

畑中 侑

株式会社ミガロ.

システム事業部 システム2課

[Delphi/400]

FastReportへの効率的な 帳票レイアウトコンバート

- はじめに
- 効率的な帳票レイアウトコンバート手法
- おわりに



略歴
1983年7月6日生まれ
2006年 京都産業大学 法学部卒業
2006年4月 株式会社ミガロ. 入社
2006年4月 システム事業部配属

現在の仕事内容
システムの受託開発を担当しており、要件確認から納品・フォロー、保守作業に至るまで、システム開発全般に携わっている。

1.はじめに

アプリケーションのマイグレーションにはさまざまな目的があるが、最近の傾向としてはシステム基盤の最新 OS への対応や、新しい IT 技術、たとえばスマートデバイス活用の検討などが挙げられる。

2016年7月にリリースされた Delphi/400 10 Seattle では、Windows 10 への正式対応や、マルチデバイス開発環境が強化されており、Delphi/400 のマイグレーションを検討・実施された企業は非常に多い。こうしたマイグレーションの際、事前に検討すべき課題の1つに帳票ツールがある。

Delphi 7 以前のバージョンでは、帳票ツール「QuickReport」により帳票機能を作成していることが多い。しかしバンドルされる帳票ツールが変わり、QuickReport が標準サポートされなくなつてから、帳票機能を別の帳票ツールに作り直す必要が生じてきた。

このような作り直しの手間が、マイグレーションに踏み切れない理由になる場合もある。この課題に対する解決策の1つとして、本稿では QuickReport から Delphi 最新版に付属している帳票ツール「FastReport」へのコンバートを題材に効率的な手法を説明する。

2.効率的な帳票レイアウト コンバート手法

通常、帳票ツールを変更する場合には、帳票レイアウトを新しい帳票ツールで作成し直すことになる。そうした作業はプログラムの移行に時間を要するが、QuickReport から FastReport へコンバートする場合には、FastReport で用意されているコンバート機能を利用することで、効率的に移行できる。大きな流れとしては、コンバート作業の環境を準備し、移行用のプロジェクトを作成してコンバート処理を実行する。

本稿では、QuickReport が付属する最終バージョンの Delphi 7 から、

FastReport が最初に付属した Delphi XE3 へのマイグレーション環境で検証している。

2-1. コンバート環境の準備

コンバートを実施する作業環境として、QuickReport を使用していた Delphi 開発環境（本稿では Delphi 7）に FastReport のコンポーネントをインストールする。

手順は、次のとおりである。

① FastReport コンポーネントの「.bpl」ファイル作成

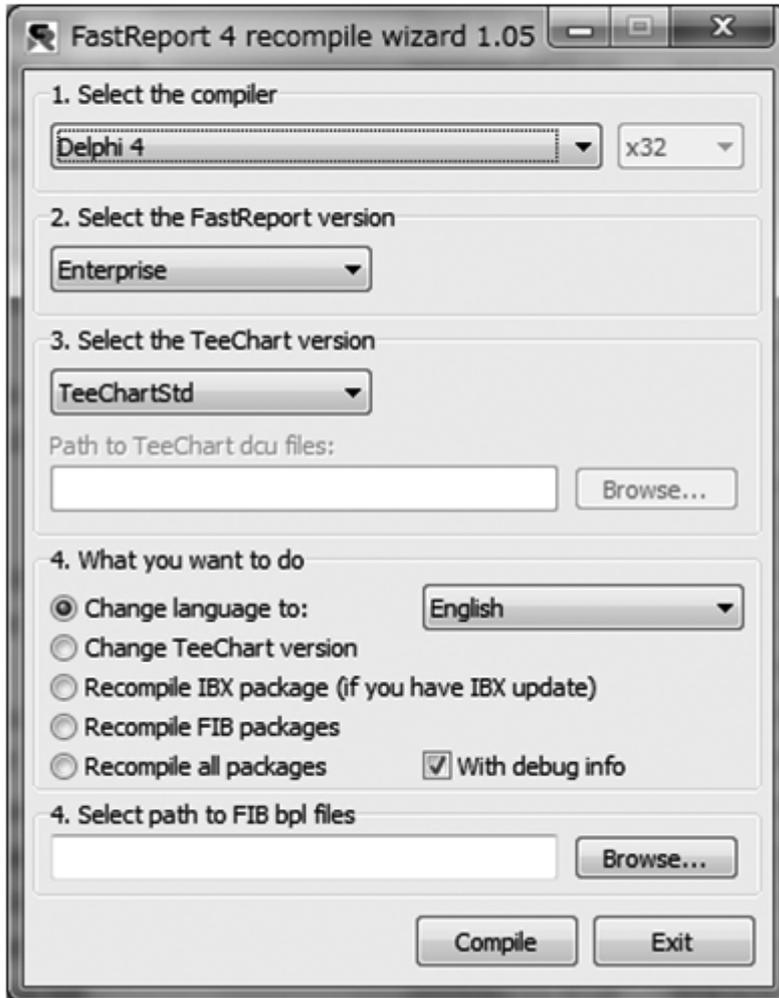
FastReport のインストールフォルダ内に、「recompile.exe」という実行ファイルが存在する。もしくは、Windows ボタンから「すべてのプログラム | FastReports | FastReport」内を見ると、「Recompile Wizard」が見つかる。

