

日本JasperServerユーザ会 (JJSUG)
勉強会発表資料

WebアプリでJasperReportsLibraryを使った事例

2013年1月27日
(株)アップセル 吉田悟

1:アプリの概要

- ・小さなウェブアプリアプリです。
- ・15店舗の売店・飲食店の売上集計・管理システム
- ・各店舗が、CSV形式になっている売店のレジデータを本社にアップロードする。
- ・1日1回、毎日アップロードされるデータを集計し、約10帳票の管理資料を作成する。
- ・管理資料はエクセルシートに出力する。

← JasperReportに感謝

(アプリの画面イメージ)



アプリの動作イメージ

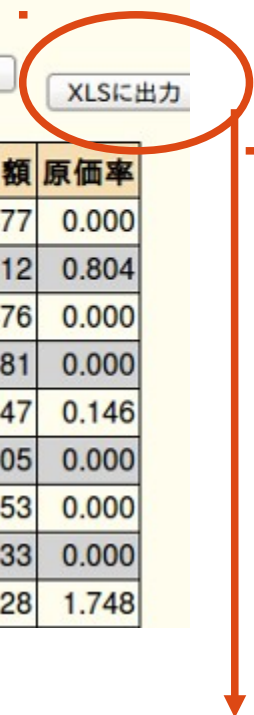
営業日報システム
 Smile making System
 ログアウト
logged in

(画面イメージ)

売上一覧表

所属: レストランA 対象日付(下限): 2012/09/01 対象日付(上限): 2012/09/30
 検索
XLSに出力

商品コード	商品名	売単価	仕入単価	売上数	売上金額(税抜)	原価	粗利額	原価率
6411000000404		0	0	1	477	0	477	0.000
6411000000763	ハーブキャンディー	480	230	2	572	460	112	0.804
4988375000596		0	0	1	476	0	476	0.000
6411000000251		0	0	1	381	0	381	0.000
9981234563848	ソフトクリーム	300	92	4	2,515	368	2,147	0.146
4985902025817		0	0	1	505	0	505	0.000
4909900400254		0	0	1	553	0	553	0.000
4977103905012		0	0	1	133	0	133	0.000
6411000000749	福袋1000000円	1,000	500	2	572	1,000	-428	1.748



売上一覧表

(Excelシート)

店舗 : レストランA

集計期間 : 2012/09/01 から 2012/09/30 まで

null

商品コード	商品名	販売単価	仕入単価	販売個数	販売金額	原価	粗利額	原価率
6411000000404		0.00	0.00	1	477	0	477	0.000
6411000000763	ハーブキャンディー	480.00	230.00	2	572	460	112	0.804
4988375000596		0.00	0.00	1	476	0	476	0.000
6411000000251		0.00	0.00	1	381	0	381	0.000

2:動作環境(ソフトウェア)

<サーバー>

OS :Ubuntu12.04

DB :PostgreSQL9.1

アプリケーション開発言語:Java7(OpenJDK1.7)

アプリケーションサーバー:Glassfish3.1.2CommunityEdition(JavaEE6)

