シルバー賞

Delphi/400とDelphiを利用したIBM i 資源の有効活用
ーIBM i で作成した帳票配布、Delphi + OfficeエンジンによるExcel→PDFバッチ変換

小山 祐二 様
澁谷工業株式会社
経営情報システム部 主任

澁谷工業株式会社
http://www.shibuya.co.jp/

パッケージプラントを主力製品とする東証名証１部上場の機械メーカー。特に、国内外の大手飲料メーカーに採用されているボトリングシステム製造では、世界トップの地位を確立している。また、半導体製造装置、医療機器などの製造も行っている。

はじめに

澁谷工業は1931年創立、1949年設立の会社で、今日まで多くのお客様に支えられ、2011年に創立80周年を迎えることができた。当社では「カスタマーファースト」を貫き、お客様のニーズにあわせたパッケージングプラントを“ターンキー（直ぐに稼働できる状態）”で提供するビジネスを主体としている。

当社の基幹システムにおける既存帳票作成方法や帳票配布方法は、主にIBM iネイティブ環境による帳票（以下「SPOOLF」）とクイックレポート（以下「ツール」）を利用している。しかし、このSPOOLFとツールは利用するうえで強みもあるが、弱みも多い。

そこで本稿では、「Delphi/400とDelphiを利用したIBM i 資源の有効活用」と題して、既存方法とは別の帳票作成や配布方法を紹介する。

具体的には、IBM i でExcelをベースとした帳票を作成し、それをDelphi/400を利用してエンドユーザーに配布する方法、さらには「Delphi + Office エンジンによる Excel → PDF バッチ変換」方法を解説する。

開発経緯

最初にツールにおける帳票作成について簡単に説明する。このツールはヘッダー、明細、フッターごとにセグメントを分け、それらにデータをセットし、帳票を作成するものである。

このツールはレイアウトを作成するうえで、視覚的に非常に理解しやすいものである。しかし、このツールを利用するには、細かな罫線制御やセル内のフォント等の制御をすべてアプリケーション側で行う必要があり、この点が考慮点となる。

当時、帳票作成方法の選択肢に関しては、(1) ツール (2) SPOOLF (3) 新たな帳票作成ツールの導入、があった。

しかし (2) を利用すれば、逆に工数が膨らみ、帳票作成における柔軟度も下がることになる。また (3) を導入するコストもかけられなかったため、このツールを使い続けていたのが現状である。

そこで、他のさまざまな方法を模索する中で、やはりクライアント側ではなく、サーバー側（IBM i ）で処理できないかと考えた。

検証の結果、IBM i でJavaを利用してIFS（IBM iにおけるファイル格納領域）にExcelを作成できることがわかった。ここではその作成方法の詳細には触れないが、Excel自体の既存機能により（条件付き書式や各種セルの書式設定等）、前述の考慮点で挙げた制御が可能となった。すなわち、アプリケーション側で細かな制御の必要なく、帳票を作成できる。

苦労した点／その解決方法

ここでいくつかの問題点が出てきた。それは、IBM i で作成した帳票を、いかにエンドユーザーに配布するかという

図1

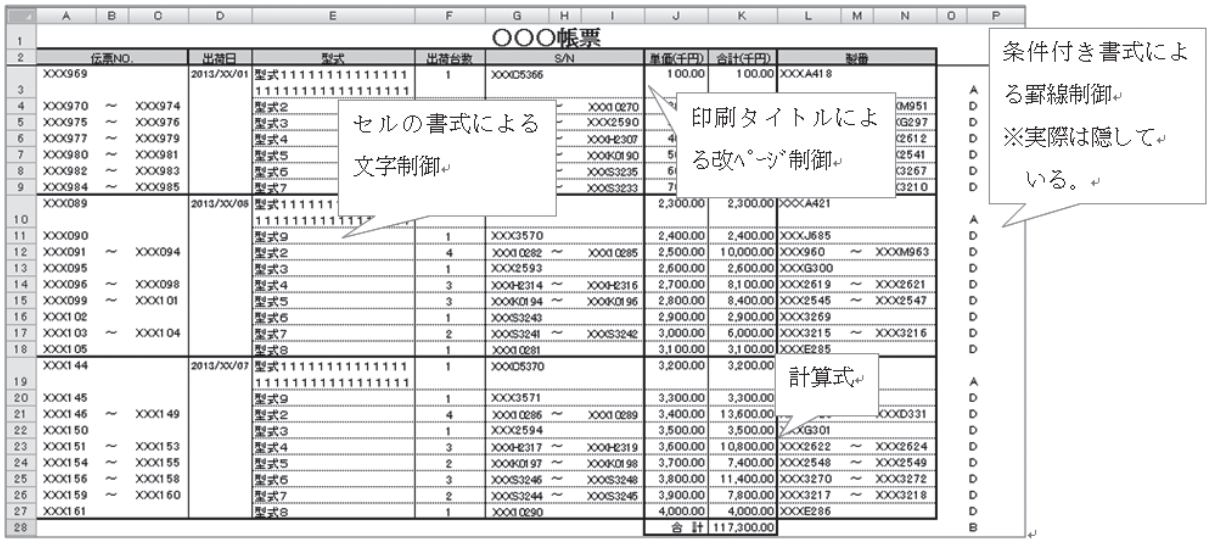
【FTPでIFSのファイルをクライアントPCにGETする方法】

```
open 「IBM i接続先」
user 「IBM iユーザー」 「IBM iパスワード」
binary
quote site namefmt 1
get  「IFSのパス」¥○○○.xls  C:¥□□□.xls
quit
```

