

NEW

Visual Studio.NET/.NET Framework関連ツール新製品レビュー

PRODUCTS



WebReportCafe for .NET

PDFに特化した帳票システム構築ツール



精進湖計算機
立中 秀樹 TATENAKA, Hideki

問
合
先

株式会社テンアートニ

TEL : 03-5298-2924

FAX : 03-5298-2865

URL : <http://www.10art-ni.co.jp/>

MAIL : ws-sales@10art-ni.co.jp

Technology Tools

- Visual Basic .NET
- Visual C# .NET
- SQL Server 2000
- Oracle 9i
- Access 2002
- ASP.NET
- Internet Information Services
- Other:

Environment

ターゲットOS

Windows 2000/XP/Server 2003

対応開発環境

Visual Studio .NET 2003

クライアント (PDF)

Adobe Reader 4.0以上

価格

Standard Edition

(シングルスレッド対応) : 99,750円

ヘルプデスクサポート : 年間25,200円

Professional Edition

(マルチスレッド対応) : 価格は要問い合わせ

PDFで帳票出力

一昔前と違い、ほとんどの人がOffice系のアプリケーションくらいは使えるスキルを持つようになりました。筆者の受注する案件でも、帳票系アプリケーションの要求が減り、紙ではなく表計算アプリケーションのデータやXMLへの出力を求めるケースが増えつつあります。

ではまったく帳票の要求がなくなったかといえばそんなことはなく、いまだに伝票や請求書類、保管が必要な会計書類などの帳票出力は業務において「必須」です。業務上で変化があるとなれば、レーザープリンタや安価なカラープリンタの登場とともに、ドットインパクトプリンタを使う連帳の用紙を使う機会が減ったことでしょう。そのためか、レポートツールもページ単位のレイアウトを行なうものが主流となってきました。罫線やタイトルなどの固定印字データも含む印刷を行なう場合、印字データをそのままプリンタに送り出す方法もありますが、Acrobat Reader (PDF) やExcelなどの印刷機能を持つアプリケーションに対応したファイル形式で出力すれば「印刷」と

「印刷データの保存」という2つの役割を果たすことができます。

今回紹介する「WebReportCafe for .NET」は、PDFでの帳票出力に特化した、Windows/Webアプリケーション対応の帳票ツールです。WebReportCafe for .NET Designer(以下Designer)で帳票デザインを行ない、アプリケーションでデータを与えて帳票を出力するという、帳票ツールとしては基本的な利用方法でPDFを作成します。

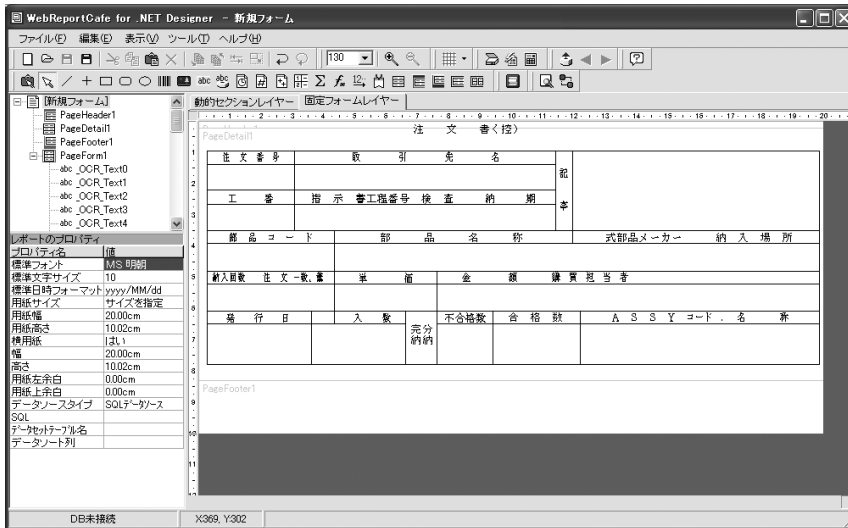
Designerの特徴

Designerは、単体のWindowsアプリケーションとして提供されています。

Designerの特徴として、レイヤー構造を持つことが挙げられます。Designerは、「固定フォームレイヤー」と「動的セクションレイヤー」という2つのレイヤーを持ち、罫線や表題などの固定的な情報と、動的に値が変わるデータ表示部とを分けてデザインすることができます(図1)。

固定フォームレイヤーに配置したデータは、生成されるすべてのページに合成されて表示されます。ヘッダーやフッターなどのセクションごとに出力

図1：WebReportCafe for .NET Designer



する必要がある場合は、動的セクションレイヤーにも固定情報を配置することができます。

そして、さらに特徴的な機能として「OCR機能」と「下絵機能」があります。これらは、すでに存在する帳票を元に帳票デザインを行なう際に威力を発揮します。

- ・ OCR機能：既存の伝票などをスキャナで読み込んでBMP形式で保存し、そのデータをDesignerに指定して解析を行なうと、データ上にある文字列と罫線を固定フォームレイヤーに配置する
- ・ 下絵機能：BMP ファイルを2つのレイヤーの下に表示する

続いて、実際にこれらの機能を利用してサンプルアプリケーションで使用する伝票のデザインを行なってみましょう。

帳票のデザイン

OCR機能を使うには、まず既存の帳票をスキャナで読み込み、白黒2値のBMPファイルを作成する必要があります。

■帳票イメージのスキャン

WebReportCafe for .NETは、直接スキャナから読み込む機能を持たないので、ここでは伝票イメージを取り込むためにOffice 2003付属の「Office Document Imaging」を使用しました^[注1]。

どのアプリケーションで読み込みを行なう場合もスキャナメーカーの用意したTWAINドライバ経由で取り込みを行なうのが一般的で、取り込みに用いるパラメータなどはTWAIN側で指定したことになります。今回は、EPSONのスキャナ (GT-700U) を使用し、TWAINドライバはEPSON TWAIN3を用いました。EPSON TWAINではいろいろなイメージタイプのデフォルト設定があり、そのなかにOCR用の設定もあります。はじめにこの設定で試したところ、文字認識はそこそこ良好でしたが、罫線をまったく認識してくれません。そこで、イメージタイプを線画に設定し、DesignerのOCR機能 (図2) が最適化されている400dpiで取り込むようにしてスキャンを行ない、Designerでの解

注1) Office Document Imaging では、画像ファイルをTIFF形式でしか出力できないため、Windows 付属のペイントに画像をカット&ペーストして画像イメージをBMP形式で保存しました。

図2：DesignerのOCR機能



析処理がもっとも機能するように、2値化の閾値を手動設定しながらスキャンと解析を繰り返してみました。前掲の図1がOCR機能で解析後、生成された伝票デザインです。

もっとも結果が良かった伝票デザインの背景に、下絵機能で元の伝票のカラーキャン画像を表示した画面が図3になります。OCR機能で認識に成功した罫線や文字がほぼ正確に配置されているのがわかります。

■データオブジェクトの配置

罫線の足りない部分を足すなど整形した後、誤字や文字の大きさを修正すれば、固定フォームレイヤーのデザインは完了です (図4)。

続いて、動的セクションレイヤーにデータオブジェクトを配置してゆきます。今回作成する伝票は、一品一葉のタイプなので、ページヘッダー (Page Header1) にデータオブジェクトを配置します。Detail (PageDetail1) は、複数行にデータが表示される場合に使用します。

Designer上のPageHeader1の領域を広げて、データオブジェクトを配置してゆきます (図5)。今回のサンプルは、注文番号、注文日、取引先名、記事、部品コード、部品名、注文数量、単価、金額にデータを印刷することになります。