

東芝テック株式会社  
TPCL プロバイダ

Version 1.0.0

ユーザーズ ガイド

September 28, 2021

備考:

**【改版履歴】**

バージョン	日付	内容
1.0.0	2021-9-28	初版

**【対応機種】**

機種	注意事項
B-EX4T1	

**【動作確認機器】**

機種	注意事項
B-EX4T1	

## 目次

1. はじめに .....	4
1.1. PC 環境のセットアップ .....	4
2. プロバイダの概要 .....	5
2.1. 概要 .....	5
2.2. メソッド・プロパティ .....	6
2.2.1. CaoWorkspace::AddController メソッド .....	6
2.2.1.1. Conn オプション .....	8
2.2.2. CaoController::GetVariableNames プロパティ .....	8
2.2.3. CaoController::AddVariable メソッド .....	8
2.2.4. CaoController::Execute メソッド .....	9
2.2.5. CaoController::OnMessage イベント .....	9
2.2.5.1. メッセージ詳細 .....	9
➤ プリンタステータス受信 .....	9
➤ その他のメッセージ受信 .....	10
2.2.6. CaoVariable::get_Value プロパティ .....	10
2.2.7. CaoVariable::put_Value プロパティ .....	10
2.3. 変数一覧 .....	11
2.3.1. CaoController クラス .....	11
2.4. エラーコード .....	13
3. コマンドリファレンス .....	14
3.1. CaoController クラス .....	14
3.1.1. CaoController::Execute ("Issue") コマンド .....	14
3.1.2. CaoController::Execute ("Feed") コマンド .....	15
3.1.3. CaoController::Execute ("GetMessage ") コマンド .....	15
3.1.4. CaoController::Execute ("GetStatus") コマンド .....	16
3.1.5. CaoController::Execute ("SendCommand") コマンド .....	16
3.1.6. CaoController::Execute ("Reset") コマンド .....	17
4. サンプルプログラム .....	19
5. 付録 .....	21
5.1. プリンタステータス .....	21

## 1. はじめに

本書は、東芝テック株式会社製(以後、東芝テック)ラベルプリンタに対しデータの書込み/読出し/印刷を行う CAO プロバイダのユーザーズガイドです。本書で扱う CAO プロバイダ(CaoProvTECTPCL.dll)を TPCL プロバイダと呼びます。

TPCL プロバイダは BCP Print For .NET を使用して東芝テック製ラベルプリンタと通信を行います。

BCP Print For .NET は ORiN2SDK に同梱されていますので、別途インストールを行う必要はありません。

また、TPCL プロバイダは BCP プリントシステム V2 または Label Format Generation Tool で作成したラベルフォーマットにのみ対応しています。

本プロバイダは下記に示すバージョンの BCP Print For .NET を使用して開発されました。バージョンが異なる場合プロバイダが正しく機能しない可能性がありますので、別途 BCP Print For .NET をインストールする場合は同じバージョンのソフトウェアをインストールしてご使用ください。