ソース1

```
var
  FromName     : string;
  ToName       : string;
  err          : string;
  strFileName  : string;
  strFilePath  : string;
begin

  strFileName :=  Application.Exename;
  strFilePath :=  ExtractFilePath(strFileName);

  FromName    :=  '/「IFS上パス」/○○○.xls' ;
  ToName      :=  strFilePath  + '□□□.xls' ;

  //IFS上Excel→指定PC保管場所
  AS4001.Connect ;
  CoIFS1.Connect ;

  CoIFS1.CopyFileFromAS(FromName,ToName,true,true,false,0) ;

  CoIFS1.DisConnect ;
  as4001.Disconnect ;

end;
```

図2



ことである。

・解決その１

実際にその方法を考えると、最初に、IBM i の NetServer 機能（クライアント側から、IFS をネットワークドライブとして認識できる機能）を利用し、IBM i をファイルサーバーとして利用することが考えられた。しかし、IBM i のコストから考えると、この利用方法は非常にコストパフォーマンスの悪い解決方法になってしまう。

・解決その２

また、IBM i で作成した Excel をメールに添付し、エンドユーザーに配布する方法も考えられる。この方法は、個人別ログイン等でエンドユーザーと送信先メールアドレスを関連付ける必要はあるが、IBM i で Java を利用することにより簡単に実現可能である。

ただし、Excel をメールに添付して送信する方法はセキュリティに問題が生じるため、その帳票の性質を十分に考慮する必要がある。

・解決その３

そこで、次に挙げられる方法として、FTP で IFS の Excel を GET する方法が考えられる。確かにこの方法は、NetServer 機能と抱き合わせて考えることにより可能である。

とはいえ、IFS の Excel を GET する方法を知っている人は少ないのではと思う。参考までに以下にその方法を記しておく。

該当 IFS フォルダーに適切な制御を与え、ネットワークドライブ接続を確立することにより、Delphi をトリガーとして IFS の該当ファイルの取得は可能である。

なお、私の知る限りこのような処理を行っている人は知らないし、一般的にもこのような処理を行っている人は少ないと予想される。【図 1】

・解決その４

そこで私が行った内容を紹介する。当社では Delphi/400 を導入しており、IBM i とやり取りを行っている。一般的な使い方でもかなり有効に活用できるが、この Delphi/400 のコンポーネントの中に、IFS を操作できる CoIFS コンポーネントがある。

当然 Delphi では、IFS をネットワークドライブとして認識させることは可能である。そして信じられないかもしれないが、Delphi と FTP の連携では多少面倒な処理も、Delphi/400 では簡単に処理可能となる。【ソース 1】

しかし、ここで新たな問題点が発生した。検証当初は、拡張子を「xlsx」タイプで Excel を作成していた。このタイプで作成した場合、かなり処理時間がかかることがわかり、実運用には不向きだという結論に至った。

そこでゼロベースで再考した結果、拡張子を「xls」タイプで Excel を作成し、処理時間を比較してみた。その結果、処理時間としては約 1/7 ～ 1/9 くらい短いことが確認できた。幸いなことに、マイクロソフトの発表によれば、拡張子 xls タイプの Excel は Office2013 も対応していることが確認できた。そこで私は、拡張子 xls タイプで Excel を作成することとした。【図 2】

