入力定義で[入力順位]を表示させて確認します。



• 使用例 (TSV 形式)

```
//入力データ指定
```

```
//商品名、商品コード、シリアル No.、発行枚数を順番に入力
```

```
MLWebComponent.PrnData =
```

```
"SG412R-ex" + "¥t" + "490310999999" + "¥t" + "RX00007802" + "¥t" + "3";
```

■データ形式

デフォルトではタブ区切り（TSV形式）でデータを指定しますが、カンマ区切りのCSV形式やスペース区切りのPRN形式でデータを指定することもできます。

• 使用例（CSV 形式）

```
//データ形式指定(CSV 形式)
MLWebComponent.PrnDataType = 1;

//入力データ指定
MLWebComponent.PrnData = "SG412R-ex,490310999999,RX00007802,3";

//括り文字""（ダブルクォーテーション）を使って改行コードも入力可能
MLWebComponent.PrnData =
    ""SG412R-ex は、" + "¥n" + "高速発行が可能な 4 インチ堅牢型プリンタです。",1"
```

• 使用例（PRN 形式）

```
//データ形式指定(PRN 形式)
MLWebComponent.PrnDataType = 2;

//入力データ指定
MLWebComponent.PrnData = "SG412R-ex 490310999999 RX00007802 3";
```

6-6

データを変数名で指定して入力する

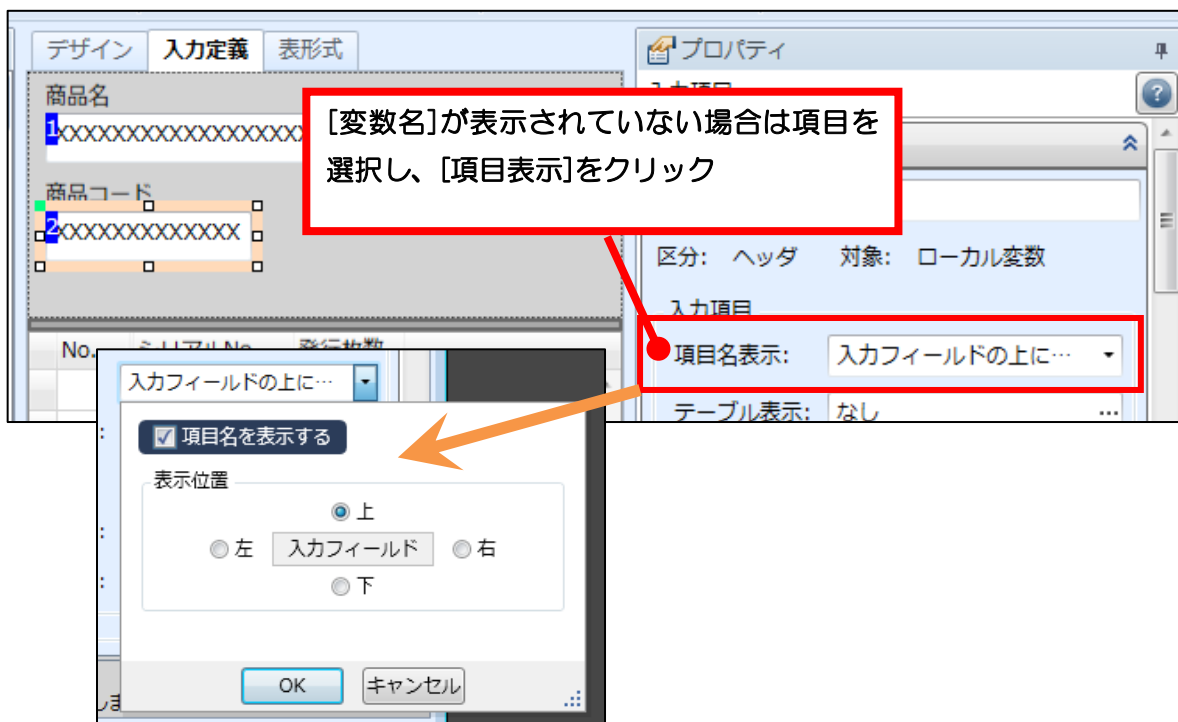
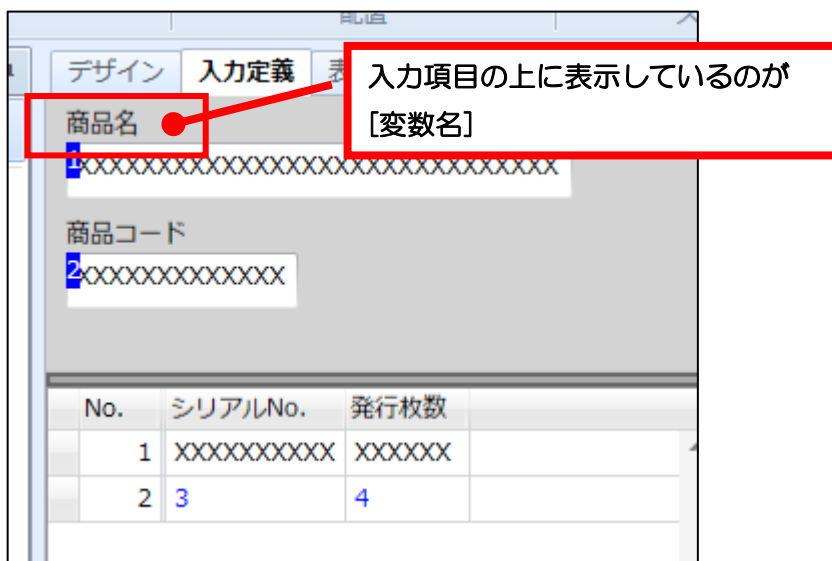
■変数名

変数名を指定してデータを入力します。変数名が共通しているレイアウトファイルを複数使う場合や入力順番が変更される可能性がある場合など、レイアウトファイルの入力順位を意識せずにデータを入力できます。

変数名を意識せず入力順位で簡単に入力したい場合は「[データを一括で入力する](#)」をご確認ください。

■変数名

まずデータ指定に必要な[変数名]を、MLデザインの入力定義で確認します。



• 使用例

```
// 「商品名」を入力
MLWebComponent.SetPrnDataField("商品名", "SG412R-ex");

// 「商品コード」を入力
MLWebComponent.SetPrnDataField("商品コード", "490310999999");

// 「シリアル No.」を入力
MLWebComponent.SetPrnDataField("シリアル No.", "RX00007802");

// 「発行枚数」を入力
MLWebComponent.SetPrnDataField("発行枚数", "3");
```

6-7

動作設定ファイルを利用する

■動作設定ファイル

MLWebComponentと同じフォルダに動作設定ファイルを配置することで、プロパティでは設定できない拡張設定が利用できます。設定値（XMLタグ）がない場合は、初期値が利用されます。

■ファイル名

MLWebComponentSettings.xml

■格納先

MLWebComponent.dllと同一フォルダ

（通常は[プログラムフォルダ] – SATO – ML Web Component）

■文字エンコーディング

Unicode (UTF-8)

■書式（記述例）

```
<?xml version="1.0"?>
<MLWebComponentSettings
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <IsLog>>false</IsLog>
  <LogFolder>D:¥Logs</LogFolder>
  <IsSheetCountError>>false</IsSheetCountError>
  <TaxRate>5.0,8.0,・・・</TaxRate>
  <AlternativeFont>>false</AlternativeFont>
  <DesignDefaultWindowsFontName>MS ゴシック</DesignDefaultWindowsFontName>
  <DesignDefaultWindowsFontSize>9</DesignDefaultWindowsFontSize>
  <IsFileForwardCompatibleError>>true</IsFileForwardCompatibleError>
  <IsEnableGcCollectForced>>false</IsEnableGcCollectForced>
  <GcCollectForcedThreshold>0</GcCollectForcedThreshold>
</MLWebComponentSettings>
```

