

第

1

章

iReport で 簡単帳票作成

PDF/Excel/HTML 対応の GUI 帳票エディタ兼実行環境

(株)ビーブレイクシステムズ <http://www.bbreak.co.jp/>
横井 朗 YOKOI Akira yokoi@bbreak.co.jp / 鹿取 裕樹 KATORI Hiroki katori@bbreak.co.jp

はじめに

業務システム構築の際に、請求書やレポートなどの帳票出力を求められる機会は多いと思います。

PDF なら Apache XML プロジェクトの FOP, Word・Excel なら Jakarta POI と、出力フォーマットに応じて個々のライブラリを習得し、それぞれに合わせて定義ファイルの作成やコーディングをするのはなかなかの手間です。そこで今回は、同一定義ファイルから PDF, HTML, Excel などでの帳票作成が可能なオープンソースの GUI エディタ兼、実行環境である iReport を紹

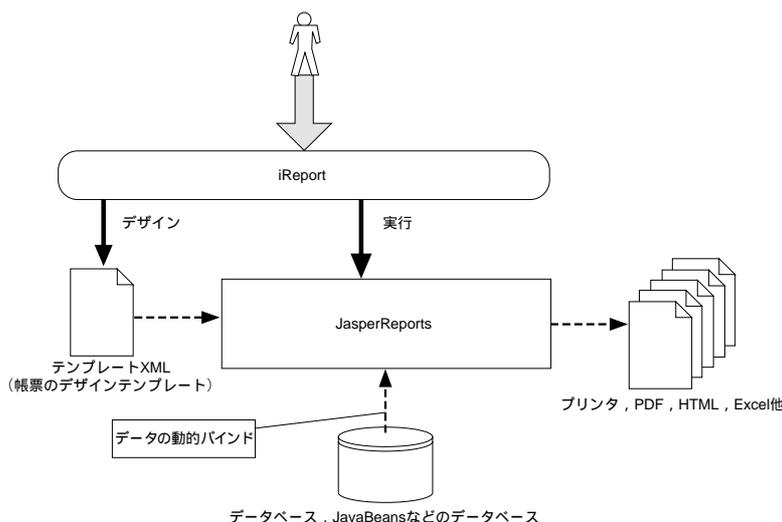
介します。

本稿では、iReport のウィザード機能を使用して帳票作成の流れを確認した後、サンプル請求書を作成しながら iReport の使用方法を解説していきます。

iReport とは

iReport は、JasperReports^{注1} のデザイン・実行環境で、SourceForge.net で開発が行われているオープンソースのツールです^{注2}。図1に示すように、iReport でテンプレートXMLを作成し、iReport 上から実行することで、JasperReports にテンプレートXMLが読み込

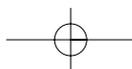
図1 iReport による帳票出力のしくみ



注1) JasperReports は、Java で記述された XML ファイルの定義ファイルから帳票を作成するオープンソースのライブラリです。JasperReports の詳細は JasperReports 開発ページ (<http://jasperreports.sourceforge.net/>) を参照してください。また、本誌 Vol.15 連載記事「Java オープンソース探索」でも紹介されています。

注2) ライセンスは GNU General Public License (GPL) を採用しています。





iReport で簡単帳票作成

第 1 章

PDF/Excel/HTML対応の
GUI帳票エディタ兼実行環境

まれ、帳票が出力されます。また、JasperReports および作成したテンプレートXMLを取り込めば、任意のアプリケーションから簡単に帳票機能を利用できます（本稿後半部でWebアプリケーションへの組み込み方法を紹介しています）。

バージョンについて

現在、iReportにはJavaバージョン（iReport-0.2系）とWindowsバージョン（iReport-0.1系）が存在します。Windowsバージョンのほうが安定しており、操作性も良いですが、Javaバージョンでのみ使用可能な機能も存在します。また、今後はJavaバージョンのみバージョンアップされる予定のため、これから使用を開始するのであればJavaバージョンをお勧めします。

なお、本稿ではiReport 0.2.2（執筆時の最新バージョン）を使用して説明します。バージョンアップその他の詳細情報はiReport 開発ページ（<http://ireport.sourceforge.net/>）を参照してください。

