

## .NET 表形式データ 印刷用 DLL (TablePrint.dll)

### 1. 概要

表形式のデータ(DB,DataGridView,DataTable,Csv)を印刷する .NET 対応の DLL です。  
FrameWork 4.0 で使用可能です。

#### ※ 対応データ

- 1.データベース (ODBC、OledDB) ODBC に対応しているので、殆どのデータベースが印刷可能です。  
例 : Access,SQLServer,MySQL,etc
- 2.DataGridView
- 3.DataTable
- 4.CSV (CSV ファイル、又は、改行で区切られた CSV 形式の文字変数)

※ 最下行に、合計の印刷が可能です。

※ オプションで、バーコード(JAN8,JAN13,Code39,QRCode)の印刷が可能。

### 2. 1 開発手順 (開発環境の前準備)

1.次の2つのファイルを、アプリの exe と同じフォルダにコピーします。

(必須) TalePrint.dll

(選択) DotNetBarcode.dll (BarCode を印刷に対応した DLL の場合)

注) 通常は、...¥bin¥Debug¥ 又は、...¥bin¥Release¥ です。

注) バーコード印刷に対応していない dll の場合は、DotNetBarcode.dll のセットは不要です。

2.上記2の Dll を、参照設定で、開発環境に登録します。

プロジェクト -->参照の追加 -->『参照』タブ

### 2. 2 開発手順 (プログラムの初期処理)

1.ソースの先頭で、インポート宣言します。

```
Imports TablePrint
```

2.フォームに、次の2つのコントロールを配置します。

```
PrintDocument
```

PrintPreviewDialog

3.印刷用のインスタンスを作成します。

```
Private _TablePrint As TablePrint.ClsTablePrint
```

```
' <<< PrintDocument、 PrintPreviewDialog コントロールを指定して New します。
```

```
_TablePrint = New TablePrint.ClsTablePrint(PrintDocument1, PrintPreviewDialog1)
```

4. 次のイベントを登録します。

```
Private Sub PrintDocument1_PrintPage(sender As Object, e As System.Drawing.Printing.PrintPageEventArgs) _  
    Handles PrintDocument1.PrintPage
```

```
_TablePrint.PrintPage(e)
```

```
End Sub
```

## 2. 3 開発手順 (DataGridView の印刷)

印刷したい DataGridView をセットすれば、印刷されます。

```
Private Sub btnデータグリッドを印刷_Click(sender As System.Object, e As System.EventArgs) _  
    Handles btnデータグリッドを印刷.Click
```

```
_TablePrint.印刷初期化("") ' 必須 : 印刷前に初期化します。
```

```
_TablePrint.SetDate(dgv) ' 必須 : 印刷したいDataGridView をセットします。
```

```
_TablePrint.Preview("データグリッドビューの印刷例です") ' 必須 : 印刷内容を、プレビューします。
```

```
End Sub
```

## 2. 4 開発手順 (DataTable の印刷)

印刷したい DataTable をセットすれば、印刷されます。

```
Private Sub btnデータグリッドにデータ表示_Click(sender As System.Object, e As System.EventArgs) _  
    Handles btnデータグリッドにデータ表示.Click
```

```
Dim pDataTable As DataTable = Nothing
```

```
_TablePrint.印刷初期化("") ' 必須 :
```

```
_TablePrint.SetDate(pDataTable) ' 必須 : 印刷したいデータテーブルをセットする。
```

```
_TablePrint.Preview("データテーブルの印刷例です") ' 必須 : 印刷内容を、プレビューします。
```

```
End Sub
```

## 2. 5 開発手順 ( CSV の印刷)

印刷時の列タイトルを、カンマ区切りで指定します。

そして、印刷したい CSV を行数文繰り返し指定します。

\_TablePrint.印刷初期化(“Ti0, T01, T02, …”) '必須:列タイトルを、カンマ区切りで指定。  
 \_TablePrint.SetDataFmCSV(“CSV一行分”) '必須:CSV一行分をセットする。  
 ……必要な行数分、\_TablePrint.SetDataFmCSV を繰り返す。  
 \_TablePrint.Preview(“CSVデータの印刷例です”) '必須:

## 2.6 開発手順 (データベースのテーブルを、ODBC 接続で印刷)

ODBC 接続でデータベースをオープンし、  
 その接続コネクションと、select 文を与えて、select された結果を印刷します。

注) O1eDB接続接続の場合は、O1eDB接続コネクションを指定します。

```
Private Sub btnPreviewDBODBC_Click(sender As System.Object, e As System.EventArgs) _
    Handles btnPreviewDBODBC.Click
    Dim _CnOdbc As Odbc.OdbcConnection
    _CnOdbc = New Odbc.OdbcConnection(“接続文字列”)
    _CnOdbc.Open()
    Dim Select文 as string=” select * from table where。。”

    _TablePrint.印刷初期化() '必須:
    _TablePrint.SetDate(_CnOdbc, Select文) '必須:ODBCのコネクションと、select文を指定
    _TablePrint.Preview(“ODBC接続 DBの印刷例です”) '必須:
End Sub
```

## 3. TablePrint.DLL の関数仕様書

プロパティ

名称	R/W	説明
BarCodeFG	R	この DLL がバーコードに対応しているかどうかを、Boolean で返す。 True:バーコード印刷が可能。 False:対応していない。
用紙設定	R/W	用紙設定を 下記のカンマ区切り文字列で取得、セットする。 用紙={A5,A4,A3,B5,B4,B3},用紙方向={横,縦},FontSize=nn,少数桁数=n, 行高=mm,列最大文字数=nn,上=mm,下=mm,左=mm,右=mm 凡例: { }(どれかの値) n(整数) mm(ミリメートル指定) 注) セットする場合、変更項目のみ記述でも OK。
MarginLeft	R/W	用紙の左マージンを mm で、指定、又は、返す
MarginRight	R/W	用紙の右マージンを mm で、指定、又は、返す
MarginTop	R/W	用紙の上マージンを mm で、指定、又は、返す
MarginBottom	R/W	用紙の下マージンを mm で、指定、又は、返す

用紙	R/W	用紙のサイズを、文字で、取得、セットする。 { "A5", "A4", "A3", "B5", "B4", "B3" }
用紙方向横	R/W	用紙方向を、取得、セットする。 True : 横 False : 縦
最大列幅文字数	R/W	列の最大幅 (半角文字数) を指定する。 これよりも、データが長すぎると、切り値締めて印刷される。
少数桁数	R/W	実数、通貨の印刷をする場合の小数点の桁数を指定します。
FontSizeData	R/W	データ印刷のフォントサイズを整数で指定、例 : 10
横頁数	R	印刷時の横方向のページ数を返す。
縦頁数	R	印刷時の縦方向のページ数を返す
CSVType	R/W	CSV データのタイプ(Integer で指定) -1: 行の先頭が " なら、"" の CSV と判断、(起動時デフォルト) 0: 文字項目を、" " で囲まない 1: " " で囲む 2: ' ' で囲む

## メソッド

関数名	説明
New	Sub New (ByRef pPD As System.Drawing.Printing.PrintDocument, ByRef pPD As System.Windows.Forms.PrintPreviewDialog) 画面に PrintDocument と、PrintPreviewDialog コントロールを配置し、 それを、指定して、New します。
New ※ 解除キーを指定して 呼び出す場合	Sub New (ByRef pPD As System.Drawing.Printing.PrintDocument, ByRef pPD As System.Windows.Forms.PrintPreviewDialog, ByVal p氏名 As String, ByVal p解除キー As String )
解除キー登録画面表示	解除キーの登録画面を表示します。 Function 解除キー登録画面表示 (ByVal pMsg01 As String, ByVal pMsg02 As String) As Boolean Rt=True : 登録に成功
印刷設定	用紙、マージン等の印刷設定の画面を表示します。 Sub 印刷設定 ()
レジストリ読込	レジストリから、印刷情報等の設定情報を読込みます。 設定を元に戻したい時に使用します。 Sub レジストリ読込 ()
印刷初期化	印刷処理を初期化する。印刷実行前に、必ず、最初に呼び出して下さい。 Sub 印刷初期化 (Optional ByVal p列タイトル As String = "", Optional ByVal p列フォーマット As String = "")