IsLog（ログ出力の有効・無効）

ログファイルを出力するかを設定します。

true	ログファイルを出力します。
false(初期値)	ログファイルを出力しません。

LogFolder（ログ出力先）

ログファイルの出力先フォルダをフルパスで指定します。

IsSheetCountError（発行枚数エラーの有効・無効）

発行枚数が指定されていない場合に、エラーとするかを設定します。MLV4 WebEngine互換設定のため、通

常は設定を変更せず、印字データで発行枚数を指定してください。

true(初期値)	PrnData プロパティまたはPrnDataArrayメソッドで発行枚数が指定されていない場合に、Output メソッドがエラーNo.802となります。
false	発行枚数が未設定でも発行エラーとせず、発行枚数ゼロでプリンタコマンドを送信します。カット動作が機能しないなど問題が発生する可能性があります。通常は使用しないでください。

TaxRate (税率の設定)

税編集で使用する税率を設定します。設定値はTaxRateプロパティをご参照ください。

緊急対応用として用意している設定値です。通常はTaxRateプロパティで指定してください。

AlternativeFont (代替フォント使用の有効・無効)

レイアウトで使用しているWindowsフォントがない場合に、代替フォントを使用するか設定します。

true	代替フォントが利用されます。 デザイン時に指定したフォントと異なるフォントで印字されるため、文字の形状やサイズ、自動改行位置が異なります。通常は発行環境に該当フォントをインストールするか、フォントを変更するなどの対応を行ってください。
false(初期値)	OutputメソッドがエラーNo.600となります。

DesignDefaultWindowsFontName (デフォルトフォントのフォント設定)

代替フォントに使用されるデフォルトフォントのフォントをWindowsフォント名で設定します。

本設定がない場合は、OSのデフォルトフォントが利用されます。

DesignDefaultWindowsFontSize (デフォルトフォントのフォントサイズ設定)

代替フォントに使用されるデフォルトフォントのフォントサイズをポイント数で設定します。

本設定がない場合は、OSのデフォルトフォントサイズが利用されます。

IsFileForwardCompatibleError (ファイルバージョンチェックの有効・無効)

レイアウトのファイルバージョンがMLWebComponentより新しい場合に発行エラーとするか設定します。

true(初期値)	発行エラー (No.61~66) となります。
false	エラーが発生しません。 レイアウトによっては、正常に印字されない危険性があります。ファイルバージョンチェックを無効にする場合は、レイアウトの新規追加・編集に充分注意し、変更後は運用前に必ず正常に印字されるかテストを行ってください。

IsEnableGcCollectForced (メモリ自動解放の有効・無効)

IEの使用メモリが閾値に達した場合にガベージコレクションによるメモリ解放 (GC.Correct) を実行します。

閾値はGcCollectForcedThresholdで指定します。

true	自動解放を行います。
false(初期値)	自動解放を行いません。

GcCollectForcedThreshold (メモリ自動解放の閾値設定)

メモリ解放を行う閾値 (IEのメモリ使用量) を設定します。

0(初期値)	発行する度にメモリ解放を実行します。 発行速度が遅延するため、IEのメモリ使用量によって、適切な値に調整してください。
1~2048(MB)	IEが指定したメモリ使用量 (MB単位) を越えた場合に、メモリ解放を実行します。

■ログファイルを出力する

動作設定ファイルでログ出力を有効にすることで、通信・発行・プリンタ制御のメソッドを実行する度に、ログを出力することができます。

ログファイルは、MLWebComponentがロードされたタイミングで「MLWebComponent_*.log」 (*GUID) のファイル名で生成され、メソッドが呼び出されるタイミングでログが出力されます。MLWebComponentがアンロードされると編集中のファイルを閉じます。

ログファイルの出力時に、フォルダ有無、権限、ディスク残量等のチェックなどはいりません。また、出力したログファイルの自動削除も行いませんので、手動またはお客様のアプリケーションで削除してください。

•書式

2014/04/18 10:23:52:123<tab>OpenPort<tab>0<tab>COM1:9600,n,8,1…

①

②

③

④

①メソッドの実行日時

YYYY/MM/DD HH:MM:SS:MMM

②メソッド名

通信 OpenPort、ClosePort

発行 Output、OutputHeader、OutputTail、SendStringData

プリンタ制御 GetStatus、Cut、SendCancel

③メソッドの戻り値

メソッドの実行結果を表す戻り値 (0: 正常終了、0以外はエラー)

SendStringDataは例外エラーの番号

④付加情報

メソッド毎に出力されるプロパティの値やステータスがタブ区切りで付加されます。

メソッド	内容
OpenPort	Setting、Protocol、Timeout
ClosePort	なし
Output	LayoutFile、Darkness、Speed、Offset、MultiCut、SortMark、

OutputHeader	HeaderTailSetting、Formoverlay、TaxRate、PrnDataType、PrnData
OutputTail	(PrnDataArray)
SendStringData	なし
GetStatus	プリンタから受信したステータス文字列
Cut	なし
SendCancel	なし

6-8

評価版ライセンスから入れ替える

■入替方法

ご購入前に期間限定版のライセンスファイルを利用している場合は、製品版のライセンスファイルに入替を行ってください。

■ライセンスファイルと認証キー

製品版の「ライセンスファイル」「認証キー」は、テクニカルサポートサイトでダウンロードしてください。ログインに必要な「ライセンスID」と「パスワード」は、製品に同梱されている「はじめに、ライセンス情報」に記載されております。

- ① テクニカルサポートサイトにログインする。

<http://dcs.sato.co.jp/mlwebcmp/support/index.asp>

- ② ライセンスファイルをダウンロードする。

■ライセンスファイルの入替

- ① サーバに配置したライセンスファイルを入れ替えます。

「MLWebComponent.lic」を上書き保存してください。

- ② 認証キーを変更します。

MLWebComponent.Authenticate(“(認証キー※)”,“(ライセンスファイルの保存先)”)

※製品版の認証キーに変更してください。

第7章

利用方法・注意事項

7-1**利用方法について**

利用方法に関する注意事項を説明します。

■ライセンスファイルと認証キーについて

ライセンスファイルは、お客様毎に個別のシリアル番号で管理しております。認証キーおよびライセンスファイルは第三者に悪用されないように管理し、セキュリティ対策を行ってください。ライセンスファイルの流用が発覚した場合は、WebComponent の使用を停止させて頂く場合もございます。

セキュリティ対策

- ライセンスファイルのファイル名、拡張子を変更する（ライセンスファイルの流用防止）
- ソースコードを HTML と分離する（ソースの直接参照防止） …など

■ライセンス認証について

MLWebComponent を使用するには、ライセンスファイルで「ライセンス認証」が必要です。使用方法はワンポイントテクニックをご確認ください。OpenPort メソッドで、ライセンス認証がされていない場合はエラーNo.31 となります。

■IE の保護モードについて

IE の保護モードはセキュリティが高く、WebComponent の一部機能が使用できません。WebComponent を利用する場合は、IE のメニューで[ツール]-[インターネットオプション]を選択し、[セキュリティ]タブで[信頼済みサイト]に開発した Web サイトを登録してご利用ください。

■IE10 以降での利用

IE10 以降は HTML5 に準拠して HTML の記述方法が厳密にチェックされます。WebComponent の動作に問題ありませんが、WebComponent を宣言する object タグは head タグに記述できないため、body タグなど別の場所へ記述してください。その他、正常に動作しない場合は、[ツール]メニューから[開発ツール]を表示させ、[ブラウザモード]や[ドキュメントモード]を変更するか、[コンソール]タブに表示されるメッセージに従って HTML の修正を行ってください。

■Windows 7 での利用

Windows 7 で .NET Framework 4.5.2 以降がインストールされていない場合は、別途 .NET Framework 4.5.2 以降をインストールしてください。

<https://docs.microsoft.com/ja-jp/dotnet/framework/install/guide-for-developers>

■.NET Framework のバージョンについて

古いバージョンの .NET Framework を利用すると、MLWebComponent が正常に動作しない場合があります。Microsoft でサポートされているバージョンをご利用ください。

<https://docs.microsoft.com/ja-JP/lifecycle/faq/dotnet-framework>

■サーバサイドアプリケーションでの利用

MLWebComponent はクライアントで動作させることを前提に開発・提供しており、マニュアル・サンプルは全てクライアントでの利用方法で記載しております。ASP.NET などサーバサイドアプリケーションでの利用は技術的には可能ですが、動作確認は行っておりません。ご利用の際は事前に十分な検証をお願いいたします。

