

JasperReports6.3.1 移行ポイント資料

2017/09/01

Atrris

JasperReports6.3.1について

- Pexaが現在利用しているJasperReportsはバージョン4.6であるが、最新のバージョンは6.3.1になっています。
- バージョン4.6では帳票編集ツールはiReportを使用していましたが、バージョン6.3.1ではJasperSoft studioに変更されています。これはiReportではベースがNetBeansであったのが、Jaspersoft StudioではEclipseをベースにするようになったためです。新たなJaspersoft studioでは、プロパティの編集パネルに若干の違いがあります。
- PDFフォントの設定がjrxml内に存在するとWarningマークが出るようになります。よって、既存の*.jrxmlは改めて再作成することになります。Warningマークを無視するならそのまま問題ありません。
- 4.6と6.3.1で作成する*.jasperファイルで若干の差異があるようなので、jasperReportsの再コンパイルは必要です。
- 6.3.1ではバーコードとしてQRコードが使用できます。(英数文字?)

JasperReports6.3.1のインストール

- 以下のファイルをダウンロードします。
 - JasperReportのダウンロード
 - <https://sourceforge.net/projects/jasperreports/files/jasperreports/JasperReports%206.3.1/jasperreports-6.3.1-project.zip/download>
 - JasperSoft Studioのダウンロード
 - <http://sourceforge.net/projects/jasperstudio/>
 - downloadボタンを押す。

バージョンアップ作業 (jasperReports関係のjarの変更)

- ダウンロードしたJasperReports6.3.1のdistフォルダから以下のファイルを抽出します。
 - jasperreports-6.3.1.jar
 - jasperreports-fonts-6.3.1.jar
- 関連するJarファイルは、Mavenを利用して取得します。(Mavenをあらかじめインストールする必要があります)
- 以下のコマンドを格納したパス下で実行します。
 - mvn dependency:copy-dependencies
 - target/dependencyの以下のjarを使用します。

• barbecue-1.5-beta1.jar	• barcode4j-2.1.jar	• batik-anim-1.8.jar
• batik-awt-util-1.8.jar	• batik-bridge-1.8.jar	• batik-css-1.8.jar
• batik-dom-1.8.jar	• batik-ext-1.8.jar	• batik-gvt-1.8.jar
• batik-parser-1.8.jar	• batik-script-1.8.jar	• batik-svg-dom-1.8.jar
• batik-svggen-1.8.jar	• batik-util-1.8.jar	• batik-xml-1.8.jar
• core-3.2.1.jar	• itext-2.1.7.js5.jar	• jcommon-1.0.23.jar
• jfreechart-1.0.19.jar	• jxl-2.6.10.jar	• poi-3.10.1.jar
• poi-ooxml-3.10.1.jar	• poi-ooxml-schemas-3.10.1.jar	• xmlbeans-2.6.0.jar
• dom4j-1.6.1.jar		
- 独自のjarも使用します。
 - iTextAsian.jar pw62fonts.jar
- 上記のファイルを各プロジェクトの以下のフォルダにコピーします。
 - /lib/jasperrepots
- /tool/lib/jasperrepotsには、以下のjarファイルをコピーします。

• ant-1.7.1.jar	• antlr-2.7.5.jar	• commons-beanutils-1.9.0.jar	• commons-digester-2.1.jar
• groovy-all-2.4.5.jar	• itext-2.1.7.js5.jar	• jasperreports-6.3.1.jar	• jfreechart-1.0.19.jar

フォントJarファイルについて

- lib/j2ee/earフォルダに以下のPexaWorks標準フォントが含まれるJarファイルを格納します。
 - <http://pexa.atrris.com/pwdemo/pw62fonts/pw62fonts.jar>
 - 独自でFont設定を行う時は、JasperSoft Studioのwindow->Preferences->Jaspersoft Studio->FontsでFont listを作成し、Exportボタンで、Font用のJarファイルを作成する必要があります。
- 本fontJarファイルは、Jaspersoft StudioのそれぞれのProjectのプロパティのjava Build PathにAdd External JARsで登録しておく必要があります。

jrxml生成用templateファイルの変更

- jrxmlファイルにて、pdf用の属性を記述すると、JasperSoft StudioでのDesign表示時Warningマークが出てしまうので、pdf用の属性を自動生成しないようにするために以下のファイルを修正します。
- /etc/template/activity/conv_ExportPrint_jrxmlComponent.ftl
 - 5,13行の以下の部分を削除します。

```
pdfFontName="{PdfFontName}" pdfEncoding="{PdfEncoding}" isPdfEmbedded="{embedded}"
```

