

iReport und JasperReports Anleitung

von Daniel Quathammer & Andre Knieschewski

Inhaltsverzeichnis

1 JasperReports installieren und konfigurieren.....	2
1.1 Voraussetzungen.....	2
1.2 JasperReports und iReport.....	3
1.2.1 iReport.....	3
1.2.2 Einstellungen.....	3
1.3 Entwurfsbereich.....	4
1.4 JasperReports als Kommandozeilen-Tool.....	5
2 Berichtsdesign.....	6
2.1 Sortierung und Filtern.....	8
2.2 Listen.....	10
2.3 Gruppierung.....	11
2.3.1 Gruppen definieren.....	11
2.3.2 Seitenumbrüche.....	13
2.3.3 Gruppensummen.....	14
2.3.4 Gruppierung ohne eigenes Band.....	15
2.3.5 Gruppierung mit virtuellen Dimensionen.....	17
2.4 Tabellengestaltung.....	17
2.4.1 Summen, Seitenzahlen.....	17
2.4.2 Seitenumbrüche steuern.....	18
2.4.3 Zeilen alternierend einfärben.....	20
2.5 Unterberichte	24
2.5.1 Konfiguration des Hauptberichts.....	24
2.5.2 Konfiguration des Unterberichts.....	29
2.5.3 Einhängen des Berichts in Edustore.....	35
2.6 Kreuztabellen.....	35
2.6.1 Bedienung des Assistenten für Kreuztabellen.....	35
2.6.2 Nachbearbeitung des Kreuztabellen-Assistenten.....	39
2.6.3 Kreuztabellen im Gruppenfuß.....	40
2.6.4 Kreuztabellen mit virtuellen Dimensionen.....	40
2.7 Diagramme.....	41
2.7.1 Einfache Diagramme.....	42
2.7.2 Diagramme nach Tabelle ausgeben.....	46
2.7.3 Diagramme mit berechneten Werten.....	46
2.8 Schriften.....	59
2.8.1 Allgemeines zu Schriften.....	59
2.8.2 Schriften in iReport einbinden.....	60
2.8.3 Default-Schriftart in iReport.....	63
2.8.4 Fonts auf dem Server	63
2.9 Effiziente Formatierung.....	65
2.10 Expression Editor.....	67
2.10.1 Bedingungen.....	68
2.10.2 Umgang mit Zeichenketten.....	69

2.10.3Umgang mit Datumsfeldern.....	69
2.10.4Besonderheiten / Workarounds.....	70
2.11Fortgeschrittene Techniken bei Feldern.....	70
2.12Einrücken von Feldern.....	71
2.13Berechnungen.....	72
2.13.1Prozentrechnung.....	72
3Nutzung in Edustore.....	73
3.1Berichtsentwicklung bei Grunddaten und Basisberichten.....	73
3.1.1Allgemein.....	73
3.1.2Datenquelle erzeugen.....	74
3.1.3Template erzeugen.....	76
3.1.4Template einbinden.....	77
3.1.4.1Reportdesign verlinken.....	77
3.1.4.2Probleme beim Deployment im Server.....	79
3.1.4.2.1Feldnamen im Berichtsdesign weichen vom Datenbank-Feldnamen ab	79
3.1.4.2.2Korrektur der Feldnamen im Berichtsdesign.....	80
3.2Zuordnung von Default-Feldbelegungen zu Stylesheets.....	80
3.3Kurzanleitung Direktaufruf von Jasper-Berichten von der Maske.....	84
3.4Spezielle Variablen.....	84
3.5Kompatibilität von iReport mit JasperReports.....	85
3.5.1.1iReport Grafiken in JasperReports.....	86
3.5.1.2Weitere Bibliotheken von iReport.....	87
4Troubleshooting.....	87
4.1Report Problem: Class not set for bucket: <<Variable>>.....	87
4.2Die jasper Datei.....	88
5Weitere Dokus.....	88

1 JasperReports installieren und konfigurieren

1.1 Voraussetzungen

- Da es ein paar Bugs in den verschiedenen Versionen gibt, empfehlen wir zu der Erstellung von JasperReport Berichten die iReport Version 4.1.3, auch wenn in Edustore die Version 5.1.0 verwendet wird. Download der passenden Version von iReport von <http://sourceforge.net/projects/ireport> (dort auf "files"->"iReport" klicken, dann erhalten Sie Links auf die jeweilige Version.
- Für Demos & Deployment auch <http://jasperforge.org/projects/jasperreports> sowie Apache ANT <http://ant.apache.org/>
- Java 1.5 JRE oder höher

Hinweis: In Edustore ist derzeit die Version 5.1.0 von JasperReports installiert. Da in der iReport Version 5.1.0 uns bereits Bugs bekannt sind empfehlen wir aus Erfahrung die iReport Version 4.1.3.

1.2 JasperReports und iReport

JasperReports ist das Programm, das fertige Berichte erzeugen kann. Es besitzt (außer einer Vorschauansicht der Berichte) keine graphische Oberfläche. Für das Entwerfen von Berichten ist z.B. iReport geeignet, bis zur Version 5.5 das "Standardwerkzeug" für JasperReports.

Hinweis: Ende 2013 wurde von der Fa. Jaspersoft bekannt gegeben, dass iReport ab Version 5.5 nicht mehr das "Standardwerkzeug" für JasperReports ist, sondern das Eclipse-Plugin JasperStudio. Es werden zwar noch Bugfixes für iReport angeboten bis Ende 2015, aber langfristig werden wir auf JasperStudio umsteigen (müssen).

Um es noch ein wenig zu veranschaulichen könnte man sagen, dass wenn ein JasperReport das Worddokument wäre, dann wäre iReport z.B. Microsoft Word.

1.2.1 iReport



Das Paket enthält bereits die Laufzeitumgebung JasperReports, diese muss also nicht separat installiert werden. Nichtsdestotrotz empfiehlt sich der Download, denn in dem JasperReports-Paket befinden sich viele lehrreiche Demo-Berichte.

Die Windows-Version beinhaltet auch einen Installer. Um diesen zu nutzen sind Admin-Rechte erforderlich.

Alternativ bzw. unter Linux ist iReport ohne Installation direkt nach dem Entpacken lauffähig. Wenn Sie auch Scriptlets erstellen wollen, müssen Sie nicht nur eine Java-Laufzeitumgebung (JRE) installieren, sondern ein JDK, und die Datei \$JAVA_HOME/lib/tools.jar nach iReport- \llcorner Versionsnr. \ggcorner /ireport/libs kopieren. Danach müssen Sie iReport neu starten.

