

前坂 誠二

株式会社ミガロ.

システム事業部

[Delphi/400] FastReportを使用した帳票作成テクニック —FastReport応用

- はじめに
- レポートウィザードを使用したデザイン作成
- グラフを用いた帳票の作成
- 画像ファイルを用いた帳票の作成
- 帳票作成プログラムの応用テクニック
- 最後に



略歴
1989年3月21日生
2011年3月 関西大学 文学部卒
2011年4月 株式会社ミガロ. 入社
2011年4月 システム事業部配属

現在の仕事内容
Delphi/400 を利用したシステム開発や保守作業を担当。Delphi、Delphi/400 の開発経験を積みながら、日々スキルを磨いている。

1.はじめに

FastReport は、Delphi/400 Version XE3 で新たにバンドルされた帳票作成ツールである。FastReport を用いると、誰でも容易に帳票出力プログラムを作成できる。基本的な帳票作成の方法については『Migaro.Technical Report 2013』の「FastReport を使用した帳票作成入門」にわかりやすく解説されているので、ぜひ参考にさせていただきたい。本稿は、FastReport を使用し、グラフや画像ファイルを用いた帳票出力の方法や FastReport の実践的なテクニックについて紹介する。

本稿ではまず、第2章でレポートウィザードを使用したデータベース帳票の作成方法を紹介する。次に、第3章・第4章では、第2章で作成した帳票デザインを基に、グラフや画像ファイルを挿入する方法を紹介する。第5章では、FastReport で帳票出力プログラムを作成する際に活用できる帳票作成プログラムの実践的なテクニックを紹介する。

なお、本稿で使用しているプログラム例は Delphi/400 Version XE3 を使用し、FastReport はバンドル版を使用している。また、FastReport のバージョンは 4.12.13 である。

2.レポートウィザードを使用したデザイン作成

FastReport を使用し、データベース帳票を新規作成する場合、一般的な方法としては、デザイン画面上にバンドオブジェクトを配置し、TfrxMemoView コンポーネントなどのコンポーネントを1つずつ貼り付けて、項目設定を行っていく。もちろんこの方法でレポートデザインは作成できるが、項目ごとに1つずつコンポーネントを貼り付ける作業を手間を感じることもある。今回紹介するレポートウィザードを使用した作成方法はそういった手間を省くことができ、非常に簡単な手順でデータベース帳票の基盤が作成できる。また、用紙レイアウトや出力させたい項目も手順にそって設定で

きる。FastReport を初めて使用する方でも、悩むことなく、デザインの作成が可能である。

今回は帳票の作成例として、【図1】のような“売上一覧表”を作成する。この例では、営業所名ごとにグループ化を行い、グループごとの小計、そして最終レコードのあとに合計を表示させる。

レポート作成の準備

まず、作成の準備として、印刷フォーム画面に TfrxReport コンポーネントを貼り付ける。今回出力する帳票はデータベースから取得した値を出力するため、TfrxDBDataSet コンポーネントも併せて貼り付け、DataSet プロパティに出力したいデータセットを紐づけておく。

レポートウィザードの使用

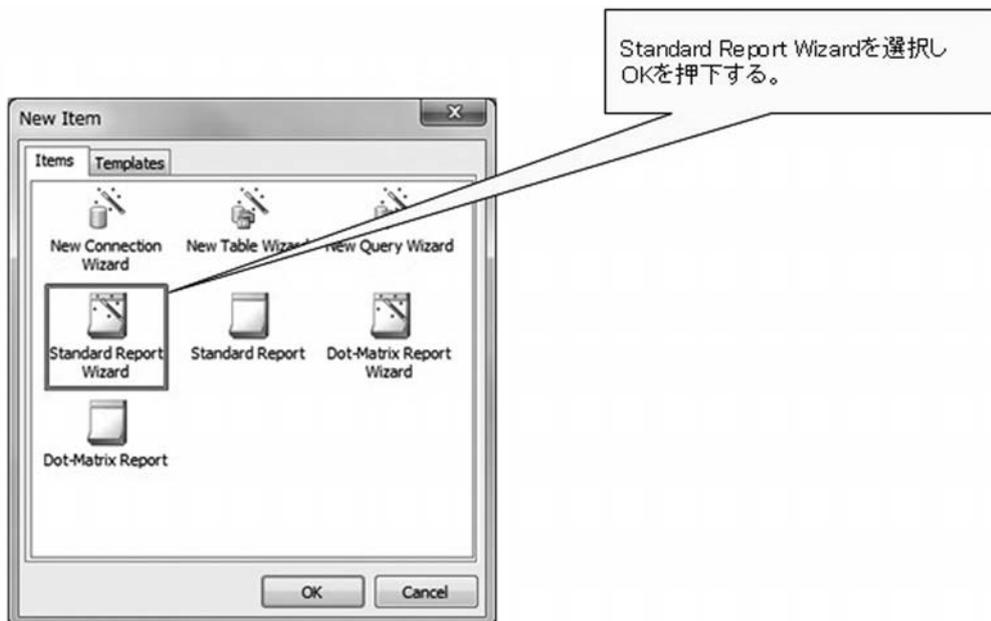
レポートウィザードを使用するには、まず、Delphi の開発画面で貼り付けた frxReport1 をダブルクリックし、レポートデザイナ画面を起動させる。レポートデザイナ画面が起動した後、「File」

図1

売上一覧表(2014年)					
名称	住所1	住所2	電話番号	FAX番号	金額
東京営業所					
株式会社足立商店	東京都足立区	1-1-2	XXX-XXXX-XXX	XXX-XXXX-XXX	1,000
株式会社足立興業	東京都足立区	1-2-3	XXX-XXXX-XXX	XXX-XXXX-XXX	2,000
株式会社荒川商店	東京都荒川区	2-2-3	XXX-XXXX-XXX	XXX-XXXX-XXX	3,000
株式会社荒川工業	東京都荒川区	1-2-1	XXX-XXXX-XXX	XXX-XXXX-XXX	5,000
株式会社板橋商店	東京都板橋区	5-2-3	XXX-XXXX-XXX	XXX-XXXX-XXX	18,000
				小計:	29,000
大阪営業所					
株式会社池田商店	大阪府池田市	1-2-3	XXX-XXXX-XXX	XXX-XXXX-XXX	1,000
株式会社泉商店	大阪府泉大津市	12-2-3	XXX-XXXX-XXX	XXX-XXXX-XXX	2,000
株式会社泉興業	大阪府泉大津市	13-2-3	XXX-XXXX-XXX	XXX-XXXX-XXX	3,000
株式会社佐野商店	大阪府泉佐野市	13-2-3	XXX-XXXX-XXX	XXX-XXXX-XXX	8,000
株式会社泉佐野商店	大阪府泉佐野市	13-2-3	XXX-XXXX-XXX	XXX-XXXX-XXX	15,000
				小計:	29,000
青森営業所					
株式会社津軽商店	青森県津軽市	1-2-2	XXX-XXXX-XXX	XXX-XXXX-XXX	100
株式会社北津軽商店	青森県北津軽市	1-3-2	XXX-XXXX-XXX	XXX-XXXX-XXX	400
				小計:	500
広島営業所					
株式会社安芸商店	広島県安芸市	2-1-2	XXX-XXXX-XXX	XXX-XXXX-XXX	900
株式会社江田島商店	広島県江田島市	1-4-2	XXX-XXXX-XXX	XXX-XXXX-XXX	800
				小計:	1,700
高知営業所					
株式会社香美商店	高知県香美市	3-1-5	XXX-XXXX-XXX	XXX-XXXX-XXX	2,910
株式会社四万十商店	高知県四万十市	1-1-5	XXX-XXXX-XXX	XXX-XXXX-XXX	10,000
				小計:	12,910
				合計:	73,110