	<p>列タイトル：カンマ区切りで、表の列タイトルを指定します。</p> <p>指定しなかった場合、データが持っている列の名称を使用します。</p> <p>だから、通常は、無指定でも問題ありません。</p> <p>列フォーマットの書式は、下記を参照して下さい。</p>
印刷列タイトル	<p>印刷時の列タイトルを明示的に指定します。印刷データを蓄積する前に呼び出します。</p> <p><b>Sub 印刷列タイトル (ByVal p列タイトル As String)</b></p> <p>注) 指定しない場合は、それぞれのデータの列名を印刷します。</p>
印刷列フォーマット	<p>印刷時の列フォーマットを明示的に指定します。印刷データを蓄積する前に呼び出します。</p> <p><b>Sub 印刷列フォーマット (ByVal p列フォーマット As String)</b></p>
SetDataFmCSV	<p>一行分の印刷データを CSVデータをセットする。</p> <p><b>Sub SetDataFmCSV (ByVal pCSV As String)</b></p> <p>注) 複数行ある場合は、これを連続して呼び出します。</p>
SetDataFmCSV	<p>一行分の印刷データを 文字配列にして渡し、データをセットする。</p> <p><b>Sub SetDataFmCSV (ByRef pLnData() As String)</b></p>
SetData	<p>改行で区切られた複数行のCSVデータを、纏めて渡して、セットする。</p> <p><b>Function SetData (</b>  <b>ByRef pCSV As String,</b>  <b>Optional ByVal p列タイトル As String = "") As Integer</b></p> <p>pCSV : 各行は、改行 (cr lf) で区切られている。  各項目は、カンマ区切り  行の先頭が、" で始まる場合は、""で囲まれたCSVと判断する。</p> <p>p列タイトル：  ヌル：CSVの一行目がタイトル行だと判断する。  指定：CSVの一行目から、データ行だと判断する。</p> <p>Rt: 取得した行数を返す。</p>
SetDataFmCSVFile	<p>CSVファイル を指定して、印刷データをセットする</p> <p><b>Function SetDataFmCSVFile (ByRef pCSVFileName As String,</b>  <b>Optional ByVal p列タイトル As String = "") As Integer</b></p> <p>pCSV : 各行は、改行 (cr lf) で区切られている。  各項目は、カンマ区切り  行の先頭が、" で始まる場合は、""で囲まれたCSVと判断する。</p> <p>p列タイトル：  ヌル：CSVの一行目がタイトル行だと判断する。  指定：CSVの一行目から、データ行だと判断する。</p> <p>Rt: 取得した行数を返す。</p>
SetDate	<p>データテーブルを指定して、印刷データをセットする。</p>