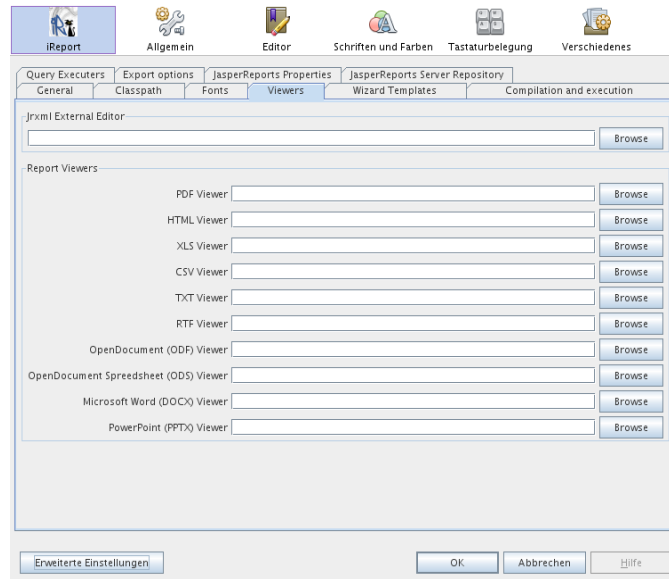
Start unter Windows: iReport- \llcorner Versionsnr. \ggcorner /bin/ireport.exe

Start unter Linux: iReport- \llcorner Versionsnr. \ggcorner /bin/ireport

1.2.2 Einstellungen

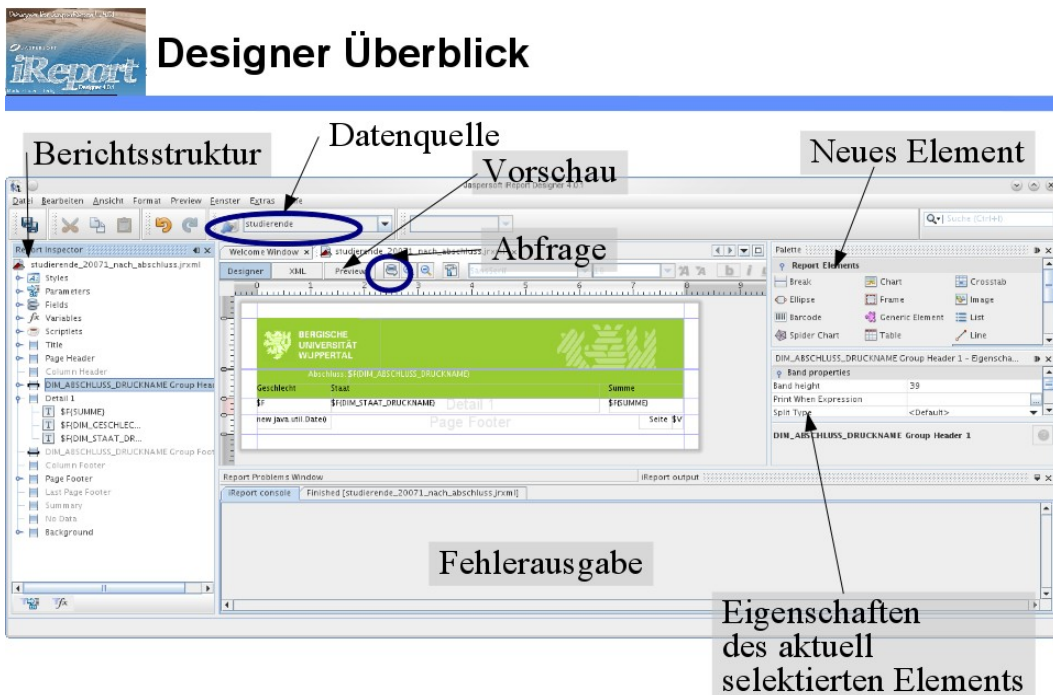
Als erstes sollten Sie die Anwendungen einstellen, mit denen iReport generierte Berichte anzeigen soll:

Im Menü Extras-> Optionen wählen Sie im Reiter "iReport"- "Viewers" die entsprechenden Anwendungen:



1.3 Entwurfsbereich

Der Entwurfsbereich bzw. "Designer" hat einen recht komplexen Aufbau. Sie können die Fenster und Seitenbereiche zwar recht flexibel anpassen, aber hier ist die Auslieferung:



© 2011 D. Quathamer

Die wichtigen Bereiche sind hervorgehoben:

- Berichtsstruktur: Welche Seitenbereiche gibt es?
- Datenquelle: Welche Datenbank wird genutzt?

- Abfrage: Was ist die konkrete Abfrage für den Bericht?
- Elemente: Welche Elemente kann ich einfügen, und welche Eigenschaften haben sie?
- Vorschau: wie sieht der Bericht "gedruckt" aus?
- Fehlerausgabe: Jeder Bericht wird "kompiliert", dazu gibt es eine Fehlerausgabe
Im Reiter "Preview" kann man sich eine Vorschau des Ausdrucks anzeigen lassen.

1.4 JasperReports als Kommandozeilen-Tool

In der JasperReports-Distribution befindet sich ein ANT-Script, das mehrere Berichte erzeugen kann.

```
cd demo/samples
```

Beispiel xmldatasource:

```
cd xmldatasource
```

```
ant -p
```

Die Ausgabe ist:

```
Buildfile: build.xml
Shows how the built-in XPath-based data source could be used for reporting out of XML data.
Main targets:

clean          Deletes all the generated files.
compile        Compiles the XML report design and produces the .jasper file.
csv           Generates a CSV version of the report by converting the .jrprint file.
decompile      Decompile a .jasper file into a .jrxml file.
docx          Generates a DOCX version of the report by converting the .jrprint file.
fill          Fills the compiled report design with data and produces the .jrprint file.
html          Generates an HTML version of the report by converting the .jrprint file.
javac         Compiles the java source files of the test application.
jxl           Generates an XLS version of the report by converting the .jrprint file using the J
ExcelApi library.
ods           Generates an ODS version of the report by converting the .jrprint file.
odt           Generates an ODT version of the report by converting the .jrprint file.
pdf           Generates a PDF version of the report by converting the .jrprint file.
pptx          Generates a PPTX version of the report by converting the .jrprint file.
print         Sends the report in the .jrprint file to the printer.
rtf           Generates an RTF version of the report by converting the .jrprint file.
test          Runs the sample.
view          Launches the report viewer to preview the report stored in the .jrprint file.
viewDesign1  Launches the design viewer to preview the master compiled report design.
viewDesign2  Launches the design viewer to preview the subreport compiled report design.
viewDesignXml1 Launches the design viewer to preview the master XML report design.
viewDesignXml2 Launches the design viewer to preview the subreport XML report design.
viewXml      Launches the report viewer to preview the generated report stored in an XML file.
writeApi     Generates a .java version of the report by converting the .jrxml file.
writeApiXml  Generates a .jrxml version of the report by executing the API version of the repor
t design.
xhtml        Generates an XHTML version of the report by converting the .jrprint file.
xls          Generates an XLS version of the report by converting the .jrprint file using the P
OI library.
xlsx         Generates an XLSX version of the report by converting the .jrprint file.
xml          Generates an XML version of the report by converting the .jrprint file.
xmlEmbed     Generates an XML version of the report by converting the .jrprint file.
Default target: test
```