view :JSF2(JavaServerFaces)+HTML+CSS+PrimeFaces3.2

IDE :NetBeans7.2

レポート(Exceシート)作成ツール:iReport4.7 + JasperReport

<クライアント>

ブラウザ :IE or Firefox

サーバー環境は全てOSS

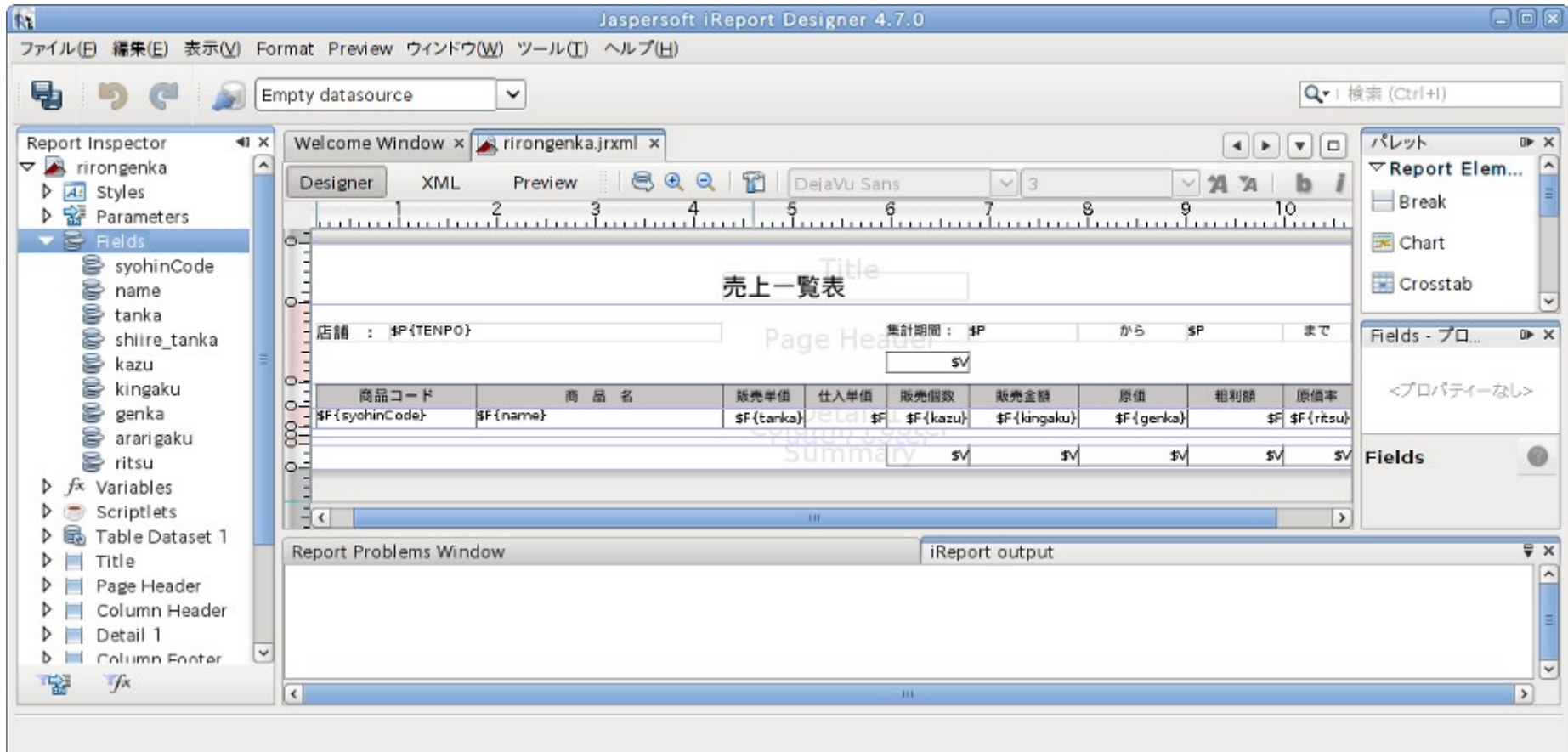
(ライセンスの難しいところはよくわかりませんが・・・)

