

 無料電子ブック

学習

jasper-reports

Free unaffiliated eBook created from
Stack Overflow contributors.

#jasper-
reports

.....	6
.....	7
Jasper.....	7
2: JasperReports .jrxml.jasper.....	8
Examples.....	8
IDE.....	8
Apache Ant.....	10
Java.....	11
Apache Maven.....	11
3: pdf.....	13
.....	13
Examples.....	13
IDE.....	13
JasperSoft Studio.....	13
Java.....	14
JasperPrintjrxml.....	14
JasperPrintjrxml1.....	15
4: xls / xlsx.....	16
Examples.....	16
Java.....	16
.....	16
5:.....	19
.....	19
.....	19
Examples.....	20
.....	20
datasoure.....	20
6:.....	21
Examples.....	21
.....	21
.....	21
.....	

.....	21
7:	23
.....	23
Examples.....	23
IDE.....	23
JasperSoft Studio	23
JavaJasperReport.....	24
.....	24
.....	24
.....	24
.....	25
.....	26

You can share this PDF with anyone you feel could benefit from it, downloaded the latest version from: [jasper-reports](#)

It is an unofficial and free jasper-reports ebook created for educational purposes. All the content is extracted from [Stack Overflow Documentation](#), which is written by many hardworking individuals at Stack Overflow. It is neither affiliated with Stack Overflow nor official jasper-reports.

The content is released under Creative Commons BY-SA, and the list of contributors to each chapter are provided in the credits section at the end of this book. Images may be copyright of their respective owners unless otherwise specified. All trademarks and registered trademarks are the property of their respective company owners.

Use the content presented in this book at your own risk; it is not guaranteed to be correct nor accurate, please send your feedback and corrections to info@zzzprojects.com

1: jasper-reportsのい

Javaでレポートをするために、[JasperReports Java API](#)をするライブラリがいくつかあります。

- [DynamicReports](#)
- [ダイナミックジャスパー](#)

このライブラリ/フレームワークは、レポートのテンプレート *jrxml* ファイルをするとしないので、

バージョン

JasperReports ライブラリ

バージョン	
6.3.0	2016620
6.2.0	2015-11-11
5.6.0	2014-05-27
5.5.0	20131024
5.0.4	2013-03-26
5.0.0	2012-11-12
4.8.0	2012-11-05
4.7.0	2012-07-02
4.6.0	2012-05-21
4.5.0	2011-12-06
4.1.1	2011-04-18
4.0.0	2010-12-31
3.7.6	2010-10-27
3.7.5	2010-09-22
3.7.0	2009-12-08
3.6.0	2009-08-31

バージョン	
3.5.3	2009729
3.5.0	2009325
3.1.4	2009-02-10
3.1.2	2008114
3.1.0	2008-09-17
3.0.1	200887
3.0.0	2008-05-19
2.0.5	2008-03-12
2.0.3	2007-12-12
2.0.0	2007-08-14
1.3.4	20070611
1.3.0	2006-12-22
1.2.8	2006-11-14
1.2.0	2006-02-06
1.1.0	2005-10-21
1.0.3	20051010
1.0.0	2005-07-20
0.6.8	2005-05-31
0.2.3	2002-02-06

レポートをデザインするための **IDE**

のバージョンのデザイナーは *Eclipse* についています *Jaspersoft Studio*。

のバージョンのデザイナーは *NetBeans iReport Designer* についていました。

iReport Designer のバージョンはしたアプリケーションでした - [iReport Classic](#)

Examples

インストールまたはセットアップ

JasperReports ライブラリ

*JasperReports*はオープンソースのJavaベースのレポートツールです。 *JasperReports*ライブラリは、[リリースの Jaspersoft Community](#)からダウンロードできます。

のリリースではlibフォルダにあるサードパーティのjarファイルがされていない、らはのリポジトリからダウンロードするがあり、pom.xmlのために。 Mavenは、ターゲット/フォルダのなものをむすべてののをするためにできます。

```
mvn dependency:copy-dependencies
```

Jaspersoft StudioIDE

[Jaspersoft Studio](#)は、iReport Designerをきえるために、EclipseプラットフォームにされたJasperReportsののクライアントです。

iReport デザイナーIDE

[iReport Designer](#)はJasperReportsののレポートデザイナーです。バージョン5.6.020145リリースはこのバージョンでした。ベンダーのサポートは2015にしました。

JasperReport Communityの

JasperReports ライブラリにするよくある

- [よくある](#)

ソースコード

- [JasperReportsライブラリのソースコード](#)

チュートリアル

- [チュートリアルポイント](#)
- [JasperReportsのガイド](#)

サンプル

- サンプル
-
- [コミュニティ Wiki](#)

のバグトラッカー

- [バグトラッカー](#)

ワークフロー

ジャスパーレポートのフローはのとおりです。

1. レポートをし、レポートレイアウトをするjrxmlファイルを作ります。 jrxmlはシンプルなテキストエディタをしてすることができますが、はIDEJasperSoft StudioまたはiReportをしてレポートのをスピードアップし、レイアウトをにします。
2. レポートjrxmlをコンパイルして、.jasperファイルまたはJasperReportオブジェクトを作ります。このプロセスは.classファイルとコンパイルされる.javaファイルとできます。
3. レポートをし、パラメータとデータソースをレポートにして、オブジェクトJasperPrintを作ります。JasperPrintも.jprintファイルにできます
4. JasperPrintを、および/またはエクスポートする。もなフォーマットは、pdf、excel、word、html、cvsなどとしてサポートされています。