■プリンタのスリープモードについて

「CL4NX-Jシリーズ」「CL6NX-Jシリーズ」「PW208NX/PW208」をご利用の際、プリンタがスリープモードになっていると、出力メソッド([Output](#)、[OutputHeader](#)、[OutputTail](#)、[Cut](#)、[SendCancel](#))がエラー11になる場合があります。下記のいずれかの方法で対応してください。

1. スリープモードをOFFに変更する。
スリープモードの変更方法は、プリンタの取扱説明書をご参照ください。
<https://www.sato.co.jp/support/printer/>
2. プリンタファームウェアを更新する。(PW208NX/ PW208のみ)
担当営業、またはお客さまヘルプデスクにご連絡ください。
<https://www.sato.co.jp/contact/>

■Windows フォントの全角スペースが.NET Framework のバージョンによって異なる

Windows フォントでシャープ有効時に全角スペースの幅が.NET Framework のバージョンによって若干異なります。PC の入替えや OS のバージョンアップにより、.NET Framework が 4.6.2 より前のバージョンから、4.6.2 以降にバージョンアップする場合、全角スペースが複数使われていると印字位置がずれる可能性があるため、自動改行の位置など印字内容に影響がないかご確認ください。

7-2

プログラミングについて

プログラミングに関する利用方法・注意事項を説明します。

■カット動作に関して

MLWebComponent は初期設定でカットを行いません。

カット動作が正常に行われない場合は、下記をご確認ください。また、レイアウトファイルで連番変数を利用している場合は、MultiCut プロパティは利用できません。Cut メソッドでカット動作の制御を行ってください。

- 印字データに発行枚数が指定されていますか？
発行枚数が指定されていない場合、ラベルは 1 枚だけ発行されますがカット動作は行われません。
ワンポイントテクニック「[データを一括で入力する](#)」「[データを変数名で指定する](#)」を参考に正しくデータが設定されているかご確認ください。

■初回起動時・レイアウトファイルロード時の遅延

MLWebComponent で使用している.NET Framework のライブラリがロードされるため、アプリケーション起動後に初めて MLWebComponent を呼び出す時、初めてレイアウト情報を読み込む時 (Output メソッドや GetPriter メソッドなど) に処理が遅延することがあります。

運用上で遅延が障害となる場合は、アプリケーションの起動時に MLWebComponent で仮のレイアウトを呼び出すなどの方法で回避をお願い致します。

■Windows フォント (WPF) の使用について

Windows フォント (WPF) を MLWebComponent、MLPreviewComponent で利用する場合、発行またはプレビュー実行中に、お客様のアプリケーションで.NET Framework(WPF)の文字描画 (DrawingContext.DrawText などの文字描画 API) を同時に実行しないでください。Windows 内部で発生する現象により、まれに入力した文字と異なる文字が印字される可能性があります。

7-3

インターフェースについて

接続方法に関する利用方法・注意事項を説明します。

■LAN

- 指定方法について

[Setting プロパティをご参照ください。](#)

- 接続・切断について

発行する度に OpenPort メソッド・ClosePort メソッドを繰り返す場合は、ClosePort メソッド実行後に 200ms 以上経過してから次の OpenPort メソッドを実行してください。間隔を空けずに OpenPort メソッドを実行すると2重接続が発生する可能性があります。

- 2重接続

他のアプリケーションがプリンタを使用している時、Openport メソッドは成功しますが、出力メソッド (GetStatus、Output、SendStringData、Cut、SendCancel) でエラー (戻り値7, 9, 10) が発生する可能性があります。接続後すぐに上記メソッドでエラーが発生する場合は、ClosePort メソッドを実行し、ウェイト処理を入れるなどして再接続して下さい。

- 送信前の状態確認

出力メソッド (Output、SendStringData) を実行する前に GetStatus メソッドで必ずプリンタの状態をご確認下さい。GetStatus メソッドでプリンタ状態の確認を行わずに出力メソッドを実行した場合、プリンタの状態によっては送信した印字データが消失したり、プリンタの印字が停止する恐れがあります。

- 送信後の状態確認

Output メソッド実行直後に ClosePort メソッドを実行すると、OS や Pr など環境によって送信データが途切れる恐れがあります。そのため、ClosePort メソッドを実行する前に、GetStatus メソッドでプリンタとの通信状態を確認してください。

- 通信プロトコルについて

ステータス4は、Output メソッド実行時にデータの到達を確認する ACK/NAK 確認が行われません。送信後に GetStatus メソッドでプリンタの状態を確認してください。

■USB

- 指定方法について

[Setting プロパティをご参照ください。](#)

- 送信前の状態確認

出力メソッド (Output、SendStringData) を実行する前に GetStatus メソッドで必ずプリンタの状態をご確認下さい。GetStatus メソッドでプリンタ状態の確認を行わずに出力メソッドを実行

した場合、プリンタの状態によっては送信した印字データが消失したり、プリンタの印字が停止する恐れがあります。

- **通信プロトコルについて**

[ステータス4を指定してください。](#)

- **64bit 版での利用**

64bit 版 Windows 上では 32bit アプリケーションとして動作する時のみ使用可能です。

- **排他制御**

USB 接続時は排他制御できません。アプリ内の複数スレッドや複数のアプリで、同時に発行処理が行われないように運用にご注意ください。

- **Setting プロパティの[プリンタ機種]について**

レイアウト選択の名称	設定値
L'esprit(レスプリ)T408v/R408v	Lesprit408v
L'esprit(レスプリ)T412v/R412v	Lesprit412v
Et Vie(エヴィ)EV208R	EV208R
Et Vie(エヴィ)EV212R	EV212R
Et Vie(エヴィ)EV208R(SATOC)	EV208R
Et Vie(エヴィ)EV212R(SATOC)	EV212R
パートロニクス CF408T	CF408T
パートロニクス RT308R	RT308R
スキャントロニクス HA212R	HA212R
スキャントロニクス HA224R	HA224R
スキャントロニクス CL4NX-J 08	CL4NX-J 08
スキャントロニクス CL4NX-J 12	CL4NX-J 12
スキャントロニクス CL4NX-J 24	CL4NX-J 24
SeaTa(シータ) CT4-LX DT203/TT203	CT4-LX-J 203
SeaTa(シータ) CT4-LX DT305/TT305	CT4-LX-J 305
L'esprit (レスプリ) HC4-LX DT203/TT203	HC4-LX-J 203
L'esprit (レスプリ) HC4-LX DT305/TT305	HC4-LX-J 305
スキャントロニクス SG408R-ex	SG408R-ex
スキャントロニクス SG412R-ex	SG412R-ex
スキャントロニクス SG424R-ex	SG424R-ex
スキャントロニクス BF408R	BF408R
スキャントロニクス BF412R	BF412R
CL6NX-J 08	CL6NX-J 08
CL6NX-J 12	CL6NX-J 12
スキャントロニクス SG608R	SG608R

スキャントロニクス SG612R	SG612R
スキャントロニクス SG112T/R	SG112T/R
スキャントロニクス GN412T	GN412T
SATOC(サトック)ST308R	ST308R
SATOC(サトック)ST312R	ST312R
Petit lapin(プチラパン)PW208NX/PW208mNX/ PW208/PW208m	PW208
スキャントロニクス SG408R	SG408R
スキャントロニクス SG412R	SG412R
スキャントロニクス SG424R	SG424R
CL4NX(203dpi)	CL4NX(203dpi)
CL4NX(305dpi)	CL4NX(305dpi)
CL4NX(609dpi)	CL4NX(609dpi)
CT4-LX DT203/TT203	CT4-LX 203
CT4-LX DT305/TT305	CT4-LX 305
CL6NX(203dpi)	CL6NX(203dpi)
CL6NX(305dpi)	CL6NX(305dpi)

- **Setting プロパティの[シリアル No.]について**

[シリアル No.]は、Windows 上で確認できます。

1. 電源が投入されたプリンタを USB 接続する。
2. コントロールパネルからデバイスマネージャを起動する。
3. ユニバーサル シリアル バス コントローラーの USB 印刷サポートを選択する。
4. 右クリックしたメニューからプロパティを開く。
5. 詳細タブのプロパティから「デバイス インスタンス バス」を選択する。
6. 表示された値の末尾がシリアル No.に該当する。

(例) USB¥¥VID_XXXX&PID_XXXX¥¥(8桁のシリアル No.)

