

尾崎 浩司

株式会社ミガロ.

システム事業部 システム3課

# Delphi/400: 帳票開発ノウハウ公開

業務に必須の帳票開発。Delphi/400 で使用できる帳票ツールや PDF ファイル出力連携など、帳票開発手法の具体例を紹介する。

- はじめに
- 帳票レイアウト上での「計算項目」の活用
- 画像ファイルの取り扱い
- 出力先プリンタの指定
- XMLファイルをソースとした帳票出力
- PDFファイルの作成
- 最後に



略歴

1973年8月16日生まれ  
1996年三重大学工学部卒  
1999年10月株式会社ミガロ.入社  
1999年10月システム事業部配属

現在の仕事内容  
ミガロ.入社以来、主にDelphi/400  
を利用した受託開発を担当している。

## 1. はじめに

“帳票”を出力するには、どのような方法があるだろうか？

IBM i ユーザーであれば、「OUTQ」(スプール)を使用した帳票が一般的であろう。また、以前からの Delphi/400 ユーザーであれば、「QuickReport」を使用した帳票出力を行っている場合もあるだろう。では、現在の Delphi/400 にバンドルされた標準レポートツールは何であろうか？それは「Rave Reports」である。

本稿では、この Rave Reports を中心に、帳票開発に関するノウハウをトピックス形式で紹介したいと思う。

なお、Rave Reports の基本的な使用方法については、ミガロ.のホームページに詳しく紹介しているのでそちらを参照していただきたい。

Rave Reports 講座

<http://www.migaro.co.jp/contents/>

[products/delphi400/tips/report/3\\_01/index.html](http://products/delphi400/tips/report/3_01/index.html)

## 2. 帳票レイアウト上での「計算項目」の活用

Rave Reports の特徴は、QuickReport 同様、Delphi/400 の DataSet を直接使用できることである。

DataSet には、データベースのフィールド情報を保持しているため、各フィールドをそのまま帳票出力項目として、レイアウト設計に使用できる。さらに、レポート側に「計算項目」を定義することで、実体のフィールドと同様に、レイアウト設計で使用可能になる。例えば、データベース上に「数量」と「単価」しかフィールドを持たない場合でも、「金額」フィールドを「数量×単価」という計算項目として定義できる。

ここでは、この帳票レイアウトにおける計算項目を取り上げ、いかに活用できるかを見ていこう。

最初は、作成手順である。Rave Repo

rts の帳票レイアウト設計は、専用の「レポートデザイナー (Rave Reports Designer)」を使用して行う。【図1】

まず、作成するレポートにある Parameters プロパティに、新規パラメータ Amount を定義しよう。この Amount が帳票で出力する計算項目となる。

次に、計算項目に使用する計算式を定義する。ツールパレットから、CalcOp コンポーネントをレポートに貼り付けよう。【図2-①】

CalcOp コンポーネントには計算式が指定できるので、下記のように定義する。【図2-②】

- ・ DestParam : 出力用計算項目  
(ここでは「Amount」を指定)
- ・ Operator : 算術演算子を指定  
(ここでは「乗算」を指定)
- ・ Src1DataField、Src2DataField :  
演算を行うフィールドを指定  
(ここでは「数量」「単価」の各フィールドを指定)

これで定義は完了だ。あとは、出力用フィールドとして DataText コンポーネントを貼り付け、Amount フィールドを指定すれば、帳票出力時に「数量×単価」から「金額」が計算され、結果が帳票出力されるというわけである。【図3】

このように、あらかじめ帳票レイアウトに計算項目を使用し、計算式を埋め込むことで、有効な帳票が実現する。その際、帳票出力プログラムは、Rave ReportsにDataSetを渡すだけでよいのである。

### 3. 画像ファイルの取り扱い

画像ファイルの帳票出力は、IBM i の OUTQ からは難しいと思われるが、Rave Reports を使用すると容易に実現できる。ここでは、帳票出力において画像をセットする手順を紹介しよう。

Rave Reports での画像出力には、Bitmap コンポーネントを使用する。このコンポーネントには、画像ファイルを指定する FileLink プロパティが用意されているので、ここに出力する Bitmap ファイルを指定すればよい。

では、プログラムから動的にファイルを指定するにはどうすればよいだろうか？

Rave Reports で作成するレポートは、Delphi と同様にイベントを使用することができる。つまり、画像ファイルが出力される直前に発生するイベント OnBeforePrint で、出力したい画像ファイルを指定すればよいのである。【図4】