さまざまなレポートバンドの

タイトル

このバンドは、のめにされます。 isTitleNewPage="true"することで、のページとしてでき isTitleNewPage="true"

ページヘッダー

タイトルバンドがされている、のページをくページのめにされ、サマリーバンドが isSummaryWithPageHeaderAndFooter="false"されているは、

ヘッダー

これはページのバンドのにされます。

このセクションは、されたデータソースのレコードにしてりされます。のバンド1、2 ..nをつことがされ、のようにされる

```
Row 1
  detail 1
  detail 2
  detail n
Row 2
  detail 1
  detail 2
  detail n
```

フッター

バンドがするページのバンドのにされます。デフォルトのはページのページフッターのですが、これは `isFloatColumnFooter="true"` してのバンドのレコード

ページフッター

これは、タイトルページバンド、サマリーバンドページフッターなし、およびのページフッターがされているのはサマリーバンドをくページのにされます。

のページフッター

これは、のページフッターではなく、のページにされますページフッターのないサマリーバンドでない。

`isSummaryNewPage="true"` がされ、 `isSummaryWithPageHeaderAndFooter="true"` はページヘッダーとフッターでしいページのレポートのにされ `isSummaryWithPageHeaderAndFooter="true"`

グループヘッダー

このセクションは、グループがされるたびにバンドのにグループがされているにされます。

グループフッター

このセクションは、グループがされるに、バンドにグループがされているにされます。

バックグラウンド

このバンドは、のすべてのバンドのバックグラウンドとしてすべてのページにされます。

データなし

これは、データソースがされなかった、またはデータソースが0レコードで、`whenNoDataType="NoDataSection"`がされているにのみされます。

Jasperレポートファイル

- `.jrxml`はレポートデザインファイルです。はがなXMLであり、`JasperReport`オブジェクトにし、`.jasper`としてされます
- `.jasper`のコンパイルされたバージョンである`.jrxml`とにロードすることができる
`JasperReport`データでされるができてオブジェクト
- `.jrprint`はシリアルされた`JasperPrint`オブジェクトで、すでにデータでいっぱいになっていて、、、およびまたはのフォーマットへのエクスポートのためにロードすることができます。
- `.jrpxml`のXMLの`rappresentativo`ある`JasperPrint`それはし、そのりすためにアンマーシャルすることができるオブジェクト
`JasperPrint`オブジェクトを

オンラインでjasper-reportsのいをむ <https://riptutorial.com/ja/jasper-reports/topic/3594/jasper-reportsのい>

