



**Kostenloses eBook**

**LERNEN**

**jasper-reports**

Free unaffiliated eBook created from  
**Stack Overflow contributors.**

**#jasper-  
reports**

# Inhaltsverzeichnis

Über.....	1
<b>Kapitel 1: Erste Schritte mit Jasper-Berichten.....</b>	<b>2</b>
Bemerkungen.....	2
Versionen.....	2
JasperReports- Bibliothek.....	2
IDE zum Entwerfen von Berichten.....	3
Examples.....	4
Installation oder Setup.....	4
<b>JasperReports-Bibliothek.....</b>	<b>4</b>
<b>Jaspersoft Studio (IDE).....</b>	<b>4</b>
<b>iReport Designer (IDE).....</b>	<b>4</b>
<b>JasperReport Community-Ressourcen.....</b>	<b>4</b>
Häufig gestellte Fragen zu JasperReports-Bibliothek.....	4
Quellcode.....	4
Tutorials.....	4
Proben.....	5
Verweise.....	5
Offizieller Bug Tracker.....	5
Arbeitsablauf.....	5
Die unterschiedlichen Berichtsbereiche verstehen.....	5
<b>Titel.....</b>	<b>5</b>
<b>Kopfzeile.....</b>	<b>5</b>
<b>Spaltenüberschrift.....</b>	<b>6</b>
<b>Detail.....</b>	<b>6</b>
<b>Spaltenfußzeile.....</b>	<b>6</b>
<b>Fußzeile.....</b>	<b>6</b>
<b>Letzte Seitenfußzeile.....</b>	<b>6</b>
<b>Zusammenfassung.....</b>	<b>6</b>
<b>Gruppenkopfzeile.....</b>	<b>7</b>

<b>Gruppenfußzeile</b> .....	<b>7</b>
<b>Hintergrund</b> .....	<b>7</b>
<b>Keine Daten</b> .....	<b>7</b>
Jasper-Berichtsdateiformate.....	7
<b>Kapitel 2: Als PDF exportieren</b> .....	<b>8</b>
Bemerkungen.....	8
Examples.....	8
Mit IDE (Integrierte Entwicklungsumgebung).....	8
<b>JasperSoft Studio</b> .....	<b>8</b>
Mit Java.....	9
<b>Einzelnen JasperPrint (einzelnes Jrxml) in eine Datei exportieren</b> .....	<b>9</b>
<b>Exportieren Sie mehrere JasperPrint-Dateien (mehrere Jrxml) in eine einzelne Datei</b> .....	<b>10</b>
<b>Kapitel 3: Bericht ausfüllen</b> .....	<b>11</b>
Parameter.....	11
Examples.....	11
Mit IDE (Integrierte Entwicklungsumgebung).....	11
<b>JasperSoft Studio</b> .....	<b>11</b>
Füllen Sie die JasperReport-Vorlage mit Java.....	12
<b>Allgemeine Anforderungen</b> .....	<b>12</b>
<b>Verwenden einer Datenbankverbindung</b> .....	<b>12</b>
<b>Verwenden einer benutzerdefinierten Datenquelle</b> .....	<b>12</b>
<b>Ohne Datenquelle, unbenutztes Detailband</b> .....	<b>13</b>
<b>Kapitel 4: Export nach xls / xlsx</b> .....	<b>14</b>
Examples.....	14
Mit Java.....	14
Autofilter für Spalten hinzufügen.....	14
<b>Kapitel 5: Font-Erweiterungen</b> .....	<b>17</b>
Examples.....	17
Erstellen und Verwenden von Font-Erweiterungen.....	17
<b>Was sind Font-Erweiterungen?</b> .....	<b>17</b>

<b>Standardschrifterweiterung</b> .....	<b>17</b>
<b>Häufige Probleme</b> .....	<b>17</b>
<b>Kapitel 6: Kompilieren Sie JasperReports .jrxml zu .jasper</b> .....	<b>19</b>
Examples.....	19
Mit IDE (Integrierte Entwicklungsumgebung).....	19
Mit Apache Ant.....	21
Mit Java.....	22
Mit Apache Maven.....	22
<b>Kapitel 7: Unterberichte verwenden</b> .....	<b>24</b>
Parameter.....	24
Bemerkungen.....	24
Examples.....	25
Übergabe der Verbindung zum Unterbericht; Rückgabewerte zurück zum Masterbericht.....	25
Datenübergabe an Unterbericht übergeben.....	25
<b>Credits</b> .....	<b>26</b>



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](mailto:info@zzzprojects.com)

---

# Kapitel 1: Erste Schritte mit Jasper-Berichten

## Bemerkungen

Es gibt mehrere Bibliotheken, die [JasperReports Java API](#) zum Erstellen von Berichten mit Java verwenden:

- [DynamicReports](#)
- [DynamicJasper](#)

Diese Bibliotheken / Frameworks können Berichte "on fly" mit oder ohne Verwendung der Berichtsvorlage ( *jxml*- Datei) erstellen.

## Versionen

### *JasperReports*- Bibliothek

Ausführung	Veröffentlichungsdatum
6.3.0	2016-06-20
6.2.0	2015-11-11
5.6.0	2014-05-27
5.5.0	2013-10-24
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

Ausführung	Veröffentlichungsdatum
3.7.0	2009-12-08
3.6.0	2009-08-31
3.5.3	2009-07-29
3.5.0	2009-03-25
3.1.4	2009-02-10
3.1.2	2008-11-04
3.1.0	2008-09-17
3.0.1	2008-08-07
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	2007-06-11
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	2005-10-10
1.0.0	2005-07-20
0,6,8	2005-05-31
0,2,3	2002-02-06

## ***IDE zum Entwerfen von Berichten***

Die aktuelle Version des Designers basiert auf *Eclipse* : *Jaspersoft Studio* .