Delphi ＋ Officeエンジンによる Excel→PDFバッチ処理

ある意味、裏技的な処理を紹介する。多くのユーザーが利用している Office というソフトがある。ご存知の通り、Office2007 以降では PDF 変換機能が備わっており、一度該当ファイルを開き、保存する時に PDF を選択すれば簡単に PDF を作成することができる。

しかし、その該当ファイルを開くことなく、バッチ的に PDF を作成する方法はないだろうか？ うれしいことに、マイクロソフトはこれらの情報を公開している。（※）

この情報を参考に、Delphi で該当 Excel を PDF にバッチ変換できることを検証できた。参考までに、そのコーディングおよび実行結果を記しておく。【ソース 2】【図 3】

この検証結果をベースに、Java を利用して IFS に作成した Excel を、Delphi/400 を利用してクライアントに配布、そして IFS に作成した Excel をベースに「Delphi ＋ Office エンジンを利用して Excel → PDF バッチ処理」変換し、それをクライアントに配布することを可能とした。

※ Saving Workbooks to PDF and XPS Formats in Excel2007
http://msdn.microsoft.com/en-us/library/bb407651.aspx?cs-save-lang=1&cs-lang=vb#code-snippet-5

DelphiでExcelとPDFを作成しなかった理由

今回は、IBM i で Excel を作成する方法、およびその Excel から Office エンジンを利用してバッチ的に PDF を作成する方法を紹介した。しかし、Delphi でも Excel や PDF の作成は可能である。

では、なぜ Delphi で Excel や PDF を作成しなかったか？ その理由として、Delphi で Excel を作成する場合は OLE で作成するため、多少の処理時間がかかることが挙げられる。またクライアント側で処理するのではなく、他のプロシージャーとともに IBM i で処理することがよいのではと思う。

また、PDF 作成も無償サードパーティー製品で作成可能である。ただし、罫線の制御やフォントの制御を加味して帳票を作成するには、少し労力が必要である。他方、有償サードパーティー製品を利用するのは１つの手段であり、当然それなりのコストも発生する。

なお、今回の組み合わせによる帳票作成は、当社は Office2007 導入済みということもあり、追加投資を必要としなかった。

メッセージと課題

以上を踏まえたうえで、私は IBM i で Excel を作成することを決断した。そして、その Excel から PDF を作成することとした。

そこで危惧されることは「Java は敷居の高い言語？」と感じる人が多いはずである。しかし、私はそんなことはないと断言する。なぜなら、今回の Java はそんなに難しい考え方を行っていないこともあるが、RPG 畑で育ってきた私自身が実装している。本稿を読んだ人はぜ

ソース2

```
var

  F_ExcelApp  : Variant  ;
  F_ExcelVer  : Currency ;
  F_ExcelBook : Variant  ;

  F_ErrMessage: String   ;

const
  PDF_TYPE    = $00000000;
  QUALITY_TYPE= $00000000;

begin

  //ファイル存在チェック
  if not FileExists(F_FromExcel)  then
  begin
    MessageBox(HWND(Nil),'ターゲットファイルが存在しません。処理を中止します。',
               Nil, MB_OK) ;
    Exit ;
  end;

  //Ｅｘｃｅｌ操作開始
  F_ExcelApp  := CreateOleObject( 'Excel.Application' );

  //Ｅｘｃｅｌバージョン取得
  F_ExcelVer  := StrToFloatdef(F_ExcelApp.Application.Version,0.0);

  //Ｅｘｃｅｌ⇒Ｐｄｆ変換処理
  if F_ExcelVer >=12.0 then

    //Office2007以上
  begin

    F_ExcelBook :=F_ExcelApp.Application.WorkBooks;
    F_ExcelBook :=F_ExcelApp.Application.WorkBooks.Open(F_FromExcel);

    //変換処理
    try

      F_ExcelBook.ExportAsFixedFormat(
          PDF_TYPE,                 //  Type_                    : XlFixedFormatType;
          F_ToPdf ,                 //  Filename                 : OleVariant;
          QUALITY_TYPE,             //  Quality                  : OleVariant;
          True,                     //  IncludeDocProperties     : OleVariant;
          True ,                    //  IgnorePrintAreas         : OleVariant;
          EmptyParam,               //  From                     : OleVariant;
          EmptyParam,               //  To_                      : OleVariant;
          False,                    //  OpenAfterPublish         : OleVariant;
          EmptyParam                //  FixedFormatExtClassPtr: OleVariant)
          );

    except

      on E: Exception do
      begin
        F_ErrMessage:=e.Message;
        MessageBox(HWND(Nil),PChar(F_ErrMessage), Nil ,MB_OK) ;

      end;

   end;

  end

  else
  begin
    //Office2007以下
    MessageBox(HWND(Nil),
    'Office2007以降のバージョンがインストールされていません。処理を中止します。'
     Nil, MB_OK) ;

  end;

  //Ｅｘｃｅｌ操作終了
  F_ExcelApp.Application.quit;

end;
```