これを起動すると、【図1】のウィザード画面が表示される。

このウィザード画面で上から順に設定値を指定し、最後に "Compile" ボタン

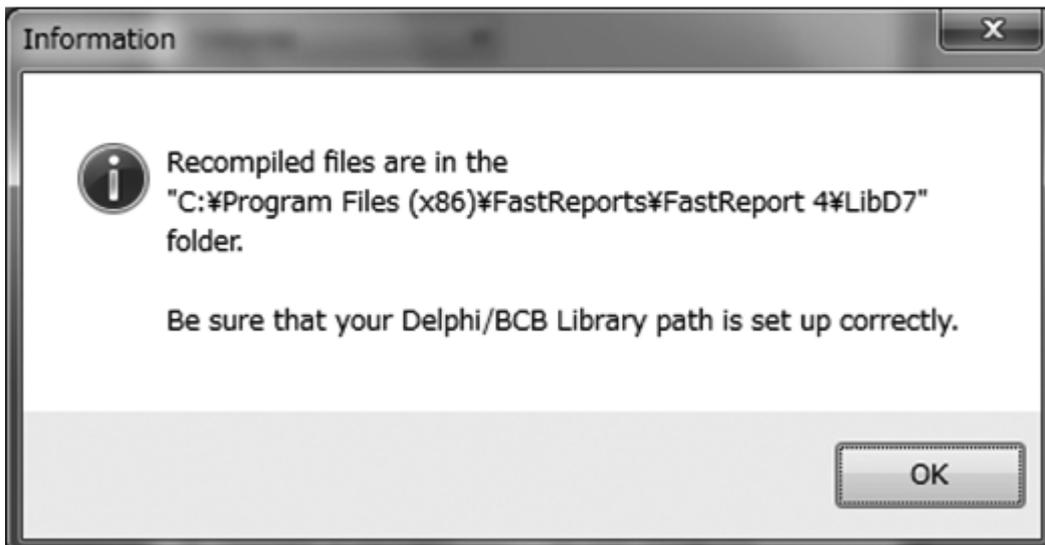
☒1

FastReport 4 recompile wizard



☒2

Information



を押すことで、「FastReport 4」フォルダ内に「LibD*」というフォルダが作成され、各種「.bpl」ファイルを一括で作成できる（「*」は指定バージョンに応じてバージョン値が入る）。

