MLOCX ユーザのための MLComponent 移行ガイド

株式会社サトー

2022 年 11 月 25 日

はじめに

この度は、「Multi LABELIST Component」(以下 MLComponent)をご利用いただき誠にありがとうございます。MLComponent は、弊社製汎用ラベル・タグ発行ソフトウェア「Multi LABELIST V6」(以下 MLV6)の資産を利用し、お客様のアプリケーションにラベル・タグ発行機能を追加するために開発した.NET コンポーネントです。

MLV6 で作成したレイアウトファイルをもとに、自由度の高いラベル／タグ発行システムを構築していただくために MLV6 の一部の機能は省かせていただきましたが、USB、LAN、COM(シリアルポート)、および弊社製プリンタドライバと、出力デバイスを問わない設計が可能です。ステータス監視機能をサポートしており、プリンタがどのような状態になっているか取得することができます。

本マニュアルでは旧製品の Multi LABELIST OCX（以下 MLOCX）から MLComponent へ移行する際に必要なプロパティやメソッドの変更点をご説明しております。

用途別の利用方法については、「MLComponent テクニックマニュアル」をご参照ください。
プロパティ・メソッド毎の詳細な説明は、「MLComponent リファレンスマニュアル」をご参照ください。

◆本マニュアルは以下の環境で作成しています。

ソフトウェア	Visual Studio 2022 Professional (Version 17.3.6)
	Multi LABELIST V6（Ver.6.0.0.0）
	Multi LABELIST Component（Ver.6.0.0.0）
OS	Windows 10 Enterprise 20H2

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下さい。
- 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。

目次

はじめに 2

ご注意 2

第１章 変更機能 5

1.コントロールを入れ替える 6

■開発環境と.NET Framework 6

■入替え方法 6

■MLComponent.XmlSerializers.dll を配置する 9

■マルチプロセスからマルチスレッドへの移行 11

2.変更する機能を確認する 13

■プロパティー覧 13

■メソッド一覧 14

3.プロパティを変更する 15

■Setting 15

■Protocol 15

■ExOutputAckCheck 15

■OutBufferCount 16

■LayoutFile 16

■OutCut 16

■Siwake 16

■TaxRate 16

4.プロパティを削除する 17

■PrnPath 17

■MemoryCard 17

■COMMode 17

5.メソッドを変更する 18

■OpenPort 18

■Output 18

■ExOutput 19

■ExOutputB 20

■ExInput、ExInputCount 21

■GetPrinter 21

■GetHeaderLayoutFile 21

■GetTailLayoutFile 22

6.メソッドを削除する 23

■AboutBox 23

第２章 新機能 25

7.新機能を確認する 26

■プロパティー覧 26
■メソッドー覧 27
8.新しいプロパティを利用する 28
■Setting 28
■PrnData、PrnDataType 28
■MultiCut 28
■EjectCut 29
■SortMark 29
■HeaderTailSetting 29
■HeaderFile 29
■TailFile 29
■Formoverlay 29
■LayoutNameCaption 29
■TotalQtyCaption 29
■TaxRate 30
■Version 30
9.新しいメソッドを利用する 31
■Output 、SetPrnDataArray 31
■OutputHeader 31
■OutputTail 31
■SendStringData 32
■SendRawData 32
■GetInputFields 32
■GetPrnDataArray 32
■EnumerateBluetoothDevices 32
■AuthenticateBluetoothDevice 32

第１章

変更機能

1 コントロールを入れ替える

■入替方法

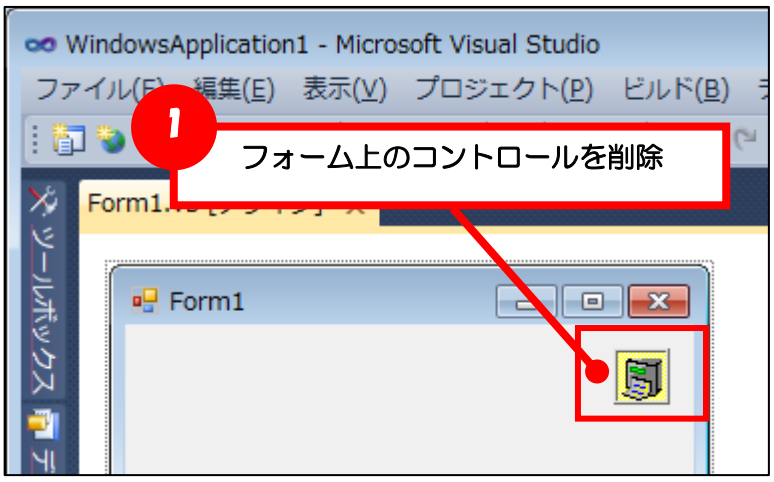
MLOCX はフォーム上にコントロールを配置して利用していましたが、MLComponent は参照の追加を行いコード上でインスタンスを作成して利用します。

■開発環境と.NET Framework

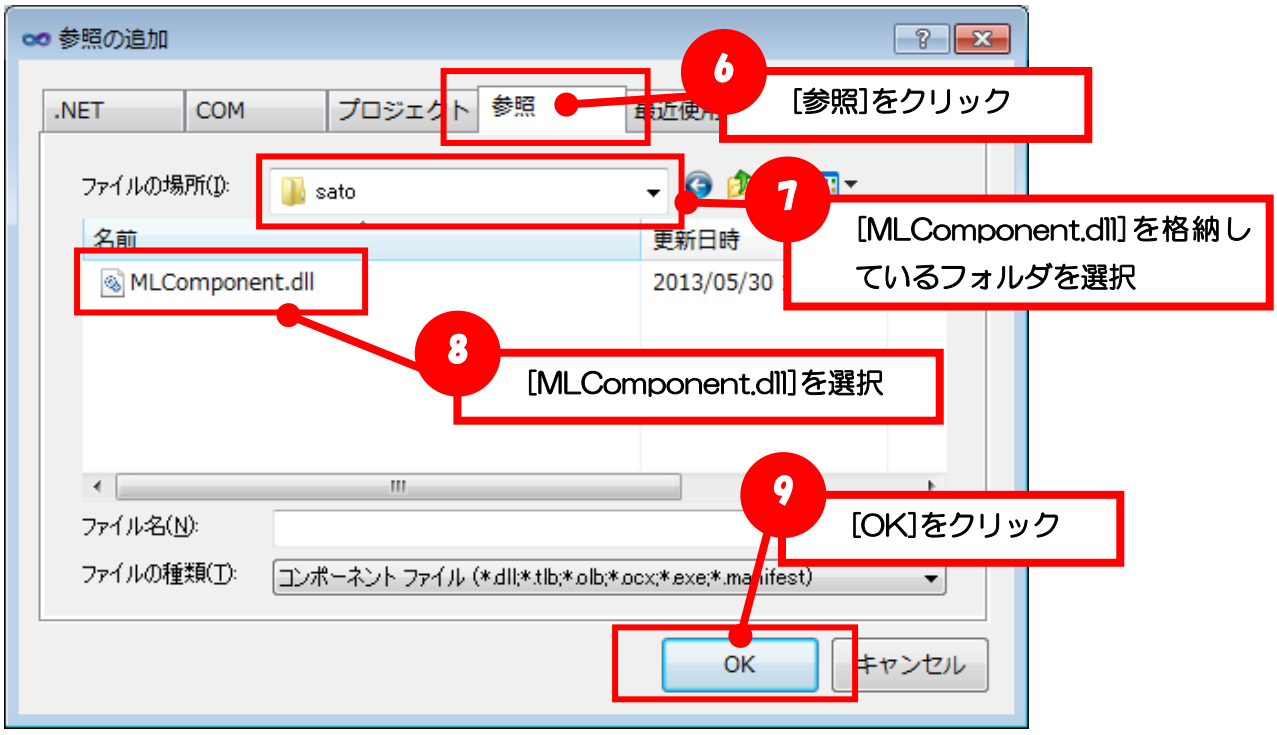
MLOCX は ActiveX コントロールとして提供していましたが、MLComponent は.NET Framework 4.8 で開発した.NET コンポーネントとして提供しています。そのため、お客様が開発するアプリケーションを.NET Framework 4.8 以降で開発いただく必要がございます。