2: JasperReports .jrxmlを.jasperにコンパイルする

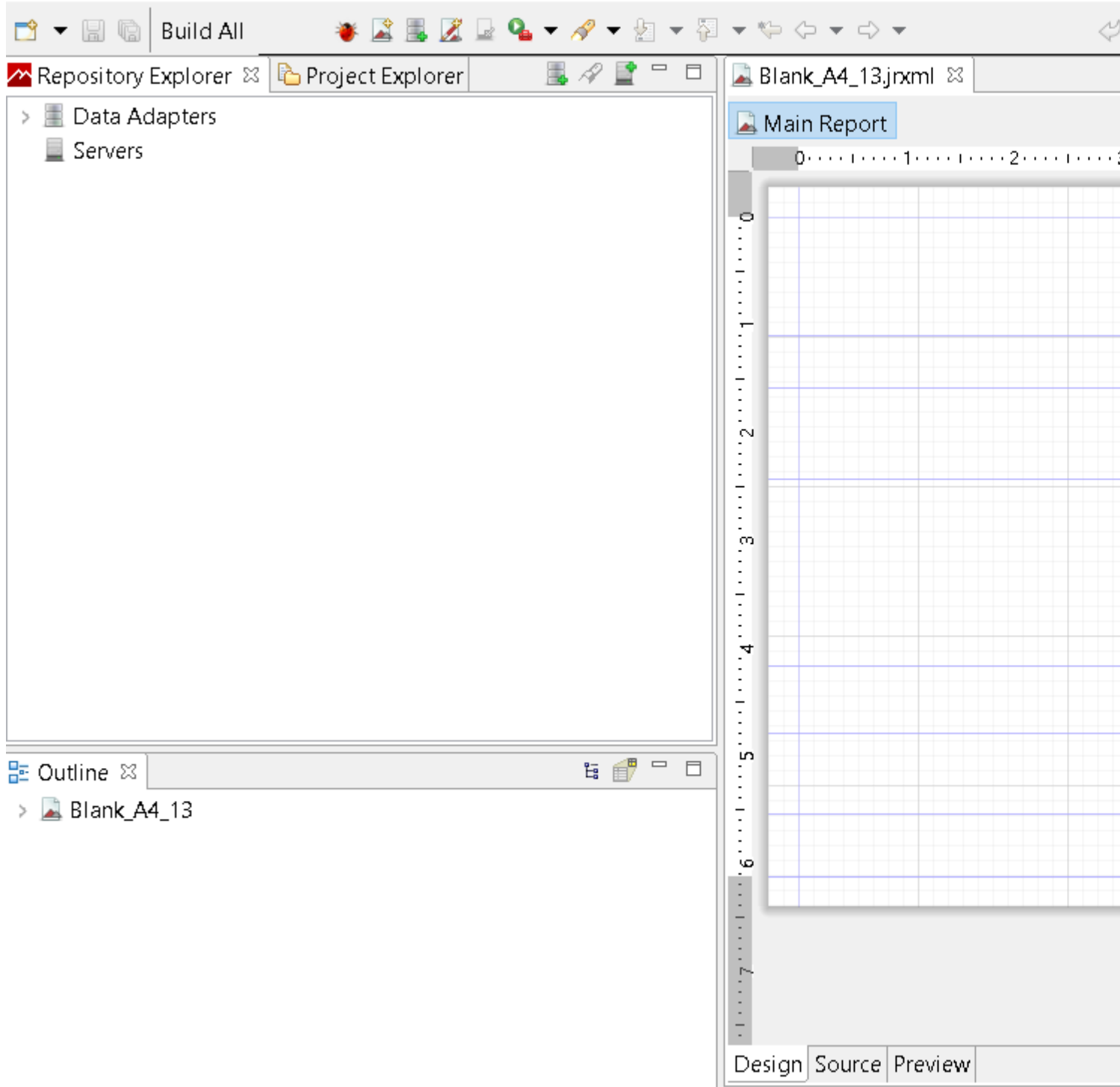
Examples

IDE

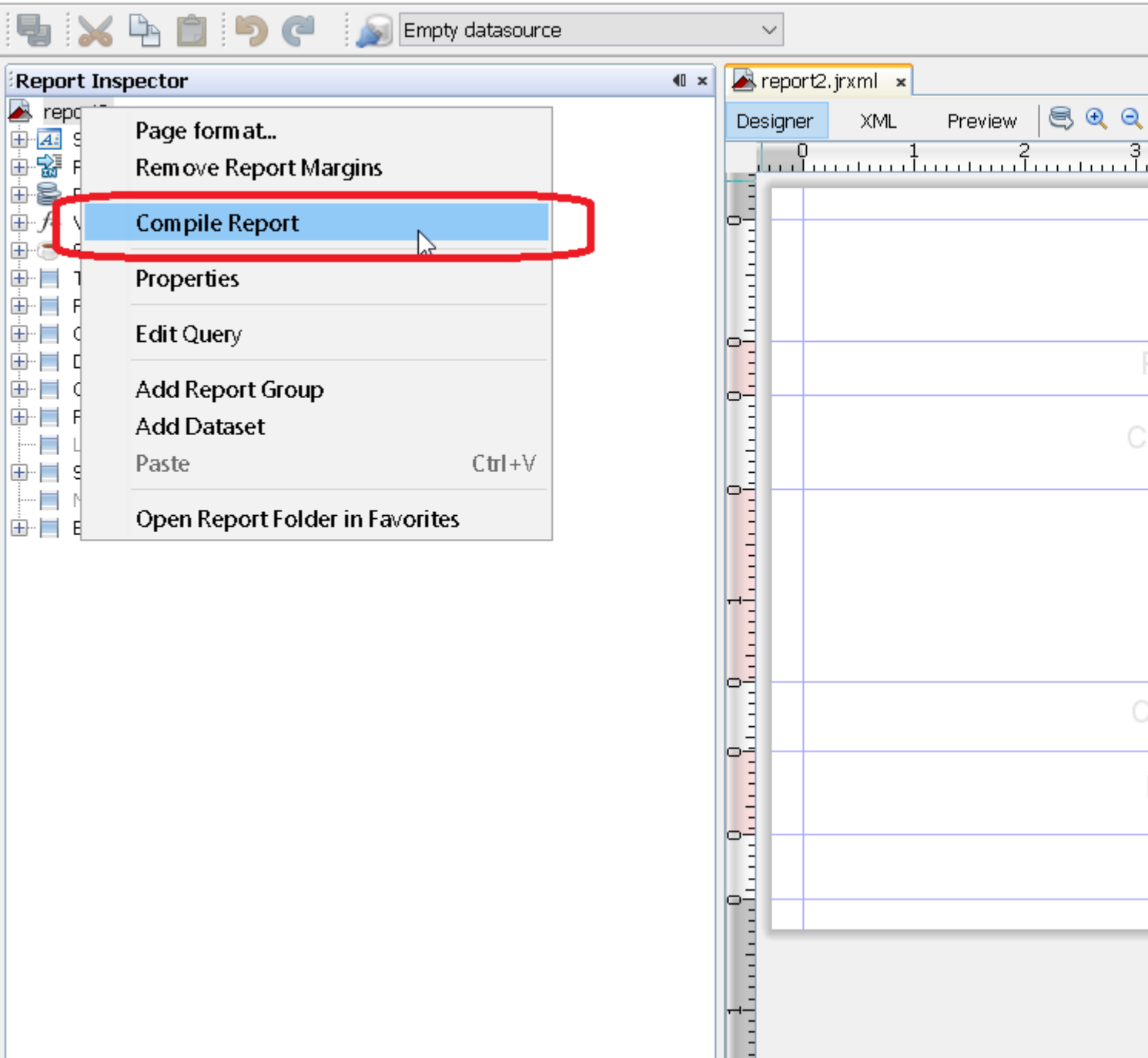
IDE [Jaspersoft Studio](#) [JSS](#) または旧バージョンの [iReport Designer](#) では、プレビューをすだけです。