Es werden, je nach Auswahl pdf-, Excel- oder andere Berichte erzeugt. Mit dem Target "test" werden alle Targets durchlaufen.

```
ant test
```

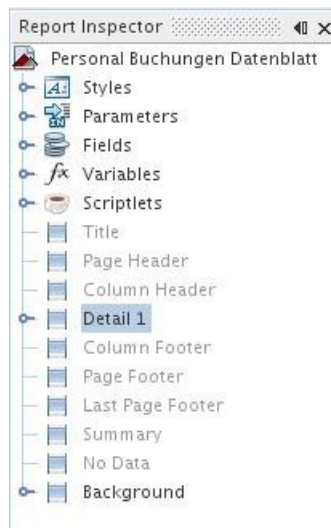
Achtung: manche Demos brauchen eine laufende HSQLDB (siehe demo/hsqldb).

2 Berichtsdesign

Das Erstellen eines Berichtes erleichtert iReport wesentlich. Dennoch wird man sich damit etwas beschäftigen müssen, um ein vernünftiges Ergebnis zu erhalten.

Zuerst muss man den Bericht öffnen. Das geschieht über „Datei“ → „Open“ → Bericht auswählen → auf „Öffnen“ klicken. Nun noch oben links die für diesen Bericht erstellte Datenquelle auswählen und man kann loslegen.

iReport bezeichnet die Ausgabebereiche als Bänder. Ein Band ähnelt einer Zeile. Es reicht über die komplette Seitenbreite. Die Höhe kann dabei selbst bestimmt werden und wird dem Informationsgehalt angepasst. Das Band „Detail 1“ ist so etwas wie das Hauptband, in dem dann die Informationen zeilenweise ausgegeben werden. Die linke Spalte könnte z.B so aussehen:



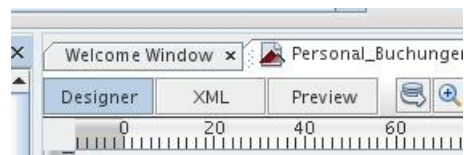
Hier sieht man, dass das Band „Detail 1“ aktiv ist und die anderen wie „Title“ noch nicht. In den Gruppen „Fields“ und „Variablen“ befinden sich die Felder (Spalten) aus der Ergebnistabelle und interessante Variablen wie Seitenzahl, die man nun einfach in das bestehende Band „Detail 1“ per „Drag and Drop“ ziehen kann.



Wenn man z.B. in der Gruppe Fields das Feld „Name“ hat und dies in das Band „Detail 1“ zieht, befindet sich ein Platzhalter im Entwurfsbereich:



Wenn Sie dann auf „Preview“ klicken, wird eine Vorschau des Berichts angezeigt.



Es werden alle Namen aufgelistet, die in der Datenquelle vorhanden sind:



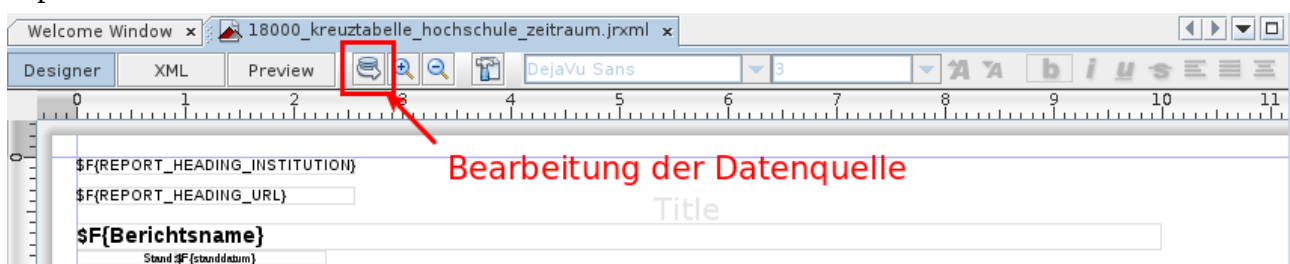
Sie können das Layout der Ausgabe (Schriftart, Größe) über die Eigenschaften des Feldes variieren.

Um wieder zurück in den Entwicklermodus zu kommen klickt man einfach wieder auf den Reiter "Designer".

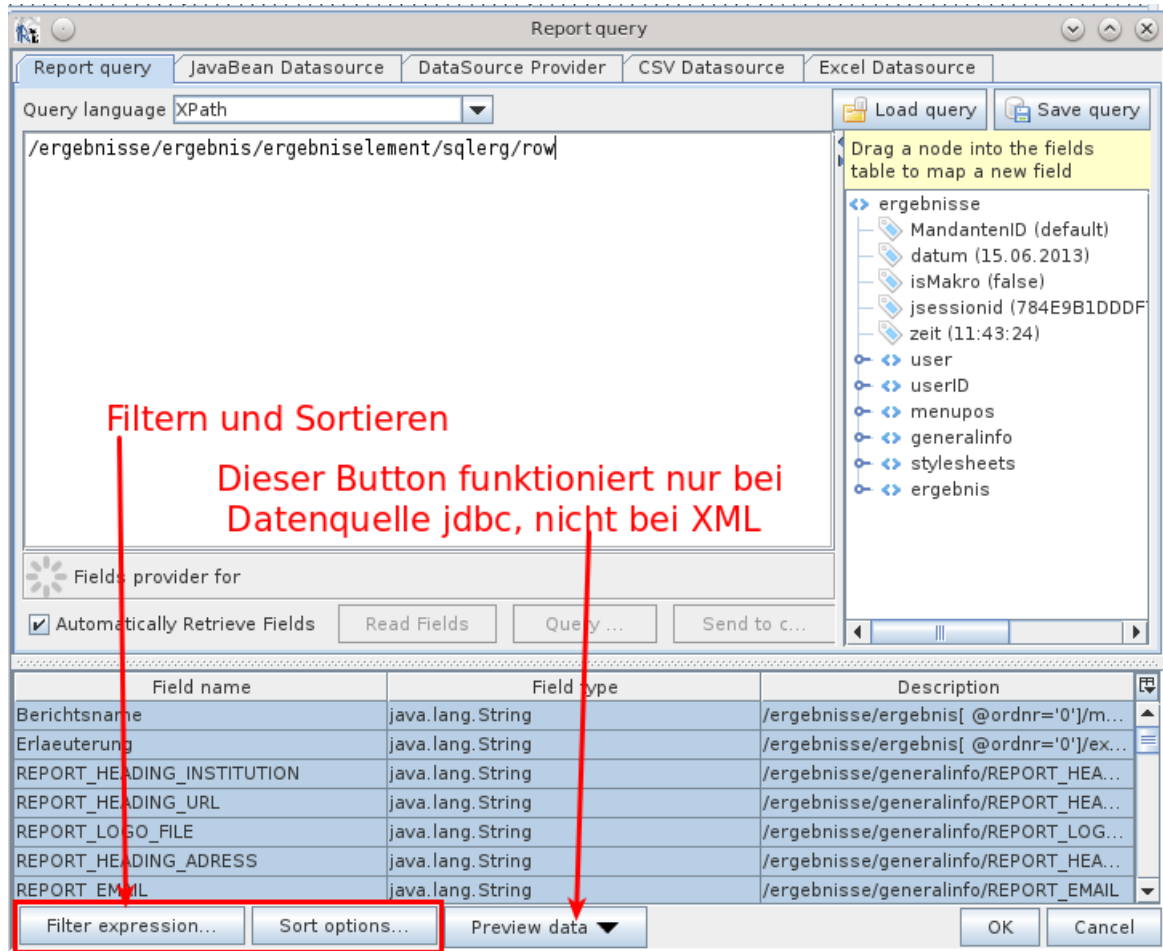
Die Bänder „Title“, „Page“ usw. kann man mit einem Rechtsklick und dann „Add Band“ hinzufügen und auch wieder mit einem Rechtsklick und dann „Delete Band“ wieder entfernen. Diese Bänder haben bestimmte Eigenschaften. Z.B. Band „Titel“ wird nur einmal auf der ersten Seite abgebildet, Band „Page Header/Footer“ auf jeder Seite am Anfang/Ende.