以下に、ウィザード画面の設定内容を確認する。

(1) Select the compiler

開発環境のバージョンを選択する。本稿では「Delphi7」を指定。

(2) Select the FastReport version

FastReport のバージョンを選択する。本稿では「Enterprise」を指定。

(3) Select the TeeChart version

FastReport で使用するグラフ「TeeChart」のバージョンを選択する。本稿では「TeeChartStd」を指定。

(4) What you want do

リコンパイル目的を選択する。本稿では「Recompile all packages」を指定。

(4) Select path to FIB bpl files

レポート内のデータソースで FIB 接続を用いる場合に選択する。本稿では不要なので設定はしない（連番 (4) が続いているが、画面表記に合わせている）。

各項目を設定し、「Compile」ボタンを押すことで、コンパイル処理が自動で始まり、【図 2】の確認メッセージが表示されれば完了である。

この操作で、「FastReport 4」フォルダ内に「LibD7」というフォルダが作成され、その中に各種「.bpl」が存在することが確認できる。【図 3】

② FastReport 関連コンポーネントの登録

Delphi を起動し、メニューバーより「コンポーネント | パッケージのインストール…」を順番に選択し、「デフォルトプロジェクトオプション」のウィンドウを起動する。

中段の「追加 (A) …」ボタンより、先ほど「LibD7」フォルダ内に生成された各種「.bpl」を順次追加する。追加が終われば、「OK」ボタンを押下して、「デフォルトプロジェクトオプション」を終了する。【図 4】

③ ライブラリパスの追加

Delphi 開発環境のメニューバーより、「ツール | 環境オプション…」を順番に選択し、「環境オプション」のウィンドウを起動する。

次に「ライブラリ」タブに切り替えて、「ライブラリパス (B)」の右側にある参照ボタンを押下し、「ディレクトリ」のウィンドウを起動する。「ライブラリパスの一覧」に先ほどの「LibD7」フォルダのパスを加える。【図 5】

具体的には、中断の参照ボタンを押下し、「LibD7」フォルダのパスを指定すると、「追加 (A)」ボタンがアクティブになるので押下する。その後、「OK」ボタンで「ディレクトリ」「環境オプション」の各ウィンドウを終了する。

2-2. コンバート用プロジェクトの新規作成

QuickReport から FastReport へ帳票レイアウトをコンバートするには、FastReport に用意されている「ConverterQR2FR」ユニットを利用する。そのため、この作業ではコンバート用にプロジェクトの作成が必要となる。

① プロジェクトの新規作成

Delphi 開発環境を起動し、メニューバーより「ファイル | 新規作成 | アプリケーション」を選択し、新規プロジェクトを作成する。次に初期生成された Unit1 のデザインに対して、任意のフォームサイズに変更し下記のコンポーネントを配置する。

Button ツールパレット = Standard frxReport

ツールパレット = FastReport 4.0

最後に、コンバート対象の TQuickRep をプロジェクトに追加する。本稿では「QuickFrm」とする。【図 6】

フォームに配置したボタンを押下することで、コンバート対象の TQuickRep の帳票レイアウトを FastReport のデザインファイルにコンバートする。

プロジェクトの保存内容と各プロパティ設定は、【図 7】のとおりである。

● プロジェクト

保存ファイル名 : ConvertSample.dpr

● 初期生成フォーム (TForm1)

Name プロパティ : frmMain
保存ファイル名 : MainFrm.pas

● 配置した Button

Name プロパティ :
btnConvertQuickReport

Caption プロパティ :

ConvertQuickReport

● 配置した frxReport

● Name プロパティ : frxReport

② コンバート手続きの実装

MainFrm のソース部を開き、uses 節に「ConverterQR2FR」とコンバート対象の QuickFrm を追加する。続いて、btnConvertQuickReport の OnClick イベントにコンバート手続きを実装する。【ソース 1】

TConverterQR2FR の変数を定義し、Source プロパティにコンバート対象の TQuickRep、Target プロパティに TfrxReport を指定したあと、Convert 手続きによりコンバートが実行される。その後、SaveToFile 手続きにより FastReport のデザインファイルとして保存する。

③ コンパイルして EXE を生成する

プロジェクトをコンパイルし、エラーがないことを確認し、EXE ファイルを生成する。QuickReport のバージョンによっては未定義の変数が TConverterQR2FR で宣言されているため、エラーとなる場合はコメントアウトする。【ソース 2】