JasperReportsのデザインファイル `.jrxml` にコンパイルされます `.jasper` と同じフォルダに `.jrxml` エラーがしない。

のは、[JSS](#)の「レポートのコンパイル」ボタンをすことです



*iReport*のレポートインスペクタからびされたコンテキストメニュー「コンパイルレポート」をします



Apache Antで

```
<target name="compile" description="Compiles report designs specified using the 'srcdir' in
the <jrc> tag." depends="prepare-compile-classpath">
  <mkdir dir="./build/reports"/>
  <taskdef name="jrc" classname="net.sf.jasperreports.ant.JRAntCompileTask">
    <classpath refid="project-classpath"/>
  </taskdef>
  <jrc
    srcdir="./reports"
    destdir="./build/reports"
    tempdir="./build/reports"
    keepjava="true"
  </jrc>
</target>
```

```
        xmlvalidation="true">
        <classpath refid="sample-classpath"/>
        <include name="**/*.jrxml"/>
    </jrc>
</target>
```

Apache Antビルド・ツールをシステムにしくインストールするがある

Javaで

それはコンパイルすることはですが、`.jrxml`にファイルを、`.jasper` Javaコードをしてファイル、これがコンパイルすることによってされ、パフォーマンスヒットき、`.jrxml` IDEをしてファイルを。これをにおいて、`.jrxml` ファイルをコンパイルするには、`JasperCompileManager`をのようになります。

```
JasperCompileManager.compileReportToFile(
    "designFile.jrxml", //Relative or absolute path to the .jrxml file to compile
    "compiled.jasper"); //Relative or absolute path to the compiled file .jasper
```

Apache Mavenの

[Alex Nederlof](#)による[JasperReports-plugin](#)は、された[org.codehaus.mojojasperreports-maven-plugin](#) プラグインのいです。

プラグインのはななです

```
<build>
  <plugins>
    <plugin>
      <groupId>com.alexnerlof</groupId>
      <artifactId>jasperreports-plugin</artifactId>
      <version>2.3</version>
      <executions>
        <execution>
          <phase>process-sources</phase>
          <goals>
            <goal>jasper</goal>
          </goals>
        </execution>
      </executions>
      <configuration>
        <sourceDirectory>src/main/resources/jrxml</sourceDirectory>
        <outputDirectory>${project.build.directory}/jasper</outputDirectory>
      </configuration>
    </plugin>
  </plugins>
</build>
```

Mavenでのコンパイルのコマンド

```
mvn jasperreportsjasper
```

ジャスパーファイルは、`${project.build.directory}/jasper`フォルダー—例えば、`/target/jasper`にされます。

オンラインでJasperReports .jrxmlを.jasperにコンパイルするをむ <https://riptutorial.com/ja/jasper-reports/topic/4943/jasperreports--jrxmlを-jasperにコンパイルする>

3: pdfへのエクスポート

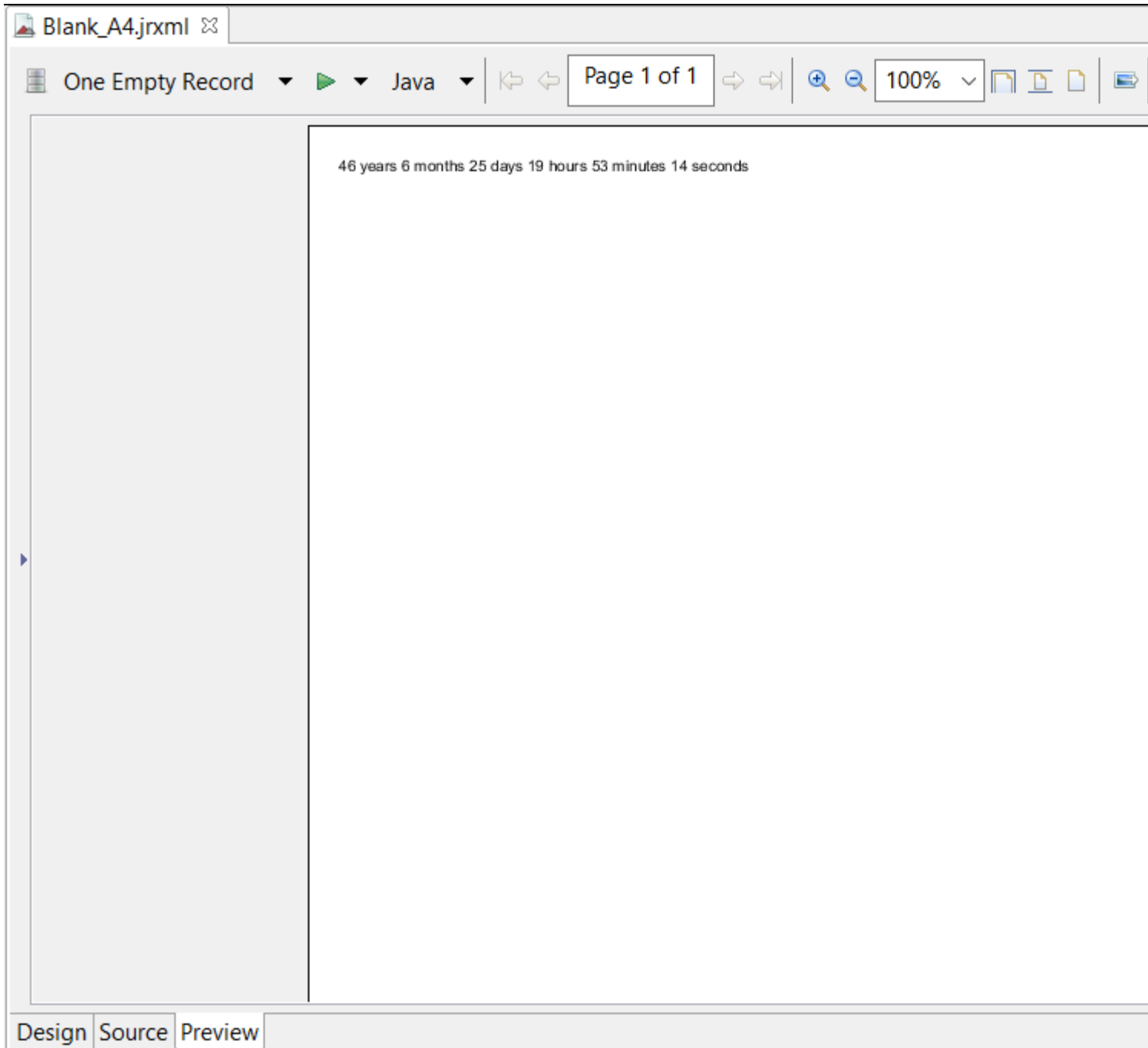
pdfでフォントをしくレンダリングするには、 `classpath`をにするがあります。