Die vorherige Version des Designers basiert auf *NetBeans* : *iReport Designer* .

Die erste Version von *iReport Designer* war eine unabhängige Anwendung - *iReport Classic*

## Examples

### Installation oder Setup

---

## JasperReports-Bibliothek

*JasperReports* ist ein Open Source Java-basiertes Reporting-Tool. Die *JasperReports*-Bibliothek kann für die neueste [Version](#) von der [Jaspersoft Community](#) heruntergeladen werden.

In den letzten Versionen werden die Jars von Drittanbietern im Ordner lib **nicht** verteilt. Sie müssen aus öffentlichen Repositorys heruntergeladen werden. Abhängigkeiten finden Sie in der Distributed `pom.xml`. Maven kann verwendet werden, um alle Abhängigkeiten einschließlich der vorübergehenden im Ziel- / Abhängigkeitsordner abzurufen.

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

---

## Jaspersoft Studio (IDE)

[Jaspersoft Studio](#) ist der offizielle Design-Client für JasperReports - der auf der Eclipse-Plattform basiert - ersetzt iReport Designer.

---

## iReport Designer (IDE)

[iReport Designer](#) ist der vorherige Berichtsdesigner für JasperReports. Version 5.6.0 (veröffentlicht im Mai 2014) war die letzte offizielle Version; Lieferanten-Support endete Ende 2015.

---

---

## JasperReport Community-Ressourcen

### Häufig gestellte Fragen zu JasperReports-Bibliothek

- [FAQ](#)

### Quellcode

- [JasperReports Library-Quellcode](#)

### Tutorials



- [Tutorials Punkt](#)
- [JasperReports Ultimate Guide](#)

## Proben

- [Beispielreferenz](#)

## Verweise

- [Offizielle Dokumentation](#)
- [Community-Wiki](#)

## Offizieller Bug Tracker

- [Bug Tracker](#)

## Arbeitsablauf

Der Arbeitsablauf in Jasper-Berichten ist:

1. Entwerfen Sie den Bericht, erstellen Sie die jrxml-Datei, die das Berichtslayout definiert. Die jrxml kann mit einem einfachen Texteditor erstellt werden. Normalerweise wird jedoch eine IDE (JasperSoft Studio oder iReport) verwendet, um die Berichtsentwicklung zu beschleunigen, aber auch um eine visuelle Ansicht des Layouts zu erhalten.
2. Kompilieren Sie den Bericht (jrxml), um eine .jasper-Datei oder ein [JasperReport](#)- Objekt zu erhalten. Dieser Vorgang kann mit einer `.java` Datei verglichen werden, die mit `.class` kompiliert wird.
3. [Füllen Sie den Bericht aus](#) , übergeben Sie Parameter und eine Datenquelle an den Bericht, um das [Druckobjekt](#) JasperPrint zu generieren, das auch in einer `.jprint` Datei `.jprint` kann
4. JasperPrint anzeigen, drucken und / oder exportieren. Die meisten Commons-Exportformate werden als PDF, Excel, Word, HTML, CVS usw. unterstützt.

## Die unterschiedlichen Berichtsbereiche verstehen

---

### Titel

Diese Band wird einmal zu Beginn des Berichts angezeigt. Sie kann als erste Seite verwendet werden, indem das Attribut `isTitleNewPage="true"`

---

### Kopfzeile

Dies wird am Anfang jeder Seite angezeigt, ausgenommen die erste Seite, wenn die `isSummaryWithPageHeaderAndFooter="false"`

verwendet wird, und die letzte Seite, wenn die Zusammenfassung mit der Einstellung `isSummaryWithPageHeaderAndFooter="false"`

---

## Spaltenüberschrift

Dies erscheint vor dem Detailbereich auf jeder Seite.

---

## Detail

Dieser Abschnitt wird **für jeden Datensatz** in der gelieferten Datenquelle iteriert. Es ist zulässig, mehrere Detailbänder (Detail 1, Detail 2, Detail n) zu haben, die wie folgt wiederholt werden

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

---

## Spaltenfußzeile

Dies wird unterhalb des Detailbereichs auf jeder Seite angezeigt, auf der Detailbereich vorhanden ist. Die Standardeinstellung ist das Seitenende (vor dem Seitenfuß), aber Sie können zum letzten Detailbereich (letzter Datensatz) wechseln, indem Sie das Attribut `isFloatColumnFooter="true"`

---

## Fußzeile

Dies wird am unteren Rand jeder Seite angezeigt, mit Ausnahme von Titelband, Zusammenfassungsband (ohne Seitenfuß) und dem letzten nicht zusammenfassenden Band, wenn die letzte Seitenfußzeile verwendet wird.

---

## Letzte Seitenfußzeile

Dies erscheint auf der letzten Seite (wenn nicht zusammenfassender Bereich ohne Seitenfuß) anstelle des normalen Seitenfußes

---

## Zusammenfassung

Dies wird am Ende des Berichts auf der neuen Seite `isSummaryNewPage="true"` wenn `isSummaryNewPage="true"` ist, und mit Kopf- und Fußzeile der Seite, wenn

isSummaryWithPageHeaderAndFooter="true"

---

## Gruppenkopfzeile

Dieser Abschnitt wird angezeigt, wenn vor jeder Detailänderung eine Gruppe definiert wird, wenn sich der Gruppenausdruck ändert.

---

## Gruppenfußzeile

Dieser Abschnitt wird angezeigt, wenn *vor* der Änderung des Gruppenausdrucks nach dem Detailbereich eine Gruppe definiert wird.

---

## Hintergrund

Dieses Band wird auf jeder Seite als Hintergrund für alle anderen Bands angezeigt.

---

## Keine Daten

Dies wird nur angezeigt, wenn keine Datenquelle übergeben wurde oder die Datenquelle leer ist (0 Datensätze) und `whenNoDataType="NoDataSection"` ist.