2014/08/20 15:33:38 Page 1

図2



New」を選択する。すると、【図2】の画面が表示されるので、Standard Report Wizardを選択する。次に以下のStep1～Step5で帳票の出力設定を選択する。

Step1: 使用したいデータセットを選択する。【図3】

Step2: 使用したいデータセットの項目を選択する。【図4】

Step3: グループ化したい項目を選択する(任意)。【図5】

Step4: 用紙の向き、項目の配置を選択する。【図6】

Step5: レポートのスタイルを指定する。【図7】

Step1～Step5で出力設定を選択後、最後にFinishボタンを押下する。あとは、レポートタイトルや明細タイトルのキャプションを設定すれば、簡単に帳票デザインの基盤が完成する。【図8】

データのグループ化について

今回は、営業所名でグループ化した帳票の出力を行う。グループ化を行うには、コンポーネントパレットからInsert Bandをクリックし、Group Header Bandを選択する。【図9】

すると、ダイアログ画面が表示されるのでData field またはExpression 欄に、グループ化したい項目を設定する。【図10】

なお、本稿で、このグループ化設定は、レポートウィザード(【図5】で記載)にて行っている。

Group Header Bandを使用すると、データをグループ化するだけでなく、ドリルダウン設定も行うことができる。ドリルダウン設定を行うと、プレビュー画面でグループごとにデータの表示・非表示を切り替えることができる。【図11】
【図12】

また、このプレビュー画面で行った表示・非表示の操作は印刷時にも影響がある。つまり、ドリルダウン設定を行った状態で、印刷を実行すると、現在プレビュー画面で表示している内容がそのまま印刷されるということである。

ドリルダウン設定を行うには、レポートデザイナー画面のGroup Header Bandを選択し、DrillDownプロパティを

Trueに変更するだけで設定が可能である。また、ExpandDrillDownプロパティをTrueにすると、初期表示時に全てのグループが展開された状態で表示する。グループのデータが非表示の状態、グループの小計などGroup Footer Bandに配置した項目のみ表示させたい場合は、ShowFooterIfDrillDownプロパティをTrueにする。【図13】

合計の表示

データベースの値を使用した合計値を表示するには、TfrxMemoViewコンポーネントを使用し、テキストに“[SUM(<データセット名.フィールド名>)]”と記述する。また、計算範囲については、コンポーネントの配置場所によって決定する。例えば、ページごとの合計値を表示させたい場合は、Page Footer Bandに配置することでページごとの合計値を表示させることができる。全レコードの合計値を表示させたい場合はFooter Bandに配置することで表示させることができる。

ただし、このような計算項目をレポートデザイナー画面で使用し、さらにGroup Header Bandのドリルダウン機能も併せて利用している場合は、注意が必要である。レポートデザイナー画面の計算項目は、表示しているデータのみを計算結果に含めるので、ドリルダウン機能でデータを非表示にしている場合、意図していない計算結果になる可能性がある。そのため、ドリルダウン機能を併せて利用する場合は、Delphiソース内でTClientDataSetの内部計算項目などを使用して、計算処理を行い、その計算値をセットさせるといった工夫が必要となる。【図14】

3. グラフを用いた帳票の作成

本章では、グラフを用いたデータベース帳票の作成手法について説明する。グラフを使用すると、視覚的に出力内容が表現され、文字データばかりの帳票よりも出力内容が把握しやすくなる。FastReportでは、グラフを用いた帳票出力プログラムも通常の帳票出力同様、容易に作成することができる。では、その作成方法について紹介しよう。

新規ページの作成

本章では、例として第2章で作成した帳票の最終ページに【図15】のようなグラフを使用した帳票出力を行う。まず準備として、レポートデザイナー画面の「File | NewPage」を選択する。すると、Report TreeにPage2という新しいページが作成される。今回は、このPage2にグラフの出力を行う。【図16】

グラフの出力

グラフを出力するには、TfrxChartViewコンポーネントを使用する。レポートデザイナー画面のコンポーネントパレットからChartObjectを選択し、TfrxChartViewコンポーネントを画面に貼り付けると、Chart Editorが表示される。【図17】

TfrxChartViewコンポーネントは、Chart Editorでグラフの種類選択や詳細設定を行う。Add Seriesボタンを押すと、グラフの種類を選択するダイアログが表示されるので、ここで使用したいグラフの選択を行う。今回は、棒グラフを使用するため、BarからNormalを選択する。【図18】

グラフの種類を選択すると、Chart Editorに先ほど追加したグラフがツリー形式で表示される。今回出力するグラフは、全営業所での顧客の売上Top5を降順に表示させる。

まずは、グラフのタイトル設定を行うため、Chartを選択し、プロパティのTitleを選択する。そして、Textをダブルクリックすると入力画面が表示されるので、そこでタイトル名を設定する。【図19】

次にグラフの設定を行う。Barをクリックし、DataSetには、対象であるfrxDBDataSet1を選択する。そして、ValuesのLabelには顧客名の項目、Yには、売上金額の項目を選択する。Other Optionsでは、並び順や上位何件まで表示させるかが設定できる。今回はSort orderを、Descendingに設定し、TopN Valuesには6を設定する。【図20】

なお、TopN Valuesを6と設定すると、6件目のデータには、上位5件を除いた項目の合計値が表示され、TopN captionの値が名称として設定される(【図15】参照)。