Examples

IDE

JasperSoft Studio

プレビューで、のをクリックしてレポートをし、エラーがなければエクスポートメニューをにし、エクスポートボタンディスクイメージをクリックして「Export As Pdf」をします。



Javaで

エクスポートするには、[レポート](#)をして **JasperPrint** オブジェクトをやるがあります。

の **JasperPrint** の **jrxml** をファイルにエクスポートする

```
// 1. Create exporter instance
JRPdfExporter exporter = new JRPdfExporter();

// 2. Set exporter input document
```

```
exporter.setExporterInput(new SimpleExporterInput(jasperPrint));

// 3. Set file path for exporter output
exporter.setExporterOutput(new SimpleOutputStreamExporterOutput("/path/filename.pdf"));

// 4. Create configuration instance
SimplePdfExporterConfiguration configuration = new SimplePdfExporterConfiguration();

// 5. Associate configuration with exporter
exporter.setConfiguration(configuration);

// 6. Fill export and write to file path
exporter.exportReport();
```

のJasperPrintのjrxmlを1つのファイルにエクスポートする

のステップだけがのセットとなります

```
List<JasperPrint> jasperPrintList = new ArrayList<>();
jasperPrintList.add(jasperPrint1);
jasperPrintList.add(jasperPrint2);

JRPdfExporter exporter = new JRPdfExporter();
exporter.setExporterInput(SimpleExporterInput.getInstance(jasperPrintList));
```

りのはじめです

```
exporter.setExporterOutput(new SimpleOutputStreamExporterOutput("/path/filename.pdf"));
SimplePdfExporterConfiguration configuration = new SimplePdfExporterConfiguration();
exporter.setConfiguration(configuration);
exporter.exportReport();
```

のについては、 [SimplePdfExporterConfiguration API](#)をしてください。

オンラインでpdfへのエクスポートをむ <https://riptutorial.com/ja/jasper-reports/topic/4190/pdfへのエクスポート>

4: xls / xlsxにエクスポート

Examples

Javaで

xlsxにエクスポートする

```
try (InputStream inputStream = JRLoader.getResourceInputStream(path)) { // read report as
input stream
    JasperReport jasperReport =
JasperCompileManager.compileReport(JRXmlLoader.load(inputStream)); // compile report

    Map<String, Object> params = new HashMap<>(); // init map with report's parameters
    params.put(JRParameter.REPORT_LOCALE, Locale.US);
    params.put(JRParameter.IS_IGNORE_PAGINATION, true);
    JasperPrint jasperPrint = JasperFillManager.fillReport(jasperReport, params, connection);
    // prepare report - passs parameters and jdbc connection

    JRXlsxExporter exporter = new JRXlsxExporter(); // initialize exporter
    exporter.setExporterInput(new SimpleExporterInput(jasperPrint)); // set compiled report as
input
    exporter.setExporterOutput(new SimpleOutputStreamExporterOutput(destFile)); // set output
file via path with filename
    SimpleXlsxReportConfiguration configuration = new SimpleXlsxReportConfiguration();
    configuration.setOnePagePerSheet(true); // setup configuration
    configuration.setDetectCellType(true);
    exporter.setConfiguration(configuration); // set configuration
    exporter.exportReport();
}
```