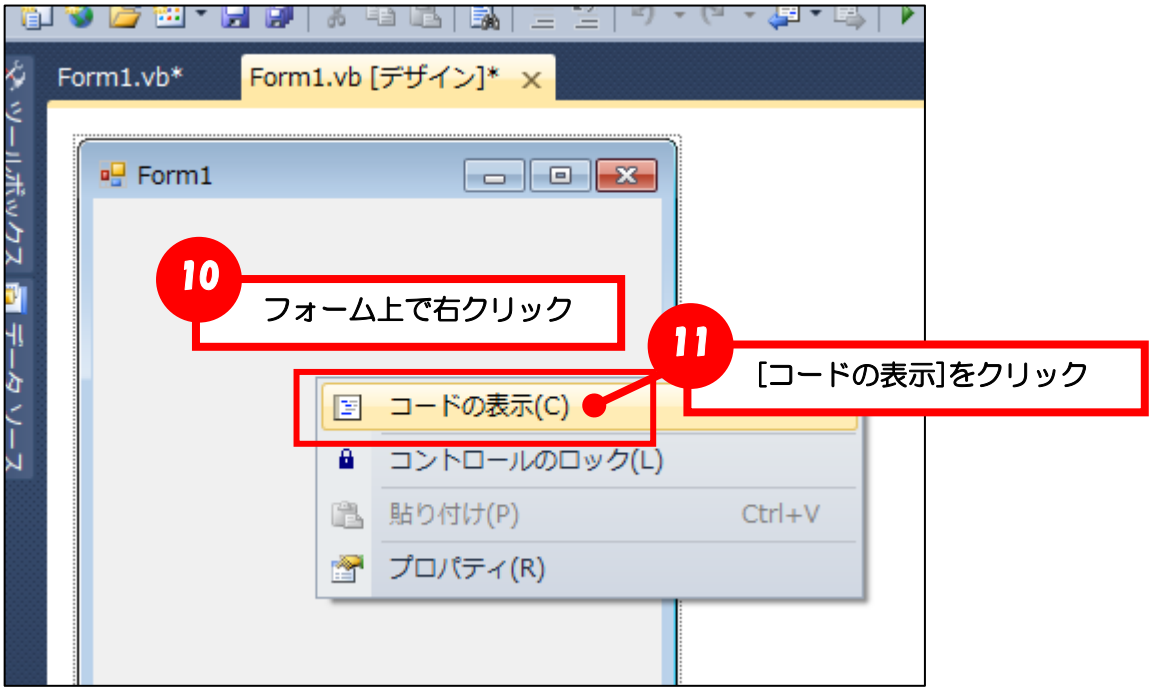
Visual Studio 2019 以前のバージョンではご利用いただけませんのでご注意ください。

■入れ替え方法




4
[プロジェクト]をクリック
プロジェクト(P)
Windows フォームの追加(F)...
ユーザー コントロールの追加(U)...
コンポーネントの追加(N)...
モジュールの追加(M)...
クラスの追加(C)...
新しい項目の追加(W)... Ctrl+Shift+A
既存項目の追加(G)... Ctrl+D
プロジェクトから除外(J)
5
[参照の追加]をクリック
参照の追加(R)...
サービス参照の追加(S)...
プロジェクト ツールボックス アイテムを最新の情報に更新(T)
WindowsApplication1 のプロパティ(P)...


参照の追加
.NET
COM
プロジェクト
参照
6
[参照]をクリック
ファイルの場所(I):
sato
7
[MLComponent.dll] を格納しているフォルダを選択
名前
更新日時
MLComponent.dll
2013/05/30
8
[MLComponent.dll]を選択
ファイル名(N):
ファイルの種類(T):
コンポーネント ファイル (*.dll;*.tlb;*.olb;*.ocx;*.exe;*.manifest)
9
[OK]をクリック
OK
キャンセル



