

FIELD NOTES: 書を持って街へ出よう [AI](#) [RSS](#) [Twitter](#)

カテゴリー

[ニュース](#)[製品情報](#)[技術調査](#)[プログラミング](#)[開発環境](#)[システム運用](#)[IP電話](#)[HOWTO](#)

プロフィール

fet PLUS

合同会社フィールドワークス(<http://www.field-works.co.jp/>)プログラマ兼代表

最新タイトル

[\[システム運用\] Jimdoでソースコードを整形して表示するならPygmentsが便利](#)

[\[技術調査\] IPAmj明朝フォントの研究](#)

[\[プログラミング\] OCamlで作成した拡張モジュールでセグメンテーション違反発生](#)

[\[IP電話\] 現在契約できるプロバイダフリーのIP電話](#)

[\[システム運用\] VMware ESXi 5.0でVNC接続を利用する方法](#)  
[\[HOWTO\] iCalとiPhoneをTODO込みで同期をとる方法](#)

[\[プログラミング\] jarファイルとjniライブラリのインストール方法](#)  
[\[プログラミング\]](#)

[Windows用PHP拡張モジュールを最短でビルドする方法](#)

[\[製品情報\] \[ニュース\] LL言語向けPDF帳票ツールField ReportsがJavaに対応しました](#)

[\[技術調査\] オープンソースのExcel出力ライブラリ](#)

[<前の日](#) | [次の日>](#) | [記事を書く](#) | [記事を書く](#)

2011-06-29

CSVから帳票を生成するツールを作ってみました [💬](#) [📧](#) [📧](#) [📧](#)[製品情報](#) | [編集](#) | [編集](#)

弊社のLL言語用PDF帳票ツールField Reportsを使って、CSVファイルを元にPDF帳票を生成するツールを試作してみました。同様の処理はMS WordとExcelの「差し込み印刷」機能を使ってもできますが、このツールには以下の特長があります。

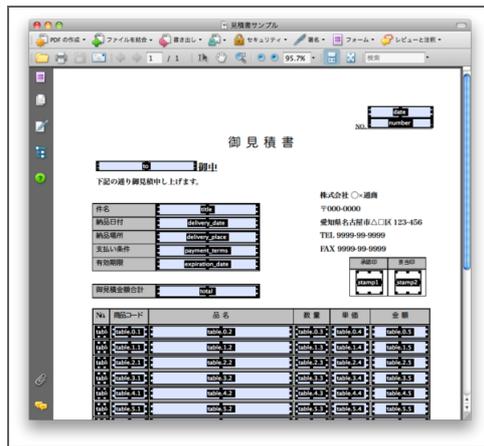
- 雛形の作成手段がWordに限定されない（PDF化できる文書ならなんでも使える）。
- Linux, Mac OS X環境で動作する（逆に、現在のところWindows環境では動作しません）。
- 他のツールと組み合わせて処理を自動化するのが比較的簡単（例えば、どこかのWebサイトから取得したHTMLからデータを抜き出して、レポートを自動生成するなど）。
- サーバサイドでの運用も問題ない（マイクロソフトでは、Officeをサーバサイドで使用することはお勧めしていないようです：[Officeのサーバサイドオートメーションについて](#)）。

## テンプレートの準備

テンプレート（雛形）となるPDFファイルを用意します。

今回は、OpenOffice.orgで請求書のフォーマットを作成し、「PDFとしてエクスポート...」でPDF化しました。

次に、作成したPDFをAdobe Acrobatで開き、テキストを流し込みたい位置にフォーム・フィールドを配置していきます。



## CSVファイルの準備

PDFテンプレートに流し込むデータを作成し、CSV形式のファイルとして保存します。

CSVファイルの1行目はヘッダ行として使用し、データは2行目以降に置きます。

ヘッダ行の見出しは、PDFテンプレートに配置したフィールド名と一致させます。

テキストの文字コードはSJISとします。

```
"date","number","to","title","delivery_date","delivery_place","payment_terms","expiration_date","table.0.0","table.0.1","table.0.1","平成23年1月22日","10R0001","△△△惣業株式会社","肉じゃがの材料","平成23年1月22日","貴社指定場所","銀行振込","発行から3ヶ月以内",1,"N00"
```

## CSV→PDF変換プログラムの作成

CSV→PDF変換プログラムをPythonで作成しました。実験的な位置づけのツールなので、エラー処理などはほとんど考慮していません。

基本的にはヘッダ行の見出しとデータ行の値を単純に組み合わせて辞書の要素を作成しているだけですが、2次元テーブルの要素名を"table.0.0"の形式で表現していて、

Field Reportsで利用するためには、

```
{'table': {'0': {'0': '値'}}}
```

のような形式に変換する必要があり、dset()関数でその処理を行っています。

```
#!/usr/bin/env python
# coding: utf-8

import sys
```

```

import csv
from field import reports

param = {
    "template": [
        {"*": {"src": "./mitumori.pdf", "rows": 10}}
    ],
    "style": [
        {"*.stamp1": {"icon": "./stamp.png"}}
    ]
}

def dset(d, keys, value):
    if len(keys) == 0:
        return value
    else:
        if d.has_key(keys[0]):
            dset(d[keys[0]], keys[1:], value)
        else:
            d[keys[0]] = dset({}, keys[1:], value)
    return d

def read_csv(fname):
    result = []
    reader = csv.reader(file(fname, "rb"))
    header = reader.next()
    for row in reader:
        item = {}
        for key, value in zip(header, row):
            if key and value:
                keys = unicode(key, 'sjis').split('.')
                dset(item, keys, unicode(value, 'sjis'))
        result.append(item)
    return result

if __name__ == "__main__":
    if len(sys.argv) == 3:
        context = read_csv(sys.argv[1])
        param['context'] = context
        reports.set_log_level(3)
        reports.render(param, sys.argv[2])
    else:
        print("usage: %s <csvfile> <outfile>" % (sys.argv[0], ))

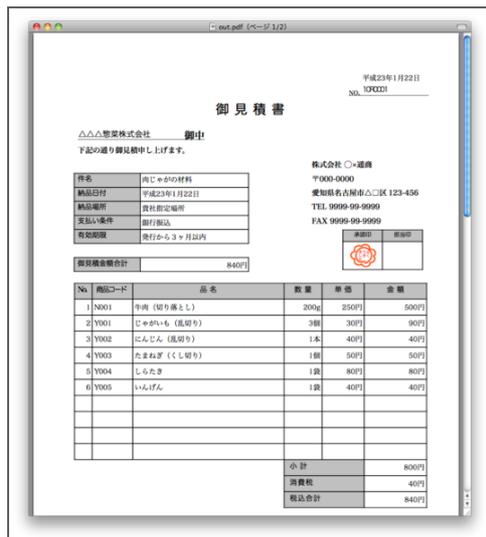
```

## 処理結果

使い方は、コマンドラインから以下のコマンドを実行します。

```
$ python csv2pdf.py data.csv out.pdf
```

作成されたPDFを以下に示します。



今回の例では1ページの帳票を作成しましたが、データが複数行あれば複数ページのPDFが作成されます。

いいね!

いいね!

Permalink | コメント(0) | トラックバック(0) | 11:36