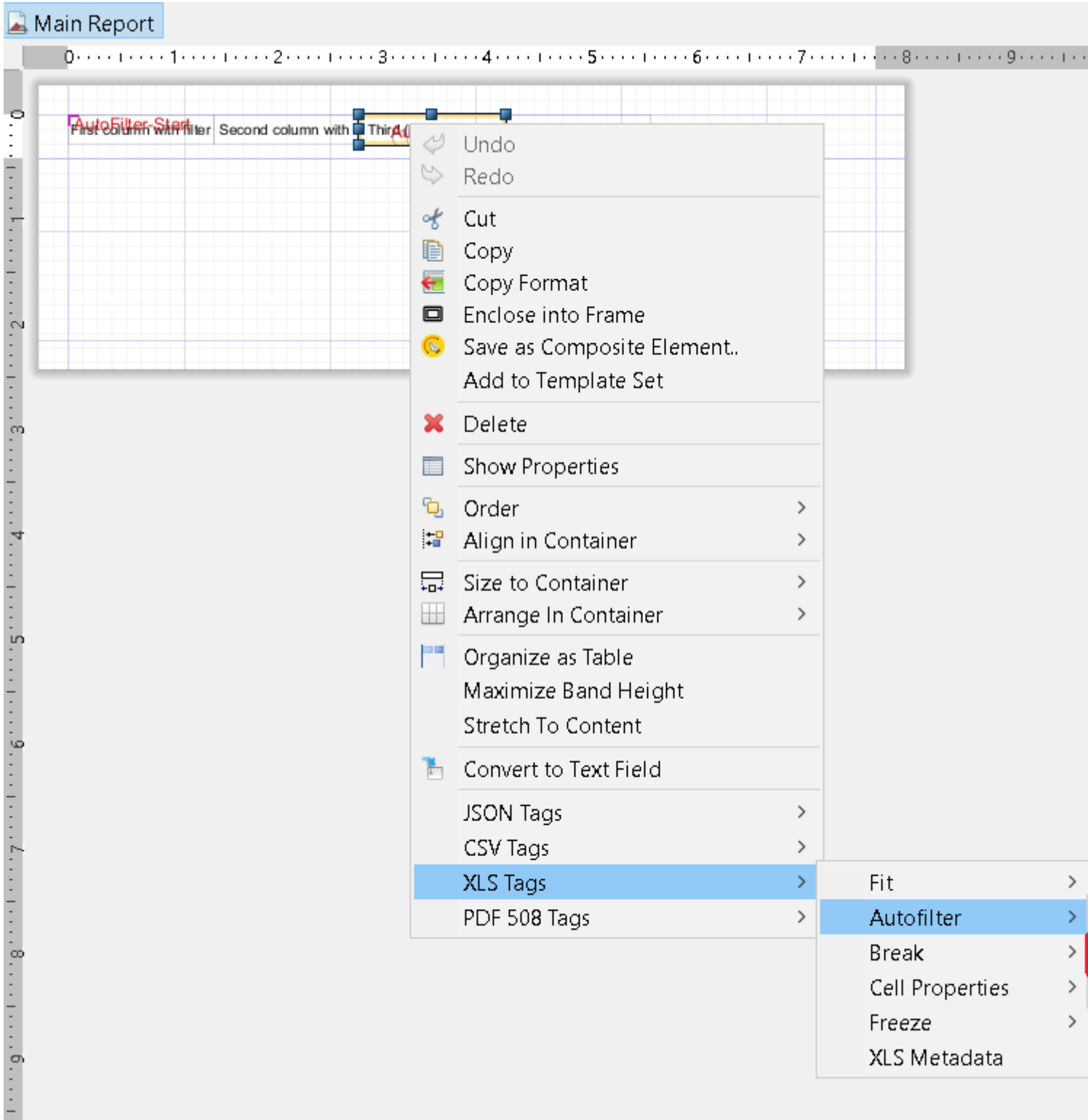
のオートフィルタの

[net.sf.jasperreports.export.xls.auto.filter](#) プロパティをすると、されたxlsファイルにautofilterを
できます。

```
<columnHeader>
  <band height="30" splitType="Stretch">
    <staticText>
      <reportElement x="0" y="0" width="100" height="20">
        <property name="net.sf.jasperreports.export.xls.auto.filter" value="Start"/>
      </reportElement>
      <text><![CDATA[First column with filter]]></text>
    </staticText>
    <staticText>
      <reportElement x="100" y="0" width="100" height="20"/>
      <text><![CDATA[Second column with filter]]></text>
    </staticText>
    <staticText>
      <reportElement x="200" y="0" width="100" height="20">
        <property name="net.sf.jasperreports.export.xls.auto.filter" value="End"/>
      </reportElement>
      <text><![CDATA[Third (Last) column with filter]]></text>
```

```
</staticText>
<staticText>
  <reportElement x="300" y="0" width="100" height="20"/>
  <text><![CDATA[Fourth column without filter]]></text>
</staticText>
</band>
</columnHeader>
```

このプロパティは、コンテキストメニューのヘルプをして *Jaspersoft Studio* であることも、 *jrxml* ファイルをであることによってですることもできます。



オンラインでxls / xlsxにエクスポートをむ <https://riptutorial.com/ja/jasper-reports/topic/5008/xls---xlsxにエクスポート>

5: サブレポートの

パラメーター

パラメータ	
parametersMapExpression	パラメータきのマップ。
サブレポートパラメータ	nameとvalueのペア subreportParameterExpression で。です。サブレポートにはいくつかのパラメータをすことができます
connectionExpression	データをすための。
dataSourceExpression	データソースをすための。
サブレポートの	サブレポートのパス/URI、さらにはJasperReportオブジェクト。
returnValue	とのペア。です。サブレポートからいくつかのをしてマスターレポートバックにすことができます

- なレポートをすために、サブレポートをすことができます。のレポートをすことは、サブレポートをすもう1つのです。
- `<subreport>`をす、`<subreport>`はマスターレポートのとしてされます。
- **subreportExpression**パラメータのは、*JasperReports*サーバであると、*JasperReports*フレームワークIDEでまたはするのAPIだけでなります。

*JasperReports Server*の、のようになります。

```
<subreportExpression><![CDATA["repo:subreport.jrxml"]]></subreportExpression>
```

*JasperReports*エンジンだけです

```
<subreportExpression><![CDATA["/somePath/subreport.jasper"]]></subreportExpression>
```