3: 帳票書式作成

3-1: 作成物の概要

レポート書式作成ツールiReportを使ってExcel表の書式を作成

<<今回作成した帳票レイアウト>>

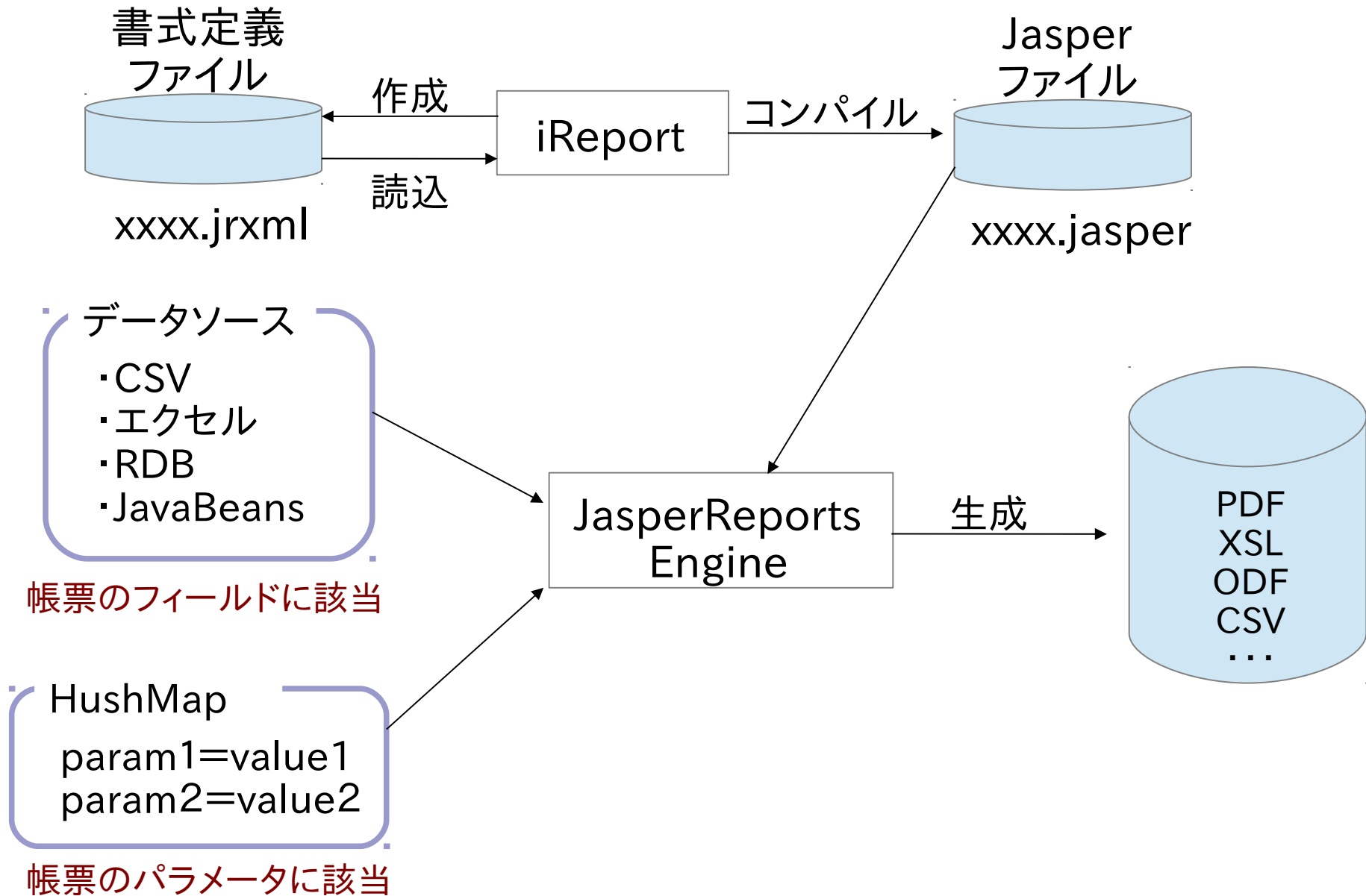


・おおよその作業の流れ

レポートインスペクターでフィールド(Field)とパラメータ(Parameters)と変数(Variables)を作る。
→デザインウィンドウに配置する。 →コンパイルする。 →実行環境に配備する。