### Jasper-Berichtsdateiformate

- `.jrxml` ist die Berichtsentwurfsdatei. Das Format ist in XML, das vom `.jrxml` lesbar ist. Es kann in ein `JasperReport` Objekt `JasperReport` und als `.jasper` gespeichert `.jasper`
- `.jasper` ist die kompilierte Version von `.jrxml` kann direkt in ein `JasperReport` Objekt geladen werden, das mit Daten gefüllt werden kann
- `.jrprint` ist das serialisierte `JasperPrint` Objekt, ein Bericht, der bereits mit Daten gefüllt ist und geladen werden kann, um gedruckt, angesehen und / oder in das gewünschte Format exportiert zu werden.
- `.jrpxml` ist das XML- `JasperPrint` eines `JasperPrint` Objekts. Es kann geändert und anschließend aus dem Marshall genommen werden, um das `JasperPrint` Objekt `JasperPrint`

Erste Schritte mit Jasper-Berichten online lesen: <https://riptutorial.com/de/jasper-reports/topic/3594/erste-schritte-mit-jasper-berichten>

---

# Kapitel 2: Als PDF exportieren

## Bemerkungen

Um **Schriftarten** in PDF - **Dateien** korrekt darzustellen, sollten immer (in classpath)

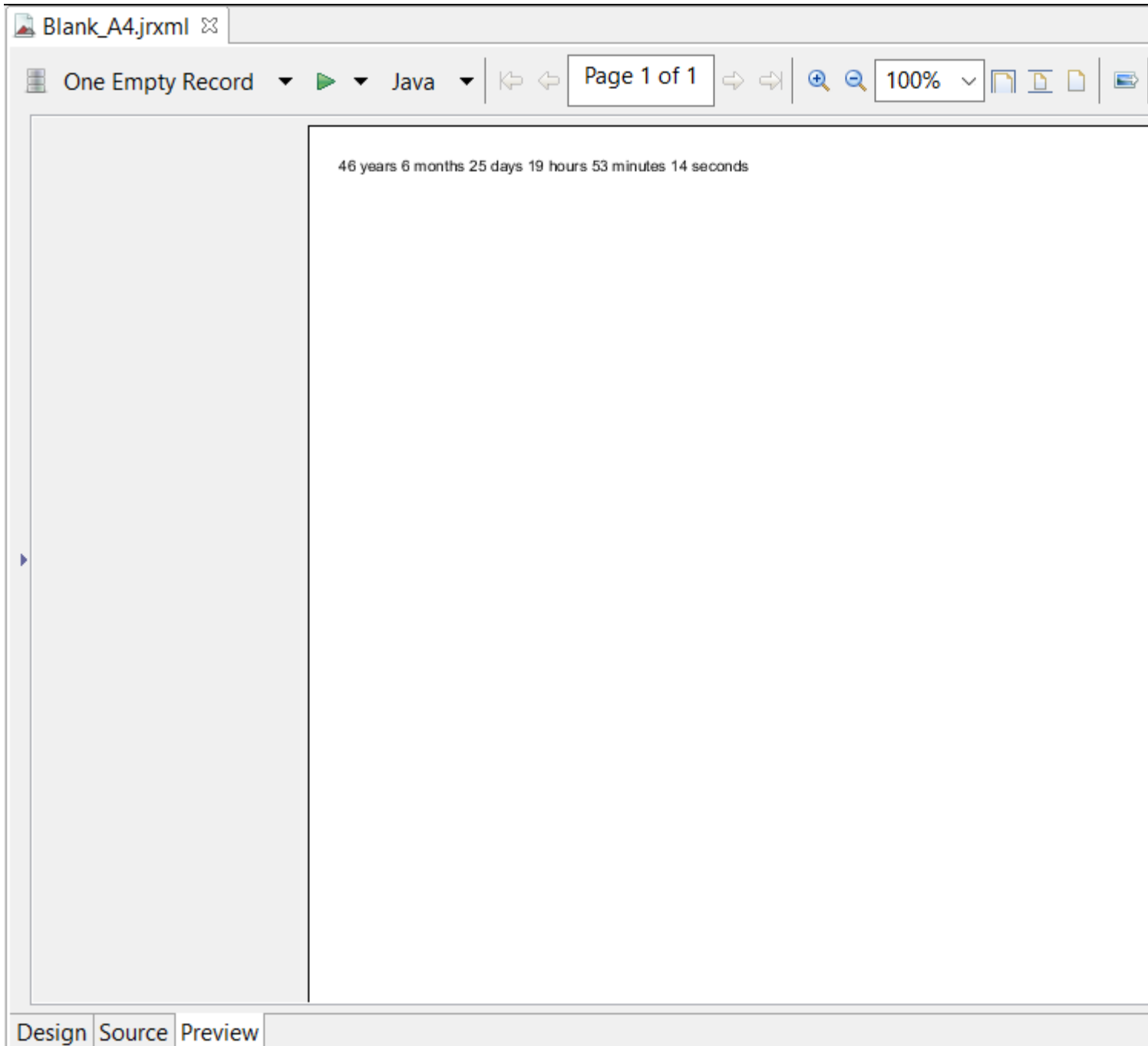
## Examples

Mit IDE (Integrierte Entwicklungsumgebung)

---

## JasperSoft Studio

Führen Sie in der Vorschau den Bericht aus, indem Sie auf den grünen Pfeil klicken. Wenn keine Fehler angezeigt werden, wird das Exportmenü aktiviert. Klicken Sie auf die Exportschaltfläche (Disk-Image) und wählen Sie "Export as Pdf".



## Mit Java

Um ein a zu exportieren, müssen Sie [den Bericht ausfüllen](#) , um das [JasperPrint](#)- Objekt zu erhalten.