@AndreasDietrichのきなは *JasperServer*にありますがサブレポートのをつけることができません

- いくつかのから、サブレポートは、`<subreport>`のけをりてマスターレポートからびすことなく、のレポートとしてできます。サブレポートはにレポートです。

Examples

サブレポートにをす。をマスターレポートにす

これはマスターレポートのスニペットです。2つのパラメータとたとえば、*jdbc* がサブレポートにされています。1つのがサブレポートからマスターレポートにされます。このは、マスターレポートでできます

```
<subreport>
  <reportElement x="0" y="80" width="200" height="100"/>
  <subreportParameter name="someSubreportParameter">

<subreportParameterExpression><![CDATA[ ${P{someMasterReportParamter}} ]></subreportParameterExpression>

  </subreportParameter>
  <subreportParameter name="anotherSubreportParameter">
    <subreportParameterExpression><![CDATA["Some text - constant
value"]]></subreportParameterExpression>
  </subreportParameter>
  <connectionExpression><![CDATA[ ${REPORT_CONNECTION} ]></connectionExpression>
  <returnValue subreportVariable="someVariableInSubreport"
toVariable="someVariableInMasterReport"/>
  <subreportExpression><![CDATA[" ${SUBREPORT_DIR} +
"subreport.jasper" ]></subreportExpression>
</subreport>
```

サブレポートに **datasoure** をす

これはマスターレポートのスニペットです。データソースは、net.sf.jasperreports.engine.data.JRBeanCollectionDataSource コンストラクタのけをりてサブレポートにされます。

```
<field name="someFieldWithList" class="java.util.List"/>
<!-- ..... -->
<subreport>
  <reportElement x="0" y="0" width="200" height="70"/>
  <parametersMapExpression><![CDATA[ ${P{REPORT_PARAMETERS_MAP}} ]></parametersMapExpression>

<dataSourceExpression><![CDATA[net.sf.jasperreports.engine.data.JRBeanCollectionDataSource( ${F{someFieldWithList}} ) ]></dataSourceExpression>

  <subreportExpression><![CDATA[" ${SUBREPORT_DIR} +
"subreport.jasper" ]></subreportExpression>
</subreport>
```

オンラインでサブレポートのをむ <https://riptutorial.com/ja/jasper-reports/topic/5452/サブレポートの>

6: フォント

Examples

フォントのと

IDEをしてフォントをします。については、[iReport](#)または[Jaspersoft Studio](#)のマニュアルをしてください。フォントはですることもできます。

フォントとはですか

`textElement`をすると、フォントをできますされていないは、デフォルトのフォント `SansSerif`がされます

```
<textElement>
  <font fontName="DejaVu Sans"/>
</textElement>
```

フォントメトリックのため、アライメントなどをし、しくフォントをレンダリングするには、フォントは、`JVMJavamacchine`にマッピングするがあります。あなたはJVMにフォントファイルをインストールすることができますが、これはされません

JasperReport Ultimate Guideから

フォントエクステンションからしたフォントのみをすることをくします。これは、レポートがにされるときにフォントがアプリケーションでできるようにするためのものためです。システムフォントをすると、そのフォントがインストールされていないのあるしいマシンにしたときに、レポートがしくしなくなるリスクがにします

デフォルトのフォント

JasperReportsは、デフォルトのフォントをしますmavenディストリビューションjasperreports-fonts.jarを。これをクラスパスにすると、のフォントをせずにのfontNameをできます

DejaVu Sans
デジャブセリフ
DejaVu Sans Mono

な

pdfitextでフォントをするにすべき

- PDFにきすときに、テキストがしくされない、がされず、りしやサイズがしくない、フォントがされないがあります。
- の.ttf サポートされていますか [OpenType](#)、フォントはにをレンダリングできますかすべてのフォントがすべてのをUTF-8でレンダリングするわけではありません。
- しいエンコーディングがiTextにされていますかわしいまたはに エンコーディングIdentity-Hしてください。これはしいPDFにはおめであり、なるエンコーディングをさせることができます。
- フォントがインストールされていないでも、コンピュータでされたPDFがコンテンツをできるように、フォントがめまれていますかフォントが14のタイプ1フォントの1つではない、にそれがめまれます。

ジャスパーレポートでされているiTextのバージョンはすべてのフォントをレンダリングしないことにしてください。 `ttf` フォントをテストし、エンコーディングをすることができます。