BCP Print For .NET V2.0.18

### 1.1. PC 環境のセットアップ

TPCL プロバイダを使用するためには 2.1 概要に従って TPCL プロバイダを登録してください。

図 1-1 にプロバイダとラベルプリンタの接続例を示します。ラベルプリンタには Ethernet で接続してください。

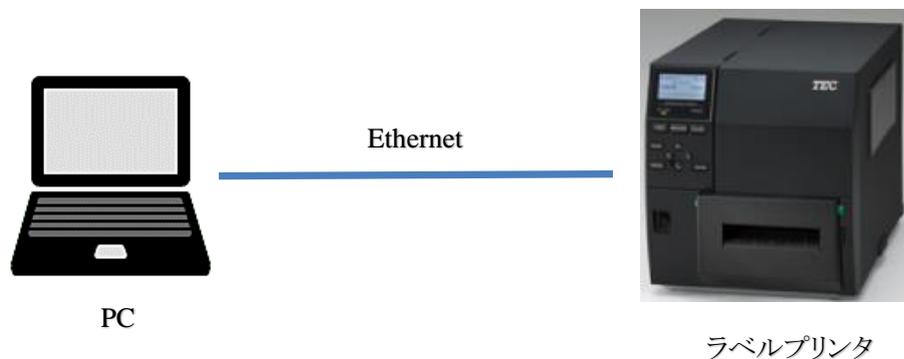


図 1-1 接続イメージ図

## 2. プロバイダの概要

### 2.1. 概要

TPCL プロバイダは、東芝テック製ラベルプリンタに対し BCP Print For .NET を使用してデータの書込み/読出し/印刷を行う CAO プロバイダです。そのファイル形式は DLL(Dynamic Link Library)であり、CAO エンジンから使用時に動的ロードされます。TPCL プロバイダを使用するには ORiN2SDK をインストールするか下表を参照してレジストリ登録を行う必要があります。

表 2-1 TPCL プロバイダ

ファイル名	CaoProvTECTPCL.dll
ProgID	CaoProv.TEC.TPCL
レジストリ登録 <sup>1</sup>	RegistAsm.bat CaoProvTECTPCL.dll
レジストリ登録の抹消	UnregistAsm.bat CaoProvTECTPCL.dll

<sup>1</sup> バッチファイル(RegistAsm.bat や UnregistAsm.bat )は、 {ORiN2 インストールフォルダ}\¥DotNet¥BAT 以下に存在します。

## 2.2. メソッド・プロパティ

### 2.2.1. GaoWorkspace::AddController メソッド

ラベルプリンタに接続してデータの書き込み/読出し/印刷を行うコントローラオブジェクトを作成します。オプション文字列でラベルプリンタに接続する際のパラメータを指定します。



```
AddController ( <bstrCtrlName:BSTR>,<bstrProvName:BSTR>,  
                <bstrPcName:BSTR>, [<bstrOption:BSTR>] )
```

```
<bstrCtrlName>    : [in] コントローラ名  
<bstrProvName>    : [in] プロバイダ名. 固定値 ="CaoProv.TEC.TPCL"  
<bstrPcName>      : [in] プロバイダの実行マシン名 (未使用)  
<bstrOption>     : [in] オプション文字列
```



```
// Create engine  
var eng = new CCaoEngine();  
// Create workspace  
var ws = eng.AddWorkspace("SampleWorkSpace","");  
  
// Connection option settings  
var option = @"Conn=TCP:192.168.1.1:8000, Timeout=5, LfmFilePath=C:¥TestLabel.lfm ";  
  
// Connect to printer  
this.ctrl = ws.AddController("Sample","CaoProv.TEC.TPCL","",option);
```

以下にオプション文字列に指定するリストを示します。

表 2-2 GaoWorkspace::AddController のオプション文字列

オプション	意味
Conn=<接続パラメータ>	必須. 通信形態とその接続パラメータを設定します。 (参照:2.2.1.1)
Timeout[=<タイムアウト時間>]	任意. ラベルプリンタからの応答を待つ時間(秒)を指定します。 (デフォルト:30) 1~9999 : 指定された時間(秒)ラベルプリンタからの応答を待ちます。 0 : タイムアウトは発生しません。 ※発行モードが発行完了復帰モードの場合のみ有効な値です。
IssueMode[=<発行モード>]	任意. ラベル発行時のモードを指定します。(デフォルト:1) 1 : 送信完了復帰モード 2 : 発行完了復帰モード

