

# LightReport2 Documentation

Version 2.2.2

1999-2000 Copyright(c) Takeshi Kanno

## 1. はじめに

---

LightReportコンポーネントは「軽くてシンプルでオープンソース」をコンセプトにして作られた印刷コンポーネントです。

"必要な部分はコードを書く"ことを前提に作られたコンポーネントのため、QuickReportと比べると機能は劣りますがその分軽く作られており、Exeファイルのサイズも小さく押さえられます。

また、膨大なプロパティの中から必要な機能を探し出す手間や、コンポーネント間の複雑な依存関係・細かいバグ等に悩まされることが無いため、若干のコーディングが必要である点を差し引いても、開発効率はむしろアップするはずです。

さらに、ソース公開でカスタマイズしやすいように作られていますので、細かな制御が必要な業務システムの帳票（特に伝票類）を作成する場合にも役立つと思います。

サンプルプログラムを見ていただければわかると思いますが、ほぼあらゆる形の帳票を作成する事が出来ます。また、バージョン2.2からは、文字間隔・行間隔を独自に指定できるTLRVariableTextコンポーネントや郵便番号カスタマーバーコードを出力するTLRPostBarItemコンポーネントが追加され、さらに多種多様の帳票が作成可能になりました。サンプルプログラムの内容については、このドキュメントの「サンプルプログラムの紹介」をご覧ください、また、このコンポーネントを拡張したい場合はこのドキュメントの「開発者ガイド」をご覧ください。

## 2. TLightRepコンポーネント

---

### 2. 1. TLightRep = class(TComponent);

TLightRepは印刷制御を行うコンポーネントです。TLightRep上にパネル・アイテムを追加することにより、複雑な帳票を簡単に出力することができます。また、うまく印刷できない場合のために TextRectやStretchDrawといった描写メソッドが用意されています。さらに、Previewメソッドを使用することにより、任意のCanvasにPreviewを表示することができます。

## 3. TLightRepコンポーネントのプロパティ

---

### 3. 1. property Title: TCaption;

Titleプロパティは印刷中にプリントマネージャに表示されるテキストを指定します。

### 3. 2. property Printer: TPrinter;

Printerプロパティは印刷に使用されるプリンタを返します。

### 3. 3. property PageNumber: integer;

PageNumberプロパティは現在印刷中のページ数を示します。印刷中でない場合は0を返します。

### 3. 4. property PrintCanvas: TCanvas;

PrintCanvasプロパティは現在印刷中のCanvasを返します。印刷中のみ有効なプロパティです。

### 3. 5. property PaperSize: TPaperSize;

PaperSizeプロパティは印刷を行う用紙の大きさを指定します。psNone以外に設定した場合は、自動的にコンポーネントの大きさが設定され、プリンタの設定も自動で行われます。それ以外の用紙を使用したい場合・または特殊な設定が必要な場合は、用紙サイズを psNoneに設定し、OnSetupPrinterイベントで独自に設定を行います（自動では行われません）。

### 3. 6. property Orientation: TPrinterOrientation;

Orientationプロパティは印刷する向きを指定します。

### 3. 7. property MarginTop: integer;

MarginTopプロパティはレポートの上方向のマージンを指定します。

### 3. 8. property MarginLeft: integer;

MarginLeftプロパティはレポートの左方向のマージンを指定します。

### 3. 9. property MarginRight: integer;

MarginRightプロパティはレポートの右方向のマージンを指定します。

### 3. 10. property MarginBottom: integer;

MarginBottomプロパティはレポートの下方向のマージンを指定します。

## 4. TLightRepコンポーネントのメソッド

---

### 4. 1. procedure BeginDoc;

BeginDocメソッドは印刷を開始します。

### 4. 2. procedure BeginPreview;

BeginPreviewメソッドは、プレビュー開始時に呼び出され、レポートをプレビューモードに設定します。

### 4. 3. procedure Print;

Printメソッドは、TLightReport上に配置されたコンポーネントの印刷を行います。複数ページの印刷を行う場合は、このメソッドを複数回呼び出します。procedure EndDoc; EndDocメソッドは、印刷を終了します。