ここでは、PictureFileStr というパラメータを使用しているので、レポートの Parameters プロパティに PictureFileStr というパラメータを追加しておく。これで Delphi ソースから値を指定することが可能になる。

Delphi ソースの記述例を示す。【ソース1】

これは、Edit1 コンポーネントに指定された Bitmap ファイル名を、レポートのパラメータ PictureFileStr に設定し、印刷する手順である。本プログラムを実行すると、画面に示すように選択したファイルが帳票出力される。【図5】

この仕組みを応用することにより、画像を埋め込んだ帳票が容易に作成できる

ので、グラフィカルで利用価値の高い帳票開発にぜひ挑戦していただきたい。

### 4. 出力先プリンタの指定

Rave Reports で作成した帳票をそのまま印刷しようとする、デフォルトで「印刷指示」画面が表示される。この画面では、出力方法や印刷枚数、出力先プリンタ等の指定が行えるため大変便利であるが、残念ながらすべて英語表記となっている。【図6】

では、この印刷指示面を使用せずに帳票出力するにはどうすればよいだろうか？

#### 帳票出力プログラム

帳票出力に関する設定情報は、TRvS system コンポーネントで管理できる。つまり、このコンポーネントを使用することで、印刷指示画面の設定自体をプログラムで行うことが可能になる。

設定箇所は、次の通りである。【図7】

- DefaultDest : 出力先の指定 (プレビュー / 印刷)
- SystemSetup/ssAllowSetup : 「印刷指示」画面の表示有無

これで印刷指示画面を経由せずに、帳票出力が可能になる。

ソース例を示す。【ソース2】

このソースは、例えば、Delphi のフォーム上に TragioGroup (コンポーネント名: rgSelect として定義) を配置し、選択値に応じて「プレビュー」あるいは「印刷」を切り替える場合の記述である。

#### 出力先プリンタの指定

これで、印刷指示画面を経由せずに、Rave Reports が使用可能となるわけだが、このままでは印刷時には、常に「通常使うプリンタ」へ出力されてしまう。では「出力先プリンタ」を切り替えるにはどうすればよいだろうか？

「出力先プリンタ」を切り替えるロジックを記述するには、デバイス定義ユニットである RPDevice を Uses 節に追加し、RPDev オブジェクトを使用すればよい。

RPDev オブジェクトには、SelectPrinter メソッドが用意されているので、ここに出力したいプリンタ名を指定すればよいわけである。

ソース例を示す。【ソース3】

Windows 標準の「印刷用ダイアログ」(TPrintDialog) を表示し、選択されたプリンタに帳票を出力するソース例である。

### 5. XML ファイルをソースとした帳票出力

データ連携といえば、以前は CSV ファイルを使用するのが一般的であったと思う。現在でも利用する機会は多いと思うが、近年は、より柔軟な定義が可能な XML ファイルを使用することも多いだろう。【図8】

ここでは、XML ファイルを使用して帳票出力する手法を紹介したい。

XML を Delphi で扱う方法はいくつかあるが、今回は TClientDataSet を使用する方法を紹介しよう。

TClientDataSet はこれまで何度か紹介しているが、クライアント PC のメモリ上に展開可能な DataSet である。実は、TClientDataSet には、XML を読み込んで DataSet として利用する方法がある。DataSet となれば、Rave Reports でそのまま使用可能になるというわけである。

では、どのようにすれば XML ファイルを DataSet として使用できるのだろうか？

#### 変換ファイル(拡張子.xtr)

これを実現するには、TXMLTransformProvider というコンポーネントを使用すればよい。このコンポーネントは、XML 文書から DataSet へ変換するためのものである。その変換には、専用の変換ファイル(拡張子.xtr)を使用する。

実は Delphi には、XML ファイルから変換ファイルを作成するためのマッピングツールが付属している。Delphi 開発画面上で[ツール]→[XML マッパー]を選択すると、画面のようなツールが起動する。【図9】