オンラインでフォントをむ <https://riptutorial.com/ja/jasper-reports/topic/5773/フォント>

7: レポート

パラメーター

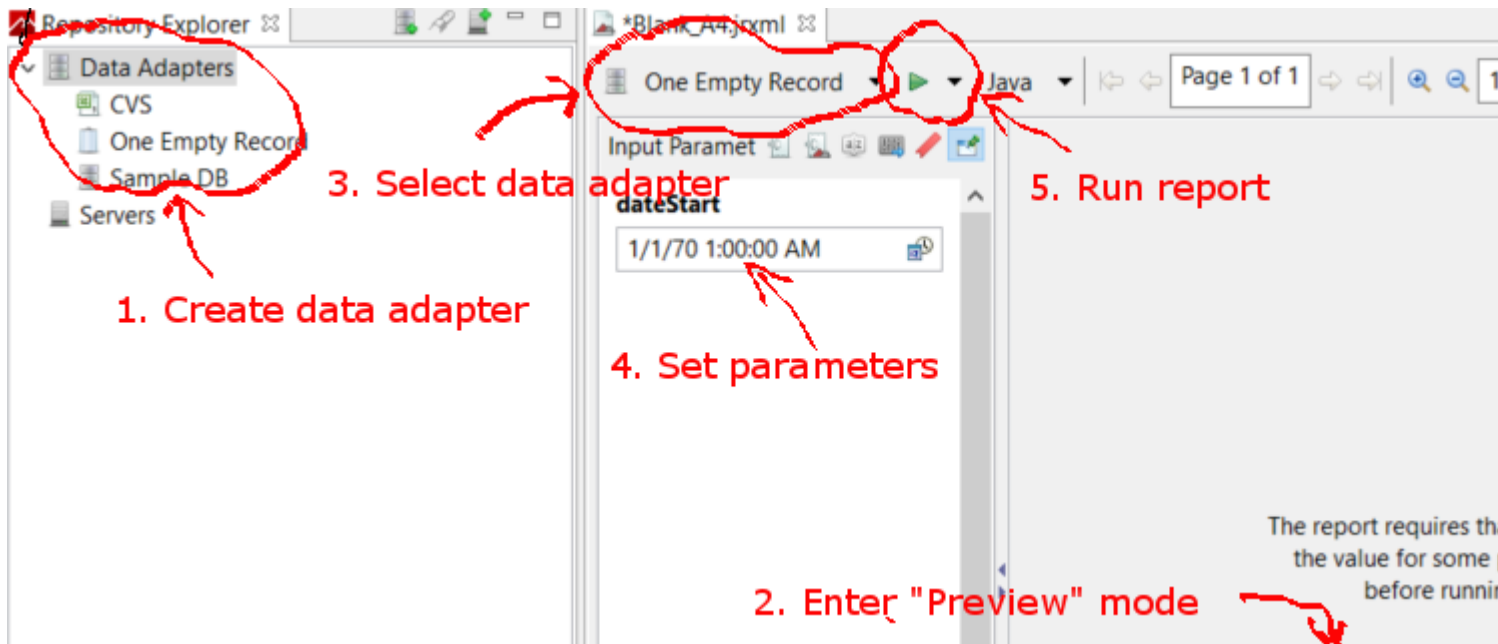
パラメーター	コラム
ジャスパープリン ト	のフォーマットにエクスポートすることができるフィルプロセスの
reportTemplate	コンパイルされたデザインファイル <code>.jasper</code>
パラメーター	されている、 <code>\$P{key}</code> によってレポートのになることができるパラメーターMapは、
	net.sf.jasperreports.engine.JRDataSource
	データベース java.sql.Connection

Examples

IDE

JasperSoft Studio

1. レポートをするためにデータソースまたはデータベースがなは、[データアダプタの]をして[データアダプタ]をクリックして、リポジトリエクスプローラでデータアダプタをします。
2. [プレビュー]タブをしてプレビューモードにりますするのあるエラーはありません
3. のデータソースをしますデータソースがなは、[のレコードを1つ]
4. にじてパラメータをする
5. の「レポートをする」をクリックしてレポートをします。



JavaをしてJasperReportテンプレートをりつぶす

の

データのになく、すべてのレポートは、レポートテンプレートとパラメータマップのパスになります。は、のすべてのでされます。

```
// Parameters passed into the report.  
Map<String, Object> parameters = new HashMap<>();  
  
// Arbitrary parameter passed into the report.  
parameters.put("KEY", "Value");  
  
// The compiled report design.  
String path = "path/to/template.jasper";
```

.jrxml ファイルをすると、ほとんどのではなのコンパイルステップがします。レポートをするに .jrxml をするカスタムソフトウェアをしていないをにまたはするなどは、のにすように .jasper ファイルをします。

データベースの

```
// Establish a database connection.  
Connection connection = DriverManager.getConnection(url, username, password);  
  
// Fill the report, get the JasperPrint that can be exported to desired format.  
JasperPrint jasperPrint = JasperFillManager.fillReport(  
    path, parameters, connection);
```

カスタムデータソースの

```
// Populate this list of beans as per your requirements.
List<Bean> beans = new ArrayList<>();

// Wrap the beans in a beans in a JRBeanCollectionDataSource.
JRBeanCollectionDataSource datasource = new JRBeanCollectionDataSource(beans);

// Fill the report, get the JasperPrint that can be exported to desired format.
JasperPrint jasperPrint = JasperFillManager.fillReport(
    path, parameters, datasource);
```

データソースがないと、のバンド

```
// Fill the report, get the JasperPrint that can be exported to desired format.
JasperPrint jasperPrint = JasperFillManager.fillReport(path, parameters);
```

データがなければ、`JasperReport`の`whenNoDataType="AllSectionsNoDetail"`をすることがあります。そうでない場合は、このレポートがされます。

オンラインでレポートをむ <https://riptutorial.com/ja/jasper-reports/topic/3958/レポート>

クレジット

S. No		Contributors
1	jasper-reportsのい	Alex K , Community , Dave Jarvis , Petter Friberg
2	JasperReports .jrxml を.jasperにコンパイルする	Alex K , Dave Jarvis , Petter Friberg
3	pdfへのエクスポート	Alex K , Dave Jarvis , Petter Friberg , RamenChef
4	xls / xlsxにエクスポート	Alex K
5	サブレポートの	Alex K
6	フォント	Dave Jarvis , Petter Friberg
7	レポート	Alex K , Dave Jarvis , Petter Friberg