# Einzelnen JasperPrint (einzelnes Jrxml) in eine Datei exportieren

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

// 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();
```

---

## Exportieren Sie mehrere JasperPrint-Dateien (mehrere Jrxml) in eine einzelne Datei

Nur die ersten Schritte unterscheiden sich vom vorherigen Satz:

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

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

Die restlichen Schritte sind die gleichen:

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

Konfigurationsdetails finden Sie unter [SimplePdfExporterConfiguration-API](#) .

Als PDF exportieren online lesen: <https://riptutorial.com/de/jasper-reports/topic/4190/als-pdf-exportieren>

# Kapitel 3: Bericht ausfüllen

## Parameter

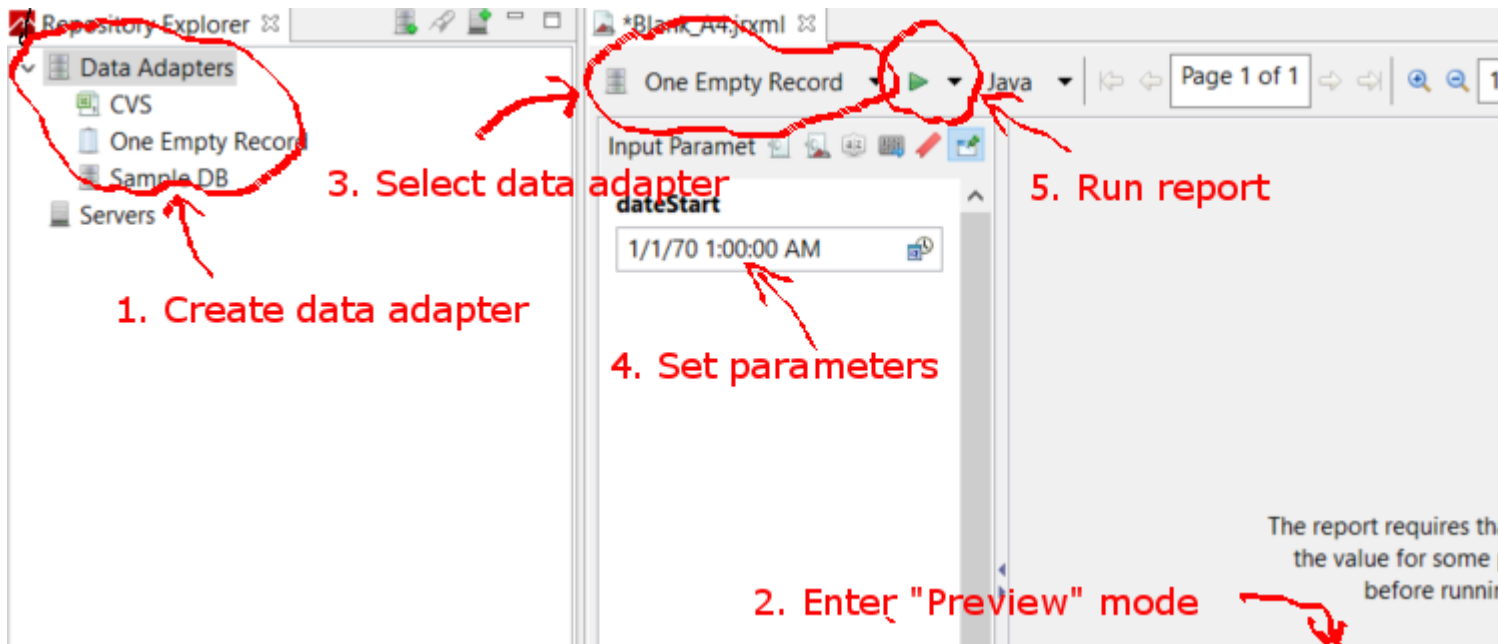
Parameter	Säule
JasperPrint	Die Ausgabe des Füllvorgangs, die in das gewünschte Format exportiert werden kann
Berichtsvorlage	Die kompilierte Entwurfsdatei <code>.jasper</code>
Parameter	Der Parameter <code>Map</code> , der, falls definiert, durch <code>\$P{key}</code> im Bericht referenziert werden kann
Datenquelle	Eine <a href="#">net.sf.jasperreports.engine.JRDataSource</a>
Verbindung	Eine Datenbankverbindung <a href="#">java.sql.Connection</a>

## Examples

### Mit IDE (Integrierte Entwicklungsumgebung)

## JasperSoft Studio

1. Wenn eine Datenquellen- oder Datenbankverbindung zum Ausfüllen des Berichts erforderlich ist, erstellen Sie Ihren Datenadapter im Repository Explorer, indem Sie mit der rechten Maustaste auf "Datenadapter" klicken und "Datenadapter erstellen" auswählen.
2. Geben Sie den Vorschaumodus, indem Sie die Registerkarte **Vorschau** Auswahl (keine Fehler in geruhen müssen vorhanden sein)
3. Wählen Sie die gewünschte Datasource aus (wenn keine Datenquelle erforderlich ist, wählen Sie "Ein leerer Datensatz" aus.)
4. Stellen Sie den Parameter wie gewünscht ein
5. Füllen Sie den Bericht, indem Sie auf den grünen Pfeil "Bericht ausführen" klicken.



Füllen Sie die JasperReport-Vorlage mit Java

## Allgemeine Anforderungen

Alle Berichte, unabhängig davon, wie die Daten dargestellt werden, nehmen einen Pfad zur Berichtsvorlage und eine Parameterzuordnung. Die Variablen werden in allen folgenden Beispielen verwendet:

```
// 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";
```

Die Verwendung einer `.jrxml` Datei erfordert einen zusätzlichen Kompilierungsschritt, der in den meisten Situationen nicht erforderlich ist. Wenn Sie keine benutzerdefinierte Software geschrieben haben, um die `.jrxml` vor der `.jrxml` des Berichts zu ändern (z. B. dynamisch Spalten hinzufügen oder entfernen), verwenden Sie die `.jasper` Datei wie in den folgenden Beispielen gezeigt.

## Verwenden einer Datenbankverbindung

```
// 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);
```



# Verwenden einer benutzerdefinierten Datenquelle

```
// 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);
```

---

## Ohne Datenquelle, unbenutztes Detailband

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

Ohne Daten muss das Attribut `whenNoDataType="AllSectionsNoDetail"` für das `JasperReport` Element gesetzt werden, andernfalls wird ein leerer (leerer) Bericht generiert.

Bericht ausfüllen online lesen: <https://riptutorial.com/de/jasper-reports/topic/3958/bericht-ausfullen>

# Kapitel 4: Export nach xls / xlsx

## Examples

### Mit Java

#### Export in das xlsx-Format

```
try (InputStream inputStream = JLoader.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();
}
```

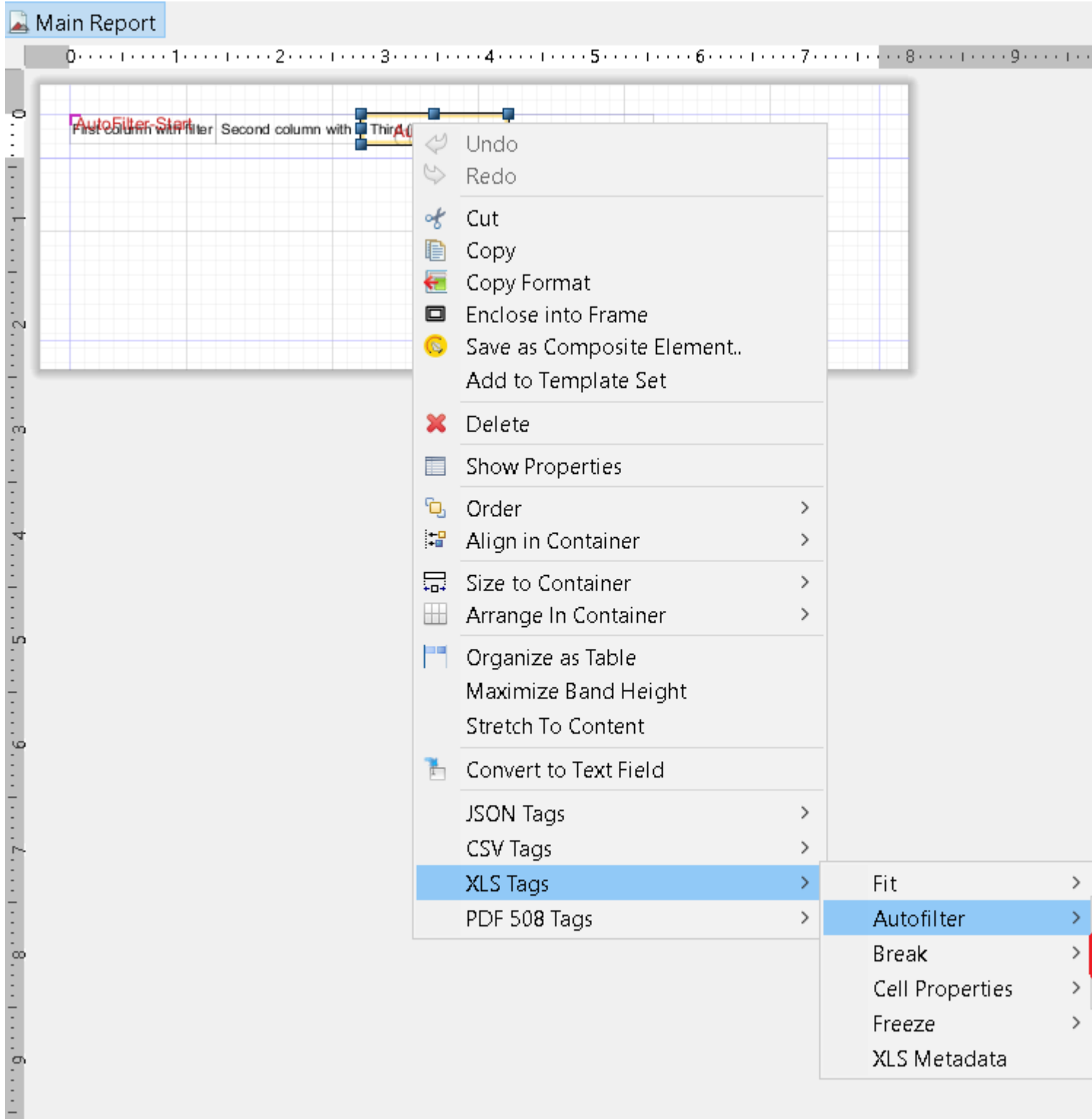
### Autofilter für Spalten hinzufügen

Die Verwendung der Eigenschaft [net.sf.jasperreports.export.xls.auto.filter](#) ermöglicht das Hinzufügen von Autofilter in der generierten XLS-Datei.