このツールより、対象となる XML ファイルを開くと、ドキュメントの構成が左側のペインに表示されるので、

図1

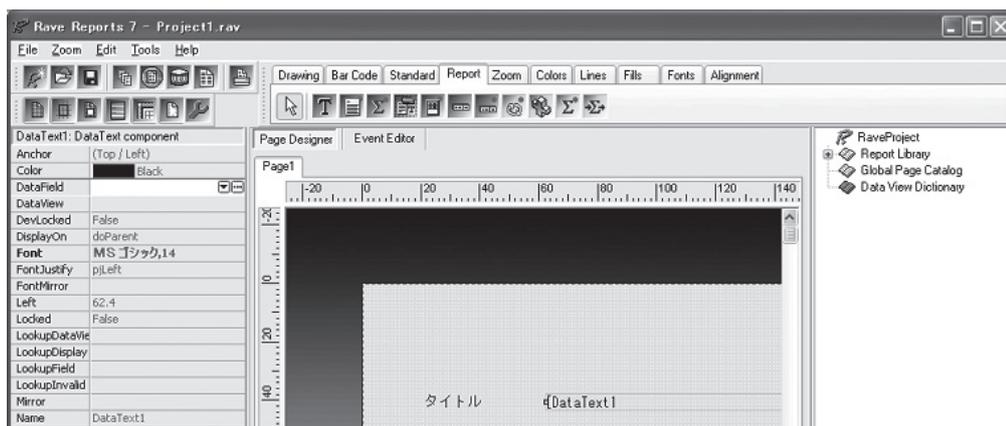


図2

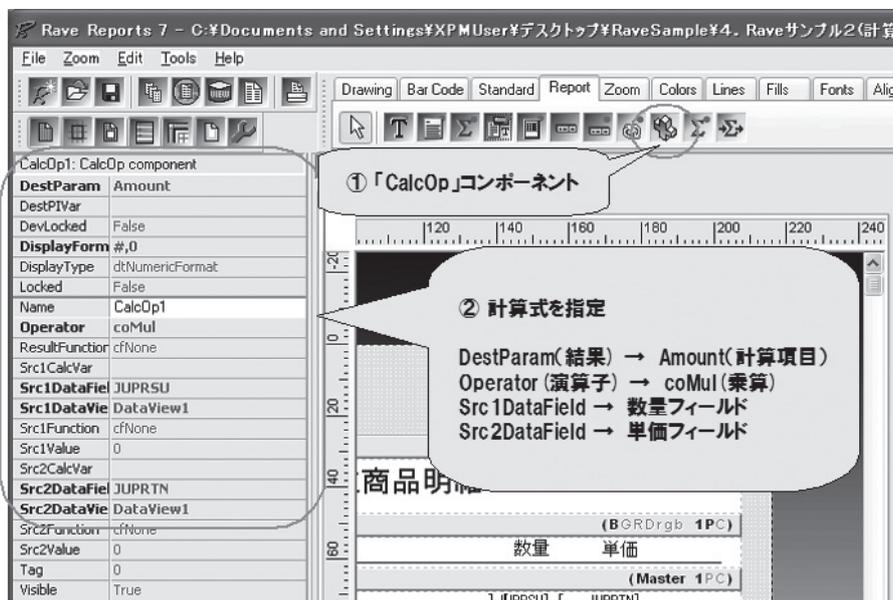
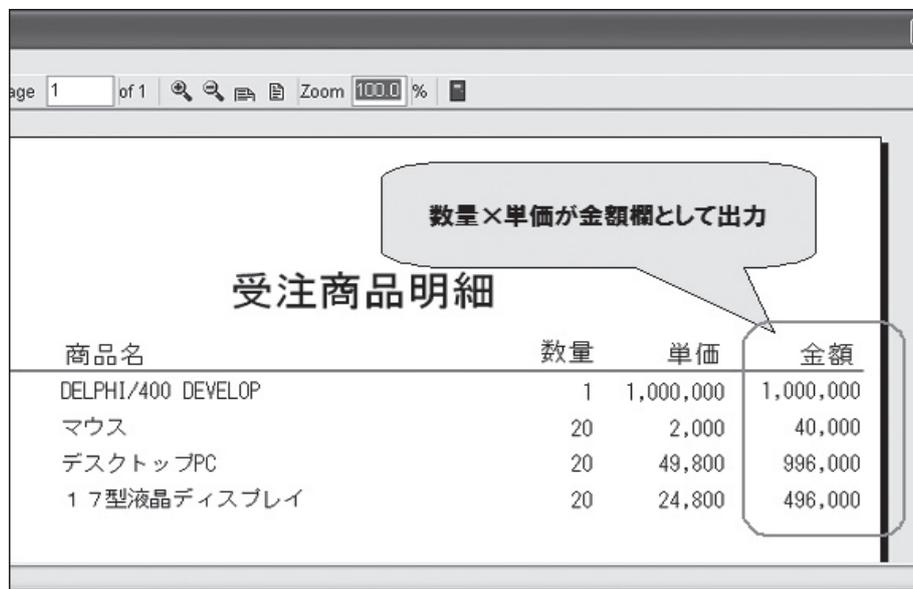