### 4. 4. procedure NewPage;

NewPageメソッドは、現在のページの印刷を終了し、改ページを行います。

### 4. 5. procedure StretchDIBDraw(Rect: TRect; Graphic: TGraphic);

StretchDIBDrawメソッドは、Graphicで指定されたグラフィックを現在のページのRectで指定した位置に印刷します。

### 4. 6. procedure TextRect(Rect: TRect; X, Y: Integer; const Text: string; Font: TFont);

Textで指定された文字列を現在のページのRectで指定した位置に出力します。

## 5. TLightRepコンポーネントのイベント

---

### 5. 1. property OnSetupPrinter: TNotifyEvent;

OnSetupPrinterイベントは、用紙サイズを psNoneに設定した場合に、印刷開始時に呼び出されます。独自の用紙を使用したい場合は、OnSetupPrinterイベントのイベントハンドラを作成してそこでプリンタ設定のコードを書くか、またはプリンタセットアップのダイアログを呼び出してプリンタの設定を行います。

OnSetupPrinterイベントはプレビュー時には呼び出されません。

## 6. TLRLayoutPanelコンポーネント

---

### 6. 1. TLRLayoutPanel = class(TCustomLRPanel);

TLRLLayoutPanelコンポーネントは、レポートのレイアウトを構成するパネルコンポーネントです。LRAlignプロパティを設定することで、位置決めを行います。また、ネストした配置をサポートしているため、複数のパネルを重ねて配置することで、複雑なレポートを作成可能です。

## 7. TLRLayoutPanelコンポーネントのプロパティ

---

### 7. 1. property LRAAlign: TLRAAlign;

**TLRAAlign = (laTop, laLeft, laRight, laClient, laBottom);**

LRAAlignプロパティは、コンテナ（親コントロール）内でコントロールを揃える方法を指定します。

一般的な帳票の場合、TLightRep上にLRAAlignをlaTopに指定したパネル(ヘッダー)、laBottomに指定したパネル(フッター)の2つを置き、間にlaClientに指定した明細部分を置く構造になります。

## 8. TLRLayoutPanelコンポーネントのイベント

---

### 8. 1. property BeforePrint: TBeforePrintEvent;

**BeforePrintEvent = procedure(Sender: TObject;  
Rect: TRect; var Status: TPrintStatus) of object;**

BeforePrintイベントは、パネルを印刷する直前に発生します。このイベントプロシージャ内で、アイテムに値を代入することにより、実行時に、レポートを変更することが出来ます。また、TPrintStatusに値を代入することで、印刷の制御を行うことが出来ます。

#### **PrintStatusの値と動作**

psNormal: 特に制御は行わない

psLineOnly: アイテムは、枠線のみ印刷される。

psEndNow: 印刷は行われない

### 8. 2. property AfterPrint: TNotifyEvent

AfterPrintイベントは、パネルを印刷した後に発生します。

## 9. TLRGridPanelコンポーネント

---

### 9. 1. TLRGridPanel = class(TCustomLRPanel);

TLRGridPanelコンポーネントは、表形式のレポートを構成するパネルコンポーネントです。LRAlignプロパティを設定することで、位置決めを行います。ColCount、RowCountプロパティに、値を設定することにより簡単に表形式のレポートを作成することができます。

## 10. TLRGridPanelコンポーネントのプロパティ

---

### 10. 1. property LRAAlign: TLRAAlign;

TLRAAlign = (laTop, laLeft, laRight, laClient, laBottom);

LRAAlignプロパティは、コンテナ（親コントロール）内でコントロールを揃える方法を指定します。

### 10. 2. property ColCount: integer;

ColCountプロパティは、表の列数を指定します。

### 10. 3. property RowCount: integer;

RowCountプロパティは、表の行数を指定します。

### 10. 4. property ReplacePanel: TLRSubPanel;

実行時のみのプロパティです。ReplacePanelプロパティに、nil以外の値を代入することにより、印刷するパネルを入れ替えることができます。

### 10. 5. property PrintDirection: TPrintDirection;

TPrintDirection = (pdHolz, pdVert);

PrintDirectionプロパティは、グリッドの印刷順序を指定します。pdHolzに設定した場合は左上=>右上=>左下=>右下の順で、pdVertに指定した場合は左上=>左下=>右上=>右下の順で印刷をおこないます。

## 11. TLRGridPanelコンポーネントのイベント

---

### 11. 1. property BeforePrint: TBeforePrintEvent;

```
TBeforePrintEvent = procedure(Sender: TObject; Rect: TRect;  
    var Status: TPrintStatus) of object;
```

```
TPrintStatus = (psNormal, psLineOnly, psEndNow);
```