オプション	意味
LfmFilePath[=<ラベルフォーマットファイルパス>]	任意. ラベルフォーマットファイル名を絶対パスで指定します. ※Issue コマンドを使用する場合は, 必ず指定してください.
LoadPCIndex[=<PC セーブ識別番号>]	任意. ラベルプリンタから読み出すセーブ識別番号(1~99)を指定します.
SystemPath[=<プリンタ情報ファイル格納パスの設定>]	任意. プリンタ情報ファイル格納パスを指定します. (デフォルト: {ORiN2SDK インストールフォルダ}¥CAO¥ProviderLib¥ToshibaTec¥TPCL¥Bin)
UsePrinter=<プリンタ番号>	任意. 使用するプリンタ番号を指定します. (デフォルト:99)  5 : B-458-TS 7 : B-858 8 : B-SX5 10 : B-SA4T 11 : B-EP2G 12 : B-EP4T 15 : B-458TR 16 : B-858TR 17 : B-SX8T 18 : B-EV4-T 19 : B-EV4-G 21 : B-EX4T1-T 22 : B-SX8R 23 : B-LV4-T 24 : B-LV4-G 28 : B-FP2D-G 32 : B-EX4T3-H 34 : B-EX6T1-T 37 : BA400T-G 38 : BA400T-T 39 : BV400-G 40 : BV400-T 99 : LFM 定義
GraphicType[=<グラフィックデータ種別>]	任意. グラフィックデータ種別を設定します. (デフォルト:0)  0 : ニブルモード(4ドット/バイト) 1 : ヘキサモード(8ドット/バイト) 2 : TOPIX 圧縮モード

オプション	意味
ControlCode[=<コマンド制御コード方式>]	任意. コマンド制御コード方式を設定します. (デフォルト:0) 0 : ESC ( 1BH ), LF ( 0AH ), NULL ( 00H ) 方式 1 : { ( 7BH ),   ( 7CH ), } ( 7DH ) 方式

### 2.2.1.1. Conn オプション

以下に Conn オプションの接続パラメータ文字列を示します.

#### 【Ethernet デバイス】

##### TCP/IP

```
"Conn=TCP:<Dest IP Address>:<Dest Port No>"
```

<Dest IP Address> : 接続先の IP アドレス.  
<Dest Port No> : 接続先のポート番号.

### 2.2.2. GaoController::GetVariableNames プロパティ

2.3 に示すシステム変数名の一覧を取得します.

#### 使用例

```
var varNames = this.ctrl.get_VariableNames();
```

### 2.2.3. GaoController::AddVariable メソッド

ラベルプリンタの通信設定やラベルフォーマットの設定値の書き込み/読出しを行うための変数オブジェクトを作成します.

#### 書式

```
AddVariable ( <bstrVariableName:BSTR>, [<bstrOption:BSTR>] )
```

<bstrVariableName > : [in] 変数名  
<bstrOption> : [in] オプション文字列(今回は使用しません)

#### 使用例

```
var controlCode = this.ctrl.AddVariable("@CONTROL_CODE", null);
```

## 2.2.4. CaoController::Execute メソッド

CaoController クラスの Execute メソッドは、コマンドを実行するためのメソッドです。各コマンドの詳細はコマンドリファレンスを参照してください。

**書式** Execute ( <bstrCommandName:VT\_BSTR>,[<vntParam : VT\_VARIANT>])

<bstrCommandName> : [in] コマンド名

<vntParam> : [in] パラメータ

## 2.2.5. CaoController::OnMessage イベント

ラベルプリンタからステータスを受信した際、CaoController クラスの OnMessage イベントとして通知します。このイベントは以下の場合に発生します。

表 2-3 OnMessage 発生タイミング

発生タイミング	詳細
発行終了時 及びエラー発生時(ステータス自動送信)	フィード時, 発行終了時(発行方法が連続発行の場合は指定枚数印字終了後, 剥離発行時は1枚印字終了後), 各種エラー時に発生します。
ステータス要求時	GetStatus コマンドでステータスを要求し, プリンタがステータスを返したときに発生します。13 バイトのステータスを取得したときに発生します。
通信障害時	バッファフルによりポートへの書き込みに失敗した場合に発生します。

※ラベルプリンタの通信設定が COM,ソケット, USB, Bluetooth のいずれかで発行モードが発行完了復帰の場合, ドライバ, ファイルの場合は OnMessage イベントは発生しません。

### 2.2.5.1. メッセージ詳細

メッセージの種類とメッセージで得られるプロパティの内容を以下に示します。

#### ➤ プリンタステータス受信

Number : 1  
 Value : 0x800A044E  
 Description : プリンタから受信したステータス文字列(5.1 参照)  
 DateTime : タイムスタンプ  
 Destination : Null  
 Source : 接続先プリンタの IP アドレス