図3



DataSet に変換したい項目をダブルクリックしていく。すべての項目を選択したら、[作成] → [XML からデータパケット] を選択しよう。すると、XML の各項目がデータパケット欄に表示されるはずだ。これでマッピングが完了である。

最後に「変換の作成とテスト」をクリックすると、図に示すように、データセットが表示されるのが分かる。(画面は、XML ファイルを、実際に DataSet に変換した結果である)。【図 10】

変換が完了したら、「保存」より変換ファイル (拡張子 .xtr) の保存を行う。これで変換ファイルの作成は終了である。

あとは、TXMLTransformProvider の XMLDataFile プロパティに処理対象となる XML ファイルを指定し、TransformRead プロパティにある TransformationFile に変換ファイルを指定の上、TClientDataSet の ProviderName プロパティに対象の TXMLTransformProvider を指定すれば完成である。

このように、XML ファイルは簡単に DataSet に変換できるため、Rave Reports において、XML ファイルを使用した帳票出力が容易であるということもご理解いただけるだろう。

## 6. PDFファイルの作成

Rave Reports には、「PDF ファイル」出力機能が用意されている。しかし残念ながら、日本語環境化での動作はサポートされていないのが実情だ。では、Delphi/400 から PDF ファイルを作成するには、どうすればよいだろうか？

最も簡単な方法は、PDF プリントドライバを使用する方法であろう。本家の Adobe Acrobat のみならず、近年は PDF 出力可能なフリーソフトも多数リリースされているので、これらを出力先プリンタに指定すれば手軽に PDF 出力が可能である (※)。

※前述の「4. 出力先プリンタの指定」で解説した技法を使用すれば、PDF プリンタの出力先指定も行える。

ただし、この手法の欠点は、クライアント PC 上に PDF プリントドライバのインストールが必須となることである。企業内の GUI アプリケーション等であ

れば統一環境の構築も検討できるであろうが、そうでない場合、クライアント PC の環境を固定化するのは困難であろう。そういった場合は別の手法を検討する必要がある。

そこで今回ご紹介する方法は、Delphi で使用可能な、PDF 出力コンポーネントである「PowerPDF」を使用する方法だ。PowerPDF は、LGPL ライブラリとしてソースコード付で公開されている。

### PowerPDF0.91 for Delphi 2009

(Delphi/400 Ver2009 以降)

[http://mam.dnsalias.net/download/delphi\\_powerpdf.html](http://mam.dnsalias.net/download/delphi_powerpdf.html)

### PowerPDF0.9

(Delphi/400 Ver2007 以前)

[http://www.est.hi-ho.ne.jp/takeshi\\_kanno/powerpdf/](http://www.est.hi-ho.ne.jp/takeshi_kanno/powerpdf/)

PowerPDF を使用すると、クライアント PC のプリンタドライバ導入状況にかかわらず、PDF ファイルの出力が可能になる。(作成された PDF ファイルの確認には、専用の閲覧ソフトが必要であることはいうまでもないだろう)。

本ツールは、QuickReport のように、Delphi フォーム上に帳票レイアウト設計が可能なコンポーネント群となっている。【図 11】

この PowerPDF の基本的な使用方法は、次の通りである。【図 12】

①プロジェクトに帳票用フォームを新規作成する。

②下記コンポーネントを貼り付ける。

- ・ TPreport : PDF 出力を管理するコンポーネント
- ・ TPRPage : PDF ファイルの 1 ページを表わすコンポーネント
- ・ TPRLayoutPanel : 単票形式のパネルコンポーネント

③TPRLayoutPanel 内に出力用コンポーネントを貼り付ける。

- ・ TPRLabel : タイトル (表題) を表わすコンポーネント
- ・ TPRTText : 複数行表示可能なコンポーネント

④ソース 4 のようなプログラムを記述する。【ソース 4】

- ・ FileName : 出力先 PDF ファイル名を指定
- ・ BeginDoc : PDF 作成の開始
- ・ Print : 指定したページの PDF を出力
- ・ EndDoc : PDF 作成の終了

なお、PDF1 ページを表わす TPRPage は、コンポーネントサイズが大きいため、TScrollBox に置いて使用すると便利であろう。

このように、PowerPDF を使用すれば帳票の PDF 化も容易である。ただし、Rave Reports とは互換性はないため、従来のプリンタ出力と PDF 出力と両方が必要となる場合は、2 種類の作り込みが必要となることにご注意いただきたい。