3: 帳票書式作成

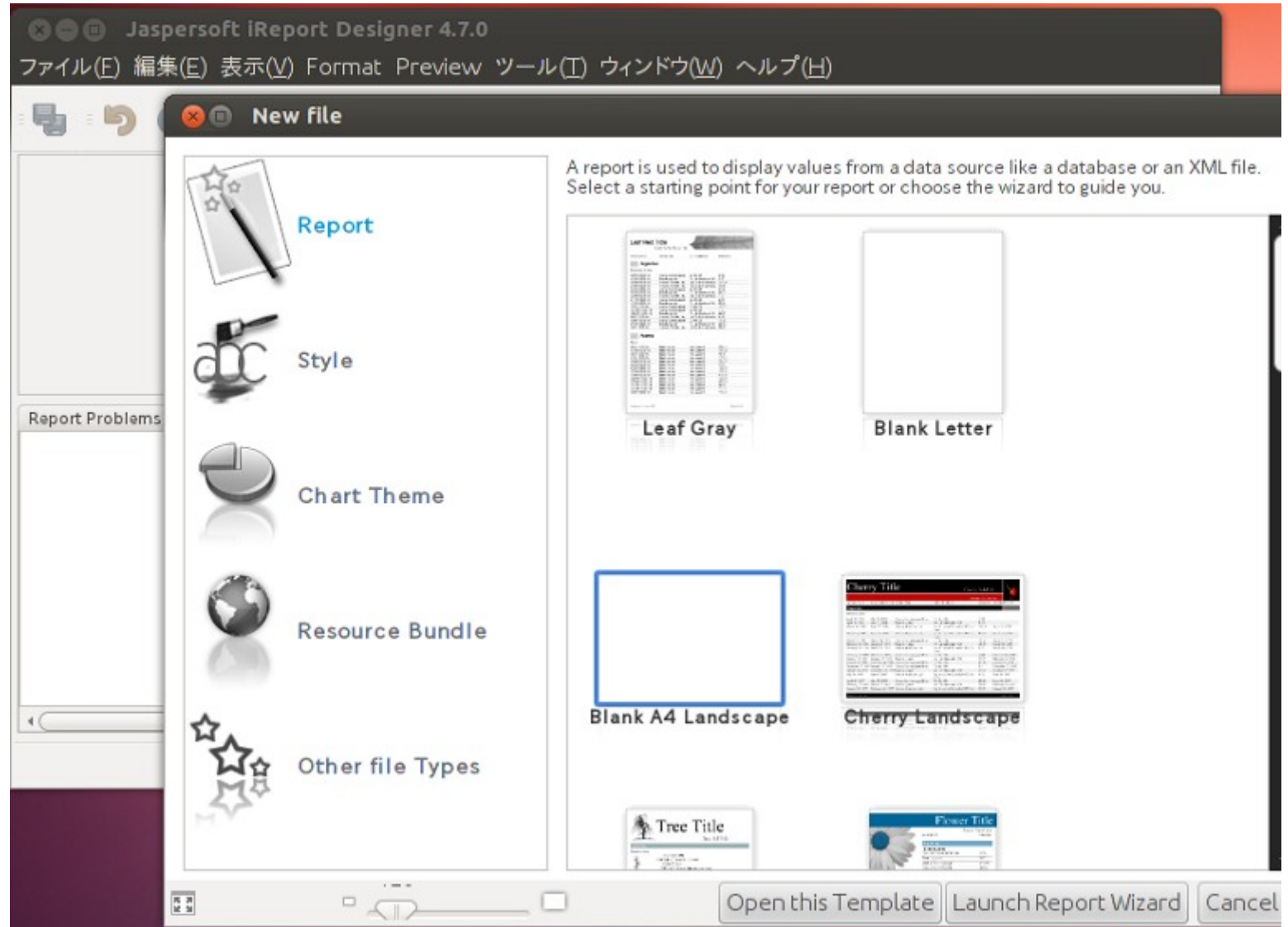
3-2: iReportとJasperReportの関連



3: 帳票書式作成