2.1 Sortierung und Filtern

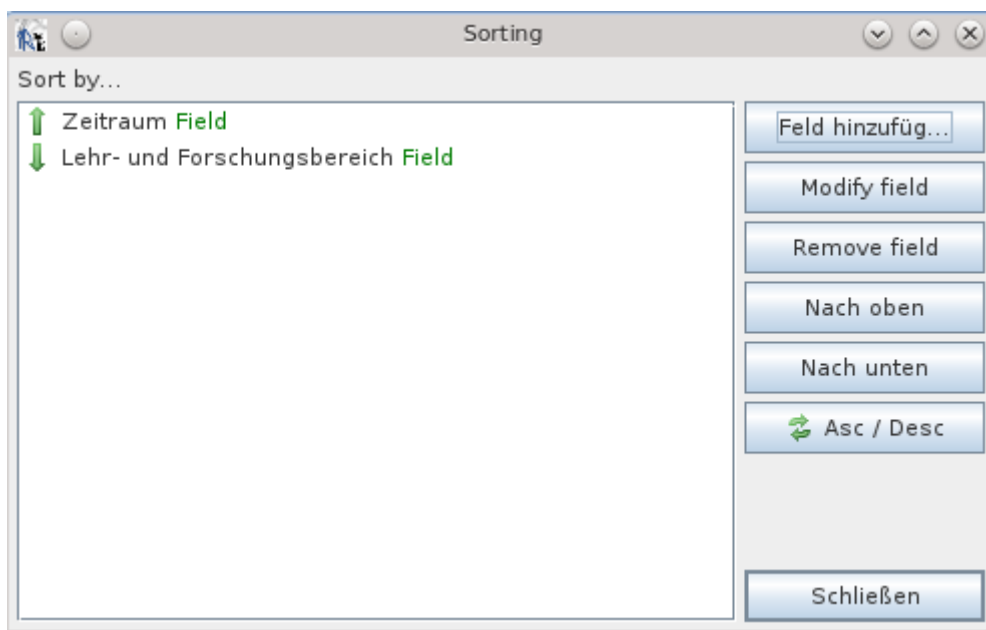
Normalerweise verarbeitet JasperReports die Datensätze zeilenweise in der Reihenfolge, in der sie bereitgestellt werden. Sie können im Entwurfsmodus die Sortierung bzw. Filterung der Datensätze anpassen:



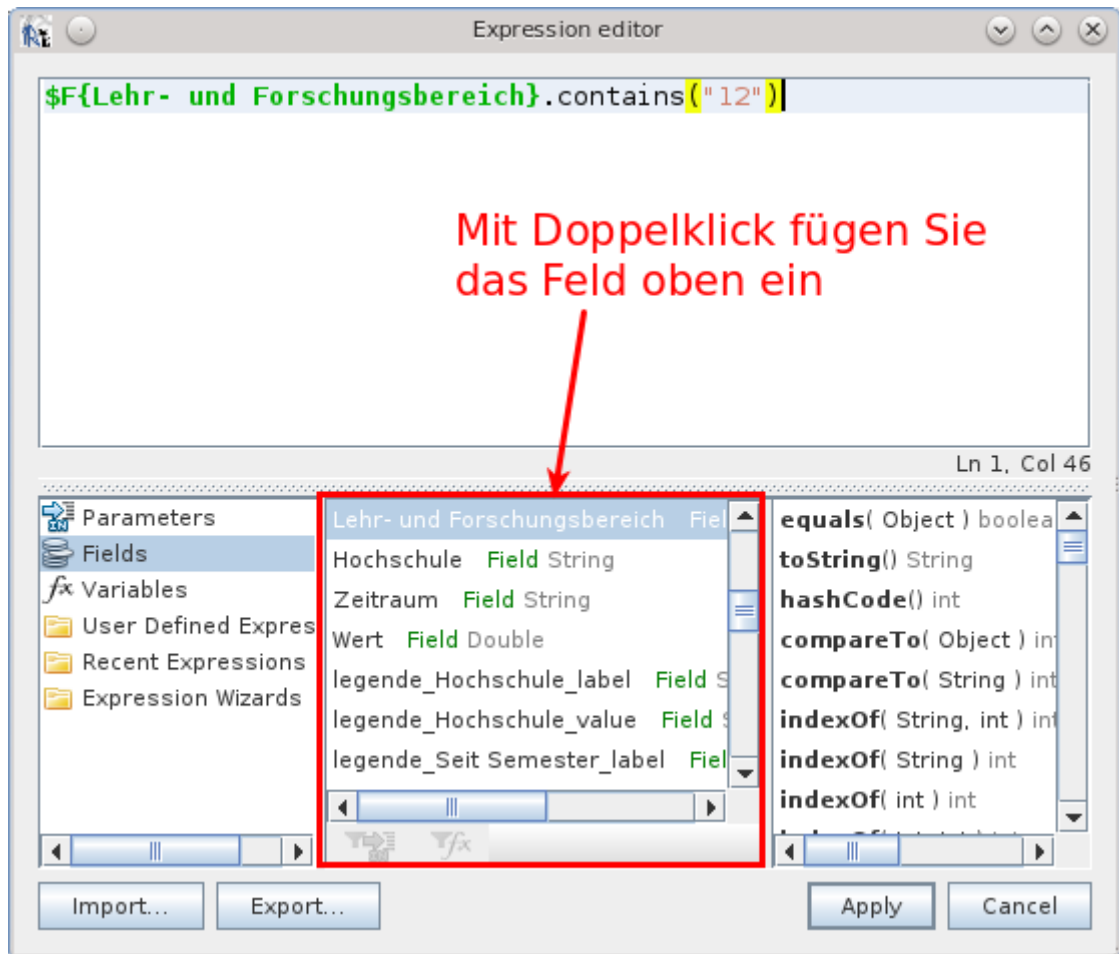
Klicken Sie auf das Icon mit der Tonne und Sie gelangen in die Einstellungen für die Datenquelle.