BeforePrintイベントは、パネルを印刷する直前に発生します。このイベントプロシージャ内でアイテムに値を代入することにより、実行時にレポートを変更することが出来ます。また、TPrintStatusに値を代入することで、印刷の制御を行うことが出来ます。

#### PrintStatusの値と動作

psNormal: 特に制御は行わない

psLineOnly: アイテムは、枠線のみ印刷される。

psEndNow: 印刷は行われぬ

### 11. 2. property OnPrintPanel: TPrintPanelEvent;

```
TPrintPanelEvent = procedure(Sender: TObject; ACol, ARow: integer;  
    Rect: TRect; var Status: TPrintStatus) of object;
```

OnPrintPanelは、パネル上の個々の要素を印刷する度に発生します。ACol,ARowパラメータが、印刷する表要素の位置をあらわします。

### 11. 3. property AfterPrint: TNotifyEvent

AfterPrintイベントは、パネルを印刷した後に発生します。

### 11. 4. property AfterPrintPanel: TNotifyEvent

AfterPrintPanelイベントは、パネル内の個々の子パネルを印刷する際に発生します。ReplacePanelプロパティにより印刷するパネルが入れ替えられている場合は、このイベントの代わりにReplacePanelプロパティで指定されたTSubPanelのAfterPrintイベントが呼び出されます。

## 12. TLRSubPanelコンポーネント

---

### 12. 1. TLRSubPanel = class(TCustomLRPanel);

TLRSubPanelコンポーネントはTLRGridPanelと組み合わせて使用し、主に合計行等でレイアウトを変更したい場合に使用します。このパネルは他のパネルと違い、TLightReport上以外の場所しか配置できません。

TLRSubPanelコンポーネントは、GridPanelプロパティを持ち、関係を持つTLRGridPanelコンポーネントを指定します。実行時に、TLRGridPanelコンポーネントのReplacePanelプロパティにTLRSubPanelコンポーネントを代入すると、TLRGridPanel上のパネルの代わりに、指定したTLRSubPanelコンポーネントを印刷することができます。くわしい使用例は、サンプルプログラムSubPanel.dprを参考にしてください。

## 13. TLRSubPanelコンポーネントのプロパティ

---

### 13. 1. property GridPanel: TLRGridPanel;

GridPanelプロパティは、自分の親となるTLRGridPanelコントロールを指定します。コントロールのサイズは、GridPanelコンポーネントのグリッドのサイズに自動的に調整されます。

### 13. 2. property BeforePrint: TBeforePrintEvent;

```
TBeforePrintEvent = procedure(Sender: TObject; Rect: TRect;  
                               var Status: TPrintStatus) of object;
```

```
TPrintStatus = (psNormal, psLineOnly, psEndNow);
```

BeforePrintイベントは、パネルを印刷する直前に発生します。このイベントプロシージャ内でアイテムに値を代入することにより、実行時にレポートを変更することが出来ます。また、TPrintStatusに値を代入することで、印刷の制御を行うことが出来ます。

#### PrintStatusの値と動作

psNormal: 特に制御は行わない

psLineOnly: アイテムは、枠線のみ印刷される。

psEndNow: 印刷は行われない

## 14. TLRItemコンポーネント

---

### 14. 1. TLRItem = class(TCustomLRItem);

TLRItemコンポーネントはレポートに印刷を行うテキストや罫線または他のコンポーネントの背景として使用する汎用コンポーネントです。表示するテキストはTextプロパティに指定します。Alignmentプロパティで左右、Layoutプロパティで上下の表示位置を設定します。HMargin・VMarginプロパティで上下・左右の余白を設定します。DisplayMaskプロパティを指定することにより、文字列のフォーマットを行うことができます。LineTop ~ LineBottomプロパティを設定することにより、罫線の印刷を行うことができます。

## 15. TLRItemコンポーネントのプロパティ

---

### 15. 1. property Align: TAlign;

Alignプロパティは、コンテナ（親コントロール）内でコントロールを揃える方法を指定します。

### 15. 2. property Alignment: TAlignment;

Alignmentプロパティを設定すると、ラベルコントロールの ClientRect 内でラベルのテキストを揃える方法を指定できます。Alignment では次の値を指定できます。

taLeftJustify: テキストはラベルの左端に揃えられる

taRightJustify: テキストはラベルの右端に揃えられる

taCenter: テキストはラベル内の横方向の中央に揃えられる

### 15. 3. property Color: TColor;

Colorプロパティはコントロールの背景色を指定します。Transparentプロパティにfalseが設定されているときにのみ有効です。

### 15. 4. property DisplayMask: string;

DisplayMaskプロパティは、Textで指定された文字列が数字の場合はFormatFloat関数・そうでない場合はFormatMaskText関数を使用してフォーマットします。詳細はDelphiのヘルプファイルのFormatFloatおよびFormatMaskText関数を参考にしてください。