## 7. 最後に

今回は、Delphi/400 から作成する帳票として、Rave Reports および PowerPDF という 2 つの帳票ツールのトピックスを紹介した。

特に、これまで IBM i の OUTQ (スプール) を使用していた方は、画面の GUI 化同様、帳票の表現力が向上するので、ぜひ一度これらのツールを試してみたい。Delphi/400 からの帳票出力が、容易でかつ効果的であることを実感いただけるであろう。

また、ミガロ . ホームページには帳票に関するトピックスも多数掲載しているので、参考になれば幸いです。

M

図4

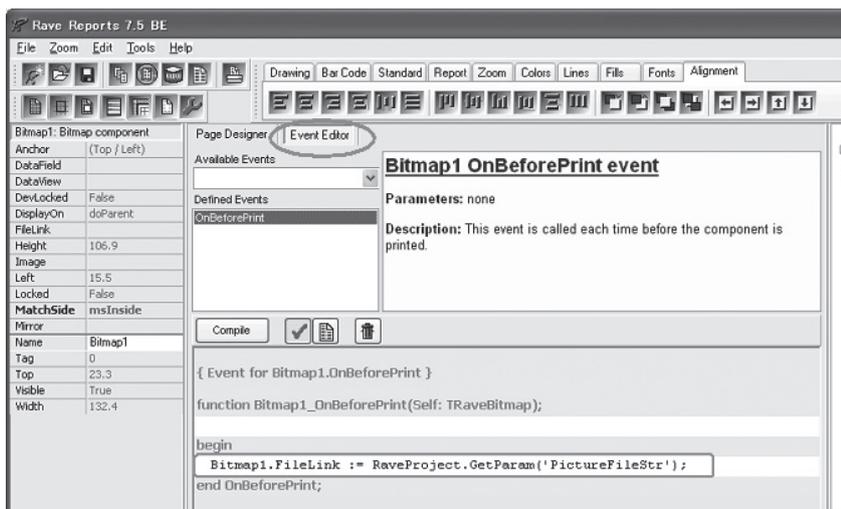


図5

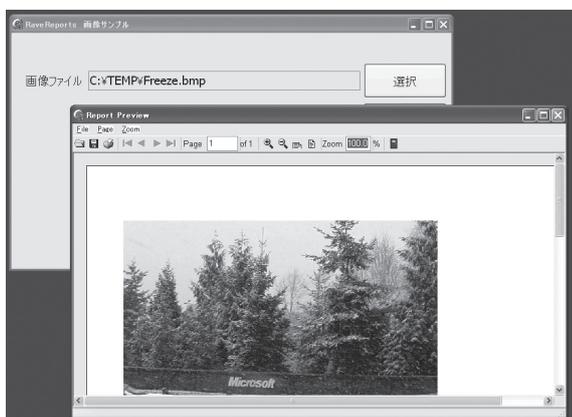


図6