Im Button Sortieren können Sie nach allen Feldern des Berichts aufsteigend bzw. absteigend sortieren.



Im Button "Filter expression" können Sie eine Formel definieren, um die Gültigkeit von Datensätzen zu definieren:



Z.B. die Expression `".contains("12")` würde auf alle Datensätze filtern, die im Feld die Zeichenkette "12" enthalten. Weitere Infos im Kapitel zum [Expression Editor](#).

2.2 Listen

Listen lassen sich einfach erstellen, indem man die jeweiligen Felder ins Detail-Band zieht. Manchmal kann es vorkommen, dass Textfelder "überlaufen", d.h. die rechteckige Box, die das Feld markiert, ist zu klein für den Feldinhalt. Beispiel:

Studienfeld	Studienbereich	Fächergruppe
A		
Agrarbiologie	Agrarwissenschaften	Agrar- und Forstwissenschaften
Ausländerpädagogik	Pädagogik, Erziehungswissenschaften	Gesellschafts- und Sozialwissenschaften
Architektur	Architektur	Ingenieurwissenschaften
Anthropobiologie	Biologie	Mathematik, Naturwissenschaften
Afrikanistik	Afrikanistik, Altamerikanistik	Sprach- und Kulturwissenschaften
Arboristik	Umwelt- und Landschaftsgestaltung	Agrar- und Forstwissenschaften
Automatisierungstechnik	Elektrotechnik	Ingenieurwissenschaften
Automobiltechnik	Fahrzeug- und Verkehrstechnik	Ingenieurwissenschaften
Asienbau	Maschinenbau	Ingenieurwissenschaften

In diesem Falle muss man das Attribut des jew. Felder (hier "Fächergruppe") "Text Field properties > Stretch with overflow" ankreuzen.

Achtung: wenn das Feld Teil einer tabellarischen Auflistung ist, muss man die anderen Element mit der Eigenschaft "Stretch type → Relative to tallest object" versehen, hier z.B. die Felder "Studienfeld" und "Studienbereich".

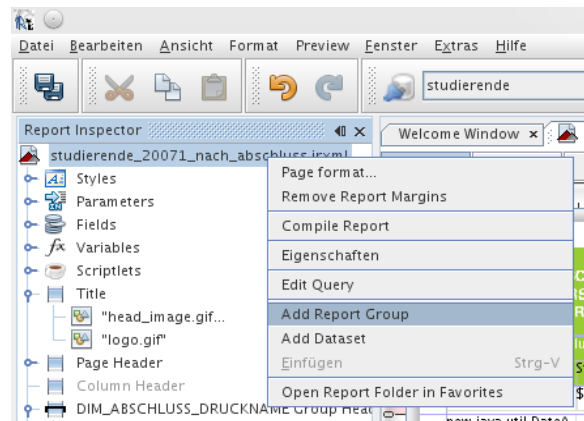
Bei vielen Zeilen ist es übersichtlicher, jede zweite Zeile farbig zu hinterlegen. Dieser Trick ist im Abschnitt Tabellengestaltung [erläutert](#).

2.3 Gruppierung

Wichtige Vorbemerkung zum Gruppieren: JasperReports setzt bei der Gruppierung voraus, dass die Datensätze der Datenquelle nach den Gruppierungsmerkmalen sortiert sind. Alle Merkmale, nach denen Sie gruppieren, sollten also entweder direkt in der Datenquelle sortiert sein, oder in iReport im [Berichtsentwurf](#).

2.3.1 Gruppen definieren

Zu den Standardbändern kann man nun auch noch selbst definierte Gruppen hinzufügen. Mit einem Rechtsklick auf den obersten Knoten → „Add Report Group“ öffnet sich ein Fenster in dem man der Gruppe einen Namen geben kann und das Objekt wählt für das die Gruppe erstellt wird.



Achtung: wenn Sie mehrere Gruppierungsebenen benötigen, dann sollte die Reihenfolge des Hinzufügens der Gruppen "top-down" verlaufen, d.h. zuerst die oberste Gruppierung hinzufügen, dann die Gruppe darunter usw.

In dem folgenden Beispiel wird die Gruppe „Monat“ gewählt und in dem nächsten Schritt mit Header und Footer hinzugefügt:

Man zieht das Feld „Namen“ in das Band „Detail“ und „Monat“ in das erstellte Band „Monat Group Header 1“. Aus der Palette rechts oben (wenn nicht vorhanden bekommt man es über „Fenster“ → „Palette“) zieht man einen „Static Text“ in das Band „Monat Group Header 1“ vor der Ausgabe des Monats, benennt das Feld „Monat:“ und noch eine Linie darunter. Zusammen sollte es so aussehen:

Monat: \${F(Monat)}	Monat Group Header 1
\${F(Name)}	Detail 1
	Monat Group Footer 1

Wenn man nun auf „Preview“ klickt erhält man als Ergebnis:

Monat: 1

Nuhr, Dieter

Monat: 2

Köster, Gabi

Krüger, Mike

Pastewka, Bastian

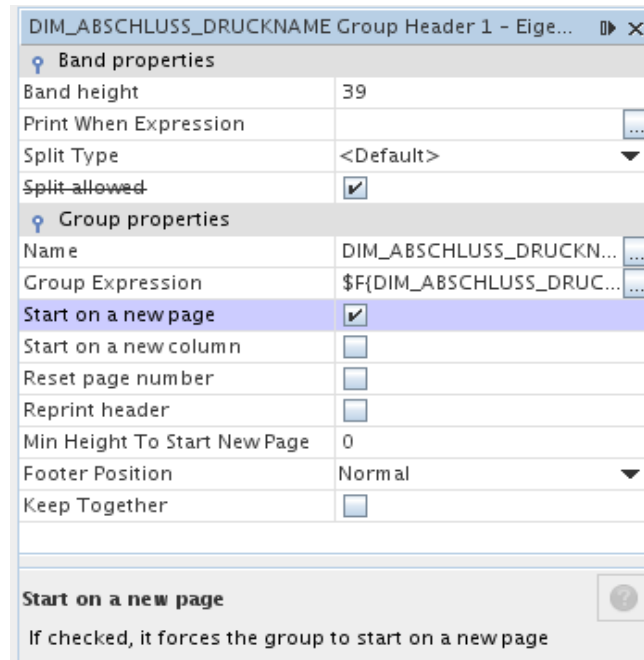
Monat: 5

Hoffmann, Rüdiger

2.3.2 Seitenumbrüche

Damit eine Gruppe nicht durch einen Seitenumbruch auseinandergerissen wird, gibt es einige Einstellungen in dem Band der Gruppe. Dazu einfach auf das jeweilige Band klicken, für welches die Einstellungen vorgenommen werden sollen und dann in „Eigenschaften“ → „Group properties“ → „Keep Together“ anklicken oder wenn man möchte „Reprint header“ um den "Header" der Gruppe am Seitenkopf zu wiederholen.

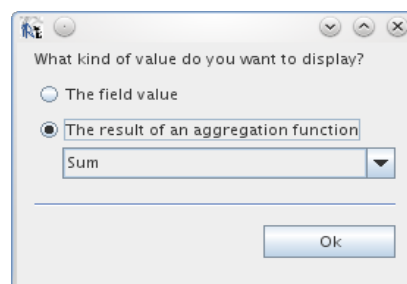
Sie können auch angeben daß vor jedem Gruppierungswechsel ein Seitenumbruch ausgeführt wird:



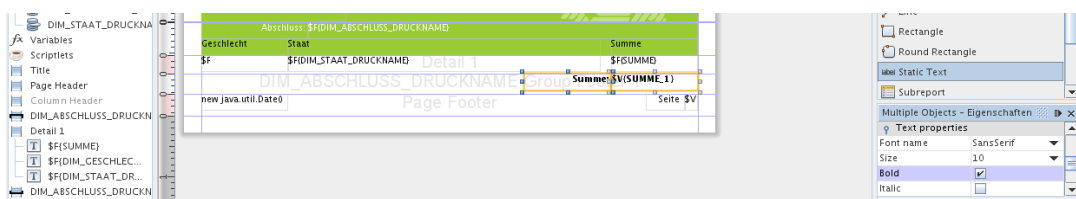
Mit "Start on a new page" wird eine neue Seite bei jedem Gruppenwechsel begonnen.

2.3.3 Gruppensummen

Gruppensummen kann man im Kopf oder Fuß einer Gruppe anlegen. Sie aktivieren z.B. den Gruppenfuß im Report Inspector, und ziehen dann das Feld Summe in den Entwurfsbereich. Beim "Fallenlassen" werden Sie gefragt, ob Sie den jew. Wert oder eine Aggregatfunktion nutzen wollen. Für Zwischensummen wählen Sie "Sum":



Im Entwurfsbereich setzen Sie noch ein Label bzw. "Static Text"-Element rechtsbündig davor, und setzen beide Elemente fett:



2.3.4 Gruppierung ohne eigenes Band

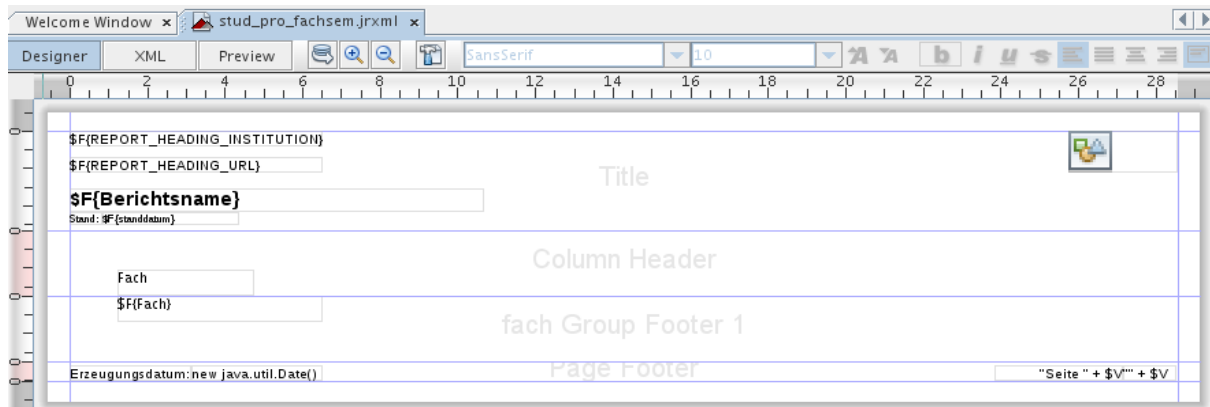
Bei der Gruppierung in JasperReports wird in der Regel ein horizontales Band für die Gruppierungsebene angelegt. Das führt dazu, daß man zeilenorientiert gruppiert. In manchen Fällen kann es gewünscht sein, in den Spalten einer Tabelle zu gruppieren. Hier ein Beispiel für das Ziel:

Studierende Datenblatt
Stand: 17.03.2012

Fach	1. FS	> 13 FS	Gesamt
Angewandte Informatik	208	0	457
Betriebswirtschaft	0	322	2035
Betriebswirtschaft BA	606	0	3005

In dem Beispiel wird in den Zeilen nach dem Merkmal **Fach** gruppiert. Zusätzlich wird noch in den Spalten nach Fachsemestern gruppiert. Um dieses Layout zu realisieren, muss man wie folgt vorgehen:

Erzeugen Sie ein Column Header Band, und ein Group Footer-Band für die Gruppierungsebene Fach. Warum Group Footer? Weil innerhalb des Fachs nochmal gruppiert werden soll, und daher erst nach dem Gruppieren nach Fach nach Fachsemestern gruppiert wird. Ein Detail-Band gibt es hier nicht.



Erzeugen Sie eine neue Variable z.B. mit dem Namen "fs1" für erstes Fachsemester, und geben Sie der Variable folgende Eigenschaften:

fs1 - Eigenschaften	
Eigenschaften	
Name	fs1
Variable Class	java.lang.Integer
Calculation	Sum
Reset type	Group
Reset group	fach
Increment type	None
Increment group	
Incrementer Factory Class	
Variable Expression	$(\{F\{Anzahl\ Fachsemester\}==1\}?\{F\{Summe\}:0$
Initial Value Expression	

Zur Erläuterung: Die Variable filtert die Datensätze gemäß dem Ausdruck in Variable Expression, und tut dies auf der Gruppierungsebene "fach". Die variable Expression ist eine für unsere Zwecke sehr nützliche Java-Kurzfassung einer IF-THEN-Bedingung, und meint:

Wenn der Inhalt des Feldes "Anzahl Fachsemester" gleich 1 ist, dann gib die Summe der jew. Studierenden aus, wenn nicht, dann gib 0 aus.