図3

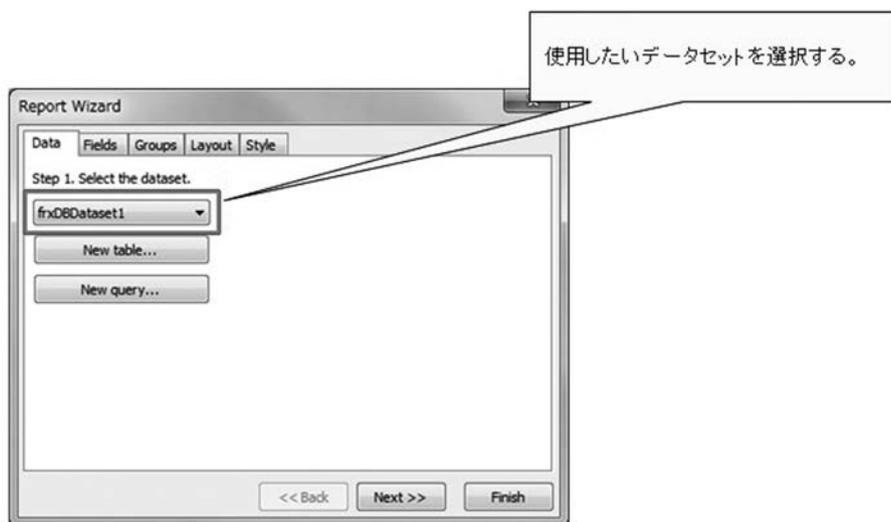


図4

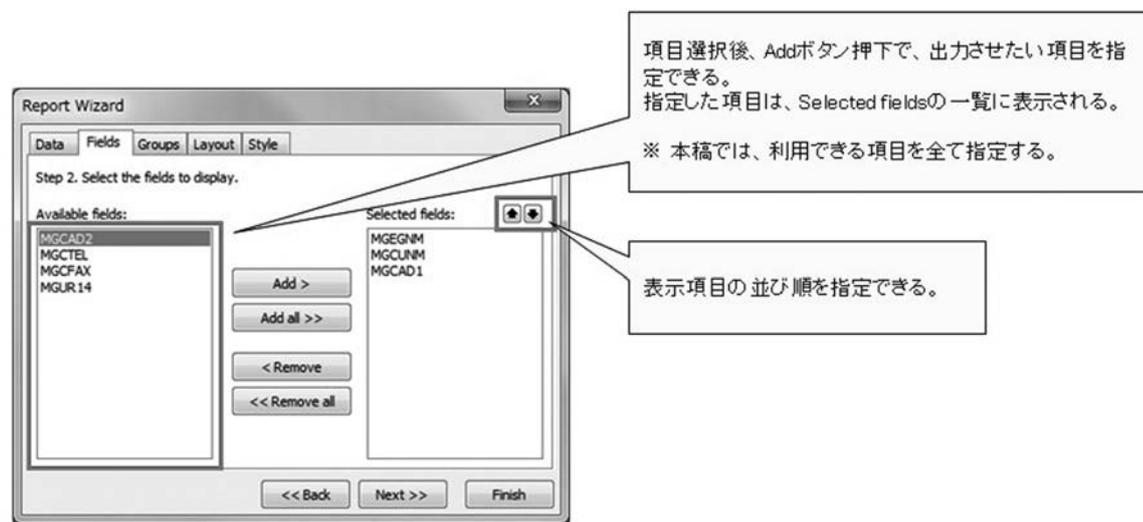


図5



以上でグラフを出力するための設定は完了である。あとは、Delphi からプレビュー処理を呼び出すだけであるが、グラフを含む帳票を出力する場合は、必ず Delphi ソースの Uses に frxChart を追加しておく必要がある。

4. 画像ファイルを用いた帳票の作成

本章では、画像ファイルを用いた帳票の作成手順について紹介する。画像ファイルの出力もグラフの出力と同様に簡単な手順で行うことができる。今回の例では、ページのフッターごとに会社のロゴを出力するプログラムを作成する。【図 21】

まず、レポートデザイナー画面で、TfrxPictureView コンポーネントを Page Footer Band に貼り付ける。【図 22】

次に、Delphi にてプレビュー処理の前に、次のロジックを記述する。変数 frxPic を TfrxPictureView で宣言し、先ほどデザイン画面で貼り付けた TfrxPictureView コンポーネントを TfrxReport の FindObject 関数で探し出し、セットする。この際、FindObject 関数の引数には、デザイン画面で貼り付けたコンポーネントの Name をセットする点がポイントである。次に、LoadFromFile 関数を使用し、出力したい画像データのパスを引数にセットする。最後に、プレビュー処理を実行すれば、画像データを使用した帳票出力の完成である。【ソース 1】

なお、今回は画像ファイルに JPEG ファイルを用いたが、他にも BMP ファイルや PNG ファイルなども使用可能である。

5. 帳票作成プログラムの応用テクニック

本章では、FastReport で帳票を作成する際に利用できるテクニックを紹介する。これらの内容を活用すれば、FastReport を使用した帳票作成プログラムの幅がより広がる。

配列で保持しているレコードデータの出力

FastReport では、データベースで取

得したレコードのデータだけでなく、Delphi プログラム上で内部保持している配列データも、Master Data Band を使用して出力することが可能である。【図 23】

Master Data Band を使用するため、Delphi のロジックで for 文などの繰り返し処理を記述せずに非常に少ないロジックで配列データの出力を行える。

まずは Delphi の開発画面で TfrxUserDataSet コンポーネントを配置し、UserName プロパティにレポートデザイナー画面で表示させたいデータセット名、Fields プロパティに出力させたい項目を設定する。【図 24】

次に、レポートデザイナー画面の設定を行う。「Report | Data」から Select Report Datasets ダイアログを開き、先ほど Delphi の開発画面の UserName プロパティで設定したデータセット名（本稿では frxUserDataSet1 と設定）にチェックを入れ、OK を押下する。【図 25】

すると、Data Tree に frxUserDataSet1 が新しく追加される。この後の手順は、TfrxDBDataSet コンポーネントを使用するときと同様に、Master Data Band を配置し、出力したい項目をその上に配置する。以上で、レポートデザイナー画面の設定は完了である。

ただし、TfrxUserDataSet の場合は、TfrxDBDataSet を使用する場合と違い、レポートウィザードでの作成ができないため、その点は注意が必要である。

最後に Delphi でデータ出力処理のロジックを記述する。データベースから値を出力する際は、ShowReport と記述するだけで、帳票出力が可能であったが、配列データを出力する際は、他にもロジックの記述が必要になる。使用するイベントは TfrxUserDataSet コンポーネントの OnCheckEOF イベントと OnGetValue イベントである。OnCheckEOF イベントでは、どのタイミングで処理を終了するかを記述し、OnGetValue イベントでは、frxUserDataSet コンポーネントの Fields プロパティで定義した変数に、どの値をセットするかを記述する。あとは、データ取得処理のあとに、帳票出力処理を呼び出せば、完成である。【ソース 2】

1行ごとに背景色を変更する方法

【図 23】のように、連続した明細データが出力された時、データを見づらく感じたことはないだろうか。そういった場合によく用いられるのが、1行ごとにレコードの背景色を分け、データを見やすくするという手法である。もちろん FastReport でもその手法を用いることができるのだが、少し工夫が必要となる。

まず、レポートデザイナー画面で TfrxMemoView コンポーネントを選択し、Master Data Band 上に配置する。配置が終わるとダイアログ画面が表示されるので、Highlight タブを選択し、条件式に「<Line#> Mod 2」と記述する。

次に、Align プロパティで baClient を選択し、Master Data Band 全体に配置されるように設定する。最後に、右クリックより Send to Back（最背面へ移動）を選択し、出力項目の背面に配置されるように設定する。【図 26】

あとは、プレビュー処理を呼び出し、出力内容を確認すると、【図 27】のように1行ごとに背景色が設定された帳票が出力される。

帳票レイアウトの分離

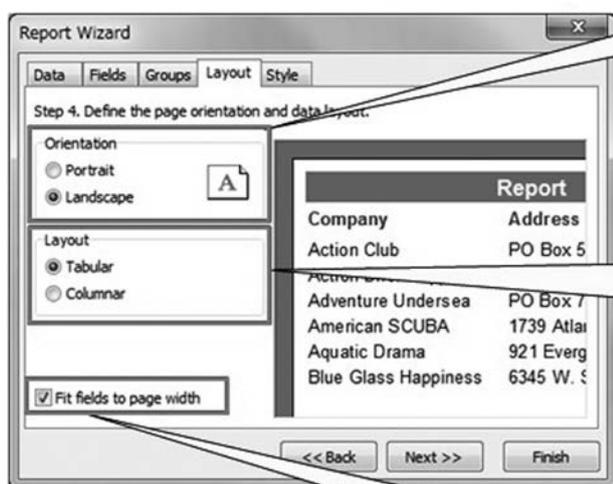
FastReport は、帳票レイアウトを実行モジュール（Exe ファイル）内に組み込んで出力する方法だけでなく、帳票レイアウトを1つのレイアウトファイルとして作成し、実行モジュールから参照することも可能である。メリットとしては、項目の配置移動などレイアウトの修正のみを行った場合に、実行モジュールの再コンパイルが不要となる点である。