なお、本稿は以下の環境で執筆しました。

OS : Windows 2000 Professional

iReport : バージョン0.2.2

JasperReports : バージョン0.5.0

JDK : Java 2 SDK 1.4.1

Ant : バージョン1.5.4

PDFビューア : Acrobat Reader 6.0

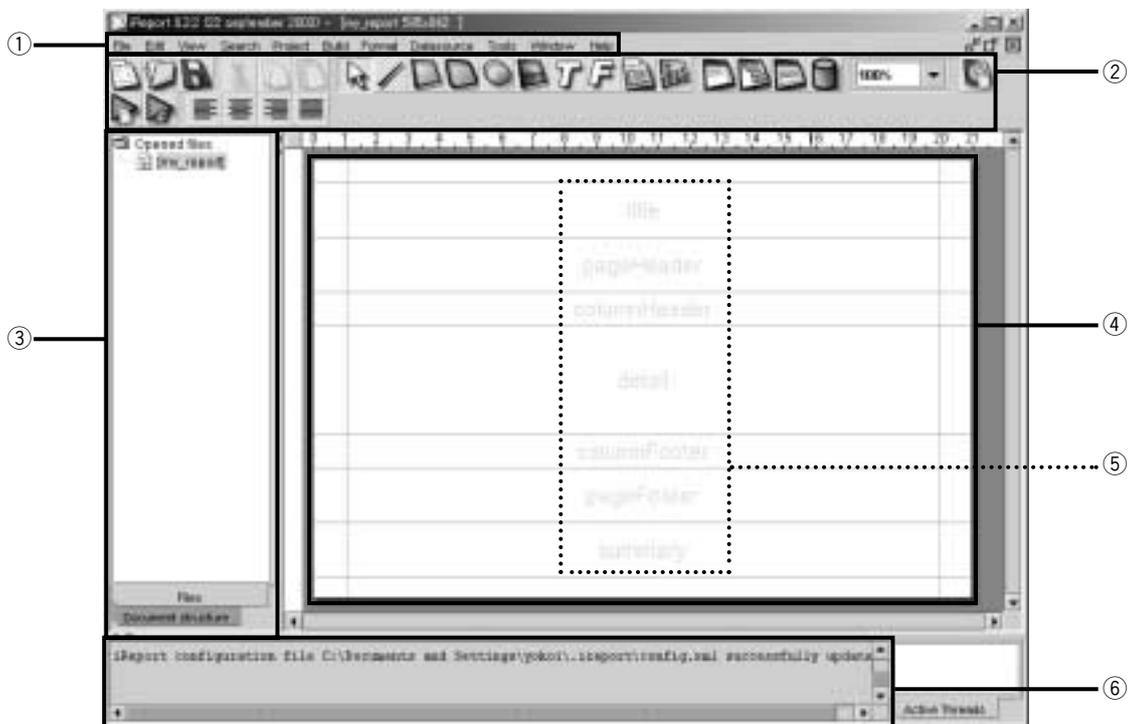
商用製品との比較

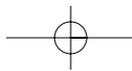
iReportは、商用製品と比較すると多少使い勝手は落ちますが、ご使用いただくと、要件次第では十分に利用できることが確認いただけるのではないかと思います。実際、筆者の担当したプロジェクトでもiReportを採用し、特に問題なく使用できています。また、現バージョンではいくつか不具合も見られるため、一部機能については直接デザインテンプレート（以降、テンプレートXML）を編集する必要がありますが、そこは今後のバージョンアップに期待しましょう。

基本操作

それではまず、iReportの画面および基本的な操作方法について説明します（図2）。iReportのメインウ

図2 iReportのメインウィンドウ





Open Sourceで
業務システム #4

iReport Jakarta POI JFreeChart
PDF, Excelデータ, グラフ出力アプリの開発

表1 アイコン一覧

| アイコン | アイコン名称 | 機能概要 |
|------|-------------------|---------------------------|
| | 保存アイコン | 編集中のテンプレートXMLを保存します。 |
| | コンポーネント選択アイコン | 編集エリアに配置されたコンポーネントを選択します。 |
| | 線分アイコン | 編集エリアに線分を配置します。 |
| | 四角形アイコン | 編集エリアに四角形を配置します。 |
| | 四角形アイコン(角丸) | 編集エリアに角が丸くなった四角形を配置します。 |
| | 円形アイコン | 編集エリアに円を配置します。 |
| | イメージアイコン | 編集エリアにイメージデータを配置します。 |
| | テキストアイコン | 編集エリアに静的テキストを配置します。 |
| | フィールドアイコン | 編集エリアにフィールドを配置します。 |
| | フィールド設定アイコン | フィールド(後述)設定ウィンドウを開きます。 |
| | データソース設定アイコン | データソース設定ウィンドウを開きます。 |
| | コンパイルアイコン | テンプレートXMLをコンパイルします。 |
| | 実行アイコン(空データソース使用) | 空のデータソースを使用して帳票出力を実行します。 |
| | 実行アイコン | データソースを使用して帳票出力を実行します。 |