■COM

- **指定方法について**

[Setting プロパティをご参照ください。](#)

- **送信前の状態確認**

出力メソッド (Output、SendStringData) を実行する前に GetStatus メソッドで必ずプリンタの状態をご確認下さい。GetStatus メソッドでプリンタ状態の確認を行わずに出力メソッドを実行した場合、プリンタの状態によっては送信した印字データが消失したり、プリンタの印字が停止する恐れがあります。

■Bluetooth

- **指定方法について**

[Setting プロパティをご参照ください。](#)

- **Bluetooth スタック**

PC 側の Bluetooth デバイスは、Microsoft 標準の Bluetooth スタック（Microsoft Bluetooth Enumerator）のみ使用可能です。その他の Bluetooth スタックでは動作保証されていません。

その他の Bluetooth スタックでは、OpenPort メソッド、EnumerateBluetoothDevices メソッド、AuthenticateBluetoothDevice メソッドは正常に動作しない可能性があります。Microsoft 標準の Bluetooth スタックに入替えてください。入替え方法はパソコンまたは Bluetooth アダプタのメーカーにご確認ください。

- **2 重接続**

他のアプリケーションがプリンタを使用している時、Output メソッドは戻り値 7（接続エラー）となります。プリンタは 1 台の Bluetooth デバイスとしか通信が出来ません。

- **送信前の状態確認**

出力メソッド（Output、SendStringData）を実行する前に GetStatus メソッドで必ずプリンタの状態をご確認下さい。GetStatus メソッドでプリンタ状態の確認を行わずに出力メソッドを実行した場合、プリンタの状態によっては送信した印字データが消失したり、プリンタの印字が停止する恐れがあります。

- **利用台数**

Windows の仕様により PC 側から同時に接続できる Bluetooth デバイスの台数は 7 台までです。7 台を超える場合は、Output メソッドは戻り値 7（接続エラー）となります。

■プリンタドライバ

- **指定方法について**

[Setting プロパティをご参照ください。](#)

- **使用できないプロパティ・メソッド**

プロパティ Protocol、Timeout
メソッド GetStatus

- **詳細設定[プリンタに直接印刷データを送る]について**

プリンタドライバの[詳細設定]で[プリンタに直接印刷データを送る]に設定している場合、OpenPort に失敗し正常に動作しません。[印刷ドキュメントをスプールし、プログラムの印刷処理を高速に行う]に設定を変更するか、プリンタドライバを使用せずインターフェース出力（LAN、USB、COM、Bluetooth）をご利用ください。

- **詳細設定[印刷ドキュメントをスプールし、プログラムの印刷処理を高速に行う]について**

プリンタドライバの[詳細設定]で「印刷ドキュメントをスプールし、プログラムの印刷処理を高速に行う」に設定している場合、選択内容によってスプールの送信方法が異なります。

「すぐに印刷データをプリンターに送る」は、Output メソッドを連続して実行するか、ClosePort メソッドで送信を完了することで、プリンタに印刷データが送信されます。

「全ページ分のデータをスプールしてから、印刷データをプリンターに送る」は、ClosePort メソッドで送信を完了することで、プリンタに印刷データが送信されます。

- **Windows 10 で発行開始が遅い**

Windows 7 では Output メソッドを実行する度に、一つ前の Output メソッドでスプールされたデータがプリンタに送信されていましたが、Windows 10 は OS の仕様変更によりスプールデータが 256KB 毎に送信されます。すぐにプリンタにデータを送信するには、Output メソッド直後に ClosePort を実行してください。

7-4

レイアウト情報について

レイアウト情報に関する利用方法・注意事項を説明します。

■入力定義

レイアウトファイルの入力定義（初期値や桁数制限、入力チェックなど）は、MLプリントの発行画面で使用される設定項目です。お客様のアプリケーションで入力定義の設定を利用する場合は、GetInputFieldsで取得してください。

■Windows フォント

文字オブジェクトで「Windowsフォント」を指定した場合、グラフィックで文字が描画されます。様々な文字の表現や自由なサイズ調整が可能ですが、送信データが大きくなり、発行が遅くなるデメリットもあります。大量の連続発行やマルチスレッドでの利用で発行が遅い場合は、文字オブジェクトを「プリンタフォント」に変更することで改善されるかご確認ください。

フォント	対応文字 *フォント種による	サイズ調整	小さな文字	データ量
プリンタフォント	○：必要充分	△：倍率指定	○：きれい	◎：小さい
Windows フォント	◎：豊富	○：ポイント指定	△：潰れる	△：大きい

■グラフィックオブジェクト

取り込んだ画像ファイルをデザイン上で拡大・縮小すると画像の一部が薄くなったり、擦れることがあります。元の画像ファイルを拡大・縮小してから取り込んでサイズを100%で利用するか、グラフィックオブジェクトの伸縮モードを「GDI 互換」に設定して改善されるかご確認ください。

■固定オブジェクト（発行速度の改善）

貼り付け文字や罫線など、データによって内容が変わらない固定オブジェクトも発行時に解析して、プリンタコマンドに変換しています。発行動作設定のその他ページで「レイアウト保存時に固定項目をプリンタコマンドに変換する（高速化）」を有効にして、レイアウトを保存することで、固定オブジェクトの数が多い場合や Windows フォントの貼り付け文字で自動改行が多く利用されている場合に、発行速度を改善させることが可能です。

■連番変数

連番変数[数値（レイアウト）][文字]を使用している場合は、プリンタドライバ出力でラベル発行を行ってください。[インターフェース出力*](#)の場合は、Outputの戻り値が「800」となります。[インターフェース出力*](#)で連番を利用する場合、アプリケーション側で連番値を指定するか、連番変数[数値（プリンタ）]をご利用ください。

レイアウトファイルに保存されている連番値の取得および変更はできません。MLV5で確認・変更を行ってください。お客様のアプリケーションで連番値の管理が必要な場合は、レイアウトファイルの連番保存方法を「なし」でご利用ください。

■多面取りラベル

[インターフェース出力](#) *で多面取りラベルを利用する場合は、1シート分の発行枚数を入力してください。1シート分を越える発行枚数を指定した場合は、Outputの戻り値が「801」となります。

*インターフェース出力

Settingプロパティに「LAN:」「USB:」「COM:」「BT:」を指定する出力方法です。

第8章

付録

8-1

サポートプリンタ

MLWebComponent のサポートプリンタについて説明します。

分類表記:

P	: Protocol プロパティ	D	: Darkness プロパティ
Spd	: Speed プロパティ	MCt	: MultiCut プロパティ
ECt	: EjectCut プロパティ	Sok	: SortMark プロパティ
F	: Formoverlay プロパティ	Cut	: Cut メソッド

■シータシリーズ

機種名	P	D	Spd	MCt	ECt	Sok	F	Cut
CT4-LX-J 203	3,4	1~10	2~8	○	○	×	○	○
CT4-LX-J 305	3,4	1~10	2~6	○	○	×	○	○

■レスブリシリーズ

レスブリ T408/R408 ⇒ レスブリ T8/R8

レスブリ T412/R412 ⇒ レスブリ T12/R12

機種名	P	D	Spd	MCt	ECt	Sok	F	Cut
HC4-LX-J 203	3,4	1~10	2~8	○	○	×	○	○
HC4-LX-J 305	3,4	1~10	2~6	○	○	×	○	○
レスブリ 408v	3,4	1~5	2~6	○	○	×	○	○
レスブリ 412v	3,4	1~5	2~4	○	○	×	○	○
レスブリ T8/R8	3,4	1~5	2~6	○	○	×	○	○
レスブリ T12/R12	3,4	1~5	2~4	○	○	×	○	○