ではまず、実行モジュールからレイアウトファイルを作成する方法であるが、レポートデザイナー画面を起動し、名前を付けて保存を行う。すると、帳票レイアウトが fr3 という拡張子のファイルで保存される。

次に、作成したレイアウトファイルを読み込むには、プレビュー処理の実行前に、LoadFromFile 関数を呼び出し、引数に先ほど保存したレイアウトファイルのパスを指定する。レイアウトファイルの参照は、このわずか1行のロジックを追加するだけで可能となる。【ソース 3】

ただし、引数に指定するパスの指定を間違えると、帳票が正しく出力されない

図6

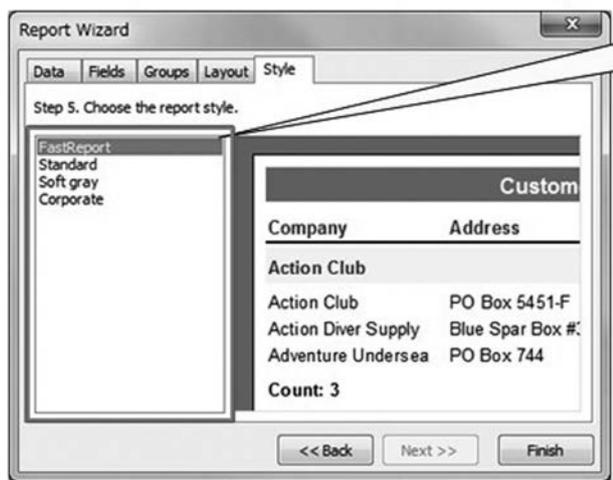


用紙の向きを設定する。
※ 本稿ではLandScapeを指定する。

項目の配置方法を指定する。
・ Tabular：出力項目を横並びで配置する。
・ Columnar：出力項目を縦並びで配置する。
※ 本稿ではTabularを指定する。

項目をページ幅に合わせるか設定する。
※ 本稿ではチェックを付ける。

図7



レポートのスタイルを指定する。
※ 本稿ではFastReportを指定する。

図8

ReportTitle: ReportTitle1					
Report					
PageHeader: PageHeader 1					
MGCUNM	MGCAD1	MGCAD2	MGCTEL	MGCFAV	MGUR14
GroupHeader: GroupHeader 1					frxDBDataset1."MGEGRM"
[frxDBDataset1."MGEGRM"]					
MasterData: MasterData1					
[frxDBDataset1."MGCUNM"]	[frxDBDataset1."MGCAD1"]	[frxDBDataset1."MGCAD2"]	[frxDBDataset1."MGCTEL"]	[frxDBDataset1."MGCFAV"]	[frxDBDataset1."MGUR14"]
GroupFooter: GroupFooter 1					
PageFooter: PageFooter 1					
					Page

レポートタイトル名や明細タイトル名は
手動で設定する。

ので注意が必要である。

6.最後に

今回は、Delphi/400 の帳票ツールの1つである FastReport の帳票作成テクニックを紹介した。FastReport はレポートウィザードの使用で、コンポーネントを貼り付ける手間もなく、容易にデザインの基盤が作成できる。また、グラフや画像ファイルの挿入も FastReport のレポートデザイナー画面でのビジュアル設計が可能であるため、Delphi でのソースをほとんど記述することなく作成できることを本稿でお分かりいただけたであろう。よって、FastReport を利用すれば、誰でも容易に帳票出力を利用したアプリケーション作成が可能である。

FastReport のバンドル版は Delphi/400 Version XE3 以降に付属しており、すぐに使用できるので、今後はさらに利用するユーザーが増えていくと思われる。その際に、本稿の内容をご活用いただければ幸いです。