2-3. コンバート処理の実行

作成した EXE ファイルを実行し、QuickReport の帳票レイアウトを FastReport へコンバートする。

EXE ファイル実行後、「Convert QuickReport」ボタンを押下すると、EXE ファイルと同階層に FastReport のデザインファイルである拡張子「fr3」のファイルが生成されたことを確認できる。続いて FastReport を起動し、生成された FastReport のデザインファイルを開いて確認する。【図 8】

QuickReport の TQuickRep が、FastReport の TfrxReportPage に変換され、TQRLabel が、TfrxMemoView に置き換えられており、レイアウトが変換されていることが確認できる。

図3

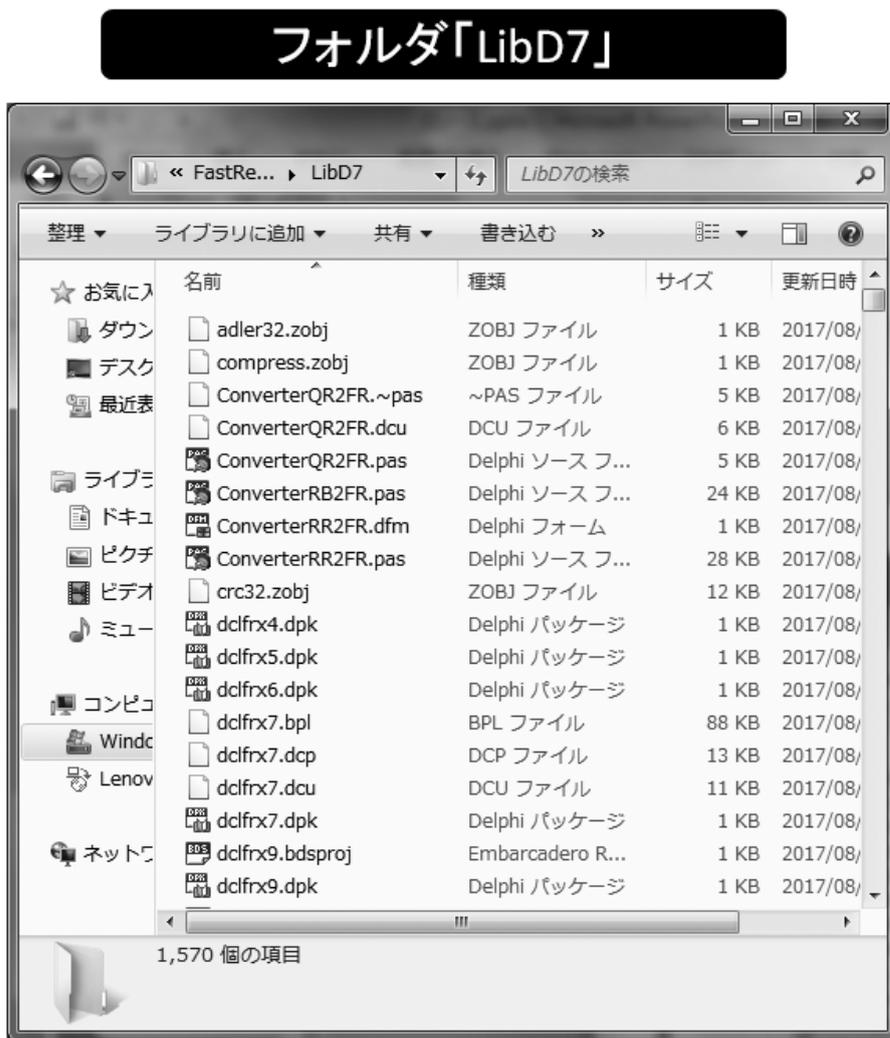


図4



しかし【図8】のように、TfrxMemo ViewのText部で一部の文字列が文字化けする場合もある。こうした文字化けが発生した場合は、コンバート後のファイルをFastReportで確認し、手動で設定などを調整する必要がある。