■スキャントロンクスシリーズ

CL4NX-J 08、CL4NX-J 08 Plus ⇒ CL4NX-J 08

CL4NX-J 12、CL4NX-J 12 Plus ⇒ CL4NX-J 12

CL4NX-J 24、CL4NX-J 24 Plus ⇒ CL4NX-J 24

CL6NX-J 08、CL6NX-J 08 Plus ⇒ CL6NX-J 08

CL6NX-J 12、CL6NX-J 12 Plus ⇒ CL6NX-J 12

機種名	P	D	Spd	MCt	ECt	Sok	F	Cut
CL4NX-J 08	3,4	1~10	2~14	○	○	×	○	○
CL4NX-J 12	3,4	1~10	2~14	○	○	×	○	○
CL4NX-J 24	3,4	1~10	2~6	○	○	×	○	○
CL6NX-J 08 (LEFT-JUSTIFY)	3,4	1~10	2~10	○	○	×	○	○
CL6NX-J 12 (LEFT-JUSTIFY)	3,4	1~10	2~8	○	○	×	○	○

<i>CL6NX-J 08</i>	3,4	1~10	2~10	○	○	×	○	○
<i>CL6NX-J 12</i>	3,4	1~10	2~8	○	○	×	○	○
<i>MR600e</i>	3,4	1~3	4,6,8	○	○	×	○	○
<i>MR610e</i>	3,4	1~3	4,6,8	○	○	×	○	○
<i>MT/MR110w</i>	3,4	1~5	3~5	○	○	×	○	○
<i>MT400e</i>	3,4	1~5	2~6	○	○	×	○	○
<i>MT410e</i>	3,4	1~5	2~6	○	○	×	○	○
<i>M-48Pro8</i>	3,4	1~3	2,4,6,8,10	○	○	×	○	○
<i>M-48Pro12</i>	3,4	1~3	2,4,6,8	○	○	×	○	○
<i>M-48Pro24</i>	3,4	1~3	2~6	○	○	×	○	○
<i>SR408</i>	3,4	1~5	2~12	○	○	×	○	○
<i>SR412</i>	3,4	1~5	2~12	○	○	×	○	○
<i>SR424</i>	3,4	1~5	2~6	○	○	×	○	○
<i>BF408R</i>	3,4	1~5	2~6	○	○	×	○	○
<i>BF412R</i>	3,4	1~5	2~6	○	○	×	○	○
<i>SG408R</i>	3,4	1~5	2~10	○	○	×	○	○
<i>SG412R</i>	3,4	1~5	2~10	○	○	×	○	○
<i>SG424R</i>	3,4	1~5	2~6	○	○	×	○	○
<i>HA212R</i>	3,4	1~5	1~4	×	×	×	○	×
<i>HA224R</i>	3,4	1~5	1~4	×	×	×	○	×
<i>GN412T</i>	3,4	1~5	3~10	○	○	×	○	○
<i>SG608R</i>	3,4	1~10	2~10	○	○	×	○	○
<i>SG612R</i>	3,4	1~10	2~8	○	○	×	○	○
<i>SG112R/T</i>	3,4	1~10	3~6	○	○	×	○	○
<i>SG408R-ex</i>	3,4	1~10	2~14	○	○	×	○	○
<i>SG412R-ex</i>	3,4	1~10	2~14	○	○	×	○	○
<i>SG424R-ex</i>	3,4	1~10	2~6	○	○	×	○	○

■SATOC シリーズ、エヴィシリーズ、パートロニクスシリーズ、FLEQV

FX3-LX、FX3-LX Plus ⇒ FX3-LX

機種名	P	D	Spd	MCt	ECt	Sok	F	Cut
<i>TR400e</i>	3,4	1~3	5~8	○	○	○	○	○
<i>TR410e</i>	3,4	1~3	4~6	○	○	○	○	○
<i>RT308R</i>	3,4	1~5	2~5	○	○	×	○	○
<i>ST308R</i>	3,4	1~3	2~10	○	○	○	○	○
<i>ST312R</i>	3,4	1~3	2~10	○	○	○	○	○
<i>EV208R</i>	3,4	1~5	2~5	○	○	×	○	○
<i>EV212R</i>	3,4	1~5	2~4	○	○	×	○	○
<i>EV208R(SATOC)</i>	3,4	1~5	2~5	○	○	×	○	○
<i>EV212R(SATOC)</i>	3,4	1~5	2~4	○	○	×	○	○

<i>CF408T</i>	3,4	1~10	2~6	○	○	×	○	○
<i>FX3-LX</i>	3,4	1~10	2~6	○	○	×	○	○

■ラパンシリーズ

PW208NX/PW208mNX、PW208m ⇒ PW208

機種名	P	D	Spd	MCt	ECt	Sok	F	Cut
<i>PW208</i>	3,4	1~10	3~6	×	×	×	○	×
<i>PW4NX</i>	3,4	1~10	2~6	×	×	×	○	×
<i>PT200m/e/j</i>	3	1~5	3~4	×	×	×	×	×
<i>PT408e</i>	3	1~5	3~4	×	×	×	×	×
<i>PT412e</i>	3	1~5	1~2	×	×	×	×	×

■タフアームシリーズ

機種名	P	D	Spd	MCt	ECt	Sok	F	Cut
<i>LR4080SR-T</i>	3,4	1~5	2~12	○	○	×	○	○
<i>LR4120SR-T</i>	3,4	1~5	2~12	○	○	×	○	○

■海外プリンタ

CL4NX(203dpi)、CL4NX Plus(203dpi) ⇒ CL4NX(203dpi)

CL4NX(305dpi)、CL4NX Plus(305dpi) ⇒ CL4NX(305dpi)

CL4NX(609dpi)、CL4NX Plus(609dpi) ⇒ CL4NX(609dpi)

CL6NX(203dpi)、CL6NX Plus(203dpi) ⇒ CL6NX(203dpi)

CL6NX(305dpi)、CL6NX Plus(305dpi) ⇒ CL6NX(305dpi)

機種名	P	D	Spd	MCt	ECt	Sok	F	Cut
<i>CT400</i>	3,4	1~5	2~6	○	○	×	○	○
<i>CT410</i>	3,4	1~5	2~4	○	○	×	○	○
<i>CL408e</i>	3,4	1~5	1~5	○	○	×	○	○
<i>CL412e</i>	3,4	1~5	1~5	○	○	×	○	○
<i>CL608e</i>	3,4	1~3	4,6,8	○	○	×	○	○
<i>CL612e</i>	3,4	1~3	4,6,8	○	○	×	○	○
<i>MB200i</i>	3	1~5	1~2	×	×	×	×	×
<i>MB400i</i>	3	1~5	3~4	×	×	×	×	×
<i>MB410i</i>	3	1~5	3~4	×	×	×	×	×
<i>XL400e</i>	3,4	1~5	5~8	○	○	○	○	○
<i>XL410e</i>	3,4	1~5	4~6	○	○	○	○	○
<i>M-5900RVe</i>	3,4	1~5	2~6	○	○	×	○	○
<i>M-84Pro-2</i>	3,4	1~5	1~5	○	○	×	○	○
<i>M-84Pro-3</i>	3,4	1~5	1~4	○	○	×	○	○
<i>M-84Pro-6</i>	3,4	1~5	1~5	○	○	×	○	○
<i>GT408e</i>	3,4	1~5	1~11	○	○	×	○	○

<i>GT412e</i>	3,4	1~5	1~11	○	○	×	○	○
<i>GT424e</i>	3,4	1~5	1~5	○	○	×	○	○
<i>CL4NX(203dpi)</i>	3,4	1~10	2~10	○	○	×	○	○
<i>CL4NX(305dpi)</i>	3,4	1~10	2~8	○	○	×	○	○
<i>CL4NX(609dpi)</i>	3,4	1~10	2~6	○	○	×	○	○
<i>CL6NX(203dpi)</i>	3,4	1~10	2~10	○	○	×	○	○
<i>CL6NX(305dpi)</i>	3,4	1~10	2~8	○	○	×	○	○
<i>CT4-LX 203</i>	3,4	1~10	2~8	○	○	×	○	○
<i>CT4-LX 305</i>	3,4	1~10	2~6	○	○	×	○	○
<i>PW208NX</i>	3,4	1~10	3~6	×	×	×	○	×
<i>PW4NX</i>	3,4	1~10	2~6	×	×	×	○	×