**➤ その他のメッセージ受信**

Number	:	2
Value	:	プリンタから受信したメッセージコード
Description	:	空文字列
DateTime		タイムスタンプ
Destination		Null
Source		接続先プリンタの IP アドレス

**2.2.6. GaoVariable::get\_Value プロパティ**

2.3 に示すシステム変数やユーザー変数に対応した値を取得します。

**使用例**

---

```
var val = (int)this.value.Value;
```

---

**2.2.7. GaoVariable::put\_Value プロパティ**

2.3 に示すシステム変数やユーザー変数に対応した値を設定します。

**使用例**

---

```
int setValue = 0;  
this.value.Value = tempValue;
```

---

## 2.3. 変数一覧

### 2.3.1. CaoController クラス

表 2-4 CaoController クラス システム変数一覧

変数名	データ型	説明	属性	
			get	put
@MAKER_NAME	BSTR	メーカー名="TOSHIBA TEC"を返します.	○	-
@VERSION	BSTR	プロバイダのバージョンを返します.	○	-
@PORT_SETTING	BSTR	通信パラメータ設定の文字列を返します.	○	-
@RECV_TIMEOUT	I4	受信タイムアウト時間(秒)を取得/設定します.	○	○
@ISSUE_MODE	I4	発行モードを返します. 1 : 送信完了復帰モード 2 : 発行完了復帰モード	○	-
@LFM_FILE_PATH	BSTR	フォーマットファイルの絶対パスを取得/設定します.	○	○
@LOAD_PC_INDEX	I4	PC セーブの識別番号を取得/設定します.	○	○
@SYSTEM_PATH	BSTR	プリンタ情報ファイル格納パスを返します.	○	-
@USE_PRINTER	I4	使用するプリンタ番号の設定値を返します. 5 : B-458-TS 7 : B-858 8 : B-SX5 10 : B-SA4T 11 : B-EP2G 12 : B-EP4T 15 : B-458TR 16 : B-858TR 17 : B-SX8T 18 : B-EV4-T 19 : B-EV4-G 21 : B-EX4T1-T 22 : B-SX8R 23 : B-LV4-T	○	-

		24 : B-LV4-G 28 : B-FP2D-G 32 : B-EX4T3-H 34 : B-EX6T1-T 37 : BA400T-G 38 : BA400T-T 39 : BV400-G 40 : BV400-T 99 : LFM 定義		
@GRAPHIC_TYPE	I4	グラフィックデータ種別を取得/設定します。 0 : ニブルモード(4ドット/バイト) 1 : ヘキサモード(8ドット/バイト) 2 : TOPIX 圧縮モード	○	○
@CONTROL_CODE	I4	コマンド制御コード方式を取得/設定します。 0 : ESC ( 1BH ), LF ( 0AH ), NULL ( 00H ) 1 : { ( 7BH ),   ( 7CH ), } ( 7DH ) 方式	○	○
@PRINTER_STATUS	BSTR	OnMessage イベントまたは GetStatus コマンドで取得したプリンタのステータス文字列を返します。 ※OnMessage イベントが未発生または GetStatus コマンド未実行でステータス文字列を取得していない場合, Null を返します。	○	-
@STATUS_AUTO_SEN D_NO	BSTR	プリンタのステータス文字列のステータス自動送信部分を返します。 ※ステータス文字列が取得できていない場合, Null を返します。	○	-
@STATUS_REQ_NO	BSTR	プリンタのステータス文字列のステータス要求部分を返します。 ※ステータス文字列が取得できていない場合, Null を返す。	○	-
@STATUS_TYPE	I4	プリンタのステータス文字列のステータス種類部分を返します。 1 : ステータス要求時 2 : ステータス自動送信時 ※ステータス文字列が取得できていない場合は Null を返す。	○	-

@STATUS_ISSUE_EXPECTED_COUNT	BSTR	プリンタのステータス文字列の残枚数(現在発行しているラベルについてまだ印字が完了していない枚数)を返します。	○	-
@RESULT_CODE	I4	プリンタから返却される結果番号を返します。	○	-