### 15. 5. property Font: TFont;

Fontプロパティは使用されるテキストのフォントを指定します。

### 15. 6. property HMargin: integer;

HMARGINプロパティは水平方向の余白（罫線と文字列の間）の設定を行います。AlimentプロパティがtaCenter以外の場合のみ有効です。

### 15. 7. property LineColor: TColor;

LineColorプロパティは罫線の色を指定を行います。

### 15. 8. property LineTop: TLineStyle;

LineTopプロパティは上罫線の指定を行います。上罫線とはコンポーネントの上端にそって描画される直線です。指定できる値は以下の通りです。

IsNone: 罫線は表示（印刷）されない

IsSingle: 罫線は標準の幅で描写される。

IsHalf: 罫線は半分の幅で描写される。

IsDouble: 罫線は2倍の幅で描写される。

### 15. 9. property LineLeft: TLineStyle;

LineLeftプロパティは左罫線の指定を行います。指定できる値はLineTopプロパティを参照してください。

### 15. 10. property LineRight: TLineStyle;

LineRightプロパティは右罫線の指定を行います。指定できる値はLineTopプロパティを参照してください。

### 15. 11. property LineBottom: TLineStyle;

LineBottomプロパティは下罫線の指定を行います。指定できる値はLineTopプロパティを参照してください。

### 15. 12. property Layout: TTextLayout;

Layout プロパティは、テキストの縦方向の配置を指定します。

tlTop: テキストはラベルの上端に表示される

tlCenter: テキストはラベル内の縦方向の中央に表示される

tlBottom: テキストはラベルの下端に表示される

**15. 13. property Text: string;**

Text プロパティは印刷する文字列を指定します。

**15. 14. property Transparent: boolean;**

Transparent プロパティは背景を透明にするかどうかを指定します。。

**15. 15. property VMargin: integer;**

VMargin プロパティは垂直方向の余白（罫線と文字列の間）の設定を行います。

Layout プロパティが taCenter 以外の場合のみ有効です。

**15. 16. property Wordwrap: boolean;**

Wordwrap プロパティは、テキストがラベルの幅より長い場合にテキストを改行するかどうかを指定します。true に指定された場合、コンポーネントの幅に入りきらない文字列は次の行に改行されて出力されます。改行は1バイト文字の場合は単語（半角スペースで区切られた）単位、2バイト文字の場合は文字単位で行われます。また、2バイト文字で行の先頭が"。","、" になった場合は一文字前で改行されます（禁則処理）。

## 16. TLRVariableTextコンポーネント

---

### 16. 1. TLRVariableText = class(TCustomLRItem)

TLRVariableTextはレポートに印刷を行うテキストを扱うコンポーネントです。  
TLRVariableTextはCharSpace, WordSpace, Leading等のプロパティを設定することで文字間隔・単語間隔・行間隔を自由に変更し、見栄えのよい帳票を作成することを可能にします。

## 17. TLRVariableTextコンポーネントのプロパティ

---

### 17. 1. property Align: TAlign;

Alignプロパティは、コンテナ（親コントロール）内でコントロールを揃える方法を指定します。

### 17. 2. property Color: TColor;

Colorプロパティはコントロールの背景色を指定します。Transparentプロパティにfalseが設定されているときにのみ有効です。

### 17. 3. property CharSpace: integer;

CharSpaceプロパティは単語間の間隔を指定します。

### 17. 4. property Font: TFont;

Fontプロパティは使用されるテキストのフォントを指定します。

### 17. 5. property Leading: integer;

Leadingプロパティは行間隔を指定します。

### 17. 6. property Text: string;

Textプロパティは印刷する文字列を指定します。

### 17. 7. property Transparent: boolean;

Transparentプロパティはアイテムを透明にするかどうかを指定します。

### 17. 8. property VMargin: integer;

VMarginプロパティは垂直方向の余白（罫線と文字列の間）の設定を行います。LayoutプロパティがtaCenter以外の場合のみ有効です。

### 17. 9. property WordSpace: integer;

WordSpaceは単語間の間隔を指定します。単語とは半角スペースで区切られた1バイト文字のことであり、文字列が全て2バイト文字の場合や半角スペースが存在しない場合は無視されます。