表2 バンド一覧

| バンド | 概要 |
|--------------|----------------------|
| background | 背景描画領域 |
| title | タイトル描画領域 |
| pageHeader | ページヘッダ表示領域 |
| columnHeader | カラムヘッダ(detail上部)表示領域 |
| detail | 詳細表示領域(繰り返し部) |
| columnFooter | カラムフッタ(detail下部)表示領域 |
| pageFooter | ページフッタ表示領域 |
| summary | サマリ表示領域 |

ウィンドウは、以下のような構成になっています。

- ①メニュー：iReportの各種操作、設定を行います。
- ②各種アイコン：帳票デザインや実行時に使用するアイコンです。代表的なアイコンについては表1を参照してください。これ以降の説明では、表1のアイコン名称を使用して説明します。
- ③プロジェクトブラウザ：現在開いているファイルの一覧が表示されます。また、タグを切り替えることでドキュメント構造が参照できます。
- ④デザイン編集エリア：帳票のデザインを行うエリアです。
- ⑤バンド：レポート上の表示形式ごとの分類です。詳細は表2を参照してください。
- ⑥出力ウィンドウ：コンパイル、実行時などiReport

注3) Apache Ant : <http://ant.apache.org/index2.html> AntはWebサイトから最新版(apache-ant-1.5.4-bin.zip)をダウンロードし、任意のディレクトリに解凍します。

のメッセージを表示するエリアです。

インストール

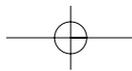
iReportおよび前提環境となるJDK(J2SDK), Antのインストールは、以下のように行います。

- ①iReport開発ページからiReport-0.2.2.zipをダウンロードし、任意のディレクトリに解凍します。これ以降、iReportの解凍先を[IREPORT_HOME]と呼びます。
- ②JDK, Ant^{注3)}をインストールし、環境変数(PATH)を設定します。Antは、iReport, HSQLDB(後述)の起動時に使用します。
- ③%IREPORT_HOME%\iReport.batを開き、IREPORT_HOME, JAVA_HOME(JDKのインストールディレクトリ), ANT_HOME(Antのインストールディレクトリ)を環境に合わせて設定します。
- ④%IREPORT_HOME%\iReport.batを実行しiReportを起動します。

事前準備

まず、本稿でのサンプルを実行する際に必要となる環境を設定します。





iReport で簡単帳票作成

PDF/Excel/HTML対応の
GUI帳票エディタ兼実行環境

第1章

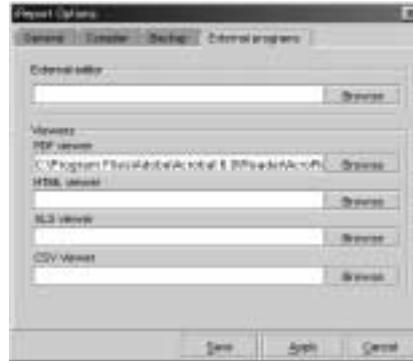
図3 追加するデータの記述

```
CREATE TABLE IREPORT(ITEMNO INTEGER PRIMARY KEY,ITEMNAME VARCHAR,PRICE INTEGER,COUNT INTEGER)
INSERT INTO IREPORT VALUES(1,'Software Design 2003年9月号',970,10)
INSERT INTO IREPORT VALUES(2,'WEB+DB PRESS Vol.16',1580,15)
INSERT INTO IREPORT VALUES(3,'Eclipse パーフェクトマニュアル',1680,5)
INSERT INTO IREPORT VALUES(4,'JAVA PRESS Vol.31',1480,10)
```