Analog kann man noch die Variable "fsgr13" für "Fachsemester > 13" anlegen.

Danach kann man die Variablen "fs1" und "fsgr13" in das Band "Fach Group Footer 1" ziehen, und noch das Feld "Summe" für die Gesamtsumme. Dann tragen wir im Column Header Band noch die Spaltenüberschriften ein. Insgesamt sieht Berichtsentwurf dann so aus:

The screenshot shows a report designer interface with the following layout:

- Title:** $\{F\{REPORT_HEADING_INSTITUTION\}$ and $\{F\{REPORT_HEADING_URL\}$
- Header:** $\{F\{Berichtsname\}$ and $\{F\{Stand: \{F\{standdatum\}$
- Column Header:**

Fach	1. FS	> 13 FS	Gesamt
$\{F\{Fach\}$	$\{V\{fs1\}$	$\{V\{fsgr13\}$	$\{V\{Summe_1\}$
- fach Group Footer 1:** (Contains the variable bands fs1 and fsgr13)
- Page Footer:** $\{F\{Erzeugungsdatum: new java.util.Date()\}$ and $\{V\{Seite\} + \{V\{Page\} + \{V\{Page\}$

Und das Ergebnis sieht so aus:

Memtext University			
Studierende Datenblatt			
Stand: 17.03.2012			
Fach	1. FS	> 13 FS	Gesamt
Angewandte Informatik	208	0	457
Betriebswirtschaft	0	322	2035

Fertig. Man könnte nun noch weitere Fachsemester-Intervalle einfügen, oder z.B. eine Summe der Studierenden in der Regelstudienzeit.

2.3.5 Gruppierung mit virtuellen Dimensionen

Wenn Sie nach Feldern gruppieren wollen, die es in Ihrer Datenquelle gar nicht gibt (z.B. nach Aggregationen von Merkmalskombinationen), dann können Sie auch Variablen mit speziellen Formeln anlegen, und diese dann in die Gruppierung aufnehmen. Im Abschnitt zu [Kreuztabellen](#) wird z.B. gezeigt, wie die Fächer Mathe, Physik und Biologie in einen neuen Wert "Naturwissenschaften" transformiert werden.

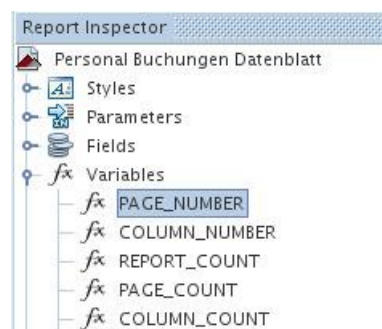
2.4 Tabellengestaltung

2.4.1 Summen, Seitenzahlen

Um es noch ein wenig interessanter zu gestalten fügt man noch das Gehalt hinzu, summiert es für die Monate und gesamt. Dazu noch die Seitenanzahl und einen Seitenzähler und eine Überschrift.

Das Gehalt bekommt man, indem man das Gehalt aus den Feldern neben den Namen zieht. Das selbe macht man auch nochmal in das Band „Monat Group Footer 1“ und wählt hier die Aggregation „Sum“ aus, um die Summe zu erhalten.

Um einen Seitenzähler zu erhalten fügt man zunächst das Band „Page Footer“ hinzu und legt dort die Variable „PAGE_NUMBER“ ab.



Unter „Eigenschaften“ → „Text field properties“ → „Evaluation Time“ wählt man hier „Now“ um die aktuelle Seitenanzahl zu bekommen und „Report“ um die gesamte Seitenanzahl zu erhalten. In dem Beispiel ist die Ausgabe z.B.: „Seite 2 von 6“.

2.4.2 Seitenumbrüche steuern

Damit die drei Personen im Monat 2 nicht durch einen Seitenumbruch auseinandergerissen werden, gibt es einige Einstellungen in dem Band der Gruppe. Dazu einfach auf das jeweilige Band klicken, für welches die Einstellungen vorgenommen werden sollen (hier „Monat“) und dann in „Eigenschaften“ → „Group properties“ → „Keep Together“ anklicken oder wenn man möchte „Reprint header“ um den "Header" der Gruppe am Seitenkopf zu wiederholen.

Beispiel im Designermodus:

Gehälter	
Monat: \${F{Monat}}	Gehalt:
\${F{Name}}	\${F{Brutto}}
Summe: \${V{Brutto_1}}	
Seite: \$ von \$	
Summe Gesamt: \${V{Brutto_2}}	

Beispiel im Preview:

Gehälter

Monat: 1	Gehalt:
Nuhr, Dieter	1.839,00 €
Summe:	1.839,00 €

Monat: 2	Gehalt:
Köster, Gabi	2.315,00 €
Krüger, Mike	3.237,00 €
Pastewka, Bastian	4.714,00 €
Summe:	10.266,00 €

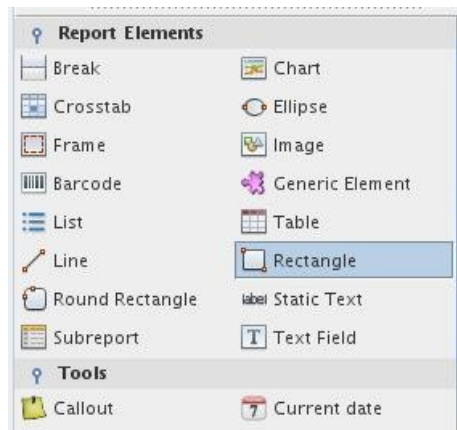
Seite: 1 von 2

Monat: 5	Gehalt:
Hoffmann, Rüdiger	3.513,00 €
Summe:	3.513,00 €
Summe Gesamt:	15.618,00 €

Seite: 2 von 2

2.4.3 Zeilen alternierend einfärben

Um nun bei mehreren Datensätzen auch noch den Überblick in den Zeilen zu behalten kann man jetzt auch noch den Hintergrund der Zeile abwechselnd farbig gestalten. Dazu Zieht man aus den „Reports Elements“ das „Rectangle“ in das „Detail 1“ Band



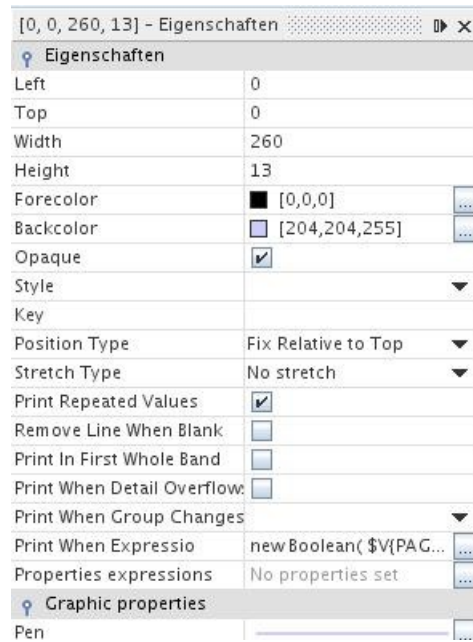
und passt es der Größe des Bandes an.