宣言

```
Dim MLComponent As New SATO.MLComponent.MLComponent
```

インスタンス名

既存の MLOCX のコントロール名と合わせるか、
宣言したインスタンス名に
コード上のコントロール名を一括変更します。

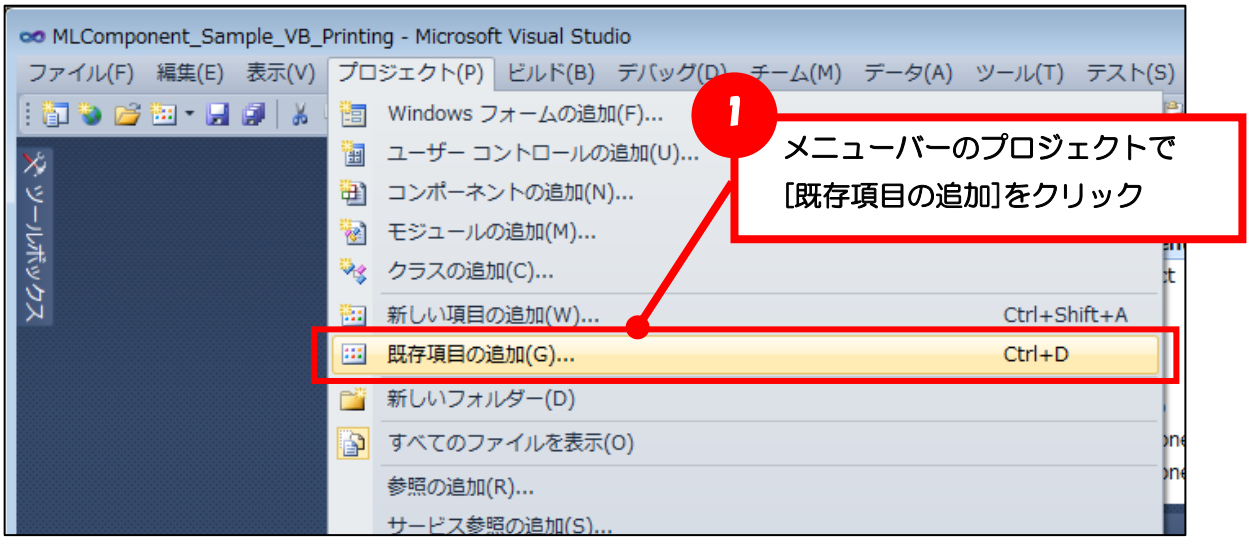
クラス名

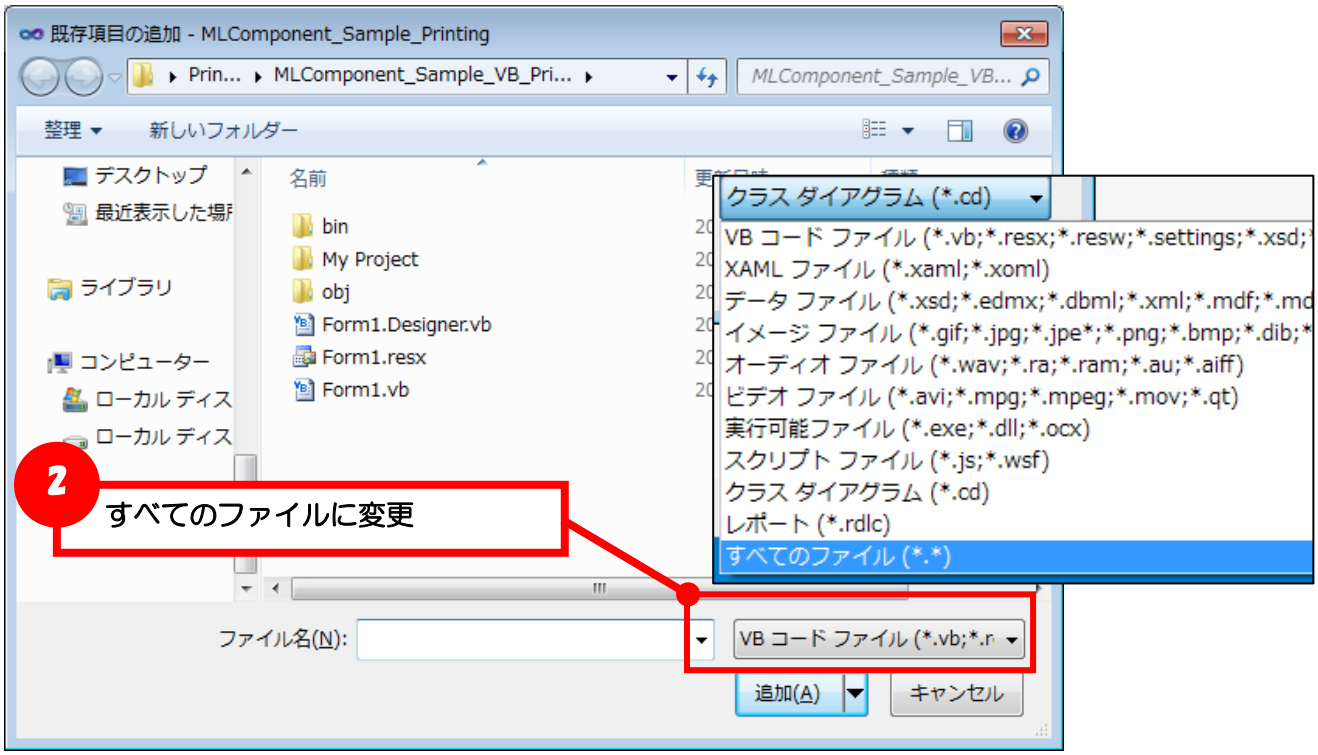
Microsoft Office の VBA で使用する場合は、「MLComponent.MLComponent」と記述してください。