※@LFM\_FILE\_PATH と@LOAD\_PC\_INDEX を設定した場合,Issue コマンドを実行する際には最後に設定した値に紐づくラベルフォーマットが優先されます。

変数名に印字データを設定するラベルフォーマットのオブジェクト名を指定することで, 印字データの設定が可能です。

表 2-5 CaoController クラス ユーザー変数一覧

変数名	データ型	説明	属性	
			get	put
印字データを設定するラベルフォーマットのオブジェクト名	BSTR	ラベルフォーマットのオブジェクトに設定する印字データを取得/設定するアイテムを作成します。 ※半角で2000桁まで設定可能。また, オブジェクトがビットマップの場合, 絶対パスでファイル名を指定してください。	○	○

## 2.4. エラーコード

TPCL プロバイダ固有のエラーコードはありませんが, BCP Print For .NET で定義されたエラーコードを返す場合があります。BCP Print For .NET のエラーについては BCP Print For .NET 付属のリファレンスマニュアル「第 6 章プリンタ詳細 2.メッセージ」をご参照ください。

また, ORiN2 共通エラーについては ORiN2 プログラミングガイドの「2.2.7.エラーコード」をご参照ください。

## 3. コマンドリファレンス

### 3.1. CaoController クラス

表 3-1 CaoController クラス コマンド一覧

コマンド	機能	ページ
ラベルプリンター操作		
Issue	ラベルプリンタに対して、ラベル発行指示を送信します。	14
Feed	ラベルプリンタに対して、フィード指示を送信します。	15
GetMessage	コードに定義された文字列を取得します。	15
GetStatus	ラベルプリンタに対して、プリンタのステータス取得指示を送信します。	16
SendCommand	ラベルプリンタに対して、コマンド文字列の送信をします。	16
Reset	ラベルプリンタに対して、送信バッファのクリア及びプリンタを初期状態に戻すコマンドを送信します。	17

#### 3.1.1. CaoController::Execute (“Issue”) コマンド

ラベルプリンタにラベル発行指示を送信します。使用するにはラベルフォーマットファイルの指定が必要です。AddController の接続オプションで LfmFilePath を指定するか、コマンド実行前にシステム変数の @LFM\_FILE\_PATH でラベルフォーマットファイルを指定してください。



Issue(<ISSUE\_COUNT>,[<CUT\_INTERVAL>])

ISSUE\_COUNT : [in] 必須. 発行枚数を指定します. 1~9999  
(VT\_I4)

CUT\_INTERVAL [in] 任意. カット間隔を指定します. 0~100 (デフォルト:0)  
(VT\_I4)

0 の場合は、カットを行いません。

B-EP2G, B-EP4G はカット機能がないため、0 を指定してください。ラベルフォーマット設定の発行方法(カットの有無)は無視されません。

戻り値 : [out] 以下の処理に成功した場合は、true,失敗した場合は false を返します。(VT\_BOOL)

通信設定	説明
COM/ソケット (送信完了復帰)	コマンドデータを送信バッファに作成します。
COM/ソケット (発行完了復帰)	発行が完了します。

**使用例**

```
var param = new int[] { 1,0 };

// If the second argument is omitted, "param" may be int.
// int param = 1;
var result = this.ctrl.Execute("Issue", param);
```

**3.1.2. CaoController::Execute ("Feed") コマンド**

ラベルプリンタにフィード(紙送り)指示を送信します。

**書式**

Feed([<CUTTING>])

CUTTING : [in] 任意. フィード(紙送り)後のカット有無を指定します。

0 : カットする (デフォルト値)

1 : カットしない

(VT\_I4)

戻り値 : [out] 以下の処理に成功した場合は, true,失敗した場合は false を返します. (VT\_BOOL)

通信設定	説明
COM/ソケット (送信完了復帰)	コマンドデータを送信バッファに作成します。
COM/ソケット (発行完了復帰)	紙送りが完了します。 プリンタから紙送り完了ステータスを受信します。

**使用例**

```
var result = this.ctrl.Execute("Feed", 1);
```

**3.1.3. CaoController::Execute ("GetMessage") コマンド**

ラベルプリンタにエラーコードやステータスコードに定義された文字列の取得要求を送信します。

**書式**

GetMessage(<PRINT\_STATUS>)

PRINT\_STATUS : [in] 必須. ラベルプリンタから受信したステータス文字列を指定します. (VT\_BSTR)

戻り値 : [out] ステータスに対応した文字列を返します. 該当する情報がない場合は, 空文字を返します.