またコンバート後の環境に合わせて、適時データセットの割り当てを補填すれば、新しいDelphi開発環境（本稿ではXE3）ですぐにFastReportのプログラムを利用できる。これでQuickReportからFastReportへの帳票レイアウトのコンバートが完了である。

3.おわりに

本稿ではQuickReportからFastReportの帳票レイアウトのコンバートを題材に効率的な手法を調査・検証しているが、QuickReportだけでなく、Rave Reportsについても類似したコンバート用ユニットは用意されている。

Delphiではバージョンによって付属される帳票ツールが変更されることがあるが、本稿の手法を使えば、最新の帳票ツールであるFastReportへ少ない作業で移行できる。帳票ツールの変更が作業課題となっている場合は、この手法で効率的なマイグレーションの実施に役立てていただきたい。

M

図5

ライブラリパスの設定



図6

QuickReportデザイン画面

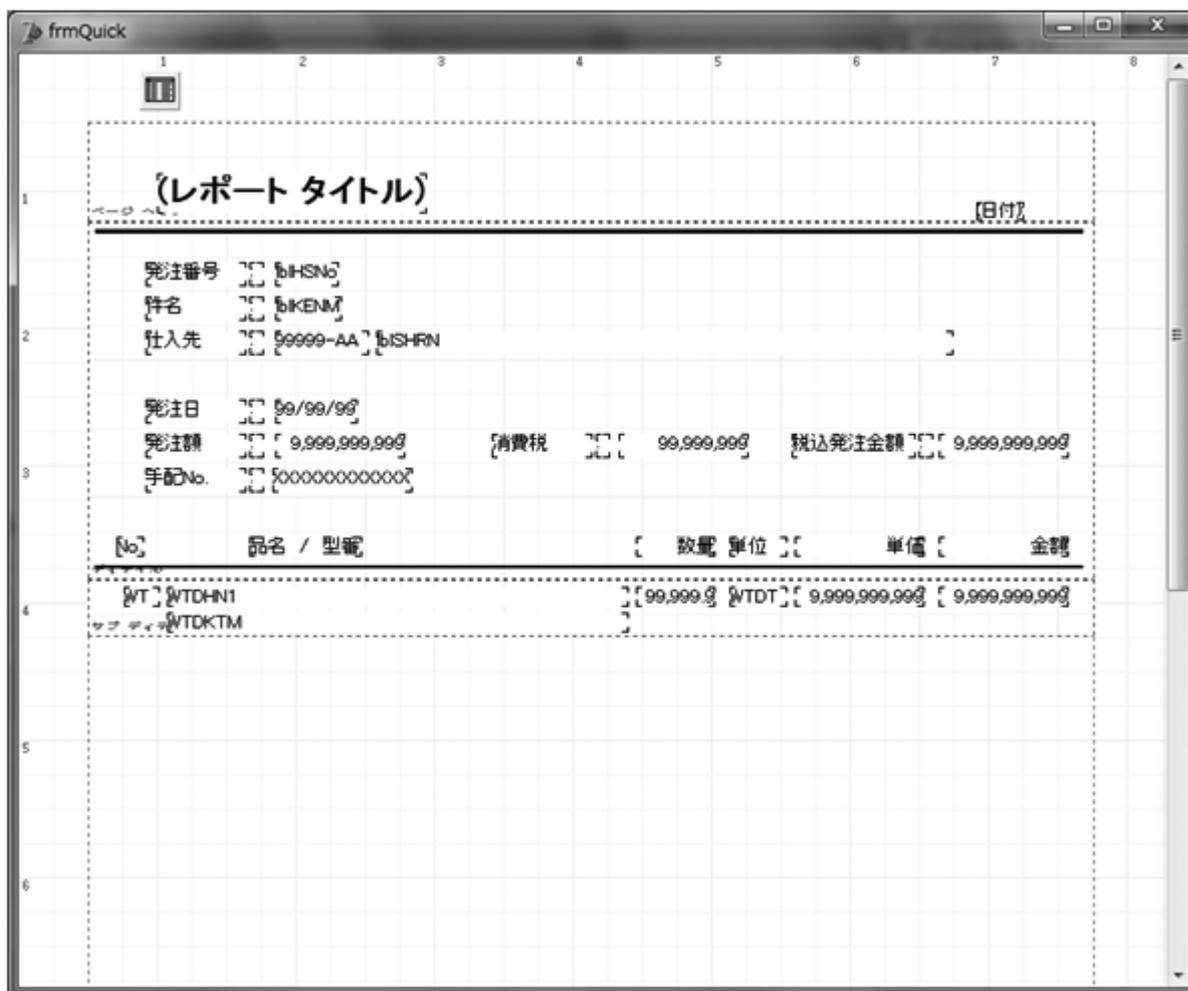


図7

プロジェクトの構成

The image shows the Visual Studio environment. On the left, the Project Manager displays a project named 'ConvertSample.exe' with a file structure including 'ProjectGroup1', 'ConvertSample.exe', 'MainFrm', and 'QuickFrm'. On the right, the WinForms Designer shows the 'frmMain' form with a 'ConvertQuick Report' button. Below it, the 'frmQuick' form is shown with a report layout.

Project Structure:

ファイル名	パス
ProjectGroup1	C:\Program Files (x86)\Bor...
ConvertSample.exe	C:\Projects\ConvertSample...
MainFrm	C:\Projects\ConvertSample...
QuickFrm	C:\Projects\ConvertSample...

frmMain Properties:

- frxReport
 - Nameプロパティ: frxReport

ConvertQuick Report Button Properties:

- Nameプロパティ: btnConvertQuickReport
- Captionプロパティ: ConvertQuickReport

frmQuick Report Content:

(レポートタイトル) [日付]

発注番号 [b]HSN []
 件名 [b]KEN []
 仕入先 [9999-AA] [b]SHRN []

発注日 [99/99/99]
 発注額 [9,999,999.99] 消費税 [99,999.99] 税込発注金額 [9,999,999.99]