3-3: 書式ファイルの新規作成

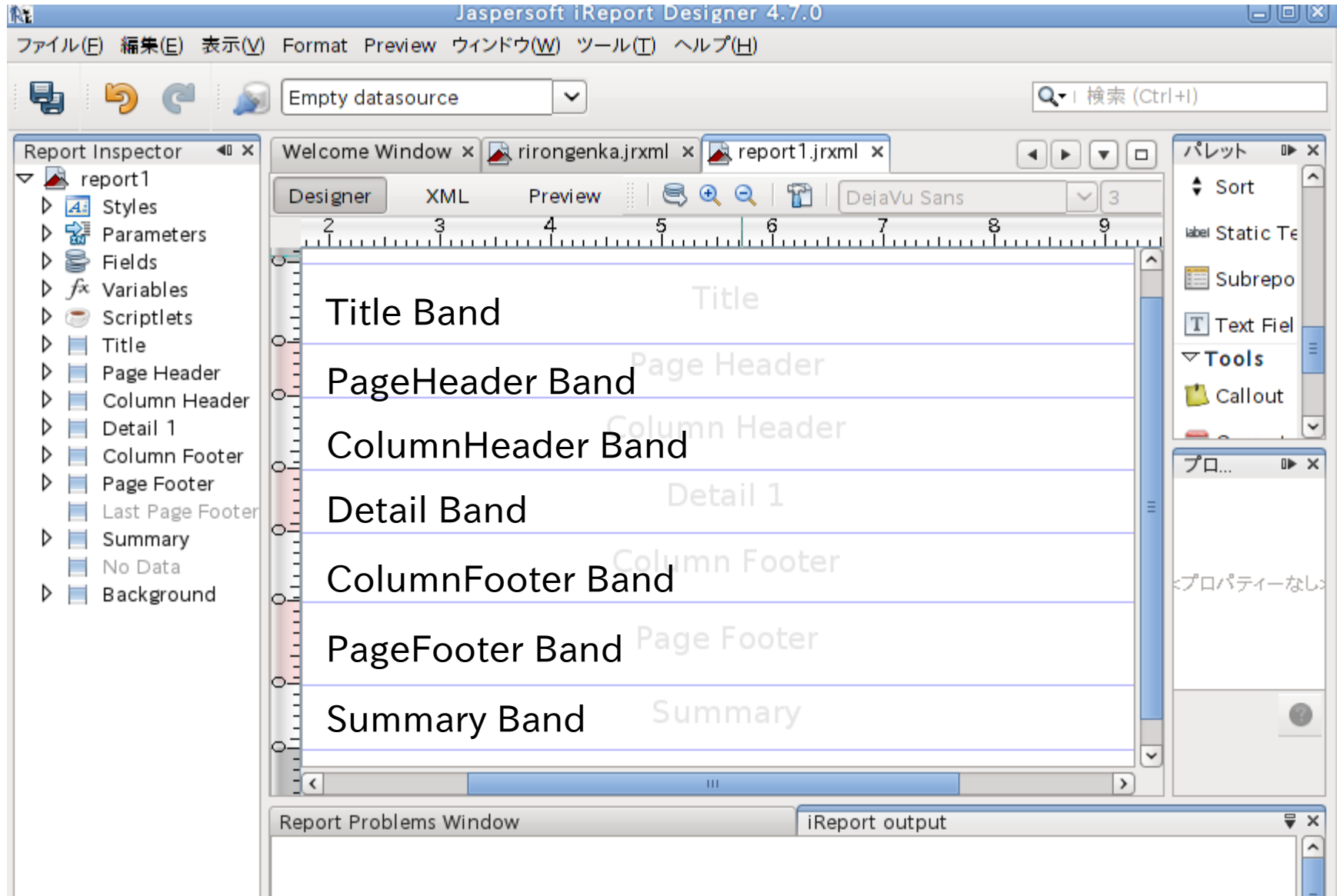
ファイル
↓
New
↓
Blank
A4Landscape
↓
Open
thisTemplate