```
<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>
```

Die Eigenschaft kann in *Jaspersoft Studio* mit Hilfe des Kontextmenüs oder manuell durch Bearbeiten der *jrxml*-Datei festgelegt werden.



Export nach xls / xlsx online lesen: <https://riptutorial.com/de/jasper-reports/topic/5008/export-nach-xls-xlsx>

[xls---xlsx](#)

---

# Kapitel 5: Font-Erweiterungen

## Examples

### Erstellen und Verwenden von Font-Erweiterungen

Erstellen Sie eine Schriftarterweiterung mithilfe der IDE. Weitere Informationen finden Sie in der Dokumentation zu [iReport](#) oder [Jaspersoft Studio](#) . Die Font-Erweiterung kann auch [manuell erstellt werden](#) .

---

## Was sind Font-Erweiterungen?

Mit einem `textElement` Sie eine Schriftart angeben (wenn nicht die Standardschriftart `SansSerif` verwendet wird).

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

Um die Font-Metrik (für Zeilenumbrüche, Ausrichtung usw.) zu berechnen und die Schriftart korrekt darzustellen, muss die **Schriftart in der JVM** (Java Virtual Macchine) zugeordnet werden. Sie können die Schriftartdatei direkt in der JVM installieren. Dies ist jedoch nicht ermutigend

Aus dem JasperReport Ultimate Guide:

Es wird dringend empfohlen, nur von Schrifterweiterungen abgeleitete Schriftarten zu verwenden, da nur so sichergestellt werden kann, dass die Schriftarten für die Anwendung verfügbar sind, wenn die Berichte zur Laufzeit ausgeführt werden. Die Verwendung von Systemzeichensätzen birgt immer das Risiko, dass die Berichte nicht ordnungsgemäß funktionieren, wenn sie auf einem neuen Computer bereitgestellt werden, auf dem diese Zeichensätze möglicherweise nicht installiert sind

---

## Standardschrifterweiterung

JasperReports bieten eine Standardschrifterweiterung (siehe Maven-Distribution "jasperreports-fonts.jar"). Wenn Sie dies zu Classpath hinzufügen, können Sie die folgenden `fontName` verwenden, ohne Ihre eigene Font-Erweiterung zu erstellen

DejaVu Sans  
DejaVu Serif  
DejaVu ohne Mono