ひチャレンジしてほしい。

今後の課題としては、Excel → PDF変換に関してはクライアント側で処理する形となってしまったが、このプロセスをサーバー側で処理できないかを検討したいと思う。

エンドユーザーの評価と今後の展望

今回紹介した方法は、まだ部分的な業務の使用にとどまっている。当社では、今でもSPOOLFで帳票出力しているものが多々あることもあり、新しい手法については、利用しているエンドユーザーの評価としては上々である。

そして今回の導入により、各プロセスのテンプレートを作成することができたため、今後の業務に展開していく予定である。

最後に

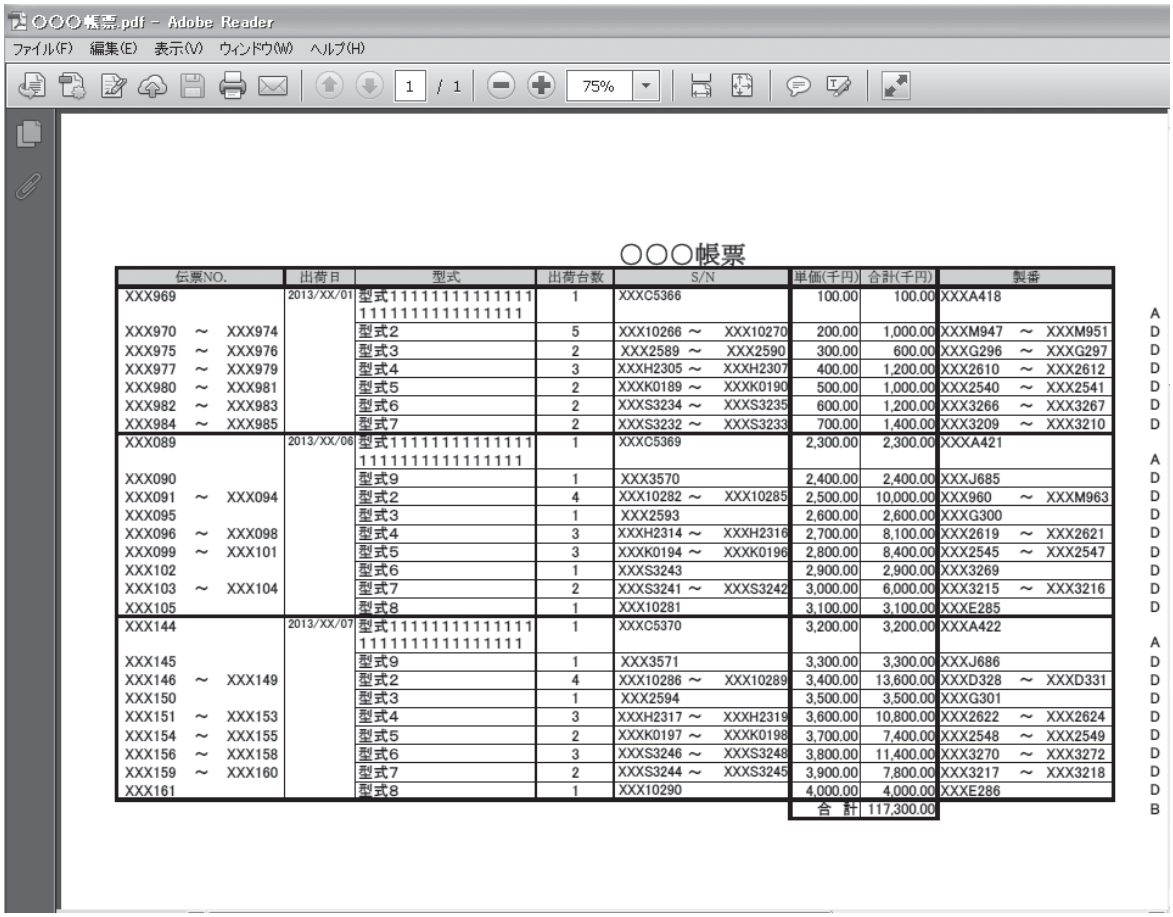
本稿を読んだ人は、今回紹介した方法により、IBM i ネイティブやネイティブ以外の資源の融合によってIBM i の資源をさらに有効に使えるという、1つの利用例があることが理解できたと思う。加えて、まだIBM i の資源を有効に使える可能性が潜んでいると考えられるため、今後もその可能性を模索していきたい。