図4 JDBC接続の設定



図5 ビューアの設定



データベースの設定

本稿では、JasperReportsにサンプルとして含まれているHSQLDBを使用します^{編注}。

- ① JasperReportsの開発ページ (<http://jasperreports.sourceforge.net/>) から JasperReportsの最新版(本稿ではjasperreports-0.5.0-project.zip)をダウンロードし、任意のディレクトリに解凍します。これ以降、JasperReportsの解凍先を[JR_HOME]と呼びます。
- ② %JR_HOME%\demo\hsqldb\test.scriptの最下行に本稿で使用するデータとして図3の記述を追加します。
- ③ 以下のコマンドを実行し、HSQLDBを起動します。

```
> cd %JR_HOME%\demo\hsqldb
> ant
```

- ④ %JR_HOME%\lib\hsqldb-1.61.jarを%IREPORT_HOME%\lib%にコピーし、HSQLDBのJDBCドライバを利用できるようにします^{注4}。

iReportのデータベース接続設定

データソースとしてDBMSを使用する場合は、以下のような設定が必要です。

- ① iReportのメニューから[Datasource]-[Connections / Datasources]を選択します。

編注) HSQLDBについては、139ページ「はじめてのHSQLDB」にも記載があります。

注4) すでにiReportを起動していた場合は、コピー後にiReportの再起動が必要になります。

- ② [Connections / Datasources]ダイアログで[New]を選択し、JDBC接続に必要な設定を入力します(図4)。HSQLDBの設定を変更していなければ、ユーザ名="sa"、パスワード不要で接続できます。
- ③ [Test]を押して接続が確認できたら、[Save]を押して[Connections properties]を抜けます。
- ④ iReportのメニューから[Build]-[Set active connection]を選択し、作成したデータベース接続設定を有効にし、[OK]を押します。

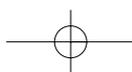
PDFビューアの設定

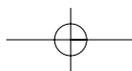
帳票出力の際に使用するビューアを設定します。

- ① iReportのメニューから[Tools]-[Options...]を選択します。
- ② [External programs]タブを開き、使用するビューアへの実行パスを設定します。本稿では、ビューアとしてAcrobat Readerを使用します(図5)。

ウィザード機能による帳票の作成

それでは、iReportのウィザード機能を使用して帳票作成の流れを確認しましょう。





Open Sourceで
業務システム #4

iReport Jakarta POI JFreeChart PDF, Excelデータ, グラフ出力アプリの開発

図6 Query



図7 Fields selection



図8 Group by...



図9 Layout



- ① iReportのメニューから[File]-[Report wizard...]を選択し、[iReport Wizard]ダイアログを開きます。
- ② Step1 (図6) [Query string]に “ select * from IREPORT ” と入力し、[Next >]を押します。
- ③ Step2 (図7) “ ITEMNAME ”, “ COUNT ”, “ PRICE ” をそれぞれ選択し、[>]を押してフィールドを登録します。
- ④ Step3 (図8) [Group 1]で “ ITEMNAME ” を選択し、[Next >]を押します。
- ⑤ Step4 (図9) レイアウトで[Columnar]-[grayC]を選択し、[Next >]を押します。他のレイアウトを選択した場合は動作させるためにデザインの修正が必要になる場合があります。
- ⑥ Step5 [Finish]を押すと、iReport上に作成された帳票が表示されます。
- ⑦ 保存アイコン () を選択し、任意のディレクトリにテンプレートXMLを保存します。
- ⑧ 実行アイコンを選択すると、作成したテンプレートXMLがコンパイルされ、帳票が出力されます。

請求書を作ってみる

ウィザード機能で帳票作成の流れを確認できたら、次にサンプル請求書(図10)の作成に移ります。

なお、本稿で使用したテンプレートXMLファイルは本誌Webサイト(<http://www.gihyo.co.jp/wdpress/>)にて公開していますので、そちらも参考にしてください。

帳票のデザイン

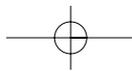
それでは、帳票のデザインを開始します。

iReportのメニューから[File]-[New document]を選択します。

[Report name]に任意の名称を入力し(図11), [OK]を押すとデザイン編集エリアが表示されます。

図12に、本稿でのデザイン編集エリアの完成イメージを示します。なお、図12中のフィールド②③、変数④は依存関係があるため、一部のフィールド、変数





iReport で簡単帳票作成

第 1 章

PDF/Excel/HTML対応の
GUI帳票エディタ兼実行環境

のみを定義した状態ではコンパイルエラーとなります。

静的文字列の配置

テキストアイコンを選択し、静的文字列を配置します(図12-①)。

配置したコンポーネントをダブルクリックするとプロパティダイアログが表示されます。[Static Text] タグ(図13)に文字列を入力します。次のステップに進む前に実行アイコンを押して、帳票が出力されることを確認してください^{注5}。