M

図9

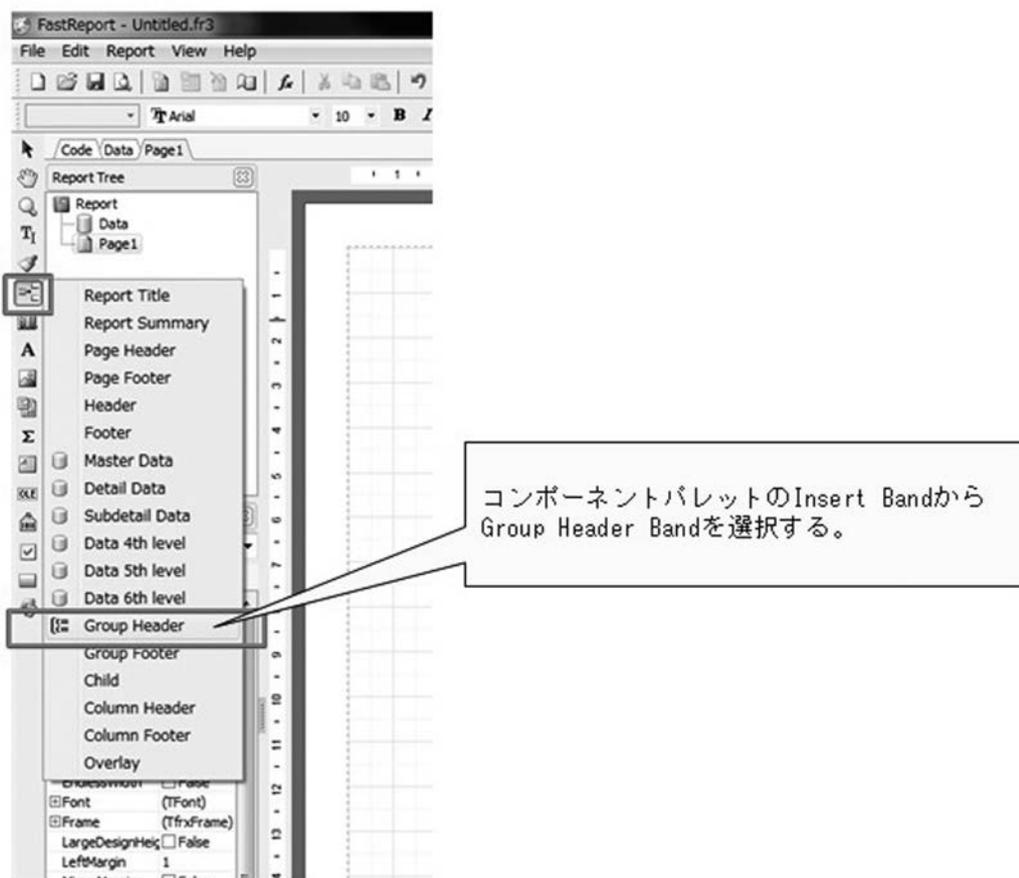


図10

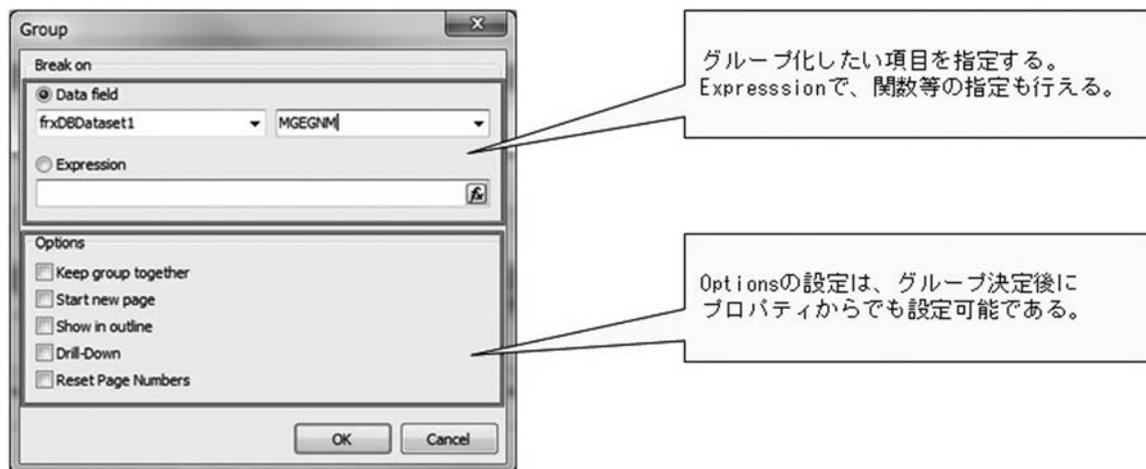


図11

売上一覧表(2014年)					
名称	住所1	住所2	電話番号	FAX番号	金額
東京営業所					小計: 29,000
大阪営業所					小計: 29,000
青森営業所					小計: 500
広島営業所					小計: 1,700
高知営業所					小計: 12,910
					合計: 73,110

営業所名をクリックすると
詳細内容の表示・非表示を切り替えることができる。
【図12】参照

2014/08/20 15:33:38 Page 1

図12

売上一覧表(2014年)					
名称	住所1	住所2	電話番号	FAX番号	金額
東京営業所					
株式会社足立商店	東京都足立区	1-1-2	XXX-XXXX-XXX	XXX-XXXX-XXX	1,000
株式会社足立興業	東京都足立区	1-2-3	XXX-XXXX-XXX	XXX-XXXX-XXX	2,000
株式会社荒川商店	東京都荒川区	2-2-3	XXX-XXXX-XXX	XXX-XXXX-XXX	3,000
株式会社荒川工業	東京都荒川区	1-2-1	XXX-XXXX-XXX	XXX-XXXX-XXX	5,000
株式会社板橋商店	東京都板橋区	5-2-3	XXX-XXXX-XXX	XXX-XXXX-XXX	18,000
					小計: 29,000
大阪営業所					小計: 29,000
青森営業所					小計: 500
広島営業所					小計: 1,700
高知営業所					小計: 12,910
					合計: 73,110

詳細内容が表示される。

2014/08/20 15:33:38 Page 1

図13



図14

ReportTitle: ReportTitle1
売上一覧表(2014年)

PageHeader: PageHeader1
名称 住所1 住所2 電話番号 FAX番号 金額

GroupHeader: GroupHeader1
[frxDBDataset1."MGEQNM"]

MasterData: MasterData1
[frxDBDataset1."MGCUNM"] [frxDBDataset1."MGCAD1"] [frxDBDataset1."MGCAD2"] [frxDBDataset1."M"] [frxDBDataset1."S"]

GroupFooter: GroupFooter1
小計: [frxDBDataset1."S"]

Footer: Footer1
合計: [frxDBDataset1."G"]

PageFooter: PageFooter1
Page

GroupFooterに小計を配置し、Footerに総合計を配置する。
DrillDown機能を使用時は、計算項目を使用せず、Delphiで計算処理を行い、セットする。