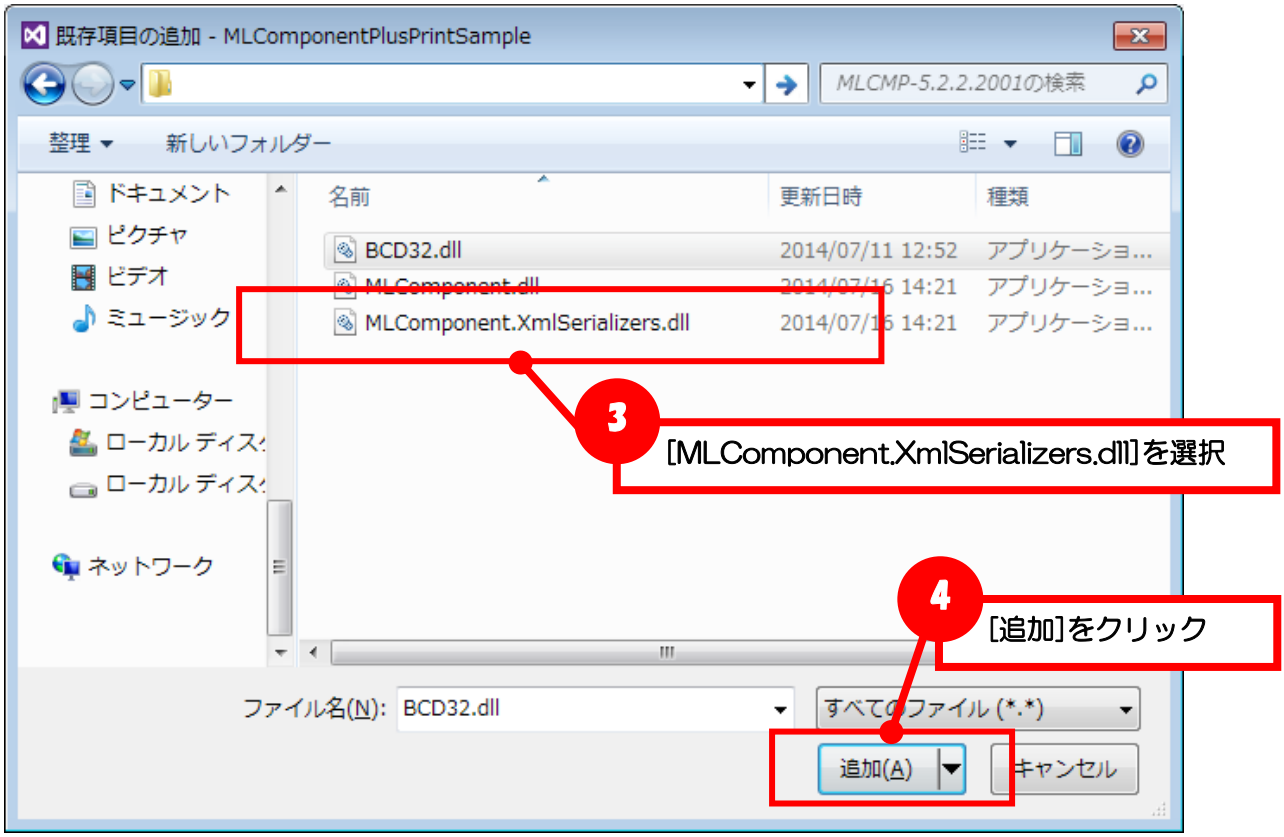
■MLComponent.XmlSerializers.dll を配置する

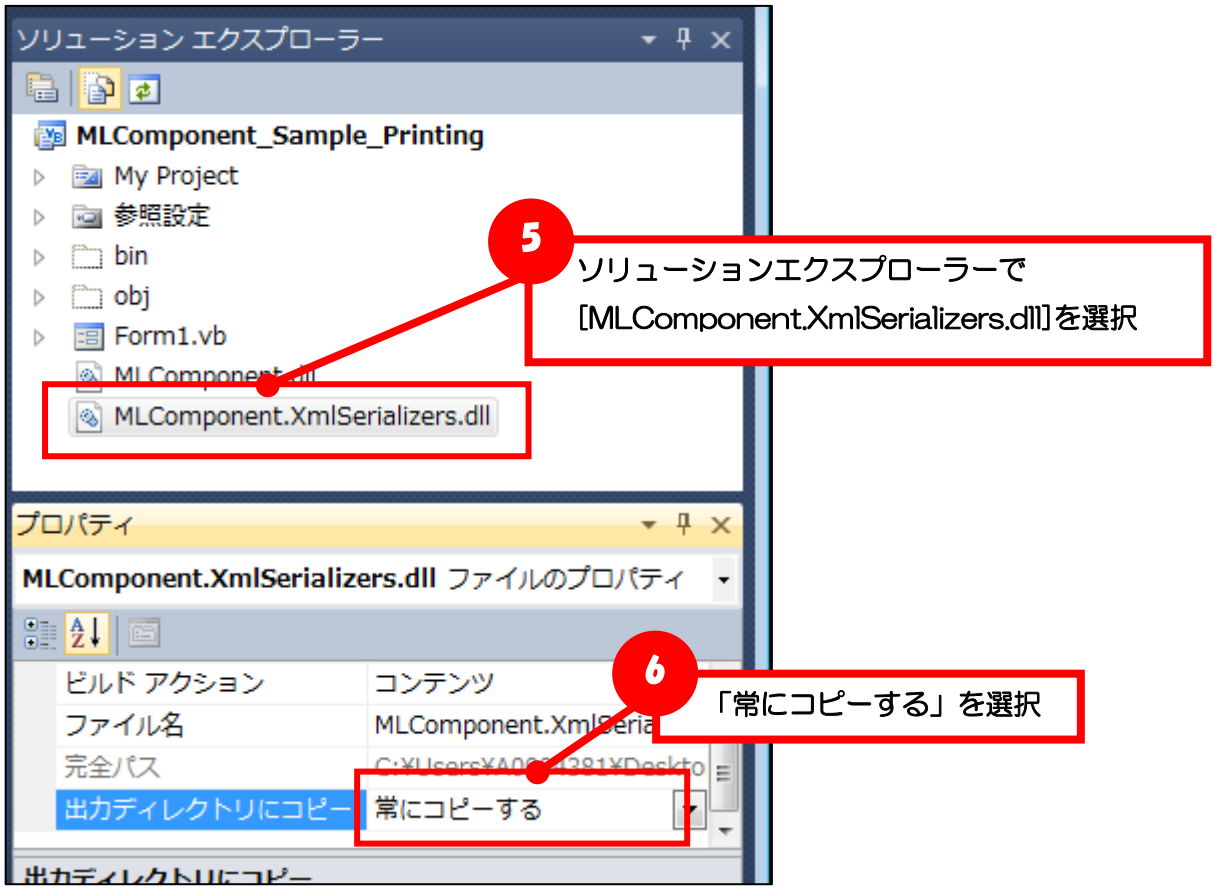
MLComponent で使用している.NET Framework のライブラリがロードされるため、アプリケーション起動後に、初めてレイアウト情報を読み込む時（Output メソッドや GetPrinter メソッドなど）に処理が遅延することがあります。

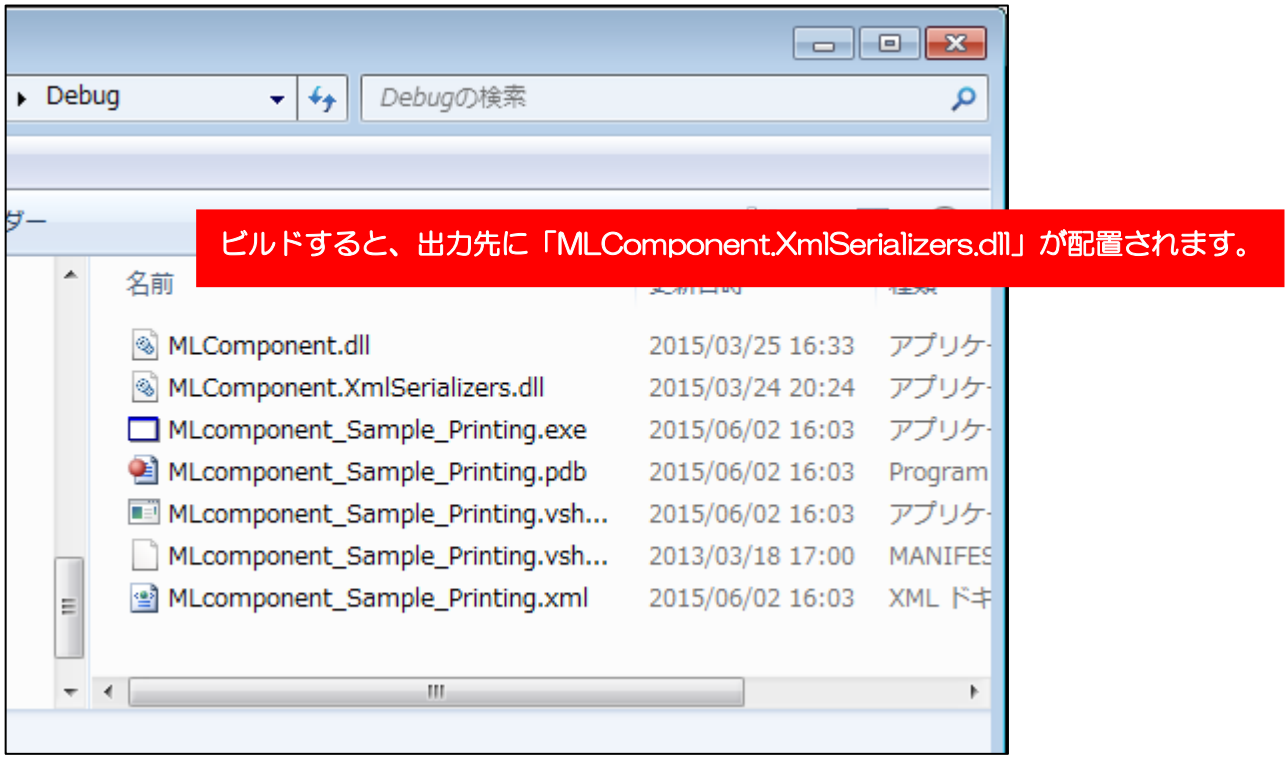
ライブラリのロード時間を改善するために、XML シリアライザ「MLComponent.XmlSerializers.dll」を MLComponent.dll と同じフォルダに配置します。例として Visual Studio でビルドする際に、MLComponent.XmlSerializers.dll を自動的に配置する方法を説明します。






既存項目の追加 - MLComponentPlusPrintSample
MLCMP-5.2.2.2001の検索
整理
新しいフォルダー
ドキュメント
ピクチャ
ビデオ
ミュージック
コンピューター
ネットワーク
名前
更新日時
種類
BCD32.dll
2014/07/11 12:52
アプリケーショ...
MLComponent.XmlSerializers.dll
2014/07/16 14:21
アプリケーショ...
3
[MLComponent.XmlSerializers.dll]を選択
4
[追加]をクリック
ファイル名(N): BCD32.dll
すべてのファイル (*.*)
追加(A)
キャンセル