8-2

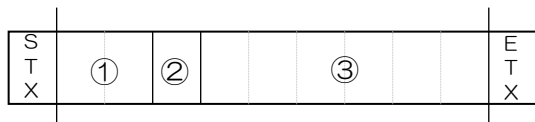
ステータス一覧

通信プロトコルのステータスについて説明します。

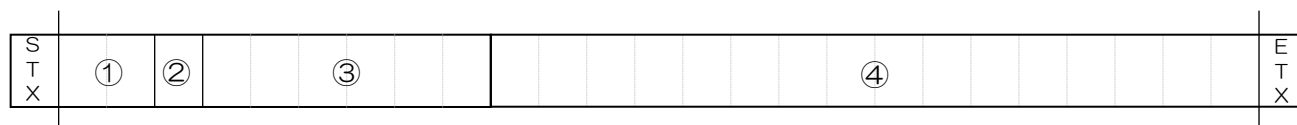
GetStatus メソッドが返すステータス文字列は、プリンタから受信したステータスの前後（STX および ETX）を省いた文字列です。

■ステータス文字列

<ステータス 3、ステータス L> (9 バイト)



<ステータス 4> (25 バイト)



- ①ステータス ID (2 バイト) Output メソッド使用時に設定した StatusID プロパティの値です。
- ②ステータス (1 バイト) プリンタの状態です。(一覧参照)
- ③残りの印字枚数 (6 バイト) 1 回の発行指示に対する残りの印字枚数です。
ステータス L 選択時は貼付完了枚数です。
- ④ジョブ名 (16 バイト) Output メソッド使用時に設定した JobName プロパティの値です。

■ステータス一覧 (ステータス 3、ステータス 4)

内 容		ASCII	16進コード	送信可否	復旧可否	
オフライン状態	エラー無し	0	30	○	○	
	リボンニアエンド	1	31	○	○	
	バッファニアフル	2	32	×	○	
	リボン/ラベルニアエンド&バッファニアフル	3	33	×	○	
	印字停止中 (エラーなし) *ステータス4のみ	4	34	○	○	
オンライン状態	受信待ち	エラー無し	A	41	○	○
		リボン/ラベルニアエンド	B	42	○	○
		バッファニアフル	C	43	×	○
		リボン/ラベルニアエンド&バッファニアフル	D	44	×	○
	印字中	エラー無し	G	47	○	○
		リボン/ラベルニアエンド	H	48	○	○
		バッファニアフル	I	49	×	○
		リボン/ラベルニアエンド&バッファニアフル	J	4A	×	○
	待機中 (ハクリ・ カット待ち)	エラー無し	M	4D	○	○
		リボン/ラベルニアエンド	N	4E	○	○
		バッファニアフル	O	4F	×	○
		リボン/ラベルニアエンド&バッファニアフル	P	50	×	○
	解析・編集 中	エラー無し	S	53	○	○
		リボン/ラベルニアエンド	T	54	○	○
		バッファニアフル	U	55	×	○

		リボン/ラベルニアエンド&バッファニアフル	V	56	×	○
	印字停止中 (エラーなし) *ステータス4 のみ	受信待ち	E	45	○	○
		印字中	K	4B	○	○
		待機中 (ハクリ・カット待ち)	Q	51	○	○
		解析・編集中	W	57	○	○
エラー検出 (オフライン状態)	受信バッファフル		a	61	×	×
	ヘッドオープン		b	62	×	○
	ペーパーエンド		c	63	×	○
	リボンエンド		d	64	×	○
	メディアエラー		e	65	×	○
	センサーエラー		f	66	×	○
	ヘッドエラー		g	67	×	×
	カバーオープン		h	68	×	○
	カードエラー		i	69	×	×
	カッタエラー		j	6A	×	×
	その他のエラー		k	6B	×	×
	カッタセンサーエラー		l	6C	×	○
	スタックフル&リファインダフル		m	6D	×	○
	RFID タグエラー		o	6F	×	○
RFID プロテクトエラー		p	70	×	○	

- バッファオーバーは、インターフェースによって発生の有無が異なります。

■ステータス一覧 (レスプリ V シリーズ、PW208NX/PW208 シリーズ、PW4NX 追加分)

内 容		ASCII	16進 コード	送信 可否	復旧 可否	
オフライン状態	バッテリーニアエンド	5	35	△	○	
	バッテリーニアエンド&リボンニアエンド	6	36	△	○	
	バッテリーニアエンド&バッファニアフル	7	37	×	○	
	バッテリーニアエンド & リボンニアエンド & バッファニアフル	8	38	×	○	
オンライン状態	受信待ち	バッテリーニアエンド	!	21	△	○
		バッテリーニアエンド&リボンニアエンド	"	22	△	○
		バッテリーニアエンド&バッファニアフル	#	23	×	○
		バッテリーニアエンド & リボンニアエンド & バッファニアフル	\$	24	×	○
	印字中	バッテリーニアエンド	%	25	△	○
		バッテリーニアエンド&リボンニアエンド	&	26	△	○
		バッテリーニアエンド&バッファニアフル	'	27	×	○
		バッテリーニアエンド & リボンニアエンド & バッファニアフル	(28	×	○
	待機中 (ハクリ・ カット待ち)	バッテリーニアエンド)	29	△	○
		バッテリーニアエンド&リボンニアエンド	*	2A	△	○
		バッテリーニアエンド&バッファニアフル	+	2B	×	○
		バッテリーニアエンド & リボンニアエンド & バッファニアフル	,	2C	×	○
	解析・編集中	バッテリーニアエンド	-	2D	△	○
		バッテリーニアエンド&リボンニアエンド	.	2E	△	○
		バッテリーニアエンド&バッファニアフル	/	2F	×	○
		バッテリーニアエンド & リボンニアエンド & バッファニアフル	@	40	×	○
エラー検出	バッテリーエラー	α	71	×	×	

- PW208NX/PW208 シリーズ、PW4NX でリボンニアエンドは検出されません。

■ステータス一覧（ステータスL追加分）

内 容		ASCII	16進 コード	送信 可否	復旧 可否	
オンライン状態	印字待ち (通過待ち)	エラー無し	M	4D	○	○
		ラベル・リボン/ラベルニアエンド	N	4E	○	○
		バッファニアフル	O	4F	×	○
		ラベル・リボン/ラベルニアエンド&バッファニアフル	P	50	×	○
	印字吸着 起動信号待 ち/貼付中	エラー無し	o	6F	○	○
		ラベル・リボン/ラベルニアエンド	p	70	○	○
		バッファニアフル	q	71	×	○
		ラベル・リボン/ラベルニアエンド&バッファニアフル	r	72	×	○
	通過中	エラー無し	s	73	○	○
		ラベル・リボン/ラベルニアエンド	t	74	○	○
		バッファニアフル	u	75	×	○
		ラベル・リボン/ラベルニアエンド&バッファニアフル	v	76	×	○
エラー検出	ラベル接続エラー（ペーパーエンド）	c	63	×	×	
	ラベル貼付エラー	l	6C	×	×	
	ラベル吸着ミス	#	23	×	×	
	能力オーバー	\$	24	×	×	
	シリンダ動作不良	%	25	×	×	
	タイマーエラー	&	26	×	×	
	ユーザー専用のエラー	+	2B	×	×	