### 17. 10. property Wordwrap: boolean;

Wordwrapプロパティは、テキストがラベルの幅より長い場合にテキストを改行するかどうかを指定します。trueに指定された場合、コンポーネントの幅に入りきらない文字列は次の行に改行されて出力されます。改行は1バイト文字の場合は単語（半角スペースで区切られた）単位、2バイト文字の場合は文字単位で行われます。また、2バイト文字で行の先頭が"。","、"になった場合は一文字前で改行されます（禁則処理）。

## 18. TLRIImageコンポーネント

---

### 18. 1. TLRIImage = class(TCustomLRItem)

TLRIImageコンポーネントは画像データを扱うコンポーネントです。TLRIImageはビットマップまたはJPEGイメージを設計時または実行時にロードすることで、画像データを印刷することが出来ます。また、susieプラグインコンポーネント等と組み合わせることで実行時にBMP、JPEG以外のイメージをロードして印刷することも可能です。

## 19. TLRImageコンポーネントのプロパティ

---

### 19. 1. property Align: TAlign;

Alignプロパティは、コンテナ（親コントロール）内でコントロールを揃える方法を指定します。

### 19. 2. property Picture: TPicture;

Pictureプロパティは、レポート上に印刷するイメージを指定します。指定できるイメージは、BitmapもしくはJPEGイメージです。設計時に Picture を設定すると画像の設定ダイアログが起動され、イメージを含むファイルを指定できます。

### 19. 3. property Stretch: boolean;

Stretchプロパティは、Picture プロパティで指定されたイメージのサイズを変更しTLRImageコンポーネントの枠に完全にはめ込みます。

## 20. TLRCustBarItemコンポーネント

---

### 20. 1. TLRCustBarItem = class(TCustomLRItem);

TLRCustBarItemは郵便番号のカスタマーバーコードを出力するためのコンポーネントです。Codeプロパティにカスタマーバーコード作成のための住所情報を設定することでカスタマーバーコードを出力することが出来ます。また、郵便番号バーコードの末尾のチェックデジットの計算も自動でおこなっています。

TLRCustBarItemはコンポーネントの高さを基準にして、バーコードイメージを拡大/縮小して印刷をおこないますが、最終的に印刷されるバーコードは、64mm～88mmの大きさになる必要があります。また、プリンタの種類によっては（インクのにじみ等で）識別できない場合もありますので、必ずテスト後に使用してください。

## 21. TLRCustBarItemコンポーネントのプロパティ

---

### 21. 1. property Code: string;

Codeプロパティは、郵便番号カスタマーバーコード作成のための住所情報を設定します。現在のところ漢字住所の解析ルーチンは備えていないため、カスタマーバーコードに必要な文字情報はプログラムで生成する必要があります。この情報は

[www.mpt.go.jp/pressrelease/japanese/yubin/0912d202.html](http://www.mpt.go.jp/pressrelease/japanese/yubin/0912d202.html)

[www.mpt.go.jp/pressrelease/japanese/yubin/0912d203.html](http://www.mpt.go.jp/pressrelease/japanese/yubin/0912d203.html)

より入手することができます。

## 22. 単票形式のレポートの作成方法

---

### 22. 1. TLightRepコンポーネントの配置

フォーム上にTLightRepコンポーネントを配置します。

### 22. 2. TLRLayoutPanelコンポーネントの配置

TLightRepコンポーネント上にTLRLayoutPanelコンポーネントを配置し、LRAlignをlaClientに設定します。

### 22. 3. TLRItemコンポーネントの配置

TLRLayoutPanelコンポーネント上に適当にTLRItemを配置し、TLRItemのFont、Alignment、Color、LineTop ~ LineBottomプロパティを設定します。固定項目についてはこの時点でTextプロパティに文字を設定します。

### 22. 4. BeforePrintイベントハンドラの作成

TLRLayoutPanelのBeforePrintイベントでTLRItemに文字列を代入したい場合（ページ数等を入れたい場合はこの方法が簡単です）以下のようなコードを書きます。

```
procedure TForm1.LRLayoutPanel1BeforePrint(Sender: TObject;  
    Rect: TRect; var Status: TPrintStatus);  
begin  
    LRItem1.Text := 'PAGE:' + IntToStr(LightRep1.PageCount);  
end;
```