ソリューション エクスプローラー
MLComponent_Sample_Printing
My Project
参照設定
bin
obj
Form1.vb
MLComponent.XmlSerializers.dll
5
ソリューションエクスプローラーで
[MLComponent.XmlSerializers.dll]を選択
プロパティ
MLComponent.XmlSerializers.dll ファイルのプロパティ
ビルド アクション
コンテンツ
ファイル名
完全パス
出力ディレクトリにコピー
常にコピーする
6
「常にコピーする」を選択



■高 DPI 設定の追加

Windows 10 以降のアプリケーションは高 DPI 設定の追加が必要です。マニフェストファイルまたは実行ファイルのプロパティを設定してください。高 DPI 設定を行っていない場合、アプリケーションの画面が実行中に小さくなる現象や Windows フォントの一部サイズが印字されない現象が発生します。

- マニフェストファイルで<dpiAware>を true または false に設定する。
  https://blogs.msdn.microsoft.com/ttanaka/2014/08/22/dpihigh-dpi-3-12503/
  ※開発アプリケーションの GUI が高 DPI を考慮していない場合は false を設定します。

- アプリケーション（*.exe）のプロパティで、互換性タブの「高い DPI スケールの動作を上書きします。」を有効にし、拡大縮小の実行元を「システム」または「システム（拡張）」を選択する。

■マルチプロセスからマルチスレッドへの移行

MLOCX はマルチスレッドに未対応のため、複数のラベル発行を並列して実行させる場合に、発行処理を別の実行ファイル（EXE ファイル）に変更し、マルチプロセスでの利用が必要でした。

MLComponent は初回起動時に.NET Framework のライブラリロードに時間が掛かるため、マルチプロセスでの発行が遅延する可能性がございます。MLComponent ではマルチスレッドに対応し、速度の向上が実現可能なため、マルチプロセスからマルチスレッドへの移行を推奨いたします。

Visual Studio でマルチスレッドを Thread クラスで実装する場合は、スレッドのネストが深くなると、Windows の不具合により、メモリが解放されない場合があります。Task クラスをご利用いただくか、発行処理スレッドのネストを見直ししてください。

■Microsoft Excel/Access(VBA)で利用する

MLOCX 同様にインストーラを利用して、MLComponent を発行環境にインストールしてください。
以下のサイトからインストーラをダウンロードしてください。
https://www.sato.co.jp/support/printertool/tool/multi-labelist-component/

詳細な利用方法は MLComponent テクニックマニュアルの「Microsoft Excel/Access(VBA)で利用する」をご参照ください。

2 変更する機能を確認する

■プロパティー覧　■メソッドー覧

MLOCX から変更された機能を説明します。

移行の分類

○：コードの変更は必要ありません。

□：コードを変更せずに動作しますが、変更を推奨します。

△：コードを変更してください。

×：コードを削除してください。

■プロパティー覧

プロパティ名	概要	移行
通信設定		
Setting	通信パラメータの設定	△
Protocol	通信プロトコルの設定	△
Timeout	通信タイムアウト値の設定	○
ExOutputAckCheck	ＡＣＫ／ＮＡＫ確認の設定	□
StatusID	ステータスＩＤの設定	○
JobName	ジョブ名の設定	○
OutBufferCount	送信データ数の取得	□
基本設定		
LayoutFile	レイアウトファイルの設定	△
PrnData	印字データの設定	○
PrnPath	プリンタ情報ファイルの設定	×
プリンタ動作設定		
Darkness	印字濃度の設定	○
Speed	印字速度の設定	○
Offset	印字補正値の設定	○
MultiCut	カット指定の設定	○
OutCut	発行終了時のカットの設定	□
Siwake	仕分けマーク印字の設定	□
特殊設定		
Formoverlay	フォームオーバーレイの設定	○
MemoryCard	ﾒﾓﾘｰｶｰﾄﾞ 登録ｸﾞﾗﾌｨｯｸの設定	×
LayoutNameCaption	「レイアウト名」の設定	○
TotalQtyCaption	「総発行枚数」の設定	○
TaxRate	税率の設定	△
互換		
COMMode	ＣＯＭ動作の設定	×

■メソッド一覧

メソッド名	概要	移行
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	ラベル発行	△
ExOutput	プリンタコマンドの送信	□
ExOutputB	プリンタコマンド（バイナリ）の送信	□
ExInput	受信データの取得	□
ExInputCount	返信データバイト数の取得	□
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	○
レイアウト情報取得		
GetPrinter	プリンタ情報の取得	△
GetHeaderLayoutFile	ヘッダ札レイアウトの取得	□
GetTailLayoutFile	テール札レイアウトの取得	□
印字データ指定		
GetInputFields	入力情報の取得	○
SetPrnDataField	入力項目の印字データ設定	○
バージョン情報		
AboutBox	バージョン情報	×

■イベント一覧（非同期モード※非推奨）

イベント名	概要	移行
発行		
OutputStatus	ラベル発行終了	□
ExOutputStatus	プリンタコマンドの送信終了	□
プリンタ制御		
RecvStatus	プリンタの状態確認終了	□
CutStatus	発行中のカット終了	□
CancelStatus	発行のキャンセル終了	□

3 プロパティを変更する

■Setting ■Protcol ■ExOutputAckCheck ■OutBufferCount ■LayoutFile

コードの変更が必要なプロパティについて説明します。

■Setting

- パラレルポート、専用ドライバが使用できなくなりました。プリンタドライバもしくは他のインターフェース出力（USB、LAN、COM）をご利用ください。

使えない記述例

```
MLOCX.Setting = "LPT1:"
MLOCX.Setting = "ODV:MR410e"
```

- 連番変数を使用したレイアウトは、インターフェース出力で使用できなくなりました。連番変数を使用する場合は、プリンタドライバ出力（DRV）に変更してください。

- 多面取り（小ラベル）を使用したレイアウトは、データが 1 シート以上になる場合、インターフェース出力では使用できなくなりました。多面取りを使用する場合は、1 シート毎にデータを指定して発行するか（例として 3 面取りの場合、発行枚数は 3 枚まで）、プリンタドライバ出力（DRV）に変更してください。

- プリンタドライバ出力時に 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。

■Protocol

- ラパン用プロトコル(LapinCOM、LapinIrDA、LapinCOM_CRC)と ReadyBusy が使用できなくなりました。Status3 もしくは Status４をご利用ください。

使えない記述例

```
MLOCX.Protocol = 2
MLOCX.Protocol = 3
MLOCX.Protocol = 4
MLOCX.Protocol = 5
```

■ExOutputAckCheck

- ExOutput、ExOutputB は互換性のために動作しますが、サポート対象外となりました。機能追加・改善は行いませんので、SendStringData、SendRawData への変更をお願い致します。
  詳しくは「メソッドを変更する」をご参照ください。

■OutBufferCount

- 非同期モードは互換性のために動作しますが、サポート対象外となりました。機能追加・改善は行いませんので、同期モードへの変更をお願い致します。

■LayoutFile

- レイアウトファイルの拡張子を mllay から mllayx に変更してください。

変更する記述例

MLOCX.LayoutFile = ″C:¥sato¥label.mllay″
⇒ MLComponent.LayoutFile = ″C:¥sato¥label.mllayx″