Jaspersoft Studioの設定

- JasperSoft Studioのダウンロードし、インストールします。
- 初期状態でMyReportプロジェクトが生成されていますので、それをそのまま利用します。
- Pexaで使用するためにPexa・フォントのjarファイルを参照することができるようにします。
 - MyReportのプロパティを選択します。
 - java Build Path->Librariesを選択します。
 - 各プロジェクトから以下のjarファイルadd External JARsで追加します。
 - lib/pexa/pexaext-share.jar
 - lib/jasperreports/pw62font.jar
- JasperSoft Studioで編集した場合、JasperSoft Studio特有のクラス名がjrxmlファイルに格納されますので、それを防ぐために以下の設定を変更します。
 - MyReportのプロパティを選択します。
 - Jaspersoft Studio->Compatibility->User Project Settingsをチェックします。
 - Don't save Jaspersoft studio propatiesをチェックします。
 - これで、JasperSoft Studio特有のクラス名は格納されません。
- jrxmlファイルをアプリケーションと関連付けをすれば、エクスプローラ・Eclipse上からjrxmlファイルをクリックすると、JasperSoft Studioが起動されてMyReportsプロジェクトに自動的に登録されます。

各プロジェクトへのJasperReports6.3.1の反映

- 以下の各プロジェクト内フォルダーの内容を削除します。
 - lib/j2ee/ear
 - lib/j2ee/webstart
 - lib/jasperreports
 - tool/lib/jasperreports
 - build/tmp (ear,warでの実行時、古いjasperReports関連のjarがあるとエラーになるのを防ぐため)
- PexaWikiからファイルをダウンロード展開し、lib,tool,etcフォルダを各プロジェクトフォルダーに上書きコピーします。Templateファイルは上書きの警告が出ますが、上書きを許可して続行します。
 - <http://pexa.atrris.com/pwdemo/jasperreports/jasper6.3.1download.zip>
- コピー後、ソースの自動生成からPexaWorksのビルドを実行します。
- すでに、iReport等のツールで編集済みの.jrxmlファイルが存在する場合、そのファイルは自動生成では上書きされませんので、注意してください。但し、JasperReports6.3.1でも該当.jrxmlファイルはそのまま利用できます。

jarファイルの格納先

- jasperreports関連のjarファイルをearに反映、またローカルプリントのためにjnlpでクライアントマシンに配給するためにlib/jasperreports下の全jarファイルを以下のフォルダにコピーする必要があります。
- earファイルに反映
 - lib/j2ee/earにコピー
- jnlpでクライアントマシンに配給
 - lib/j2ee/webstartにコピー
 - 但し、xmlbeans-2.6.0.jarはコピーしない。

build.xmlの調整

- build.xmlでのjrxmlファイルのコンパイル時、現行ではテンポラリとして.javaファイルをキープしておく設定となっていますが、.javaファイルを特にキープする必要がないのであれば、208行目のkeepjava="true"をkeepjava="false"に変更してください。

```
<taskdef name="jrc" classname="net.sf.jasperreports.ant.JRAntCompileTask"
classpathref="jasper.classpath"/>
<mkdir dir="${project.src}/print/jasper"/>
<jrc
  srcdir="${project.src}/print/jasper"
  destdir="${project.src}/print/jasper"
  tempdir="${project.src}/print/jasper"
  keepjava="false"
  xmlvalidation="true">
<classpath refid="jasper.classpath"/>
<include name="**/*.jrxml"/>
</jrc>
```

依存関係のあるJarファイルについて

- 必要なJarファイルとして、すでに記述されているものは標準的なPexaWorksソース自動生成で必要とされているもののみです。
- jaspersoft Sdutio等で修正を行った場合、他のJarファイルが必要になる場合があります。jasper6.3.1download.zip内には、jasperreports-dependencyフォルダがあります。その中には、jasperreportsで使用される可能性があるJarファイルがそろっています。必要に応じてlib/jasperreportsにJarファイルをコピーしてください。