### 22. 5. 印刷の開始

BeginDocメソッドで印刷を開始します。

```
LightRep1.BeginDoc;
```

### 22. 6. ページの印刷

必要であればTLRItemのテキストプロパティに値を代入後、Printメソッドで印刷をおこないます。

```
LRItem2.Text := 'Hello World';  
LightRep1.Print;
```

### 22. 7. 改ページ

7.次ページがある場合は、NewPageメソッドを呼び出して改ページ後再度Printメソッドを呼び出します。

```
LightRep1.NewPage;  
LRItem2.Text := 'Hello World2';  
LightRep1.Print;
```

### 22. 8. 印刷の終了

最終ページまで印刷を終えたら、EndDocメソッドで印刷を終了します。

```
LightRep1.EndDoc;
```

## 23. 表形式のレポート

---

### 23. 1. TLightRepコンポーネントの配置

フォーム上にTLightRepコンポーネントを配置します。

### 23. 2. レポートヘッダーの作成

TLightRepコンポーネント上にTLRLayoutPanelコンポーネントを配置し、LRAlignをlaTopに設定します。これがレポートヘッダーになります。このヘッダー上に適当にTLRItemを配置し、必要に応じてTextプロパティに文字列を設定し、レポートヘッダーを作成します。

### 23. 3. レポートフッターの作成

TLightRepコンポーネント上にTLRLayoutPanelコンポーネントを配置し、LRAlignをlaBottomに設定します。これがレポートフッターになります。作成したフッター上に適当にTLRItemを配置し、必要に応じてTextプロパティに文字列を設定し、レポートフッターを作成します。

### 23. 4. 明細用のTLRGrigPanelコンポーネントの配置

TLRGrigPanelコンポーネントを配置しLRAlignプロパティをlaClientに設定します。RowCountプロパティを明細の行数に設定することで、指定した行数にパネルが分割されます。段組みの帳票を作成したい場合はColCountプロパティに列数を入力します。

### 23. 5. 明細行へのTLRItemコンポーネントの配置

分割されたTLRGrigPanel上に適当な数のTLRItemコンポーネントを配置し、各プロパティを適当に設定します。

### 23. 6. OnPrintPanelイベントハンドラの作成

TLRGrigPanelコンポーネントのOnPrintPanelイベントハンドラを作成します。OnPrintPanelイベントは分割されたパネルを印刷する度に発生し、ACol, ARowパラメータは現在何番目の列 / 行を印刷しているかを示しています。ここに分割された各パネルのデータを設定するコードを記述します。以下はTTableコンポーネントのデータを基に印刷する場合の例です。

```
procedure TDBPrintForm.LRGridPanel1PrintPanel(Sender: TObject; ACol,
    ARow: Integer; Rect: TRect; var Status: TPrintStatus);
begin // 最後まで出力していたら罫線のみ印刷
  if Table1.EOF then
    Status := psLineOnly
  else
    with Table1 do
      begin
        // データをセット
        ItemNo.Text := IntToStr(RecNo);
        ItemCustNo.Text := CustNo.AsString;
        ItemCompany.Text := Company.AsString;
        ItemAddr.Text := City.AsString + Addr1.AsString;
        ItemZip.Text := Zip.AsString;
        ItemPhon.Text := Phone.AsString;
        ItemState.Text := State.AsString;
        ItemLastInvoice.Text := LastInvoiceDate.AsString;
        Table1.Next;
      end;
    end;
```

上の例では各パネルを印刷するごとにTTableのNextメソッドを呼び出してレコード

を進めています。また、最終レコードまで達していた場合は罫線のみ印刷をおこなうようにしています。あとは「単票形式のレポート」の例と同様に BeginDoc, Print, NewPage, EndDocの4つのメソッドで印刷をおこなうわけですが、TTableのようなデータセットから印刷する場合、最終的に何ページ印刷をおこなうか実行時にしかわからないという場合は若干の工夫が必要です。以下はその例です。

```
procedure TDBPrintForm.MnuPrintClick(Sender: TObject);
begin
  // テーブルを開く
  Table1.Open;
  // データが無くなるまで印刷を行う
  with LightRep1 do
  begin
    BeginDoc;
    while not Table1.EOF do
    begin
      Print;
      NewPage;
    end;
    EndDoc;
  end;
end;
```

上の例ではテーブルの末尾に達するまでPrintメソッドを呼び出しています。他にも、(レコードの件数・€RowCount)の切り上げで印刷ページ数を計算しその回数だけfor ~ Nextループで印刷をおこなう方法もあります。レコードの件数がすぐ取得できる場合はfor ~ Nextループで印刷をおこなう方法の方がコーディング的にはわかりやすい場合が多く、途中から印刷する機能等も比較的実装が簡単です。