各コンポーネント共通の注意事項ですが、コンポーネントがバンドからはみだすと枠線が赤色で表示され、コンパイル時にエラーとなりますので、バンドに納まるように配置します。バンドを変更したい場合は、プロパティダイアログの[Common] タグ中の[Band] を変更します。

フィールドの配置

フィールドとは、データベースなどから取得したデータを表示するためのデータ形式で、" \${F{フ

図 10 サンプルとして作成する請求書



図 12 デザイン編集エリアの完成イメージ

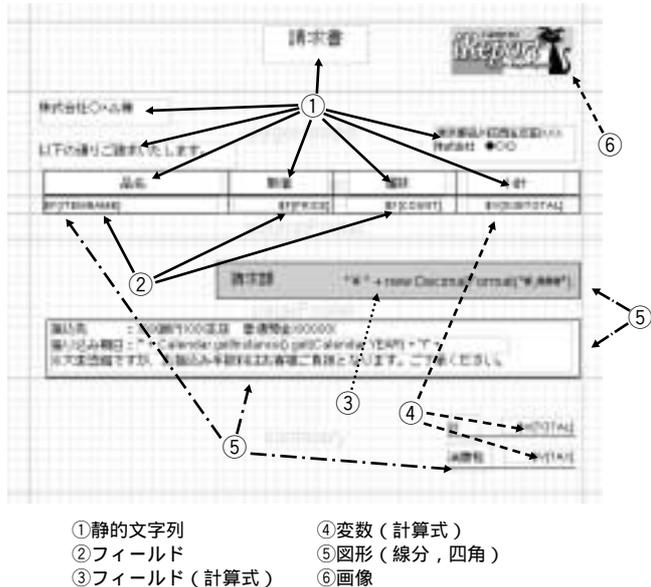


図 11 Report properties

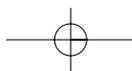


注5) 現バージョンのフォント設定では、外部フォントをGUI上から指定できないため、日本語を使用する際にはテンプレートXMLファイルを直接編集する必要があります。 タグのpdfFontNameに、使用するフォントへのパスを定義してください。本稿では以下の例のように指定し、MSゴシックを使用します(フォント設定はコンポーネントのコピー&ペーストで引き継げるので、コピー&ペーストすればXMLファイルを直接編集する必要はありません)。後述するフィールドや変数などで日本語を表示する際にも、同様の変更が必要となります。

例) 本稿でのフォント指定

```
<font fontName="MSGothic" pdfFontName="c:\winnt\fonts\msgothic.ttc,1" size="10" isBold="false" isItalic="false" isUnderline="false" isPdfEmbedded="true" pdfEncoding="Identity-H" isStrikeThrough="false" />
```





Open Sourceで
業務システム # 4

iReport Jakarta POI JFreeChart PDF, Excelデータ, グラフ出力アプリの開発

図 13 Static Text タグ



フィールド名}” で表現されます (図12-②, ③)。

●フィールド設定 (図12-②)

フィールドを使用する場合には、以下の手順で使用するフィールドを登録します (図14)。

- 1 データソース設定アイコンを押し、[Report query] を開きます。
- 2 [Report SQL query] タグを選択し、データ取得のためのSQL文を記述します。
- 3 [Read fields] を押し、SQL文により取得されたフィールドが表示されます。
- 4 レポート出力に使用するフィールドを選択し、[Register fields to report] を押しフィールドを登録します。
- 5 [Save query to report] を押し、クエリを保存します。

登録したフィールドをデザイン編集エリアに追加します。

- 1 フィールド設定アイコンを選択し、先ほど登録したフィールドが[Fields] タグに表示されていることを確認します (図15)。
- 2 フィールドアイコンを選択し、表示場所に配置します。
- 3 表示したフィールド表示領域をダブルクリックし、[Text Field] タグを選択します。
- 4 [Textfield expression] に “\$F{フィールド名}” と記述します (図16)。その際に登録したフィールドの [Class type] (図15) と [Textfield Expression Class] (図16) が一致している必要があります。

図 14 Report Query



●フィールドでのJavaプログラムの記述 (図12-③)