ただし、問題は残っている。それはIBM i の情報は他のサーバーとは異なり、非常に入手しにくい点である。この状況を打開する1つの方法として、企業間を越えたナレッジの共有が考えられる。自企業から同地区の企業へ、そして全国の企業へとその輪を広めていくことが、今後の有効な手段の1つだと思われる。

ぜひ、本稿執筆をトリガーとして、全国の企業の方々とナレッジ共有を実現させて、Delphi/400およびDelphiとともに、今まで以上にIBM i を有効かつ効率的に活用していきたい。

M

図3



○○○帳票

伝票NO.	出荷日	型式	出荷台数	S/N	単価(千円)	合計(千円)	製番	
XXX969	2013/XX/01	型式1111111111111111111111111111111	1	XXXC5366	100.00	100.00	XXXA418	A
XXX970 ～ XXX974		型式2	5	XXX10266 ～ XXX10270	200.00	1,000.00	XXXM947 ～ XXXM951	D
XXX975 ～ XXX976		型式3	2	XXX2589 ～ XXX2590	300.00	600.00	XXXG296 ～ XXXG297	D
XXX977 ～ XXX979		型式4	3	XXXH2305 ～ XXXH2307	400.00	1,200.00	XXX2610 ～ XXX2612	D
XXX980 ～ XXX981		型式5	2	XXXK0189 ～ XXXK0190	500.00	1,000.00	XXX2540 ～ XXX2541	D
XXX982 ～ XXX983		型式6	2	XXXS3234 ～ XXXS3235	600.00	1,200.00	XXX3266 ～ XXX3267	D
XXX984 ～ XXX985		型式7	2	XXXS3232 ～ XXXS3233	700.00	1,400.00	XXX3209 ～ XXX3210	D
XXX089	2013/XX/06	型式1111111111111111111111111111111	1	XXXC5369	2,300.00	2,300.00	XXXA421	A
XXX090		型式9	1	XXX3570	2,400.00	2,400.00	XXXJ685	D
XXX091 ～ XXX094		型式2	4	XXX10282 ～ XXX10285	2,500.00	10,000.00	XXX960 ～ XXXM963	D
XXX095		型式3	1	XXX2593	2,600.00	2,600.00	XXXG300	D
XXX096 ～ XXX098		型式4	3	XXXH2314 ～ XXXH2316	2,700.00	8,100.00	XXX2619 ～ XXX2621	D
XXX099 ～ XXX101		型式5	3	XXXK0194 ～ XXXK0196	2,800.00	8,400.00	XXX2545 ～ XXX2547	D
XXX102		型式6	1	XXXS3243	2,900.00	2,900.00	XXX3269	D
XXX103 ～ XXX104		型式7	2	XXXS3241 ～ XXXS3242	3,000.00	6,000.00	XXX3215 ～ XXX3216	D
XXX105		型式8	1	XXX10281	3,100.00	3,100.00	XXXE285	D
XXX144	2013/XX/07	型式1111111111111111111111111111111	1	XXXC5370	3,200.00	3,200.00	XXXA422	A
XXX145		型式9	1	XXX3571	3,300.00	3,300.00	XXXJ686	D
XXX146 ～ XXX149		型式2	4	XXX10286 ～ XXX10289	3,400.00	13,600.00	XXXD328 ～ XXXD331	D
XXX150		型式3	1	XXX2594	3,500.00	3,500.00	XXXG301	D
XXX151 ～ XXX153		型式4	3	XXXH2317 ～ XXXH2319	3,600.00	10,800.00	XXX2622 ～ XXX2624	D
XXX154 ～ XXX155		型式5	2	XXXK0197 ～ XXXK0198	3,700.00	7,400.00	XXX2548 ～ XXX2549	D
XXX156 ～ XXX158		型式6	3	XXXS3246 ～ XXXS3248	3,800.00	11,400.00	XXX3270 ～ XXX3272	D
XXX159 ～ XXX160		型式7	2	XXXS3244 ～ XXXS3245	3,900.00	7,800.00	XXX3217 ～ XXX3218	D
XXX161		型式8	1	XXX10290	4,000.00	4,000.00	XXXE286	D
				合 計		117,300.00		B