■OutCut

- 互換性のために動作しますが、OutCut プロパティに「1」を設定したときのカット動作が MLOCX と異なります。EjectCut プロパティ、MultiCut プロパティへ変更をお願い致します。

変更する記述例

MLOCX.OutCut = 0 ⇒ MLComponent.EjectCut = False
MLOCX.OutCut = 2 ⇒ MLComponent.EjectCut = False
MLComponent.MultiCut = -1
MLOCX.OutCut = 1
（排出カットを利用する場合） ⇒ MLComponent.EjectCut = True
（排出カットを利用しない場合） ⇒ MLComponent.EjectCut = False
MLComponent.MultiCut = -2 ※

※レイアウトファイルの発行時動作設定で、カット動作を「発行指示単位ごとに行う」に設定し、「排出カットを使用する」を無効にしてください。

■Siwake

- 互換性のため動作しますが、SortMark プロパティへ変更をお願い致します。

変更する記述例

MLOCX.Siwake = True ⇒ MLComponent.SortMark = True
MLOCX.Siwake = False ⇒ MLComponent.SortMark = False

■TaxRate

- [税率]を複数指定するために、設定値が数値型（Integer など）から文字列型（String など）に変更されました。設定・取得してる変数の型を変更してください。

4 プロパティを削除する

■PrnPath ■MemoryCard ■COMMode

コード上から削除が必要なプロパティを説明します。

■PrnPath

- アプリケーションと共に配布していたプリンタ毎の情報を格納したプリンタ情報ファイル（PrnObject.mlprn、PrnCommand*.mlprn（*：プリンタＮｏ））が必要なくなりましたので、プロパティを削除してください。

■MemoryCard

- MLV4 のメモリーカードマネージャー機能は MLV6 で利用できませんので、プロパティを削除してください。

■COMMode

- 旧製品 MLPROOCX で利用されていたプロパティでサポート対象外のため、プロパティを削除してください。

5 メソッドを変更する

■OpenPort ■Output ■ExOutput/ExOutputB ■ExInput、ExInputCount

コードの変更が必要なメソッドについて説明します。

■OpenPort

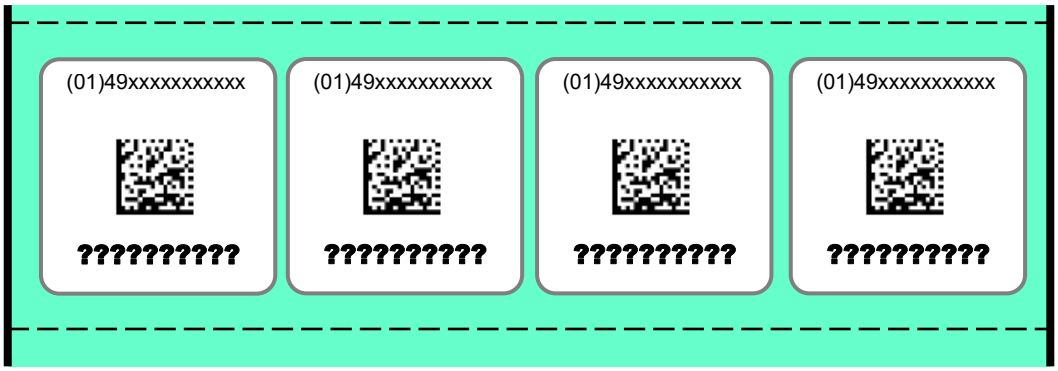
- 非同期通信モードは互換性のため動作しますが、未サポートとなりました。ご利用の際は事前に充分な検証をお願いいたします。

■Output

- 印字データに発行枚数が指定されていない場合に、エラー802 が返送されます。PrnData プロパティまたは SetPrnDataField メソッドで発行枚数を指定してください。

- MLV4 の小ラベルモードを利用したレイアウトは、プリンタドライバ以外では 1 シート以上の発行枚数を指定して発行ができません。Setting プロパティをプリンタドライバ出力に変更するか、PrnData プロパティまたは SetPrnDataField メソッドで入力する発行枚数を 1 シート以内に変更してください。

例)1 シート 4 面取りラベルの場合



変更する記述例

MLOCX.Setting = ″LAN:192.168.1.1,1024″

⇒ MLComponent.Setting = ″DRV:SATO CL4NX-J 305dpi″

MLOCX.PrnData = ″PT208e″ & vbTab & ″490310000005″ & vbTab & ″10″

⇒ MLOCX.PrnData = ″PT208e″ & vbTab & ″490310000005″ & vbTab & ″4″

- 連番変数を利用したレイアウトは、プリンタドライバ以外では発行できません。Setting プロパティをプリンタドライバ出力に変更するか、1 枚ずつ発行する処理に変更してください。

変更する記述例

MLOCX.Setting = ″LAN:192.168.1.1,1024″

⇒ MLComponent.Setting = ″DRV:SATO CL4NX-J 305dpi″

- プリンタドライバ出力時に 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。

■ExOutput

- 互換性のために動作しますが、SendStringData メソッドへ変更をお願い致します。
- 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 MLComponent も MLOCX と同様にデータ送信が保証されません。

変更する記述例

```
MLOCX.ExOutputAckCheck = True
Result = MLOCX.ExOutput( Command )
If Result ◇ 0 Then
  'エラー処理
End If
```

```
        ⇒ Try
            Result = MLComponent.SendStringData( 0 , Command , 0 , Chr(&H6) )
          Catch ex As SATO.MLComponent.MLComponentException
            'エラー処理
          End Try
```

変更する記述例

```
MLOCX.ExOutputAckCheck = False
Result = MLOCX.ExOutput( Command )
If Result = 0 Then
  Length = MLmeControl.ExInputCount
  If Length > 0 Then
    Status = MLmeControl.ExInput( 0 , Length )
  End If
Else
  'エラー処理
End If
```

```
    ⇒ Try
        Result = MLComponent.SendStringData( 0 , Command , Length , EndChr )
        'Result に格納されたデータの処理
      Catch ex As SATO.MLComponent.MLComponentException
        'エラー処理
      End Try
```

Command の内容に応じて受信する条件を、受信するバイト数 Length または特定のデータ EndChr で指定します。データを受信しない場合は、Length を「0」、EndChr を「""」(空文字)に設定してください。

■ExOutputB

- 互換性のために動作しますが、SendRawData メソッドへ変更をお願い致します。
- 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 MLComponent も MLOCX と同様にデータ送信が保証されません。

変更する記述例

```
MLOCX.ExOutputAckCheck = True
Result = MLOCX.ExOutputB(Command())
If Result <> 0 Then
  'エラー処理
End If
```

⇒

```
Try
    Result = MLComponent.SendRawData( 0 , Command() , 0 , Chr(&H6) )
Catch ex As SATO.MLComponent.MLComponentException
    'エラー処理
End Try
```

変更する記述例

```
MLOCX.ExOutputAckCheck = False
Result = MLOCX.ExOutputB(Command())
If Result = 0 Then
  Length = MLmeControl.ExInputCount
  If Length > 0 Then
    Status = MLmeControl.ExInput( 0 , Length )
  End If
Else
  'エラー処理
End If
```

⇒

```
Try
    Result = MLComponent.SendStringData( 0 , Command , Length , EndChr )
    'Result に格納されたデータの処理
Catch ex As SATO.MLComponent.MLComponentException
    'エラー処理
End Try
```