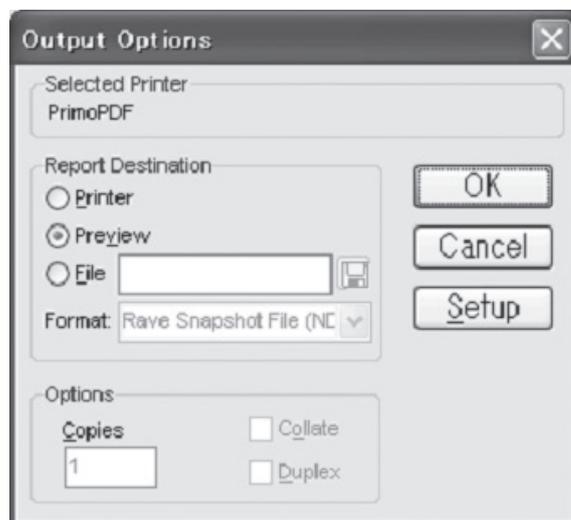


図7

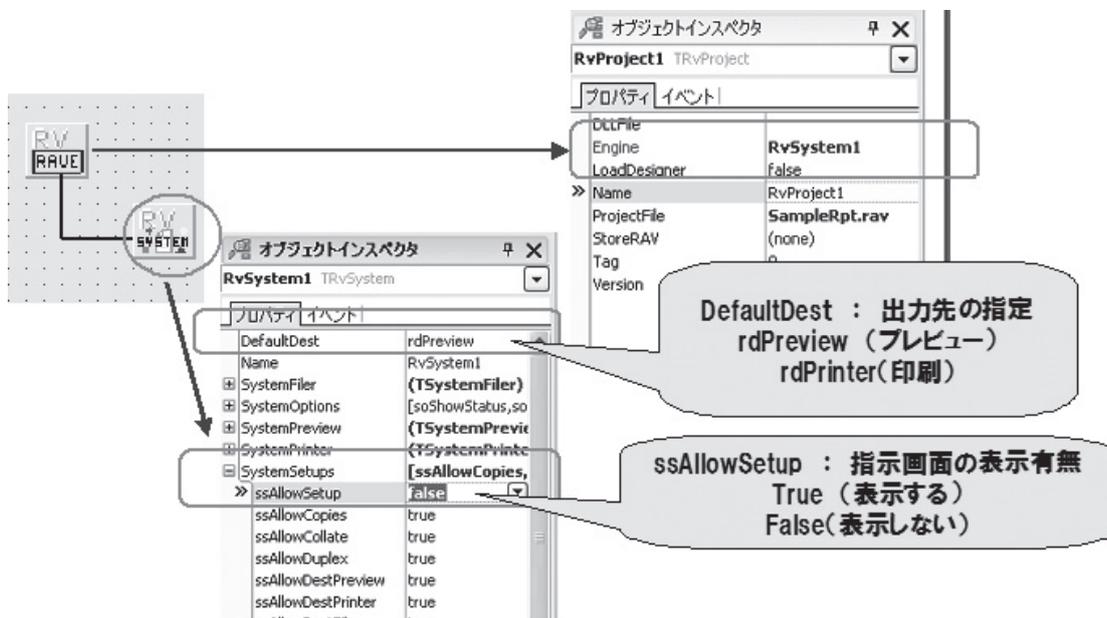


図8

```

<?xml version="1.0" encoding="ISO-2022-JP" ?>
- <住所録>
- <個人情報 ID="1">
  <名前>升村 丞</名前>
  <郵便番号>920-0967</郵便番号>
  <住所>金沢市菊川1-9-6</住所>
  <電話番号>076-261-4921</電話番号>
</個人情報>
- <個人情報 ID="2">
  <名前>北陸 太郎</名前>
  <郵便番号>920-0000</郵便番号>
  <住所>金沢市大手町1-1-1</住所>
  <電話番号>076-111-3333</電話番号>
</個人情報>
- <個人情報 ID="3">
  <名前>金沢 花子</名前>
  