3: 帳票書式作成

3-3: 書式ファイルの新規作成

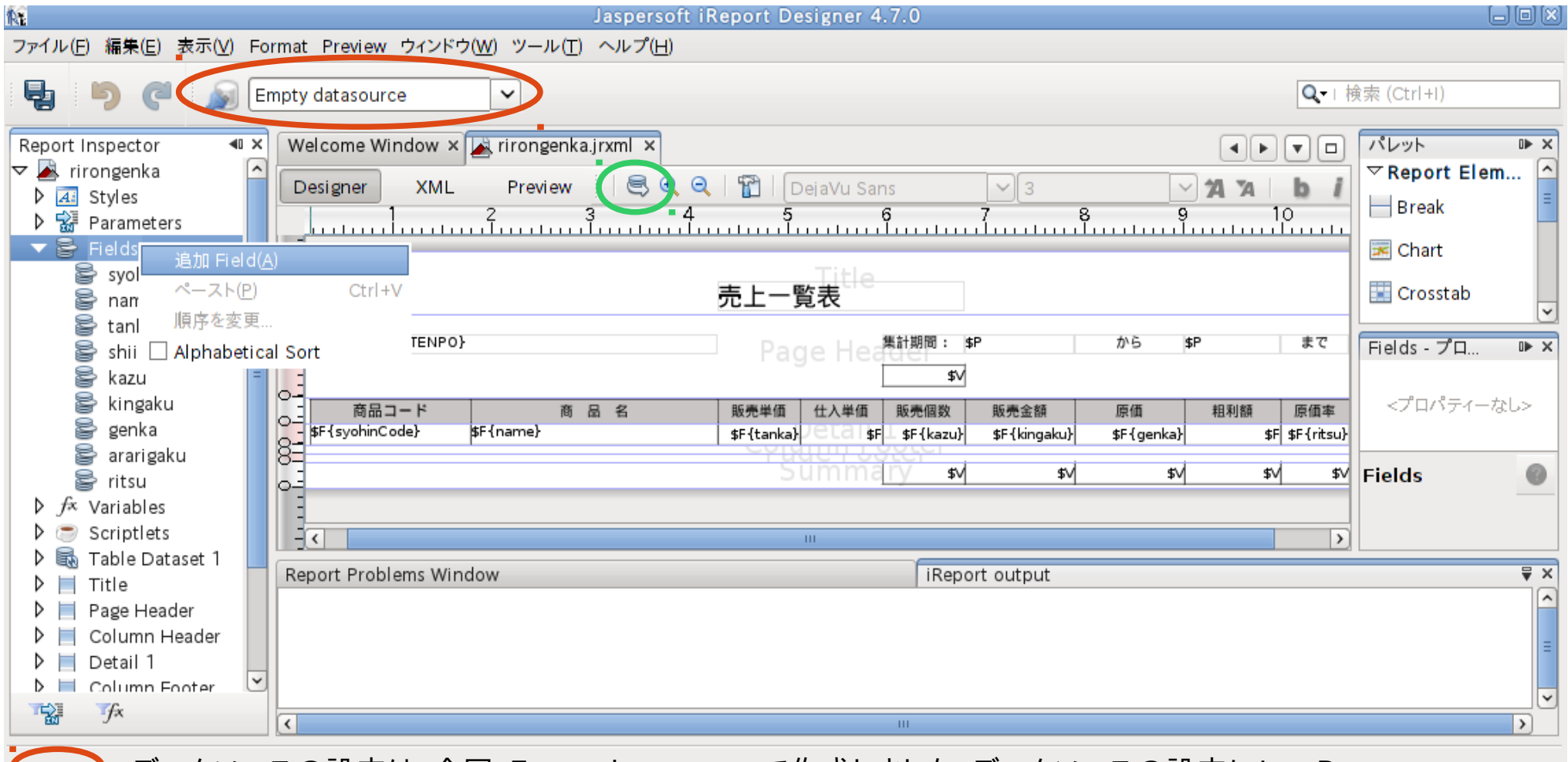
iReportのデザイン画面はバンドと呼ばれる帯域に分かれていてバンドによって印刷されるタイミング・回数が異なります。



3: 帳票書式作成

3-4: フィールドの作成

レポートインスペクタ画面で、Fieldを右クリックしてフィールドを追加します。
今回のデータソースはJavaBeansのCollectionですので、フィールド名はJavaBeansのプロパティ名と一致させました。デザインウィンドウの右にあるプロパティウィンドウで型(FieldClass)もJavaの型に合わせました。