 手配No. [XXXXXXXXXXXX]

[No]	品名 / 型電	数量	単位	単価	金額
[b] [b]TDH1	[] [99,999.9]	[b] [b]TDT	[] [9,999,999.99]	[] [9,999,999.99]	
サブディ	[b]TDKTM				

ソース1

btnConvertQuickReportのOnClickイベント

```
procedure TfrmMain.btnConvertQuickReportClick(Sender: TObject);
var
  conv: TConverterQR2FR;
begin
  //コンバート対象を生成
  frmQuick := TfrmQuick.Create(Self);
  try
    //TConverterQR2FRクラスを生成し、各種プロパティをセット
    conv := TConverterQR2FR.Create;
    conv.Source := frmQuick.RepHattyuList; //コンバート対象のTQuickRepを指定
    conv.Target := frxReport;           //自身のTfrxReportを指定
    conv.Convert;                       //変換処理の実行

    //EXEファイルと同階層にレポートタイトル名として保存
    frxReport.SaveToFile(frmQuick.RepHattyuList.ReportTitle + '.fr3');
  finally
    frmQuick.Release;
  end;
end;
```

ソース2

コメントアウト箇所(ConverterQR2FR.pas)

```
procedure TConverterQr2Fr.AddObjects(const ABand: TQRCustomband;
  const Band: TfrxBand);
var
  i: integer;
  ~~~省略~~~
  if ABand.Controls[i] is TQRShape then
  begin
    Sh := ABand.Controls[i] as TQRShape;
    Shape := TfrxShapeView.Create(Band);
    Shape.CreateUniqueName;
    case Sh.Shape of
      qrsRectangle: Shape.Shape := skRectangle;
      qrsCircle: Shape.Shape := skEllipse;
      //QuickReportのバージョンにより未定義のプロパティのためコメントアウト
      // qrsRoundRect: Shape.Shape := skRoundRectangle;
    end;
    Shape.SetBounds(Sh.Left, Sh.Top, Sh.Width, Sh.Height);
  end;
  ~~~省略~~~
end;
```

FastReportで確認