図15

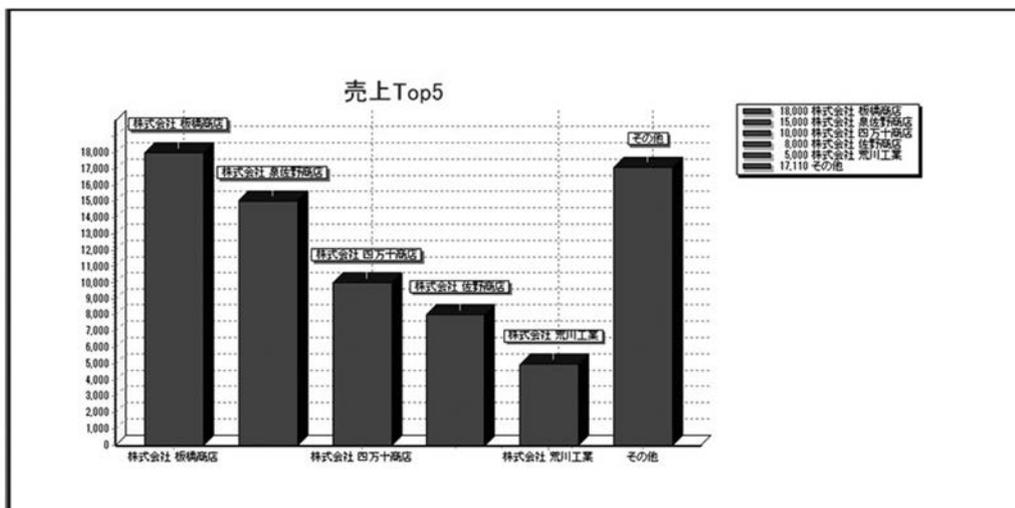


図16

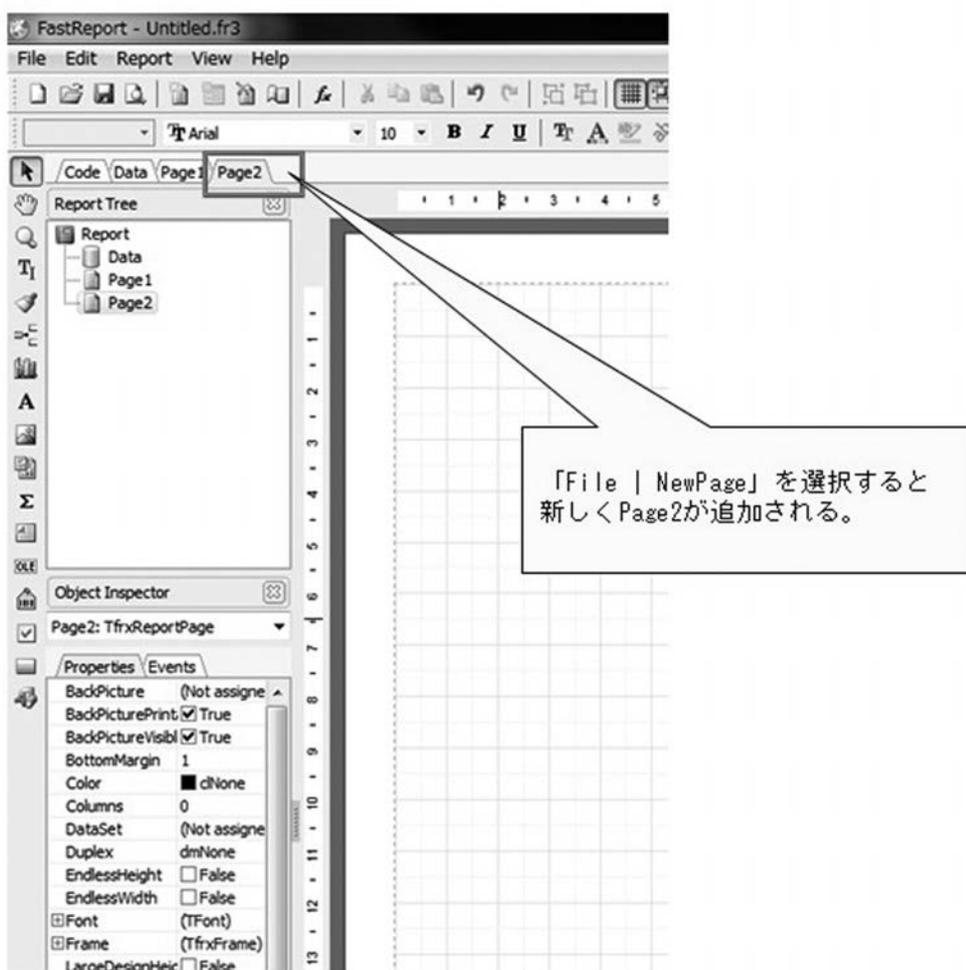


図17

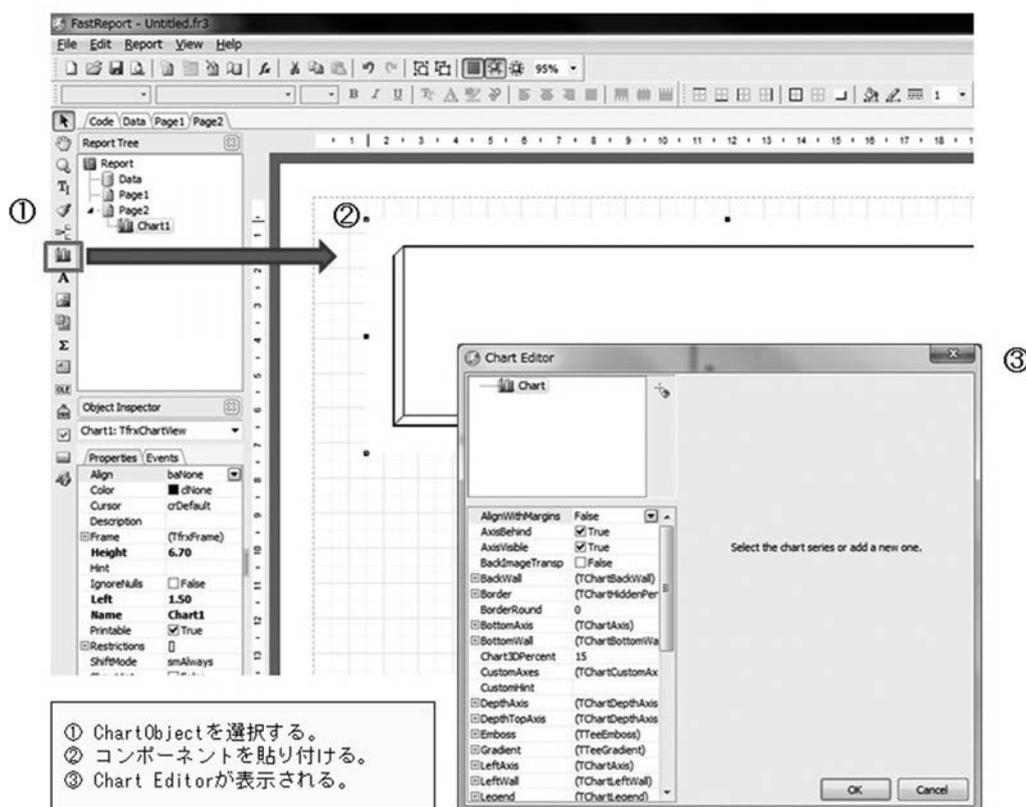


図18

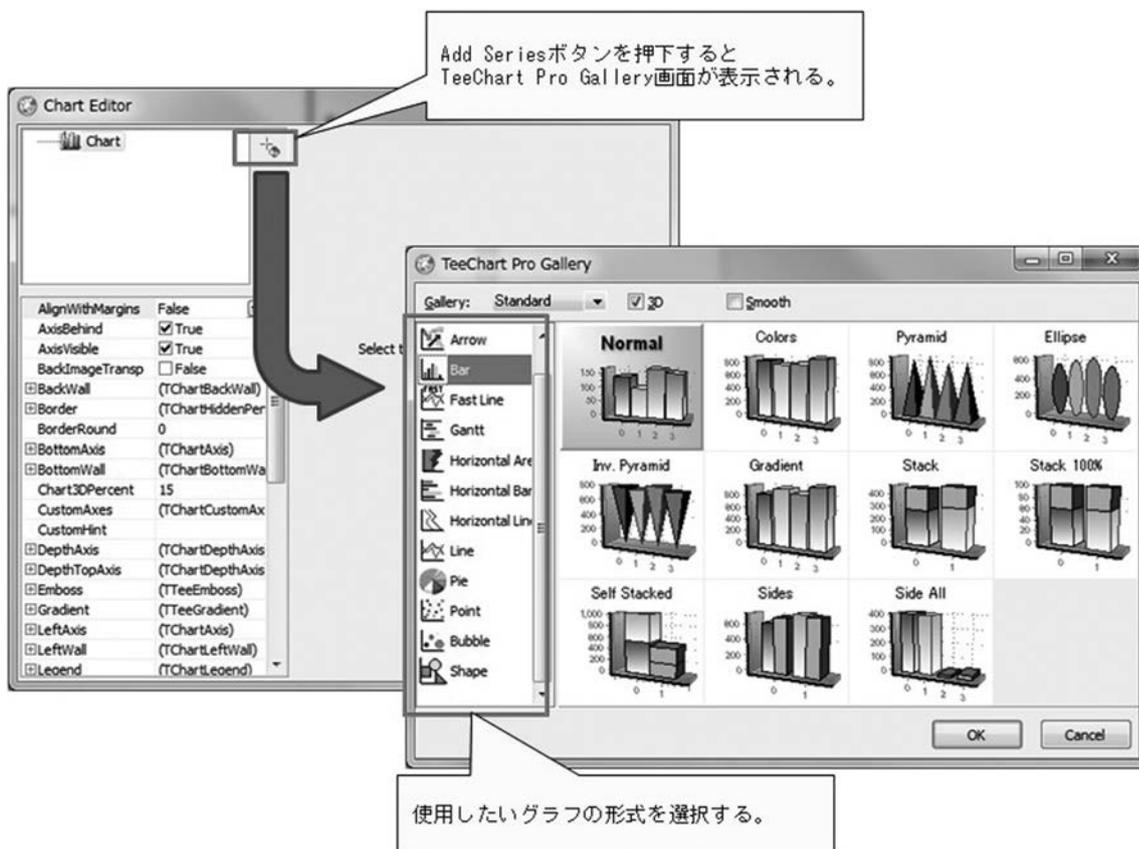


図19

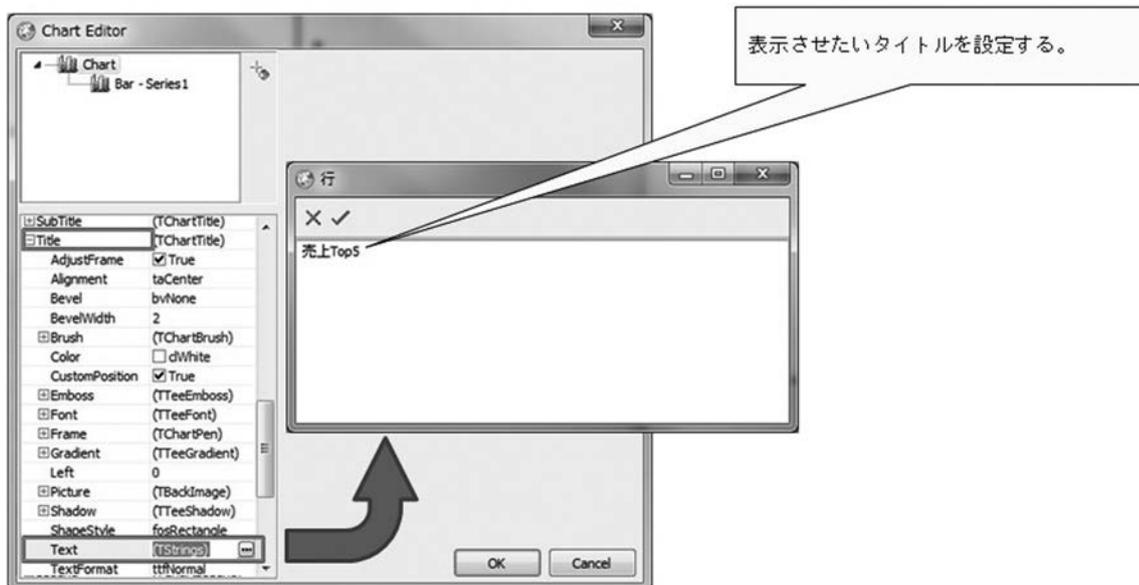


図20

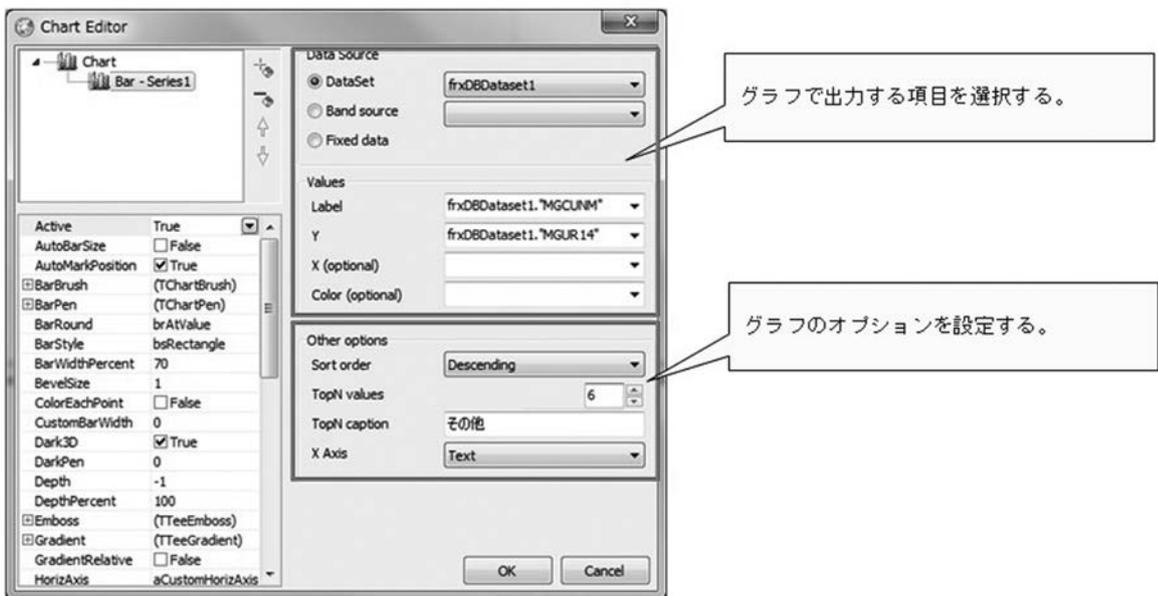
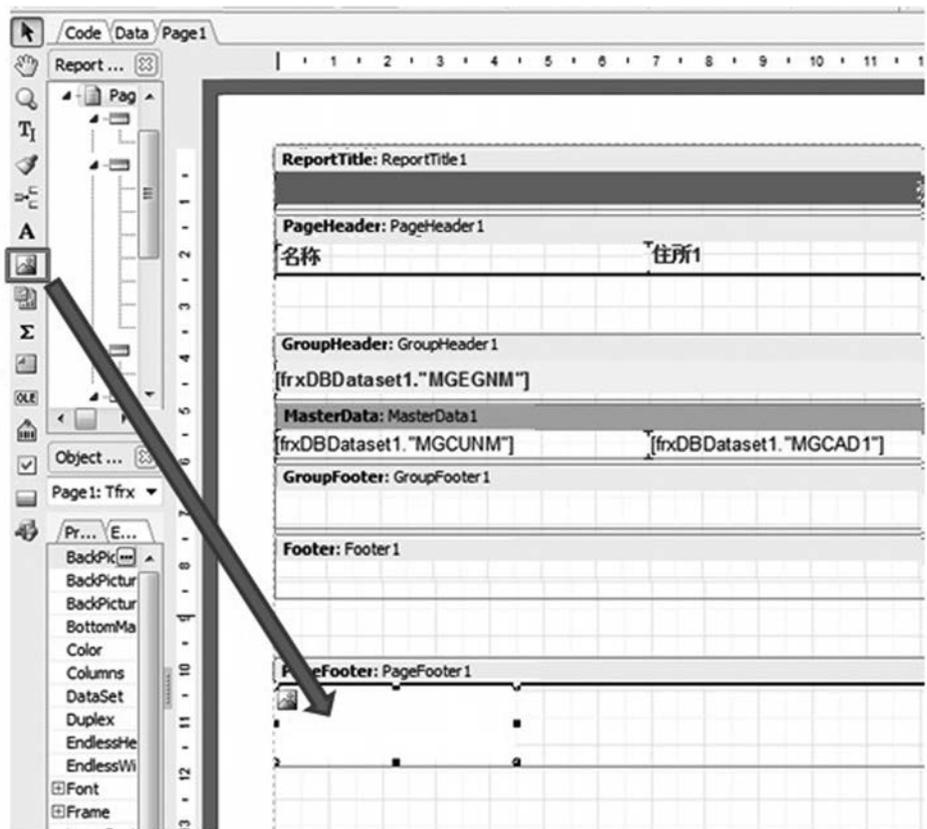


図21



図22



ソース1

```

{*****}
目的：プレビューボタン押下時処理
引数：
戻値：
{*****}
procedure TForm1.Button1Click(Sender: TObject);
var
  frxPic: TfrxPictureView;
begin
  // 画像の読み込み
  frxPic := frxReport1.FindObject('PICTURE1') as TfrxPictureView;
  frxPic.Picture.LoadFromFile('C:\Test\migaro_logo.jpg');

  // プレビュー処理
  frxReport1.ShowReport;
end;

```

保存したパスを指定する。

図23

営業所別売上一覧表 (過去5年)					
営業所名	2010年	2011年	2012年	2013年	2014年
東京営業所	1,000	1,000	2,000	3,000	4,000
大阪営業所	2,000	2,000	4,000	6,000	8,000
名古屋営業所	3,000	3,000	6,000	9,000	12,000
青森営業所	4,000	4,000	8,000	12,000	16,000
広島営業所	5,000	5,000	10,000	15,000	20,000
高知営業所	6,000	6,000	12,000	18,000	24,000
仙台営業所	7,000	7,000	14,000	21,000	28,000
博多営業所	8,000	8,000	16,000	24,000	32,000
宮崎営業所	9,000	9,000	18,000	27,000	36,000
鹿児島営業所	10,000	10,000	20,000	30,000	40,000