Command()の内容に応じて受信する条件を、受信するバイト数 Length または特定のデータ EndChr で指定します。データを受信しない場合は、Length を「0」、EndChr を「""」(空文字)

に設定してください。

■ExInput、ExInputCount

- 互換性のために動作しますが、SendStringData メソッドもしくは SendRawData メソッドの戻り値で取得する処理へ変更をお願い致します。
  変更方法は「SendStringData」「SendRawData」をご参照ください。

■GetPrinter

- 取得データが[プリンタ番号]から[プリンタ機種名]に変更されました。数値型（Integer など）で取得している場合は、文字列型（String など）に変更してください。

■GetHeaderLayoutFile

- 互換性のために動作しますが、ヘッダ札のパス情報を取得する場合は、HeaderFile プロパティへ変更をお願い致します。

変更する記述例

```
MLOCX.GetHeaderLayoutFile( HeaderLayout )
```

⇒ **HeaderLayout** = MLComponent.HeaderFile

- ヘッダ札の発行は、MLOCX では GetHeaderLayoutFile メソッドで取得したヘッダ用レイアウトファイルのパスを LayoutFile プロパティに設定して、Output メソッドで行いましたが、MLComponent では OutputHeader メソッドだけで利用できます。LayoutFile プロパティに設定されたボディ用レイアウトファイルから情報を自動的に読み取り、ヘッダ札の発行を行います。

変更する記述例

```
MLOCX.LayoutFile = "C:¥sato¥body.mllay"
Result = MLOCX.GetHeaderLayoutFile( HeaderLayout )
If Result = 0 Then
  MLOCX.LayoutFile = HeaderLayout
  MLOCX.PrnData = inputData
  Result = MLOCX.Output()
  If Result <> 0 Then
    'エラー処理
  End If
Else
  'エラー処理
End If
```

⇒ MLComponent.LayoutFile = "C:¥sato¥body.mllayx"

```
MLComponent.PrnData = inputData
Result = MLComponent.OutputHeader()
If Result <> 0 Then
  'エラー処理
End If
```

■GetTailLayoutFile

- 互換性のために動作しますが、テール札のパス情報を取得する場合は、TailFile プロパティへ変更をお願い致します。

変更する記述例

```
MLOCX.GetTailLayoutFile( TailLayout )
```

⇒

```
TailLayout = MLComponent.TailFile
```

- テール札の発行は、MLOCX では GetTailLayoutFile メソッドで取得したテール用レイアウトファイルのパスをLayoutFileプロパティに設定して、Outputメソッドで行いましたが、MLComponentでは OutputTail メソッドだけで利用できます。LayoutFile プロパティに設定されたボディ用レイアウトファイルから情報を自動的に読み取り、テール札の発行を行います。

変更する記述例

```
MLOCX.LayoutFile = "C:¥sato¥body.mllay"
Result = MLOCX.GetTalLayoutFile( TailLayout )
If Result = 0 Then
  MLOCX.LayoutFile = TailLayout
  MLOCX.PrnData = inputData
  Result = MLOCX.Output()
  If Result <> 0 Then
    'エラー処理
  End If
Else
  'エラー処理
End If
```

⇒

```
MLComponent.LayoutFile = "C:¥sato¥body.mllayx"
MLComponent.PrnData = inputData
Result = MLComponent.OutputTail()
If Result <> 0 Then
  'エラー処理
End If
```

6 メソッドを削除する

■AboutBox

コード上から削除が必要なメソッドについて説明します。

■AboutBox

- バージョン確認は Version プロパティをご利用ください。バージョン番号を文字列として返送するため、アプリケーションで自由に編集して利用可能です。

変更する記述例

MLOCX.AboutBox()

⇒ MessageBox.Show(MLComponent.Version)

7 イベントを変更する

■OutputStatus ■ExOutputStatus ■RecvStatus ■CutStatus ■CancelStatus

コードの変更が必要なイベントについて説明します。

■OutputStatus、ExOutputStatus、RecvStatus、CutStatus、CancelStatus

- イベントを利用しない同期モードに変更してください。
  非同期の動作は、標準の Task クラスなどマルチスレッドでご利用ください。

- 非同期モードは非推奨です。プログラムの改修が難しく、互換機能として暫定的に利用する場合は、十分な検証を行って頂くようにお願いいたします。

OutputStatus 記述例　※他のイベントも同様な対応となります

1．MLComponent の宣言時に「WithEvents」を付加。

```
″Dim WithEvents MLComponent As New MLComponent″
```

2．OutputStatus イベントを作成。「Handles」に「MLComponent.OutputStatus」と記述。

```
″Private Sub OutputStatus (ByVal StatusID As Integer, ByVal JobName As String,
  ByVal Result As Integer) Handles MLComponent.OutputStatus″
```

第２章

新機能

7 新機能を確認する

■プロパティー覧　■メソッド一覧

MLOCX から新規に追加された機能、拡張された機能を説明します。

■プロパティー覧

一覧の分類

◎：新しく追加された機能です。

○：拡張された機能です。

変更がない機能は一覧に載せていません。

プロパティ名	概要	分類
通信設定		
Setting	通信パラメータの設定	○
基本設定		
PrnData	印字データの設定	○
PrnDataType	印字データタイプの設定	◎
プリンタ動作設定		
MultiCut	カット指定の設定	○
EjectCut	排出カットの設定	◎
SortMark	仕分けマーク印字の設定	◎
HeaderTailSetting	ヘッダ・テール札の設定	◎
HeaderFile	ヘッダ札の取得	◎
TailFile	テール札の取得	◎
特殊設定		
Formoverlay	フォームオーバーレイの設定	○
LayoutNameCaption	「レイアウト名」の設定	○
TotalQtyCaption	「総発行枚数」の設定	○
TaxRate	税率の設定	○
バージョン情報		
Version	バージョン情報の取得	◎

■メソッド一覧

一覧の分類

◎：新しく追加された機能です。

○：拡張された機能です。

変更がない機能は一覧に載せていません。

メソッド名	概要	移行
発行		
Output	ラベル発行	○
OutputHeader	ヘッダ札発行	◎
OutputTail	テール札発行	◎
SendStringData	プリンタコマンドの送信(終了条件指定)	◎
SendRawData	プリンタコマンド（バイナリ）の送信(終了条件指定)	◎
印字データ指定		
GetInputFields	入力情報の取得	○
GetPrnDataArray	複数データの取得	◎
SetPrnDataArray	複数データの設定	◎
デバイス制御		
EnumerateBluetoothDevices	入力情報の取得	◎
AuthenticateBluetoothDevice	複数データの取得	◎

8 新しいプロパティを利用する

■Setting ■PrnData、PrnDataType ■MultiCut ■EjectCut ■SortMark

MLOCX から追加・拡張されたプロパティを説明します。

■Setting

- インターフェース出力で USB、Bluetooth に対応しました。USB はシリアル ID 指定によりプリンタドライバを利用しないラベル発行が可能です。Bluetooth は BD アドレス指定により仮想 COM ポートを必要とせずラベル発行が可能となり、デバイスの検索・ペアリングも可能です。

記述例（USB）

```
MLComponent.Setting = "USB:"
MLComponent.Setting = "USB:0000T123"
```