(VT\_BSTR)

## 使用例

```
var result = this.ctrl.Execute("GetMessage ", "testStatusBSTR");
```

## 3.1.4. GaoController::Execute ("GetStatus") コマンド

ラベルプリンターにプリンタステータスの取得要求を送信します。

## 書式

GetStatus()

戻り値 : [out] プリンタから受信したステータス文字列を返します。  
(VT\_BSTR)

通信設定	説明
COM/ソケット (送信完了復帰)	常に空文字を返します。 プリンタステータスは OnMessage イベントで 取得してください。
COM/ソケット (発行完了復帰)	プリンタから受信したステータス文字列を返 します。

## 使用例

```
var result = this.ctrl.Execute("GetStatus", null);
```

## 3.1.5. GaoController::Execute ("SendCommand") コマンド

ラベルプリンターにコマンド文字列を送信します。 SendCommand を使用して印刷指示コマンドを実行する場合、PC セーブとして登録されたコマンドデータを呼び出しておく必要があります。 AddController の接続オプションで LoadPCIndex を指定するか、コマンド実行前にシステム変数の @ LOAD\_PC\_INDEX で PC セーブの識別番号を指定してください。また、ラベルプリンター本体のリセットをした後も @ LOAD\_PC\_INDEX で PC セーブの識別番号を再度設定する必要があります。

## 書式

SendCommand(<COMMAND>)

COMMAND : [in] 必須. 送信するコマンド文字列 (コマンド制御コードは含まない) を指定します. (VT\_BSTR)

戻り値 : [out] 以下の処理に成功した場合は true, 失敗した場合は false を返します. (VT\_BOOL)

通信設定	説明
COM/ソケット (送信完了復帰)	コマンドデータを送信バッファ に作成します.
COM/ソケット	コマンドデータをポートに出力

(発行完了復帰)	します。 未出力ブロック数が 0 になるまで確認します。
----------	---------------------------------

**使用例**

```
var result = this.ctrl.Execute("SendCommand", "J1;C");
```

**3.1.6. CaoController::Execute (“Reset”) コマンド**

ラベルプリンタに Reset コマンドを送信し、送信バッファのクリア及びラベルプリンタを初期状態に戻します。

**書式**

Reset(< RESET\_TYPE>, [<WAIT>])

RESET\_TYP : [in] 任意. 初期化方式を指定します. (デフォルト:0)(VT\_I4)

- E 0 : 送信バッファのクリアのみ行ないます.
- 1 : 送信バッファのクリアとプリンタの初期化を行ないます.
- 2 : 受信バッファと受信ステータスバッファのクリアを行ないます.

WAIT : [in] 任意. プリンタの初期化を待つ時間(秒)を 1~100 の間で指定します. (デフォルト:3) (VT\_I4)

プリンタにより目安時間は異なります.

B-EX4T1-T の目安時間は、約 3 秒です.

ただし RESET\_TYPE が 0 の場合 (送信バッファのみのクリア) や COM (送信完了復帰), ソケット (発行完了復帰) 以外では待機を行なわないため、この設定は無視されます.

戻り値 : [out] 以下の処理に成功した場合は, true, 失敗した場合は false を返します. (VT\_BOOL)

通信設定	説明
COM/ソケット (送信完了復帰)	RESET_TYPE = 0: 送信バッファをクリアします. RESET_TYPE = 1: コマンドデータを送信バッファに作成します.
COM/ソケット (発行完了復帰)	RESET_TYPE = 0: 送信バッファをクリアします. RESET_TYPE = 1: プリンタの初期化が完了します.