■ステータス一覧（PT408e、PT412e、PT200m/e/j用ステータス3）

内 容		ASCII	16進 コード	送信可 否	復旧可 否	
オフライン状態	エラー無し	0	30	○	○	
	バッテリニアエンド	1	31	△	○	
	バッファニアフル	2	32	×	○	
	バッテリニアエンド&バッファニアフル	3	33	×	○	
オンライン状態	受信待ち	エラー無し	A	41	○	○
		バッテリニアエンド	B	42	△	○
		バッファニアフル	C	43	×	○
		バッテリニアエンド&バッファニアフル	D	44	×	○
	印字中	エラー無し	G	47	○	○
		バッテリニアエンド	H	48	△	○
		バッファニアフル	I	49	×	○
		バッテリニアエンド&バッファニアフル	J	4A	×	○
	剥離待ち	エラー無し	M	4D	○	○
		バッテリニアエンド	N	4E	△	○
		バッファニアフル	O	4F	×	○
		バッテリニアエンド&バッファニアフル	P	50	×	○
	解析・編集 中	エラー無し	S	53	○	○
		バッテリニアエンド	T	54	△	○
		バッファニアフル	U	55	×	○
		バッテリニアエンド&バッファニアフル	V	56	×	○
エラー検出 (オフライン状態)	受信/バッファオーバー	a	61	×	×	
	ペーパーエンド	c	63	×	○	

	バッテリーエラー	d	64	×	○
	センサーエラー	f	66	×	○
	ヘッドエラー	g	67	×	×
	カバーオープン	h	68	×	○
	その他のエラー	k	6B	×	×

■ステータス一覧の送信可否・復旧可否

- GetStatus メソッドコール時、返送されたステータス部のパラメータ（3バイト目）が [送信可否] × の場合、[送信可否] ○ のステータスが返送されるまで、印字データの送信 (Output、SendStringData) メソッドは実行しないでください。
- GetStatus メソッドコール時、返送されたステータス部のパラメータ（3バイト目）が [送信可否] が △ の場合、印字データの送信は可能ですが、バッテリー残量と印字データ（印字濃度や発行枚数など）の関係によって、ラベル・タグ発行が正常に完了出来ない場合があります。ご注意ください。
- GetStatus メソッドコール時、返送されたステータス部のパラメータ（3バイト目）が [復旧可否] ○ の場合、プリンタがエラー検出状態であってもプリンタ側のエラー状態を解除（例：ラベルの交換等）する事によりプリンタ本体で印字のリカバリが可能になります。[復旧可否] × の場合、プリンタ本体の電源再投入が必要なエラーの為、既に送信済みの印字データについては破棄されます。

■プリンタエラーの解除方法

プリンタ付属の取扱説明書をご参照ください。

<https://www.sato.co.jp/support/printer/>

8-3

プリンタ本体の通信プロトコル設定方法

プリンタ本体で通信プロトコルを設定する方法について説明します。

詳細な設定はプリンタに同梱されています「**取扱説明書**」、「**セットアップガイド**」やアクセサリ CD の「**SBPL プログラミングガイド**」をご参照ください。

<https://www.sato.co.jp/support/printer/>

プリンタ機種	インターフェース	通信プロトコル	
		ステータス 3	ステータス 4
①レスプリシリーズ (レスプリV、レスプリV-ex) T8/R8/T12/R12、T8-2/R8-2/T12-2/R12-2 T408/R408/T412/R412 T408v/R408v/T412v/R412v T408v-ex/R408v-ex/T412v-ex/R412v-ex	RS-232C	①-R	
	IEEE1284	—	①-I
	LAN(無線LAN)	①-L	
	USB	—	①-U
②EtVie シリーズ EV208R/EV212R	RS-232C	②-R	
	LAN	②-L	
	USB	—	②-U
③スキャントロニクス、TR400e/TR410e BF408R/BF412R、MR400e/MR410e、 MR600e/MR610e、MT400e/MT410e、 M-48Pro8/M-48Pro12/M-48Pro24、 M-4800Rve、MR110w/MT110w、 TR400e/TR410e	RS-232C	③-R	
	IEEE1284	—	③-I
	LAN(無線LAN)	③-L	
	USB	—	③-I
④SR400 シリーズ、SG400R シリーズ、 SG400R-ex シリーズ、SG600R シリーズ、 SG112R/T、HA200R シリーズ、 LR4000SR-T シリーズ SR408/SR412/SR424、 SG408R/SR412R/SR424R SG408R-ex/SR412R-ex/SR424R-ex SG608R、SG612R、SG112R、SG112T HA212R/HA242R LR4080SR-T/LR4120SR-T	RS-232C	④-R	
	IEEE1284	—	④-I
	LAN(無線LAN)	④-L	
	USB	—	④-U
⑤CL4NX-J、CL6NX-J CL4NX-J 08/CL4NX-J 12/CL4MX-J 24 CL6NX-J(203dpi)/CL6NX-J(305dpi)	RS-232C	⑤-R	
	IEEE1284	—	⑤-I
	LAN(無線LAN)	⑤-L	
	USB	—	⑤-U
	Bluetooth	⑤-B	

⑥SATOC ST308R/ST312R	RS-232C	⑥-R	
	IEEE1284	—	⑥-I
	LAN(無線 LAN)	⑥-L	
	USB	—	⑥-U
⑦バートロニクス RT308R	RS-232C	⑦-R	
	IEEE1284	—	⑦-I
	LAN	⑦-L	
	USB	—	⑦-U
⑧ラパンシリーズ PT200m/e/j, PT408e, PT412e (SBPL モードのみ)	RS-232C	⑧-R	—
	無線 LAN	⑧-L	—
	USB	⑧-U	—
	Bluetooth	⑧-B	—
⑨スキャントロニクス GN412T	RS-232C	⑨-R	
	LAN	⑨-L	
	USB	—	⑨-U
⑩バートロニクス CF408T	無線 LAN	⑩-L	
	USB	—	⑩-U
⑪プチラパンシリーズ PW208NX/PW208mNX/PW208/PW208m、	無線 LAN	⑪-L	
	USB	—	⑪-U
	Bluetooth	⑪-B	
⑫シータシリーズ、レスプリシリーズ (HC4-LX-J) CT4-LX-J 203/CT4-LX-J 306 HC4-LX-J 203/HC4-LX-J 306	RS-232C	⑫-R	
	LAN	⑫-L	
	無線 LAN	⑫-W	
	USB	—	⑫-U
	Bluetooth	⑫-B	
⑬FLEQV シリーズ FX3-LX/FX3-LX Plus	LAN	⑬-L	
	無線 LAN	⑬-W	
	USB	—	⑬-U
	Bluetooth	⑬-B	
⑭ラパンシリーズ PW4NX PW4NX	無線 LAN	⑭-L	
	USB	—	⑭-U
	Bluetooth	⑭-B	

■レスプリシリーズ (レスプリ V、レスプリ V-ex)

レスプリプリンタ設定ツールで設定します。

- ①-R 設定ツールを起動し、メニューの[デバイス]-[プリンタの設定]-[シリアルの設定]の[制御]で設定します。ステータス 3 は「ステータス 3」、ステータス 4 は「ドライバ専用プロトコル」に設定

します。

- ①-I 設定ツールを起動し、メニューの[デバイス]-[プリンタの設定]-[IEEE1284 の設定]の[受信モード]を「マルチ受信」に設定します。
- ①-L 設定ツールを起動し、メニューの[デバイス]-[プリンタの設定]-[LAN の設定]の[LAN モード]で設定します。ステータス3は「ステータス3モード」、ステータス4は「ENQ 応答モード」に設定します。
- ①-U 特に設定の必要はありません。

■エヴィシリーズ

EtVie プリンタ設定ツールで設定します。

- ②-R 設定ツールを起動し、メニューの[デバイス]-[プリンタの設定]-[シリアルの設定]の[制御]で設定します。ステータス3は「ステータス3」、ステータス4は「ドライバ専用プロトコル」に設定します。
- ②-L 設定ツールを起動し、メニューの[デバイス]-[プリンタの設定]-[LAN の設定]の[LAN モード]で設定します。ステータス3は「ステータス3モード」、ステータス4は「ENQ 応答モード」に設定します。
- ②-U 特に設定の必要はありません。

■スキャントロニクスシリーズ、TR400e/TR410e

- ③-R RS-232CボードのDSW1 とプリンタ本体のDSW2 で設定します。ステータス3はDSW1-7 ON、DSW1-8 OFF、DSW2-8 ON、ステータス4はDSW1-7 ON、DSW1-8 OFF、DSW2-8 OFF に設定します。
- ③-I プリンタ本体のDSW2-5 をOFF に設定します。
- ③-L プリンタ本体のDSW2 で設定します。ステータス3はDSW2-5 ON、DSW2-8 ON、ステータス4はDSW2-5 ON、DSW2-8 OFF に設定します。