記述例（Bluetooth）

```
MLComponent.Setting = "BT:000b5db4aebb"
```

- LAN の[ポート番号]が省略可能になりました。海外プリンタを利用する場合やプリンタのポート番号が変更されている特殊な運用以外では指定する必要がありません。

- COM の[ポート番号]と[ボーレート]が拡張されました。
  [ポート番号] ：1 桁以上の番号指定
  [ボーレート] ：最大値 115200

■PrnData、PrnDataType

- 文字のフォーマットが Unicode に対応しました。中国語や韓国語など多言語のデータが入力可能です。MLOCX 同様に Shift-JIS も利用可能です。

- 区切り文字のカンマ、スペースに対応しました。PrnDataType プロパティでカンマ区切りは「1」、スペース区切りは「2」を指定して利用します。カンマ区切りは括り文字「"」も利用可能です。

■MultiCut

- MLOCX では OutCut プロパティと組合せて設定した「プリンタ本体の動作に従う」動作が、MutilCut プロパティに「-1」を指定するだけで利用可能になりました。

- レイアウトファイルのカット動作設定が利用可能になりました。MutilCut プロパティに「-2」を指定して利用します。

■EjectCut

- MLOCX の OutCut プロパティから「プリンタ本体の動作に従う」機能を取り除いた機能です。論理型（Boolean など）で排出カットの有効・無効を切替えます。

■SortMark

- MLOCX の Siwake プロパティと同一機能です。名称のみ変更されました。

■HeaderTailSetting

- プリンタドライバ出力を利用した場合に、Output メソッド実行時にレイアウトファイルのヘッダ・テール札出力設定に従って自動的にヘッダ札、テール札を発行します。

■HeaderFile

- MLOCX の GetHeaderLayoutFile メソッドと同等機能ですが、値は参照型の引数ではなく、戻り値で取得します。

■TailFile

- MLOCX の GetTailLayoutFile メソッドと同等機能ですが、値は参照型の引数ではなく、戻り値で取得します。

■Formoverlay

- プリンタドライバ出力を利用し、連番変数や複数のデータを指定して発行する場合に、始めに固定項目を登録して、印字データは可変項目のみで送信する機能に対応しました。Formoverlay プロパティに「3」を指定して利用します。

■LayoutNameCaption

- LayoutFile プロパティに設定されたレイアウトファイルから自動的に名称を取得する機能に対応しました。LayoutNameCaption プロパティに「″″」(空文字)を指定して利用します。

■TotalQtyCaption

- Output メソッド実行時に自動的に発行枚数を計算する機能に対応しました。TotalQtyCaption プロパティに「0」を指定して利用します。

■TaxRate

- MLV6 では税率を最大 20 件まで登録できます。TaxRate プロパティで複数の税率を設定可能です。

■Version

- MLOCX の AboutBox メソッドで表示されていたバージョン情報が文字列で取得できます。アプリケーション開発時と違うバージョンの MLComponent が利用された場合にエラー表示するなど取得した文字列を自由に加工してご利用ください。

9 新しいメソッドを利用する

■Output、SetPrnDataArray ■OutputHeader ■OutputTail ■SendStringData

MLOCX から追加・拡張されたメソッドを説明します。

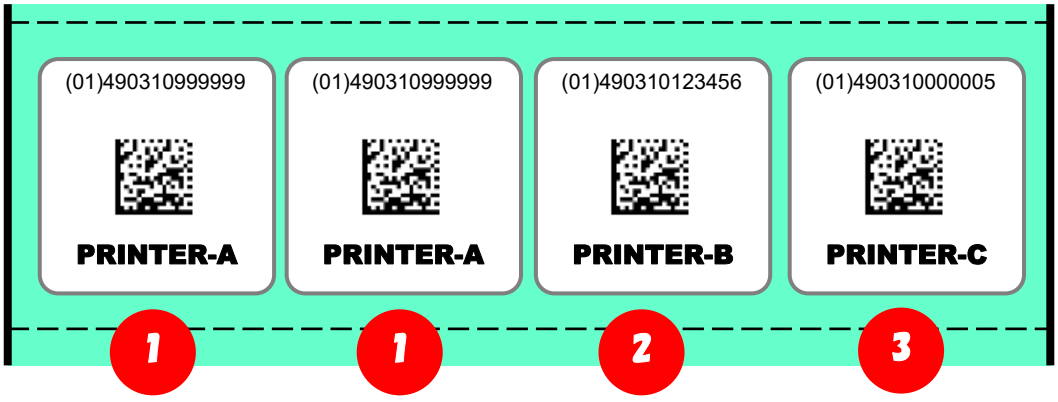
■Output 、SetPrnDataArray

- プリンタドライバ出力を利用した場合に、SetPrnDataArray メソッドで複数のデータを入力して発行する機能に対応しました。同じレイアウトファイルで異なるデータを入力する際にご利用ください。

- 多面取り（小ラベルモード）で作成したレイアウトファイルをプリンタドライバ出力で利用する場合、SetPrnDataArray メソッドで異なるデータを 1 シートに入力することができます。

記述例

```
inputData(0) = "PRINTER-A" & vbTab & "490310999999" & vbTab & "2"
inputData(1) = "PRINTER-B" & vbTab & "490310123456" & vbTab & "1"
inputData(2) = "PRINTER-C" & vbTab & "490310000005" & vbTab & "1"
MLComponent.SetPrnDataArray( inputData )
MLComponent.Output()
```



①inputData(0) ②inputData(1) ③inputData(2)

■OutputHeader

- LayoutFile プロパティに指定されたレイアウトファイルに設定されたヘッダ札を発行します。
- GetHeaderLayoutFile メソッドと Output メソッドを組み合わせた発行方法から移行してください。

■OutputTail

- LayoutFile プロパティに指定されたレイアウトファイルに設定されたテール札を発行します。
- GetTailLayoutFile メソッドと Output メソッドを組み合わせた発行方法から移行してください。

■SendStringData

- 文字列で指定したプリンタコマンド（SBPL）をプリンタに送信します。
- ExOutput メソッド、ExInput メソッド、ExInputCount メソッド、ExOutputAckCheck を組み合わせた発行方法から移行してください。

■SendRawData

- バイト配列で指定したプリンタコマンド（SBPL）をプリンタに送信します。
- ExOutputB メソッド、ExInput メソッド、ExInputCount メソッド、ExOutputAckCheck を組み合わせた発行方法から移行してください。

■GetInputFields

- 入力項目の[初期値]が取得可能になりました。レイアウトで設定された初期値をアプリケーションの初期値として利用できます。
- MLV6 で[入力チェック]の組み合わせが可能のため、GetInputFields メソッドの[入力チェック]が[入力チェック]と[その他チェック]に分かれて取得可能になりました。

■GetPrnDataArray

- 複数データ指定の SetPrnDataArray メソッドで設定されたデータを取得します。正しく設定できたか確認する際にご利用ください。

■EnumerateBluetoothDevices

- 周辺の Bluetooth デバイスを検索できます。AuthenticateBluetoothDevice と共に利用することで、アプリケーションで検索からペアリングまで実現可能です。

■AuthenticateBluetoothDevice

- 指定した Bluetooth デバイスのペアリングが実行できます。EnumerateBluetoothDevices と共に利用することで、アプリケーションで検索からペアリングまで実現可能です。