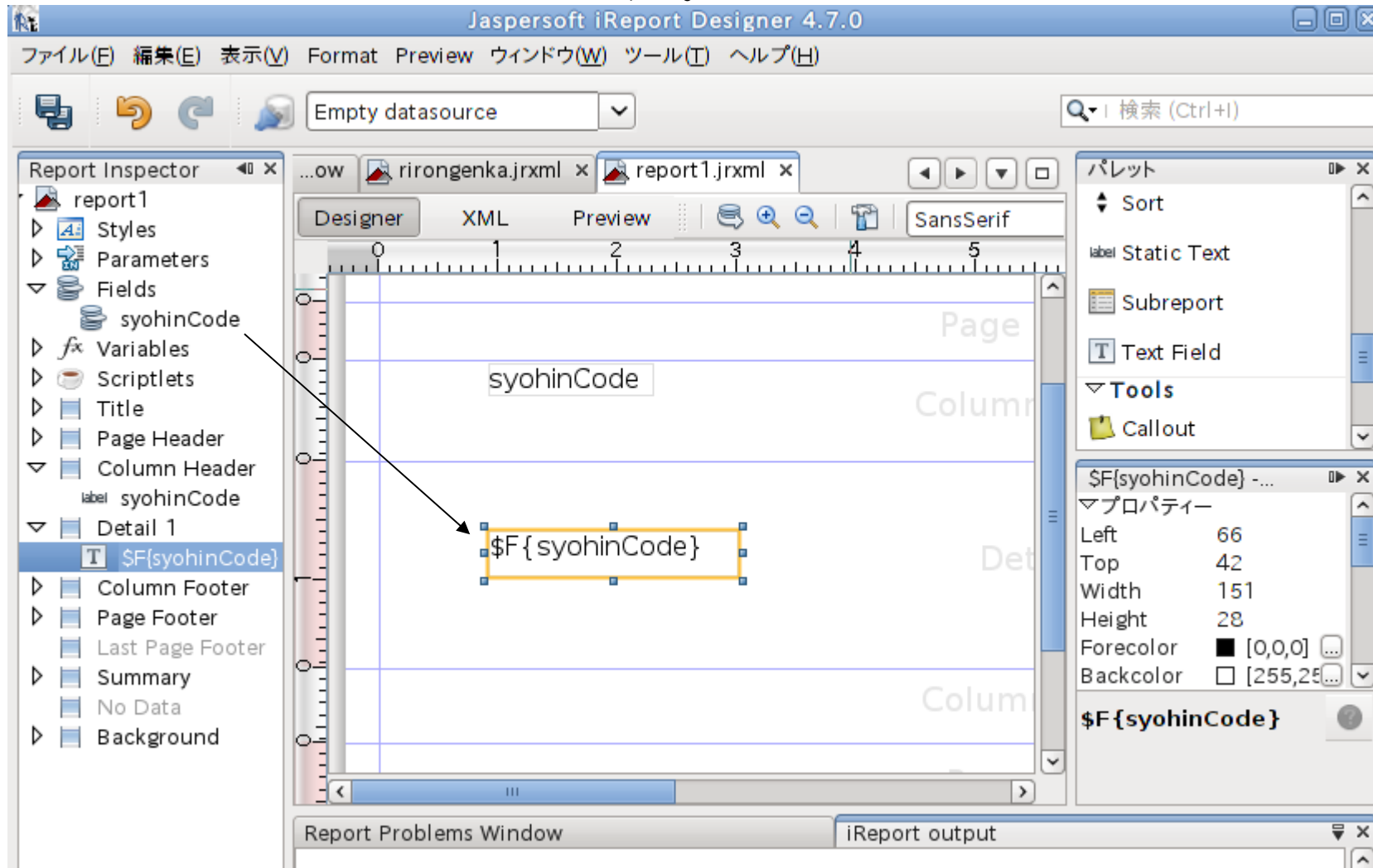
○ : データソースの設定は、今回、Emptydatasourceで作成しました。データソースの設定にJavaBeans set datasourceがあるのですが、意味も設定値もわかりません。どなたかご存知であれば教えてください。

○ : ReportqueryにもJavaBeans datasourceがあり、JavaBeansのフルパスを設定するとあるのですが、認識させることができませんでした。結果、ここも何も設定していません。

3: 帳票書式作成

3-4: フィールドの作成

Detail Bandにドラッグ&ドロップする。



Fieldsにある項目のプロパティとDetailにある項目は、同じ項目名でもプロパティは別なので念の為。

4:プログラム作成

4-1:JSF

JSFとは:HTMLの中にJSFのタグを書き、JSFタグを解釈することで動的にHTMLを生成するフレームワーク。JSFタグの中には管理Bean (Javaのクラス)とそのメソッド名を書いてその戻り値を表示できる。

JSFの一部 (〇〇〇〇.xhtml)

```
<ui:define name="selectConditionsRight">  
  <h:form>  
    <h:commandButton actionListener="#{genkaReportController.xls}" value="XLSに出力"/>  
  </h:form>  
</ui:define>
```