フィールドには、テーブルから取得したデータを表示する他に、任意のJavaプログラムを記述することもできます (図17)。

記述の際には、[Textfield expression] (図17-②) で作成されたデータ型と、[Textfield Expression Class] (図17-①) で指定した型が一致している必要があります。

本稿では、請求額の表示部分で以下のような計算式を使用しています。

```
"¥" + new DecimalFormat("#,###").format($V{TOTAL}
.intValue() + $V{TAX}.intValue()) + "-" 実際は1行
```

また、記述するJavaプログラムに独自のライブラリを使用することも可能です。その際はクラスパスにライブラリを追加し、パッケージ名から指定します。

変数の配置

変数とはレポートで使用する任意の値で、“\$V{変数名}” で表現されます (図12-④)。

- 1 フィールド設定アイコンを選択し、[Variables] タグを表示します。
- 2 [New] を押し、変数を新規に作成します。
- 3 各フィールドを入力し、[OK] を押します (図18)。図



iReport で簡単帳票作成

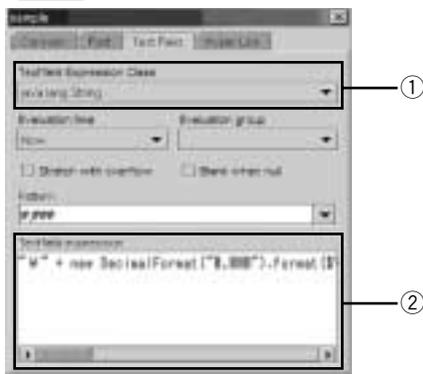
第 1 章

PDF/Excel/HTML 対応の
GUI 帳票エディタ兼実行環境

図 15 Fields タグ



図 17 Java プログラムの記述



18 の例では、同一ページに存在する $\$V\{SUBTOTAL\}$ という変数の合計値として TOTAL という Integer 型の変数を定義しています。

- ④ フィールドアイコンを選択し、デザイン編集エリアに追加します。
- ⑤ 追加したフィールド表示領域をダブルクリックし、[Text Field] タグを選択します。
- ⑥ [Text Field] タグを選択し、[Textfield expression] に “ $\$V\{\text{変数名}\}$ ” と記述します (図 19)。
- ⑦ [Pattern] を設定することで、表示形式を指定できます (図 19)。現バージョンでは表示指定が 1 パターンのみ設定可能です。本稿では小数点以下を非表示にするために、テンプレート XML の `<textField>` タグの `pattern="#,###0.00"` を `pattern="#,###"` と置換して使用しています)。

図形の配置

iReport 上から線分、四角形、四角形 (角丸)、円形が配置できます (図 12-⑤)。

図形を組み合わせることで、表などの任意の表示形式が表現できます。

図 16 Text Field タグ



図 18 Add/modify variable



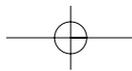
イメージの配置

イメージアイコンを選択し、イメージを配置します (図 12-⑥)。配置したコンポーネントをダブルクリックするとプロパティダイアログが表示されます。[Image] タグ (図 20) で、[Find...] を押し、任意のイメージを指定します。

請求書の生成実行

それでは、作成したテンプレート XML から帳票を作成してみましょう。

- ① コンパイルアイコンを押して、エラーがないことを確認します
iReport の出力からエラー原因が特定できない場合は、`$IREPORT_HOME` に出力されるソースファイル (テンプレート名.java) を直接コンパイルし、エラーメッセージを参照してください。
- ② コンパイルの正常終了を確認後、実行アイコンを押します



Open Sourceで 業務システム #4

iReport Jakarta POI JFreeChart PDF, Excelデータ, グラフ出力アプリの開発

図 19 Text Field タグ



図 20 画像の表示



実行アイコン (空データソース使用) を押すと、テーブルからの情報取得を行わない状態で実行されます。
③正常に実行が行なわれると、作成した請求書がPDFで表示されます

iReport の操作説明は以上で終了です。以降では、作成したテンプレートXMLをWebアプリケーションに組み込む方法について説明します。なお、Webアプリケーションに組み込む際はiReportは使用せず、実行環境であるJasperReportsのみを使用します。