## 24. サンプルプログラムの紹介

---

LightReport2を使用したレポートのサンプルプログラムの説明です。それぞれのサンプルには、印刷・プレビュー・サムネイル（縮小プレビュー）及びプリンタの切り替えの機能が備わっています。

### 24. 1. 機能デモ（ 1 ） Samp1.dpr

機能デモ 1 は、TLRItemの機能を紹介するためのデモです。TLRItemは、Alignment, Layout, HMargin, VMargin等のプロパティを使用することにより、枠内のテキストの位置を自由に設定できるため、これらの機能を使用することにより、簡単に見栄えの良いレポートを作成することができます。

### 24. 2. 機能デモ（ 2 ） Samp2.dpr

機能デモ 2 は、TLRLayoutPanel,TLRGridPanelの機能を紹介するためのデモです。

### 24. 3. 機能デモ（ 3 ） Samp3.dpr

機能デモ 3 は、ネストしたパネルの機能を紹介するためのデモです。TLRLayoutPanelを 3 枚配置して、レポートの基本レイアウトを作成し、その上に複数のTLRLayoutPanelとTLRGridPanelを配置した複雑なレイアウトのレポートです。

### 24. 4. 機能デモ（ 4 ） NewSamp1.dpr

機能デモ 4 は、バージョン2.2からの新機能(LRVariableText,LRPostBarItem)の機能を紹介するためのデモです。

### 24. 5. テーブルからの印刷その 1 DBPrint.dpr

Customer.dbの一覧を印刷するサンプルです。TDataSetから印刷する際のPrintPanelイベントの使用方法及び、レポートの制御方法を説明しています。

### 24. 6. テーブルからの印刷その 2 DBPicture.dpr

Biolife.dbの一覧を印刷するサンプルです。イメージ印刷の方法及びTLRGridPanelを使用した段組み印刷の方法について説明したサンプルです。

### 24. 7. TLRSubPanelの機能デモ SubPanel.dpr

TLRSubPanelを使用して、TLRGridPanelの中に見出し行・説明行・明細行を切り替えて印刷を行うサンプルです。

### 24. 8. 定義外の用紙で印刷を行うデモ SampHagaki.dpr

用紙サイズをpsNoneに設定して、TLightRepで定義されていないサイズの用紙に印刷を行うサンプルプログラムです。

### 24. 9. LRRountRectコンポーネント LRRountRect.pas

TLRCustomItemを継承して作成されたコンポーネントのサンプル。

## 25. 開発者ガイド

---

### 25. 1. 印刷処理のメカニズム

LightReport2の印刷処理は以下の手順で行われます。

#### 25. 1. 1. PrintItemメソッドの呼び出し

TLightRepのPrintもしくはPreviewメソッドによってPrintItemメソッドが呼び出されます。

#### 25. 1. 2. 子コントロールのメソッドの呼び出し

PrintItemメソッドは自分の子コントロールをすべて検索し、TCustomLRPanelから継承されているコントロールがあればそのPrintメソッドを呼び出します。この時引数として印刷先のTCanvas、印刷領域、印刷先のスケールを渡します。

#### 25. 1. 3. 再帰的な子コントロールのメソッド呼び出し

TCustomLRPanelのPrintメソッドは、同様に自分の子コントロールをすべて検索し印刷可能なコントロールのPrintメソッドを呼び出します。最終的に呼び出されたTCustomLRItemから継承されたコンポーネントが実際の描画処理を行います。

#### 25. 1. 4. 実際の印刷処理

LightReport2の実際の印刷処理はすべてTCustomLRItemを継承したコンポーネントから行われています。つまり、TCustomLRItemコンポーネントを継承し、Printメソッドをオーバーライドするだけで、独自の印刷コンポーネント（例えば、縦書きテキストや図形描画のコンポーネント）を作成することができます。

### 25. 2. 新しい印刷コンポーネントの作成

TLRItemコンポーネント、TLRImageコンポーネントはどちらもTCustomLRItemコンポーネントを継承して作成されています。TCustomLRItemコンポーネントを継承することで、独自の機能を持った印刷アイテムを簡単に作成することができます。以下はその手順です。