# Häufige Probleme

Probleme, die bei der Verwendung von Schriftarten in pdf (itext) zu berücksichtigen sind:

- Wenn beim Exportieren in PDF der Text nicht korrekt dargestellt wird (fehlende Teile, Zeichen nicht angezeigt, nicht korrekt umbrochen oder die Größe korrekt eingestellt), fehlen wahrscheinlich die **Schrift-Erweiterungen** .
- Ist die tatsächliche `.ttf` **unterstützt** ( [Opentype](#) ) und kann die Schrift tatsächlich den Charakter **machen**? Nicht alle Schriftarten rendern alle Zeichen in `UTF-8` .
- Ist die **richtige Kodierung** an iText übergeben? In Zweifelsfällen (oder allgemein) verwenden Sie die **Kodierung** `Identity-H` dies wird für neuere PDF-Standards empfohlen und gibt Ihnen die Möglichkeit, verschiedene Kodierungen zu mischen.
- Ist die Schrift so **eingebettet** , dass ein von mehreren Computern freigegebenes PDF den Inhalt anzeigen kann, auch wenn die Schrift nicht installiert ist? Wenn es sich bei der Schriftart nicht um eine der [14 Standard Type 1-Schriftarten](#) handelt, muss sie immer eingebettet werden.

Beachten Sie, dass die von jasper report verwendete Version von iText nicht alle Schriftarten ( [Ligaturizer-Problem](#) ) `ttf` Sie können die `ttf` Schrift und die Kodierung direkt [testen](#).

Font-Erweiterungen online lesen: <https://riptutorial.com/de/jasper-reports/topic/5773/font-erweiterungen>

---

# Kapitel 6: Kompilieren Sie JasperReports .jrxml zu .jasper

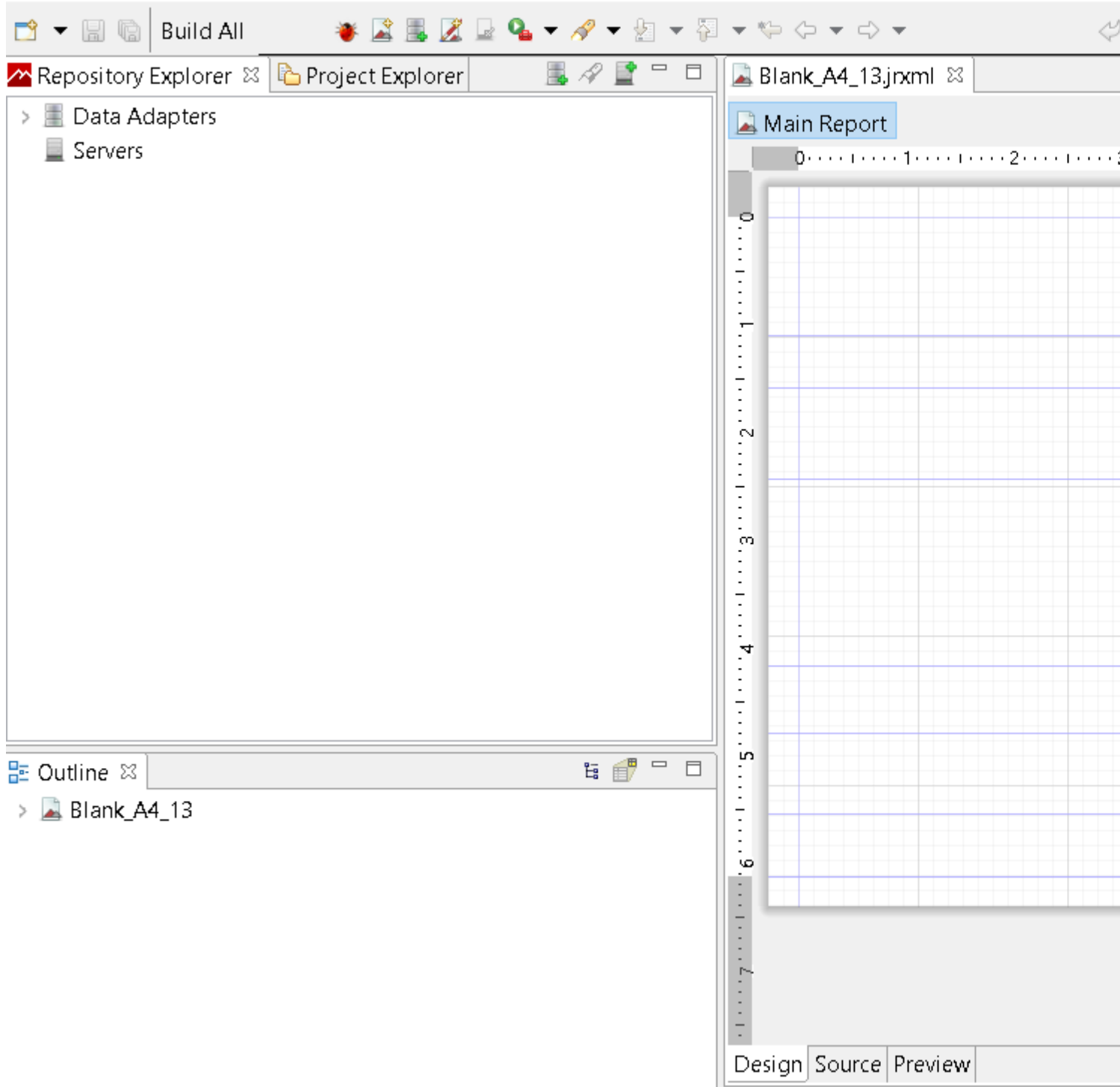
## Examples

### Mit IDE (Integrierte Entwicklungsumgebung)

In IDE [Jaspersoft Studio](#) ( *JSS* ) oder der älteren Version von [iReport Designer](#) ist es ausreichend, auf **Preview** zu **klicken** .

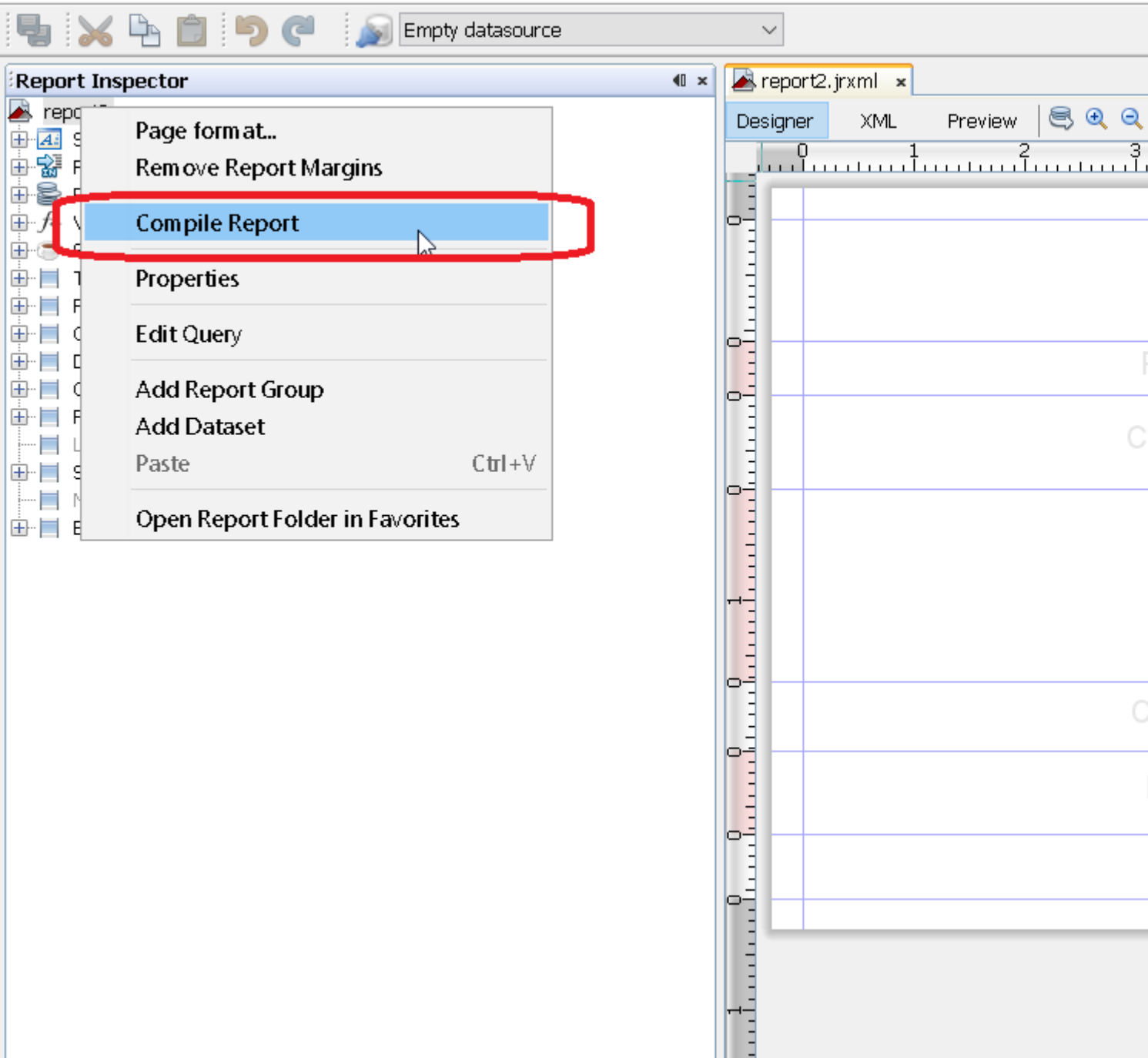
Die JasperReports-Entwurfsdatei `.jrxml` wird automatisch in `.jasper` in demselben Ordner wie `.jrxml` kompiliert, wenn **keine Fehler** vorhanden sind.

Eine andere Möglichkeit ist, die Schaltfläche "*Bericht kompilieren*" in *JSS* zu drücken



oder verwenden Sie das Kontextmenü *"Bericht kompilieren"*, das vom *Berichtsinspektor* in *iReport* aufgerufen wird





## Mit Apache Ant

