PK_JRXML2PDF_APEX2JRXML

A JRXML-report-design-generator

Introduction

Description

PK_JRXML2PDF_APEX2JRXMLis a tool which takes an APEX-page as inputs and then creates a matching report-definition in the JasperReports-Format „jrxml“. The queries from the APEX-page are transformed into datasets in the resulting jrxml, the regions from the APEX-page are transformed in according detail-bands in the resulting jrxml. A detailed description of the process is given in this document.

The resulting jrxml-report-definition can be used with PK-jrxml2pdf as runtime, but can also be used with JasperReports itself.

PK_JRXML2PDF_APEX2JRXML is part of PL-jrxml2pdf and uses some of its helper-procedures.

Disclaimer

The jrxml-report-definitions should be treated as an intermediate result, which needs to be further detailed or adjusted.

Legal Disclaimer

The program(s) and/or file(s) are supplied as is. The author disclaims all warranties, expressed or implied, including, without limitation, the warranties of merchantability and of fitness for any purpose. The author assumes no liability for damages, direct or consequential, which may result from the use of these program(s) and/or file(s).

License

The software is currently available under both MIT and GPL-licences. Choose the one which best fits your needs. Licenses are included with the software in the license-directory.

Document-version

Version	Date	Description
0.5.0.0	02.04.13	Initial version, matching package version 0.5.0.0

Installation

The tool is installed as part of PK-jrxml2pdf when you choose the APEX-installation. It only works with the APEX-installation. There are no further installation steps required.

Client-Requirements (for detailling reports)

1. Install iReport-Designer from JasperSoft (tested with version 4.7.0), **its up to you to make sure you agree the license terms**.

2. make sure the file jrxml2pdf.jar from the directory java_for_ireport is the in iReport-classpath

3. Take the generated jrxml-report-definition from the JRXML_REPORT_DEFINITIONS-table and copy them to a local jrxml-file. Edit that file with the iReport-Designer and transfer teh result back to the table JRXML_REPORT_DEFINITIONS.

Running the generator

Starting

Choose „JRXML-Generator“ from the tab-bar.

In the upcoming dialog, choose

- The application containing the page you want to create a report for
- The page in the application you want to create a report for
- The name used to store the resulting report in JRXML_REPORT_DEFINITIONS

On the next page

- Choose the regions you want to include in your report. You can choose from all regions with a currently supported region-type.

On the next page

- Choose the templates for all the possible regions of the resulting report

On the next page

- For all tabular regions you have to chosen to include in the report, choose the desired templates

On the next page

- For all formular regions you have to chosen to include in the report, choose the desired templates
- Press Finish to start the report-generator

Result

The report-generator creates new entry in the JRXML_REPORT_DEFINITIONS-table with the name given on the first page. If your page-sources (the report-queries and/or the Automatic Row

Fetch process contain references to any APEX-page-item, a parameter will have been generated for it in the rsulting report-definion. You have to pass a value for whn running the report, otherwise you get a „Not all variables bound“-exception.

Feature-Overview

Report objects

Templates

Regions

Currently supported regions are

- HTML-region with items layouted in a formular way. The query for the formualr areay is taken from the page process for queryingthe data. Parameters from the query are transformed into parameters in the JRXML-query.
- Classic reports and tabular forms
  The Region is implemented in jrxml as a table-component, the query is taken form the region-source. Parameters from the query are transformed into parameters in the JRXML-query and are passed through from the main-report to the table-component.
- Interactive-reports
  The Region is implemented in jrxml as a table-component. The primary-default report is taken as base. The columns are taken from the report, along with ordering and conditions.Parameters from the query are transformed into parameters in the JRXML-query and are passed through from the main-report to the table-component.

Space-Measurement

In reports, the space for each column is calculated from the total available space from the region-template and the space each column takes. If a column has a specfiec width set, this is used, if not the width is derived from the datatype of the accodring column.

Items-Types

All items are transformed into textfields in the report. Itemtypes which use a LoV to map a value to a text are also mapped by the generator, an according lookup-select is included in the query. This may not be optimal for performance and might be manually adjusted after the report-generation.

Some items are currently not supported:

- Shuttles
- Images in reports.
- Plugin-items