図24

frxUserDataSet1 TfrxUserDataSet

プロパティ イベント

Description

Enabled True

Fields (TStrings)

LiveBinding デザイン LiveBinding デザイン

Name frxUserDataSet1

RangeBegin rbFirst

RangeEnd reLast

RangeEndCount 0

Tag 0

UserName frxUserDataSet1

表示させたい項目を設定する。

➔

文字列リストの設定

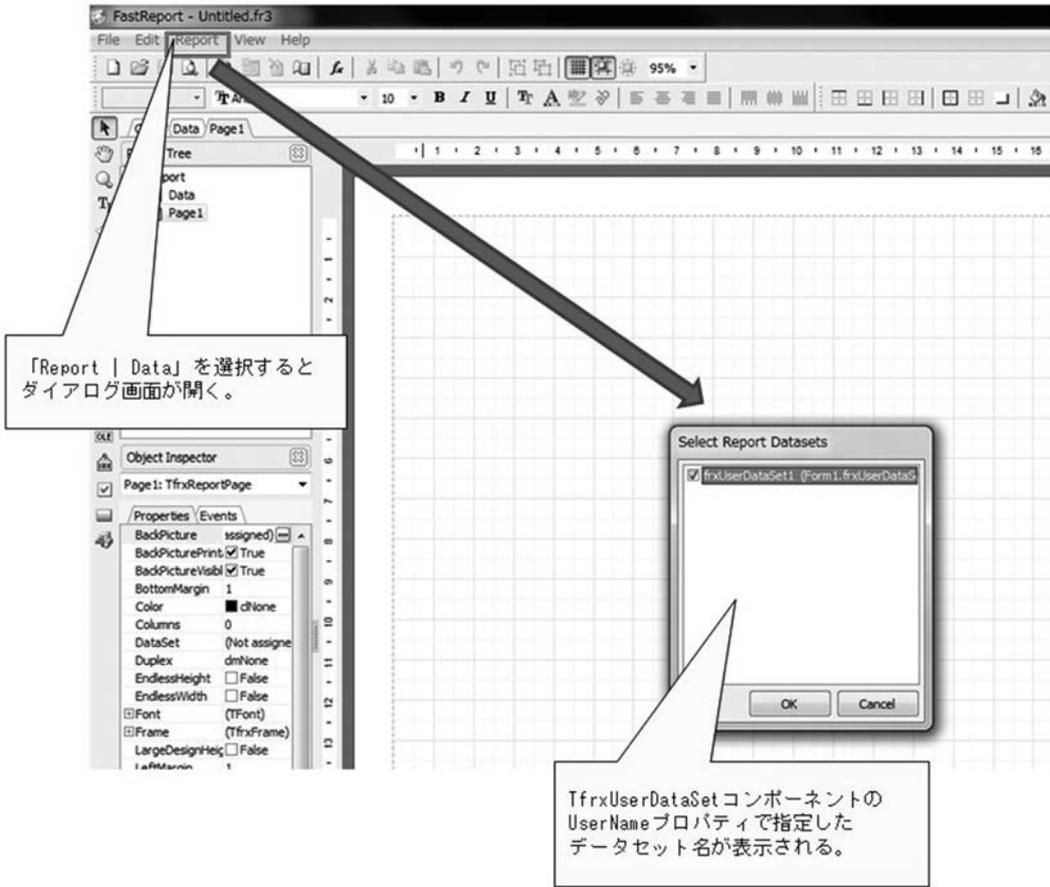
6行

- EGNAME
- URKN10
- URKN11
- URKN12
- URKN13
- URKN14

エディタ(C)... OK(O) キャンセル ヘルプ

レポートデザイナー画面で表示させたいデータセット名を設定。

図25



ソース2

```

{*****}
目的：プレビューボタン押下時処理
引数：
戻値：
{*****}
procedure TForm1.Button4Click(Sender: TObject);
begin
    // 配列データ取得処理
    GetURIRecord;

    // プレビュー処理
    frxReport1.ShowReport;
end;

{*****}
目的：データセットに配列データをセット
引数：
戻値：
{*****}
procedure TForm1.frxUserDataSet1GetValue(const VarName: string;
var Value: Variant);
begin
    if VarName = 'EGNAME' then
        Value := ArrayURIRec[frxUserDataSet1.RecNo].EGNAME // 営業所名
    else
        if VarName = 'URKN10' then
            Value := ArrayURIRec[frxUserDataSet1.RecNo].URKN10 // 2010年売上金額
        else
            if VarName = 'URKN11' then
                Value := ArrayURIRec[frxUserDataSet1.RecNo].URKN11 // 2011年売上金額
            else
                if VarName = 'URKN12' then
                    Value := ArrayURIRec[frxUserDataSet1.RecNo].URKN12 // 2012年売上金額
                else
                    if VarName = 'URKN13' then
                        Value := ArrayURIRec[frxUserDataSet1.RecNo].URKN13 // 2013年売上金額
                    else
                        if VarName = 'URKN14' then
                            Value := ArrayURIRec[frxUserDataSet1.RecNo].URKN14; // 2014年売上金額
                        else
                            Value := '';
                        end;
                    end;
                end;
            end;
        end;
    end;
end;

{*****}
目的：帳票出力終了チェック処理
引数：
戻値：
{*****}
procedure TForm1.frxUserDataSet1CheckEOF(Sender: TObject; var Eof: Boolean);
begin
    // 配列数がデータセットのレコード数を超えた場合、出力処理終了
    Eof := frxUserDataSet1.RecNo > High(ArrayURIRec);
end;

```

帳票出力メイン処理

OnGetValueイベントの処理
If文で項目名を判断し、Valueに値をセットする。
frxUserDataSet1.RecNoでデータセットの行番号を表している。

OnCheckEOFイベントの処理

図26



図27

営業所別売上一覧表 (過去5年)					
営業所名	2010年	2011年	2012年	2013年	2014年
東京営業所	1,000	1,000	2,000	3,000	4,000
大阪営業所	2,000	2,000	4,000	6,000	8,000
名古屋営業所	3,000	3,000	6,000	9,000	12,000
青森営業所	4,000	4,000	8,000	12,000	16,000
広島営業所	5,000	5,000	10,000	15,000	20,000
高知営業所	6,000	6,000	12,000	18,000	24,000
仙台営業所	7,000	7,000	14,000	21,000	28,000
博多営業所	8,000	8,000	16,000	24,000	32,000
宮崎営業所	9,000	9,000	18,000	27,000	36,000
鹿児島営業所	10,000	10,000	20,000	30,000	40,000

ソース3

```

{*****}
目的：プレビューボタン押下時処理
引数：
戻値：
{*****}
procedure TForm1.Button4Click(Sender: TObject);
begin
    // 配列データ取得処理
    GetURIRecord;
    // レイアウトの読み込み処理
    frxReport1.LoadFromFile('C:\Report\SampleReport.fr3');
    // プレビュー処理
    frxReport1.ShowReport;
end;
    
```

帳票出力メイン処理にレイアウトの読み込み処理を追加する。

保存したパスを指定する。