```

図9

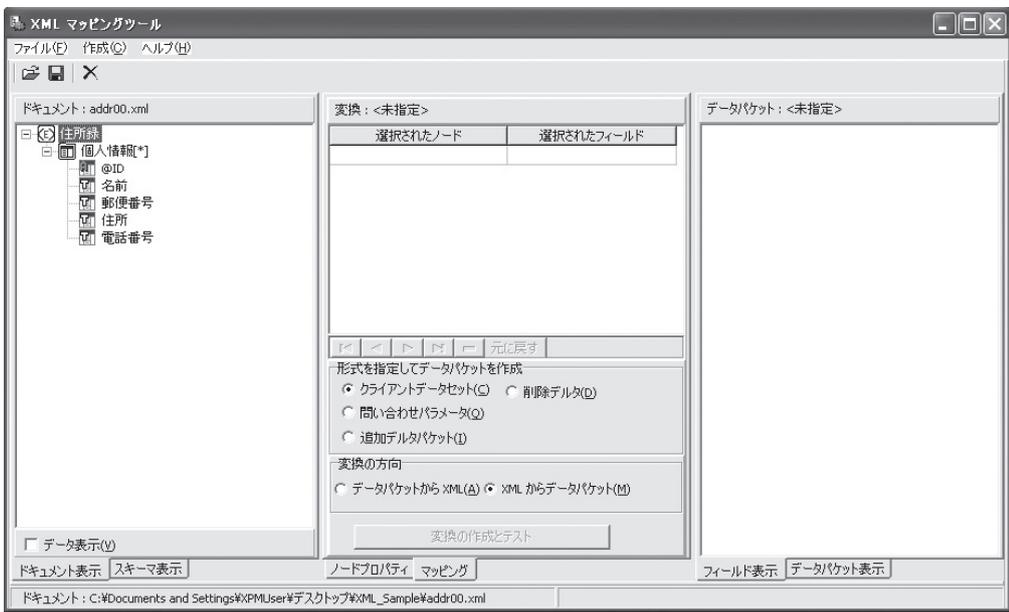


図10

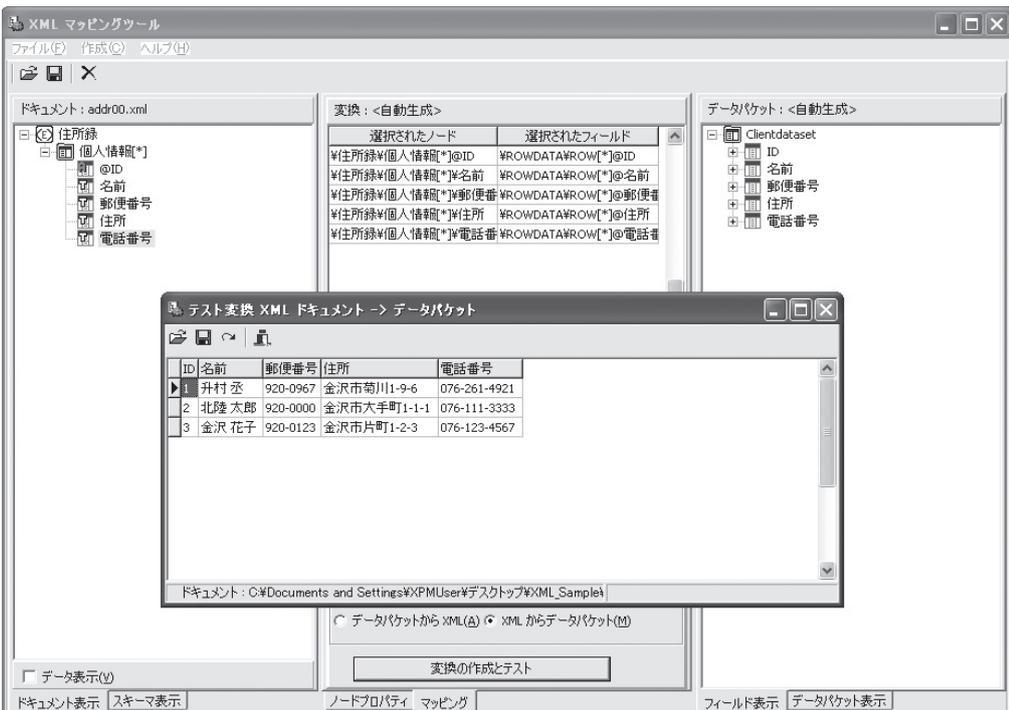


図11

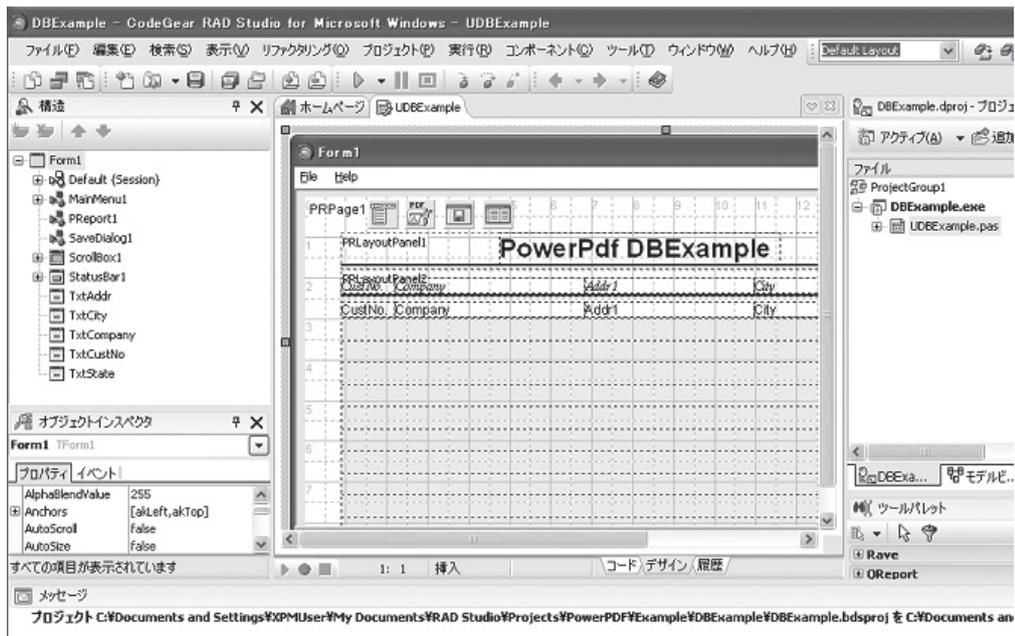
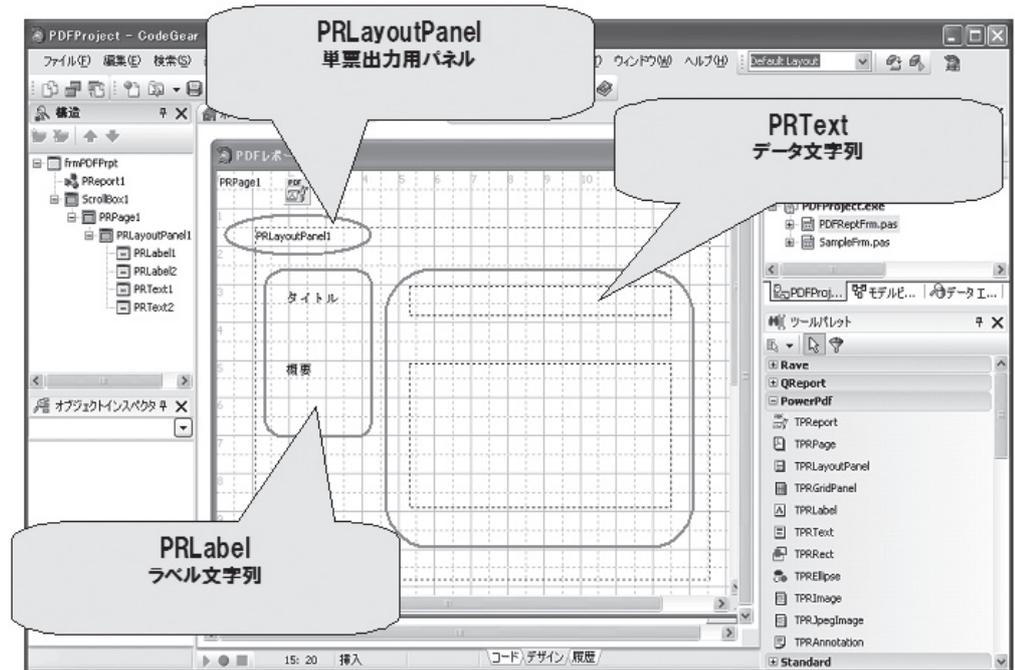


図12



## ソース1

```
ホームページ SampleFrm
. procedure TfrmSample.Button1Click(Sender: TObject);
. begin
.   with RvProject1 do
40   begin
.     //レポート選択
.     SelectReport('Report1', True);
.     // 画像ファイル名をパラメータにセット
.     SetParam('PictureFileStr', Edit1.Text);
.     // 印刷実行
.     Execute;
.   end;
. end;
```

## ソース2

```
ホームページ SampleFrm
. procedure TfrmSample.Button1Click(Sender: TObject);
. begin
.   //帳票プレビュー/印刷 選択
.   if rgSelect.ItemIndex = 0 then
.     RvSystem1.DefaultDest := rdPreview // プレビュー
40   else
.     RvSystem1.DefaultDest := rdPrinter; // 印刷
.
.   with RvProject1 do
.     begin
```

## ソース3

```
ホームページ SampleFrm
. implementation
30 . uses RPDevice, Printers;
.   [$R *.dfm]
. procedure TfrmSample.Button1Click(Sender: TObject);
.   begin
.     //印刷ダイアログを表示する
.     if not PrintDialog1.Execute then Exit;
40   . // 選択されたプリンタ名をセットする
.     RPDev.SelectPrinter(Printer.Printers[Printer.PrinterIndex], False);
.     // 印刷を行う
.     with RvProject1 do
.       begin
```

## ソース4

```
ホームページ SampleFrm
. procedure TfrmSample.Button1Click(Sender: TObject);
. begin
38 . // 保存ダイアログを表示
.   if not SaveDialog1.Execute then Exit;
40 .
.   frmPDFPrpt := TfrmPDFPrpt.Create(Self);
.   try
.     // レポート作成
.     with frmPDFPrpt do
.       begin
.         // データをセット
.         PRTxt1.Text := Edit1.Text;
.         PRTxt2.Text := Memo1.Text;
50 . // PDF出力
.         PReport1.FileName:=SaveDialog1.FileName;
.         PReport1.BeginDoc;
.         PReport1.Print(PRPage1);
.         PReport1.EndDoc;
.       end;
.     finally
.       frmPDFPrpt.Release;
.     end;
. end;
```