	<p><b>Function</b> SetDate(<b>ByRef</b> pDT <b>As</b> DataTable) <b>As</b> Integer</p> <p>Rt: 取得した行数を返す。</p>
SetDate	<p>データグリッドビューを指定して、印刷データをセットする。</p> <p><b>Function</b> SetDate(<b>ByRef</b> pDGV <b>As</b> DataGridView) <b>As</b> Integer</p> <p>Rt: 取得した行数を返す。</p>
SetDate	<p>ODBCの接続文字列と、抽出用SQL文を与えて、抽出結果を、セットする</p> <p><b>Function</b> SetDate(<b>ByVal</b> pCnStr <b>As</b> String,  <b>ByVal</b> pSQL <b>As</b> String,  <b>Optional ByVal</b> pBatFg <b>As</b> Boolean = True,  <b>Optional ByRef</b> pStatusBar <b>As</b> Label = Nothing) <b>As</b> Integer</p> <p>pBatFg = False : データが一件もなかった時に、その旨のメッセージを表示します。  pStatusBar : 作業工程を、ラベルに表示します。</p>
SetDate	<p>ODBCのコネクションと、抽出用SQL文を与えて、抽出結果をセットする</p> <p><b>Function</b> SetDate(<b>ByRef</b> pCn <b>As</b> Odbc.OdbcConnection,  <b>ByVal</b> pSQL <b>As</b> String,  <b>Optional ByVal</b> pBatFg <b>As</b> Boolean = True,  <b>Optional ByRef</b> pStatusBar <b>As</b> Label = Nothing) <b>As</b> Integer</p>
SetDate	<p>OleDbのコネクションと、抽出用SQL文を与えて、抽出結果をセットする</p> <p><b>Function</b> SetDate(<b>ByRef</b> pCn <b>As</b> OleDb.OleDbConnection,  <b>ByVal</b> pSQL <b>As</b> String,  <b>Optional ByVal</b> pBatFg <b>As</b> Boolean = True,  <b>Optional ByRef</b> pStatusBar <b>As</b> Label = Nothing) <b>As</b> Integer</p>
Preview	<p>印刷プレビューを実行する。</p> <p><b>Sub</b> Preview(<b>Optional ByVal</b> p表タイトル <b>As</b> String = "")</p> <p>p 表タイトル : 印刷時の表のタイトルを指定します。</p>
PrintPage	<p>プリントページイベントで呼び出される。</p> <p><b>Function</b> PrintPage(e <b>As</b> System.Drawing.Printing.PrintPageEventArgs,  <b>Optional ByVal</b> p表タイトル <b>As</b> String = "") <b>As</b> Integer</p> <p>印刷画面(From)に次のようなイベントを登録します。</p> <p><b>Private Sub</b> PrintDocument1_PrintPage(sender <b>As</b> Object,  e <b>As</b> System.Drawing.Printing.PrintPageEventArgs)  <b>Handles</b> PrintDocument1.PrintPage  gTablePrint.PrintPage(e)</p> <p><b>End Sub</b></p>

列の印刷書式を指定します。

必要な列のみ、指定します。

複数の列について指定したい場合は、書式をカンマ区切りで羅列します。

書式は、下記のように、各指定要素を、/ で区切り指定します。

書式： 列番号/ DataType/ SumFG/ 少数桁数

例：3列目が金額の列で、合計を取り、少数以下ゼロ、5列目は、JAN8 を印刷としたい場合

-> "3/c/1/0 , 5/B8, , , " と指定する。 注) 空白は無視する。

#### 書式指定

項目	説明
列番号	書式設定したい列番号を指定。左端の列が1番目 列番号=1～
DataType	データタイプによって、印刷書式を指定します。詳細は、下記参照。 記述を省略できます。省略した場合は、極力、データが持っているデータ型を使用します。 データ型が推定できない場合は、文字列 とみなしています。
SumFG	最後の行に、合計計算を印刷するかどうかを指定します。 0：合計を計算しない。(デフォルト) 1：合計を計算する。 記述を省略できます。省略した場合のデフォルトは、ゼロ(合計しない)です。
少数桁数	少数の桁数を指定します。 記述を省略できます。省略した場合のデフォルトは、ゼロです。

#### DataType

記号	データ	表示位置	処理内容
S	文字列	左よせ	何もしない
I	整数	右よせ	何もしない
C	通貨 (少数)	右よせ	3桁毎にカンマ区切りで、小数点表示(実数の場合も) 例 123,456.78
D	日付	左よせ	整数表現の日付型 12345678 -> 1234/56/78 と表示する。
B8	JAN8	左よせ	BarCode JAN8 を印刷
B13	JAN13	左よせ	BarCode JAN13 を印刷
B39	Code39	左よせ	BarCode Code39 を印刷
QR	QRcode	左よせ	QRCode を印刷

#### 呼び出し例

```
Private _TablePrint As TablePrint.ClsTablePrint

_TablePrint = New TablePrint.ClsTablePrint(PrintDocument1, PrintPreviewDialog1, _
```

"テスト用", "xxxx-xxxx-xxxx")

\_TablePrint. 印刷初期化("")

\_TablePrint. 印刷列タイトル("")

\_TablePrint. 印刷列フォーマット("4/c/0/0 , 6/c/1/2")

\_TablePrint. 用紙設定 = "行高=5"

\_TablePrint. SetDate(dgv)

\_TablePrint. Preview("データグリッドビューの印刷例です")

プロジェクト → 参照の追加 で、TablePrint.dll を参照追加する。

以上