**使用例**

```
var param = new int[] { 1,3 };
```

---

```
// If the second argument is omitted, "param" may be int.  
// int param = 1;  
var result = this.ctrl.Execute("Reset", param);
```

---

## 4. サンプルプログラム

以下にラベルフォーマットを使用して印刷し、プリンタステータスを取得する場合の C# のサンプルを示します。

サンプルプログラムで使用する設定は以下の通りです。実際の環境に合わせて変更してください。

IP アドレス	192.168.10.10	
ポート	8000	
ラベルフォーマットファイルのパス	C:¥tpcl_test¥Ws_PD-A_TPCL.lfm	
BCP Print For .NET インストールフォルダ	C:¥Program Files (x86)¥TOSHIBA TEC¥BCP Print For .NET 2.0 ※パスに()を含む場合、オプション文字列ではパス全体を[]で囲ってください。	
コントローラ	コントローラの接続先プロバイダ	CaoProv.TEC.TPCL
	コントローラ名	TPCL_Sample
アイテム	プリンタから返却される結果番号	@RESULT_CODE

### List 4-1

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading;
using ORiN2.ManagedCAO;

namespace Sample
{
    class Program
    {
        static void Main(string[] args)
        {
            CCaoEngine eng = null;

            try
            {
                // Create Cao objects
                eng = new CCaoEngine();
                CCaoWorkspaces wss = eng.Workspaces;
                CCaoWorkspace ws = wss[0];

                var connOpt = @"Conn=TCP:192.168.10.10:8000,
                    LfmFilePath=C:¥tpcl_test¥Ws_PD-A_TPCL.lfm,
                    SystemPath=[C:¥Program Files (x86)¥TOSHIBA TEC¥BCP Print For .NET 2.0],
                    UsePrinter=21,Timeout=5";

                // Connect to printer
                CCaoController ctrl = ws.AddController("TPCL_Sample",
                    "CaoProv.TEC.TPCL",
                    null,
                    connOpt);

                // Set event handler for message reception
                ctrl.OnMessage += new OnMessageEventHandler(OnMessageEvent);

                var param = new int[] { 1, 0 };
            }
        }
    }
}
```

```
// If the second argument is omitted, "param" may be int.
// int param = 1;
var result = ctrl.Execute("Issue", param);

// Get result code
CCaoVariable caoResultCode = ctrl.AddVariable("@RESULT_CODE", null);
var resultCode = caoResultCode.Value;
var code = (resultCode != null) ? Convert.ToString(resultCode) : string.Empty;
Console.WriteLine("@RESULT_CODE:" + code);

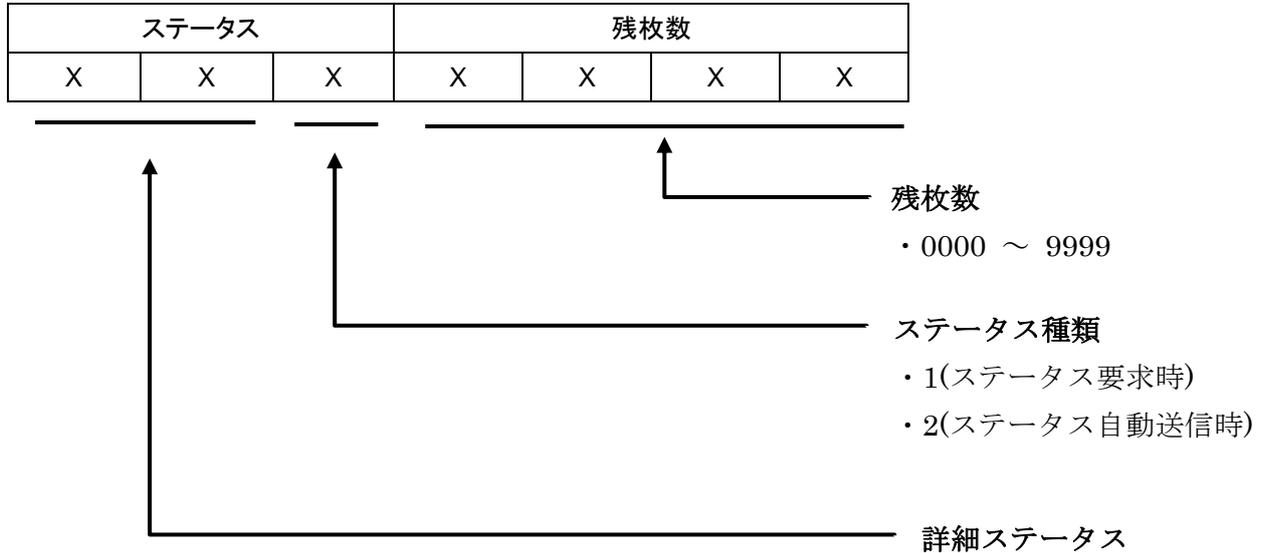
// Wait 10 seconds to accept OnMessage event
Thread.Sleep(10000);
}
catch (Exception ex)
{
    Console.WriteLine(ex.Message);
}
finally
{
    // Disconnect
    if (eng != null)
    {
        eng.Dispose();
        eng = null;
    }
}
}