```
<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>
```

Das Apache Ant-Build-Tool muss korrekt auf Ihrem System installiert sein

## Mit Java

Es ist zwar möglich, `.jrxml` Dateien mit Java-Code in `.jasper` Dateien zu kompilieren, dies führt jedoch zu einem Performance-Hit, der am besten vermieden wird, `.jrxml` Dateien mithilfe der IDE `.jrxml` werden. `.jrxml` kann das Kompilieren von `.jrxml` Dateien mit dem [JasperCompileManager](#) wie folgt durchgeführt werden:

```
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
```

## Mit Apache Maven

Das [JasperReports-Plugin](#) von [Alex Nederlof](#) ist eine gute Alternative zum angegebenen [org.codehaus.mojo:jasperreports-maven-plugin](#) plugin.

Das Hinzufügen eines Plugins ist eine typische, einfache Prozedur:

```
<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>
```

Der Befehl zum Kompilieren mit *Maven* :

```
mvn jasperreports:jasper
```

Die *Jasper*-Dateien werden im Ordner `${project.build.directory}/jasper` erstellt (z. B. in `/target/`

*jasper* ).

Kompilieren Sie JasperReports .jrxml zu .jasper online lesen: <https://riptutorial.com/de/jasper-reports/topic/4943/kompilieren-sie-jasperreports--jrxml-zu--jasper>

# Kapitel 7: Unterberichte verwenden

## Parameter

Parameter	Einzelheiten
<b><i>parametersMapExpression</i></b>	Die Karte mit Parametern. <i>Nicht benötigt</i>
<b><i>UnterberichtParameter</i></b>	Das Paar aus Name und Wert (mit <b><i>subreportParameterExpression festgelegt</i></b> ). <i>Nicht erforderlich</i> Mehrere Parameter können an den Unterbericht übergeben werden
<b><i>connectionExpression</i></b>	Verbindung zum Abrufen von Daten. <i>Nicht benötigt</i>
<b><i>dataSourceExpression</i></b>	Ausdruck für die Übergabe der Datenquelle. <i>Nicht benötigt</i>
<b><i>subreportExpression</i></b>	Der Pfad / URI des Unterberichts oder sogar das JasperReport-Objekt. <i>Nicht benötigt</i>
<b><i>Rückgabewert</i></b>	Das Paar Name und Wert. <i>Nicht erforderlich</i> Mehrere Werte können vom Unterbericht an den Hauptbericht zurückgegeben werden

## Bemerkungen

- Unterberichte können zum Erstellen komplexer Berichte verwendet werden. Die Wiederverwendung vorhandener Berichte ist ein weiteres Ziel der Verwendung von Unterberichten.
- Der Unterbericht wird als Teil des Hauptberichts angezeigt, wenn das Element `<subreport>` verwendet wird.
- Der Wert des Parameters ***subreportExpression*** unterscheidet sich bei der Verwendung auf *JasperReports Server* oder nur durch das *JasperReports*- Framework (einige APIs verwenden oder in IDE verwenden).

Für *JasperReports Server* sieht das so aus:

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

Für die Verwendung von nur *JasperReports Engine*:

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

Die großartige Erklärung von @AndreasDietrich finden Sie unter [JasperServer: Der Unterbericht](#) kann nicht gefunden werden

- Aus einigen Gründen kann der Unterbericht als gemeinsamer Bericht verwendet werden - ohne Aufruf des `<subreport>` (mithilfe des `<subreport>` -Elements). Der Unterbericht ist immer ein Bericht.

## Examples

### Übergabe der Verbindung zum Unterbericht; Rückgabewerte zurück zum Masterbericht

Dies ist ein Ausschnitt des Hauptberichts. Zwei Parameter und die Verbindung (z. B. *jdbc*) werden an den Unterbericht übergeben. Ein Wert wird aus dem Unterbericht an den Hauptbericht zurückgegeben. Dieser Wert (*Variable*) kann im Hauptbericht verwendet werden

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

<subreportParameterExpression><![CDATA[{$someMasterReportParameter}]]></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>
```

### Datenübergabe an Unterbericht übergeben

Dies ist ein Ausschnitt des Hauptberichts. Die Datenquelle wird mithilfe des Konstruktors `net.sf.jasperreports.engine.data.JRBeanCollectionDataSource` an den Unterbericht **übergeben**

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

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

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

Unterberichte verwenden online lesen: <https://riptutorial.com/de/jasper-reports/topic/5452/unterberichte-verwenden>

# Credits

S. No	Kapitel	Contributors
1	Erste Schritte mit Jasper-Berichten	<a href="#">Alex K</a> , <a href="#">Community</a> , <a href="#">Dave Jarvis</a> , <a href="#">Petter Friberg</a>
2	Als PDF exportieren	<a href="#">Alex K</a> , <a href="#">Dave Jarvis</a> , <a href="#">Petter Friberg</a> , <a href="#">RamenChef</a>
3	Bericht ausfüllen	<a href="#">Alex K</a> , <a href="#">Dave Jarvis</a> , <a href="#">Petter Friberg</a>
4	Export nach xls / xlsx	<a href="#">Alex K</a>
5	Font-Erweiterungen	<a href="#">Dave Jarvis</a> , <a href="#">Petter Friberg</a>
6	Kompilieren Sie JasperReports .jrxml zu .jasper	<a href="#">Alex K</a> , <a href="#">Dave Jarvis</a> , <a href="#">Petter Friberg</a>
7	Unterberichte verwenden	<a href="#">Alex K</a>