Webアプリケーションへの組み込み

ローカルでの帳票出力が正常に行われたら、作成したテンプレートXMLをWebアプリケーションに組み込みます。

サーバ動作環境

ここでは、以下のサーバの動作環境が整っているものとします。

- OS : Red Hat Linux 8.0
- JDK : J2SDK 1.4.1
- DBMS : PostgreSQL 7.3.4
- コンテナ : Tomcat 4.1.27
- JasperReports : バージョン0.5.0

なお、ここではTomcatインストールディレクトリを[CATALINA_HOME]とし、JasperReportsのイン

ストールディレクトリを[JR_HOME]とします。

サンプルアプリケーションのコピー

まず、JasperReportsのサンプルを実行してみましょう。\$CATALINA_HOME/webapps配下に、\$JR_HOME/demo/samples/webappsをコピーします。コピーしたディレクトリの名称をJasperReportsに変更し、Tomcatを起動します。

サンプルアプリケーションのコンパイル

Tomcat起動後、以下のようなURLにアクセスします。

http://ホスト名(IPアドレス):ポート番号/JasperReports

画面左側のcompileを選択し、コンパイル画面を開きます。

executeを押し、コンパイルします。

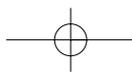
図21のように表示されればコンパイルは成功です。

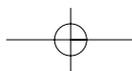
サンプルアプリケーションの実行

homeに戻りPDF outputを押し、executeを押します。設定が正常に行われていると、サンプルのPDFファイルが出力されます。

作成したテンプレートXMLの移行

ここでは、Webアプリケーションへの移行手順を簡単にするために、サンプルアプリケーションで使用したJSPファイルを書き換えて使用します。





iReport で簡単帳票作成

第 1 章

PDF/Excel/HTML 対応の
GUI 帳票エディタ兼実行環境

① 作成物のコピー

サンプルアプリケーションが正常に実行できたら、以下のディレクトリに本稿で作成したテンプレート XML (sample.xml) をコピーします。

```
$ CATALINA_HOME/webapps/JasperReports/reports
```

② JSP の書き換え

「サンプルアプリケーションのコンパイル」で使用した JSP (\$CATALINA_HOME/webapps/JasperReports/jsp/compile.jsp) を、図 22 の白地部分のように修正します。

「サンプルアプリケーションの実行」で使用した JSP (\$CATALINA_HOME/webapps/JasperReports/jsp/pdf.jsp) を、図 23 の白地部分のように修正します。JDBC の接続設定部分は環境に合わせて修正します。

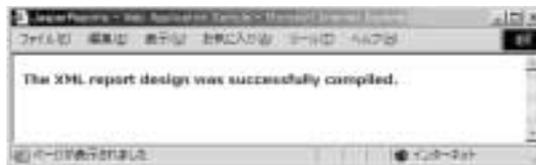
③ 日本語の表示

日本語を表示するためには、図 24 のように sample.xml の定義を変更します。

④ 実行

サンプルと同様にコンパイル 実行を行います。

図 21 実行後、表示される画面



の組み込みまでを駆け足で説明しましたがいかがでしたでしょうか。

本稿では誌面の都合上紹介できませんでしたが、iReport にはレポート中に別のレポートを埋め込む機能 (サブレポート) や JFreeChart を利用したグラフ表示など、多くの機能があります。本稿で紹介した以外の機能を試す場合は、JasperReports のデモに含まれるテンプレート XML を iReport に読み込ませ、改造するのが近道です。 **Web**

終わりに

iReport の使用方法から、Web アプリケーションへ

図 22 compile.jsp の修正

```
:
JasperCompileManager.compileReportToFile(application.getRealPath("/reports/sample.xml"));
:
```

図 23 pdf.jsp の修正

```
:
File reportFile = new File(application.getRealPath("/reports/sample.jasper"));
:
// データベースからデータを取得する場合には、JasperRunManager に connection を渡す。
Class.forName("org.postgresql.Driver");
Connection con = DriverManager.getConnection("jdbc:postgresql://localhost:5432/DEV","user","pass");

byte[] bytes =
    JasperRunManager.runReportToPdf(
        reportFile.getPath(),
        parameters,
        con
    );
:
```

図 24 日本語表示のための定義の変更 (kochi-gothic の場合)

```
<font fontName="kochi-gothic" pdfFontName="usr/share/fonts/ja/TrueType/kochi-gothic.ttf" size="12" isBold="false"
isItalic="false" isUnderline="false" isPdfEmbedded="false" pdfEncoding="Identity-H" isStrikeThrough="false" />
```