static void OnMessageEvent(object sender, OnMessageEventArgs e)
{
    var message = e.Message;
    var number = message.Number;
    var value = (int)message.Value;
    Console.WriteLine(string.Format("Message Number: {0} Value: {1} Description: {2}",
        message.Number, value, message.Description));
}
}
}
```

## 5. 付録

### 5.1. プリンタステータス

ラベルプリンタから返されるステータス文字列の構成は以下の通りです。残枚数は現在発行しているラベルについてまだ印字が完了していない枚数が設定されます。



B-EX シリーズの詳細ステータスコードを以下に示します。その他の機種のコードについては BCP Print For .NET 付属のリファレンスマニュアルを参照してください。

表 5-1 B-EX シリーズの詳細ステータス

プリンタ状態	詳細ステータス	
	ステータス 自動送信	ステータス 要求
ヘッドオープン状態時ヘッド部を閉じた	00	00
オンラインモード状態時ヘッド部を開けた	01	01
動作中(コマンド解析中, 描画中, 印字中, フィード中)	-	02
ポーズ状態	-	04
剥離待ち状態	-	05
コマンド解析中にコマンドエラーが発見された	06	06
RS-232C で通信中にパリティエラー, オーバーランエラー, フレーミングエラーのいずれかが発生した	07	07
紙送り中に紙づまりが発生した	11	11
カッター部で異常があった	12	12
ラベルが終了した	13	13

ヘッド部を開けたままフィード, 発行しようとした (FEED キーは除く)	15	15
サーマルヘッドに断線エラーが発生した	17	17
サーマルヘッドの温度が高温になりすぎた	18	18
リボンモータ用のトルク決定のためのセンサーに異常があった	21	21
リワインダー部にてオーバーフローエラーが発生した	22	22
リボンニアエンド(オンライン)*1	-	27
リボンニアエンド(ポーズ中)*2	-	28
リボンニアエンド(動作中)*1	-	29
ラベル発行が正常終了した	40	-
フィードが正常終了した	41	-
断線チェックが正常に終了した	00	-
登録用メモリへの書き込みエラーが発生した	50	50
登録用メモリのフォーマットで消去エラーが発生した	51	51
登録用メモリの空容量が足りなくて登録できない	54	54
外字, PCコマンド登録モード	-	55
ボイドパターン印字終了	60	-
RFID 書き込みエラー	61	61
RFID モジュールと通信が出来ない	62	62
RFID モジュールに使用国が設定されていない	65	65
オンラインモード状態時剥離カバーを開けた	83	83

\*1) Issue コマンドではステータスが戻りません。

このステータスを取得する場合は、事前に **GetStatus** コマンドを実行ください。

\*2) 送信完了モード時 (AddController の接続オプションで IssueMode=1 を指定) の Issue コマンドではステータスが戻りません。

このステータスを取得する場合は、事前に **GetStatus** コマンドを呼び出してください。