■スキャントロニクス SR400 シリーズ、SG400R シリーズ、SG400R-ex シリーズ、SG600R シリーズ、SG112R/T、HA200R シリーズ、LR4000SR-T シリーズ（ステータスLも選択可）

- ④-R LCD で[通信設定]-[通信プロトコル]をステータス3は「ステータス3」、ステータス4は「ドライバ専用」、Ready/Busyは「Ready/Busy」に設定します。
- ④-I LCD で[通信設定]-[通信プロトコル]を「ドライバ専用」、[受信バッファ]を「マルチ受信」に設定します。
- ④-L LCD で[通信設定]-[通信プロトコル]をステータス3は「ステータス3」、ステータス4は「ドライバ専用」で[プリンタステータス返送タイミング]を「ENQ」に設定します。
- ④-U LCD で[通信設定]-[通信プロトコル]を「STATUS4」または「STATUS5」に設定します。

■CL4NX-J、CL6NX-J

- ⑤-R LCD で[通信設定]-[RS-232C]-[通信プロトコル]を選択し、ステータス3は「STATUS3」、ステータス4は「STATUS4」に設定します。
- ⑤-I LCD で[通信設定]-[IEEE1284]-[通信プロトコル]を選択し、「STATUS4」に設定します。
- ⑤-L LCD で[通信設定]-[ネットワーク]-[設定]-[サービス]-[ポート]-[通信プロトコル]を選択し、ス

ステータス 3 は「STATUS3」、ステータス 4 は「STATUS4 ENQ」に設定します。

⑤-U LCD で[通信設定]-[USB]-[通信プロトコル]を「STATUS4」に設定します。

⑤-U LCD で[通信設定]-[Bluetooth]-[通信プロトコル]を「STATUS4」に設定します。

■SATOC ST308R/ST312R

プリンタ本体の LCD でボタンを使って設定します。

⑥-R LCD で[通信設定]-[通信プロトコル]をステータス 3 は「STATUS3」、ステータス 4 は「STATUS4」に設定します。

⑥-I LCD で[通信設定]-[通信プロトコル]を「STATUS4」、[受信バッファ]を「マルチ」に設定します。

⑥-L LCD で[通信設定]-[通信プロトコル]をステータス 3 は「STATUS3」、ステータス 4 は「STATUS4」で[プリンタステータス返送タイミング]を「ENQ」に設定します。

⑥-U LCD で[通信設定]-[通信プロトコル]を「STATUS4」に設定します。

■パートロニクス RT308R

プリンタ本体の LCD でボタンを使って設定します。

⑦-R LCD で[INTERFACE MODE]-[PROTOCOL]をステータス 3 は「STATUS3」、ステータス 4 は「STATUS4」に設定します。

⑦-I LCD で[INTERFACE MODE]-[RECEIVE BUFFER]を「MULTI」に設定します。

⑦-L LCD で[INTERFACE MODE]-[PROTOCOL]をステータス 3 は「STATUS3」、ステータス 4 は「STATUS4」で[STATUS REPLY]を「ENQ」に設定します。

⑦-U 特に設定の必要ありません。

■ラパンシリーズ

プリンタ本体の DSW1 で設定します。

⑧-R DSW1-1 OFF、DSW1-2 OFF、DSW1-3 OFF、DSW1-4 OFF に設定します。さらにプリンタ設定ツールを起動し、メニューの[プリンタ設定]-[RS-232C]の[フロー制御]で設定します。ステータス 3 は「NONE」に設定します。

⑧-L DSW1-1 OFF、DSW1-2 ON、DSW1-3 OFF、DSW1-4 OFF に設定します。プリンタが無線 LAN 仕様である必要があります。

⑧-U DSW1-1 ON、DSW1-2 ON、DSW1-3 OFF、DSW1-4 OFF に設定します。PT408e/412e のみ対応します。

⑧-B DSW1-1 OFF、DSW1-2 ON、DSW1-3 OFF、DSW1-4 OFF に設定します。プリンタが Bluetooth 仕様である必要があります。

■スキャントロニクス GN412T

プリンタ本体の LCD でボタンを使って設定します。

⑨-R LCD で[通信設定]-[通信プロトコル]をステータス 3 は「STATUS3」、ステータス 4 は「STATUS4」に設定します。

⑨-L LCD で[通信設定]-[通信プロトコル]をステータス 3 は「STATUS3」、ステータス 4 は「STATUS4」で[プリンタステータス返送タイミング]を「ENQ」に設定します。

⑨-U LCD で[通信設定]-[通信プロトコル]を「STATUS4」に設定します。

■バートロニクス CF408T

CF408T 設定ツールで設定します。

⑩-L 設定ツールを起動し、[LAN の設定]タブの[LAN モード]で設定します。ステータス 3 は「ステータス 3 モード」、ステータス 4 は「ENQ 応答モード」に設定します。

⑩-U 特に設定の必要はありません。

■ラパンシリーズ PW208NX/PW208mNX/PW208/PW208m

プリンタ本体の LCD でボタンを使って設定します。

⑪-L LCD で[通信設定]-[ネットワーク]-[サービス]-[ポート]-[通信プロトコル]を、ステータス 3 は「Status3」、ステータス 4 は「Status4」に設定します。

⑪-U LCD で[通信設定]-[USB]-[通信プロトコル]を「STATUS4」に設定します。

⑪-B LCD で[通信設定]-[Bluetooth]-[通信プロトコル]を、ステータス 3 は「Status3」、ステータス 4 は「Status4」に設定します。

■シータシリーズ、レスプリシリーズ (HC4-LX-J)

⑫-R LCD で[設定]-[通信設定]-[RS-232C]-[通信プロトコル]を選択し、ステータス 3 は「STATUS3」、ステータス 4 は「STATUS4」に設定します。

⑫-L LCD で[設定]-[通信設定]-[ネットワーク]-[設定]-[ポート]-[通信プロトコル]を選択し、ステータス 3 は「STATUS3」、ステータス 4 は「STATUS4 ENQ」に設定します。

⑫-W LCD で[Wi-Fi]-[Wi-Fi 設定]-[ポート]-[通信プロトコル]を選択し、ステータス 3 は「STATUS3」、ステータス 4 は「STATUS4 ENQ」に設定します。

⑫-U LCD で[設定]-[通信設定]-[USB]-[通信プロトコル]を「STATUS4」に設定します。

⑫-B LCD で[Bluetooth]-[通信プロトコル]をステータス 3 は「STATUS3」、ステータス 4 は「STATUS4」に設定します。

■FLEQV シリーズ FX3-LX、FX3-LX Plus

プリンタ本体の SatoSettings で設定します。

⑬-L SatoSettings で[通信設定]-[ネットワーク]-[設定]-[LAN]-[ポート]-[通信プロトコル]を、ステータス 3 は「STATUS3」、ステータス 4 は「STATUS4 ENQ」に設定します。

⑬-W SatoSettings で[通信設定]-[ネットワーク]-[設定]-[Wi-Fi]-[ポート]-[通信プロトコル]を、ステータス 3 は「STATUS3」、ステータス 4 は「STATUS4 ENQ」に設定します。

⑬-U LCD で[通信設定]-[USB]-[通信プロトコル]を「STATUS4」に設定します。

⑬-B LCD で[通信設定]-[Bluetooth]-[通信プロトコル]を、ステータス 3 は「STATUS3」、ステータス 4 は「STATUS4」に設定します。

■ラパンシリーズ PW4NX

プリンタ本体の LCD でボタンを使って設定します。

⑭-L LCD で[通信設定]-[ネットワーク]-[設定]-[Wi-Fi]-[ポート]-[通信プロトコル]を、ステータス 3 は「STATUS3」、ステータス 4 は「STATUS4 ENQ」に設定します。

- ⑭-U LCDで[通信設定]-[USB]-[通信プロトコル]を「STATUS4」に設定します。
- ⑭-B LCDで[通信設定]-[Bluetooth]-[通信プロトコル]を、ステータス3は「STATUS3」、ステータス4は「STATUS4」に設定します。