#### 25. 2. 1. TLRCustomItemの継承

TCustomLRItemを継承したコンポーネントを作成します。

#### 25. 2. 2. Printメソッドの作成

Printメソッドをオーバーライドします。Printメソッドは、いくつかのパラメータを持ち、アイテムはそのパラメータにしたがって印刷処理を行う必要があります。

```
procedure Print(ACanvas: TCanvas; ARect: TRect;  
    PrintScale: Double; APrintStatus: TPrintStatus); virtual;
```

#### パラメータの種類と説明

ACanvas: 印刷先のTCanvasをあらわす。コンポーネントはACanvasに対して描写を行う。

ARect: 印刷先の領域をあらわす。コンポーネントはARectで指定される領域に描写を行う。

APrintScale: スクリーンとプリンタの解像度の差をあらわす。

APrintStatus: psNormal, psLineOnly, psEndNowの3種類の値を持つ。

psLineOnlyが指定されたときは、コンポーネントは罫線以外は印刷できない。

psEndNowが指定されたときは、何も印刷できない。

例えば、クライアント領域上に角が丸い四角形を描くコンポーネント（TLRRoundRect）コンポーネントを作成する場合、Printメソッドは以下ようになります。

```

procedure TLRRoundRect.Print(ACanvas: TCanvas; Rect: TRect;
                             APrintScale: Double; APrintStatus: TPrintStatus);
var
  DstRect: TRect;
  RoundSize: integer;
begin
  // フラグがpsNormal以外の場合は何もしない
  if APrintStatus <> psNormal then Exit;

  // サイズを印刷先のスケールに変換
  DstRect := PrintScaleRect(Rect, APrintScale);
  RoundSize := ToPrintScale(10, APrintScale);

  ACanvas.Brush.Style := bsClear;
  // クライアント領域いっぱいに描画処理を行う
  ACanvas.RoundRect(DstRect.Left, DstRect.Top, DstRect.Right-1,
                    DstRect.Bottom-1, RoundSize,
RoundSize);
end;

```

### 25. 2. 3. Paintメソッドの作成

設計時の描画を行うために、Paintメソッドをオーバーライドします。印刷時と同じ描画を行う場合は、Paintメソッド内でPrintメソッドを呼び出します。設計時には別の描画を行う（例えば、コンポーネント名を表示する場合は、ここで記述します。下の例では、設計時にも印刷時と同じ描画を行っています。

```

procedure TLRRoundRect.Paint;
begin
  Print(Canvas, GetClientRect, 1, psNormal);
end;

```

TLRControlItemを継承したコンポーネントの作成についてより詳しく知りたい方は、サンプルプログラム「TLRRoundRect.pas」やSourceフォルダ内にあるLightRep.pasを参考にしてください。

## 26. 最後に

---

### 26. 1. 関連コンポーネントの紹介

#### 26. 1. 1. OBCBコンポーネント対応LRItem、LRVariableText(TAK様)

TAK様作成のLRItemとLRVariableTextのODBCコンポーネント対応版がODBCコンポーネントに同梱して公開されています。プロパティエディタなんかも備えられていて、さすが!!の出来栄です。

#### 26. 1. 2. LightReport2 補充コンポーネント(Ohtaka様)

Ohtaka様作成のデータベース対応のパネアイテムと追加のコンポーネントです。近日公開予定とのことです。特に角が丸い枠線が書けるLRItemRは、4つの角それぞれを丸くするかどうか指定が出来、また塗りつぶしもきれいにできるので重宝しそうです。

上記2つ以外にも作成された方がいれば是非教えてください。

### 26. 2. ライセンスその他について

LightReport2はLGPLに基づいて公開しています。つまり、プログラムの内部で使用する場合に限り、商用・非商用および無償・有償を問わず自由に使用可能ですが、このコンポーネントを改造したコンポーネントを公開する場合は、無償かつソース公開である必要があります。また、転載についても特に制限はありませんが、できたら事後でかまいませんのでメール等でご一報いただけると嬉しいです。

### 26. 3. 謝辞

このコンポーネントの作成にあたっては、FDelphi会議室の発言を始め、Delphian Worldで公開されているコンポーネント等、いろいろな方のいろいろな情報を参考にさせていただきました。関係者の皆様にお礼を申し上げます。また、TGridPanelのヒントを下さった米田 昌司様、どうもありがとうございました。ベータ版のテストに協力していただいた方々、ご意見・ご要望をメールで下さった方々にもお礼を申し上げます。

ご意見・ご要望等はtakeshi\_kanno@est.hi-ho.ne.jpまでメールしてください。