管理Bean名

メソッド名

JasperReportsのLibraryを使って
エクセルのシートを生成するメソッドです。

4:プログラム作成

4-2:管理Bean

管理Beanとは:

ブラウザから入力されたデータを受け取り、バックエンドの機能に対して処理を要求する。またバックエンドの機能からデータを入力しブラウザへ返送する為のメソッドを持つJavaクラス。JSFエンジンの管理下にある。

管理Beanの一部

ブラウザ(JSF)から呼ばれる

```
public void xls(ActionEvent actionEvent) throws JRException, IOException{
    String jasperName = "rirongenka.jasper";           //レイアウト定義ファイル
    String outputName = "rirongenka.xls";
    ReportFileFactory rff = new ReportFileFactory();
    rff.createFile(jasperName, listGenkaReport, outputName, createHashMap());
}
public HashMap createHashMap(){
    SimpleDateFormat sdf = new SimpleDateFormat("yyyy/MM/dd");
    String toDate = sdf.format(selectConditions.getToDate());
    Integer intTenpoCode = Integer.parseInt(selectConditions.getSelectedTenpo());
    Tenpo tenpo =.ejbFacade.find(intTenpoCode);
    HashMap hMap = new HashMap();
    hMap.put("TENPO", tenpo.getName());
    hMap.put("FROM_DATE", selectConditions.getFromDate());
    hMap.put("TO_DATE", toDate);
    return hMap;
}
```

データを添えて処理を依頼

JavaBeansのCollection

iReportのパラメータ名 値

4:プログラム作成

4-3:ビジネスロジック

JasperReportsのLibrayを使ってExcelシートを生成します。

```
public void createFile(String jasperName, List reportDataList, String outputName, HashMap hMap) {
    String jasperFilePath = FacesContext.getCurrentInstance().getExternalContext().getRealPath( "/resources/jasper/" +
jasperName);
    try{
        JRBeanCollectionDataSource beanCollectionDataSource=new JRBeanCollectionDataSource(reportDataList);
        jasperPrint = JasperFillManager.fillReport(jasperFilePath, hMap,beanCollectionDataSource);
        try{
            HttpServletResponse
httpServletResponse=(HttpServletResponse)FacesContext.getCurrentInstance().getExternalContext().getResponse();
            httpServletResponse.addHeader("Content-disposition", "attachment; filename=" + outputName);
            ServletOutputStream servletOutputStream=httpServletResponse.getOutputStream();
            JRXlsExporter xlsExporter=new JRXlsExporter();
            xlsExporter.setParameter(JRExporterParameter.JASPER_PRINT, jasperPrint);
            xlsExporter.setParameter(JRExporterParameter.OUTPUT_STREAM, servletOutputStream);
            xlsExporter.exportReport();
            FacesContext.getCurrentInstance().responseComplete();
        }catch(Exception e){
            logger.warning("エクセルデータを返送する際にエラーがありました" + e.getClass().toString());
            JsflUtil.addErrorMessage("エクセルデータを返送する際にエラーがありました");
            JsflUtil.addErrorMessage(e.getClass().toString());
        }
    }
}
```

5:むすびに

今回のシステムは、本当に数えきれない多数のオープンソースソフトウェアのおかげで構築することができています。

Ubuntu、PostgreSQL、OpenJDK、Glassfish、HTML、CSS、PrimeFaces、NetBeans、iReport、JasperReport等などのコミュニティーの皆さんが、先人の成果をソースコードその他の形で継承し、維持・発展させていく情熱と努力に敬意と感謝を表します。

また上記の成果物はそれを構成する自由なライセンスのライブラリーによって成り立っています。まさに数えきれない多くの方々の貢献の成果を利用させて頂いており、それらに関わる、また関わってきた全ての方々に感謝します。

先人の成果を継承・発展させていく活動の中で、ソフトウェアを作成・提供する方々が知的欲求の充足、周りの人からの感謝、報酬などの形で何らかの喜びが得られることと、利用者とソフトウェアを共有し、対話することで皆が喜びを得られることを願っています。

2013年1月27日
(株)アプセル 吉田 悟
syoshida@appcel.jp