Gehälter	
Monat: \${F(Monat)}	Gehalt:
\${F(Name)}	\${F(Brutto)}
Summe: \${V(Brutto_1)}	
Seite: \$ von \$	
Summe Gesamt: \${V(Brutto_2)}	

Die Hintergrundfarbe kann man nun in den Eigenschaften in „Backcolor“ ändern. Damit dies nun nicht in jeder Zeile erscheint, sondern abwechselnd, fügt man noch in den Eigenschaften in „Print When Expression“ den Wert `Boolean.valueOf(${V{PAGE_COUNT}.intValue() % 2 == 0 }` ein.

Zur Erläuterung:

Die Variable „PAGE_COUNT“ zählt die Anzahl der Datensätze hoch. Mit „% 2“ wird diese Variable durch 2 geteilt und nur der Restwert ausgegeben. Als Ergebnis erhält man also in jeder ungeraden Zeile eine 1. Dieses Ergebnis wird nun mit der „0“ verglichen. Wenn dieser Ausdruck/Vergleich (expression) wahr ist, wird das Objekt, in diesem Fall der Hintergrund, angezeigt. Bei diesem Ausdruck erhält also jede gerade Zeile den Hintergrund. Möchte man lieber die ungeraden Zeilen hervorheben, schreibt man einfach eine 1 anstelle der 0 hinter „==“.



Das Ergebnis bei einer ungeraden Hervorhebung sieht z.B. so aus:

Gehälter	
Monat: 1	Gehalt:
Nuhr, Dieter	1.839,00 €
Summe:	1.839,00 €
Monat: 2	Gehalt:
Köster, Gabi	2.315,00 €
Krüger, Mike	3.237,00 €
Pastewka, Bastian	4.714,00 €
Summe:	10.266,00 €
Seite: 1 von 2	

Ein weiteres Beispiel welches bereits in der Uni Mannheim benutzt wird ist die Ausgabe einer Rechnungslegungsliste. Das Beispiel zeigt nicht nur ein aufwändigeres Layout, sondern auch, wie man gruppierte Tabellen anlegt: Dabei sind die Zeilen der Tabelle im Detail 1 Band, und die Zwischensummen in den Gruppierungs-Bändern. Die Spaltenüberschriften der Tabelle befinden sich im untersten Group-Header-Band. Im Designermodus sieht das ganze folgendermaßen aus:

UNIVERSITÄT MANNHEIM		Rechnungslegungslistepersonal			
		Einzelbuchungen: \$F"\$F(Jahr)			
Buchungsstelle: Kapitel: \$F{Kapitel} Titel: \$F{Titel} Buchungsstelle: \$F - \$F{Buchungsstelle Text}					
Personal Nummer und Name		Brutto	Soz. Vers.	Zus. Vers.	Summe
KoA \$F					
\$F	\$F{Name}	\$F{Brutto}	\$F	\$F{Zusätzliche}	\$F{Brutto} + \$F
Summe Kostenart:		\$V{Brutto_1}	\$V	\$V{Zusätzliche}	\$V{Brutto_1} +
Summe Kostenstelle: \$F		\$V{Brutto_2}	\$V	\$V{Zusätzliche}	\$V{Brutto_2} +
Stand: \$F{standdatum}		Universität Mannheim			Seite: \$V/\$V

Und im Preview:

Rechnungslegungslistepersonal

Einzelbuchungen: 04.2010

Buchungsstelle: Kapitel: 1420 **Titel:** 42901 **Buchungsstelle:** 1101030001 – Pressestelle

Personal Nummer und Name	Brutto	Soz. Vers.	Zus. Vers.	Summe
KoA 62356				
Summe Kostenart:				
Summe Kostenstelle: 1101020				

Personal Nummer und Name	Brutto	Soz. Vers.	Zus. Vers.	Summe
KoA 62356				
Summe Kostenart:				

Personal Nummer und Name	Brutto	Soz. Vers.	Zus. Vers.	Summe
KoA 63629				
Summe Kostenart:				

Personal Nummer und Name	Brutto	Soz. Vers.	Zus. Vers.	Summe
KoA 63635				

Buchungsstelle: Kapitel: 1420 Titel: 42901 Buchungsstelle: 1101040001 - Stabsst Studium, Lehre GA				
Personal Nummer und Name	Brutto	Soz. Vers.	Zus. Vers.	Summe
KoA 63635				
	€	€	€	€
Summe Kostenart:	€	€	€	€
Summe Kostenstelle: 1101040	€	€	€	€
Personal Nummer und Name	Brutto	Soz. Vers.	Zus. Vers.	Summe
KoA 63635				
	€	€	€	€
Summe Kostenart:	€	€	€	€
Summe Kostenstelle: 1101041	€	€	€	€

Stand: 10.06.2010 Universität Mannheim Seite: 3/361

2.5 Unterberichte

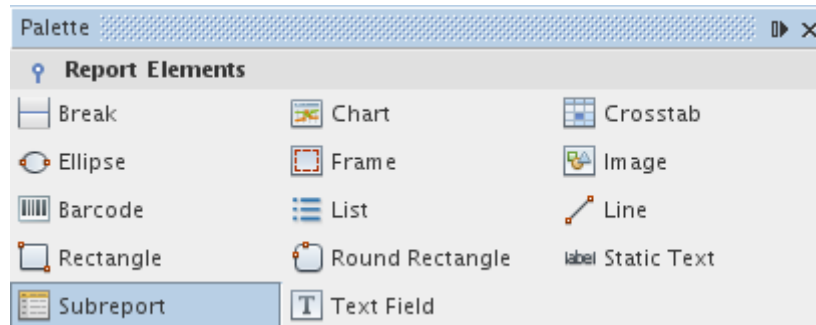
Achtung: In Edustore funktionieren Unterberichte nur, wenn der Dateiname der jrxml-Datei mit "_xmlsource.jrxml" endet. Dadurch wird dem Servlet mitgeteilt, dass die Datenquelle für den Bericht ein XML-Datenstrom ist. Dabei ist mit Performanceeinbußen zu rechnen. Außerdem muss beim Berichtsdesign die gleiche iReport Version genutzt werden wie im Server.

Im folgenden eine Anleitung zur Erstellung eines Unterberichts:

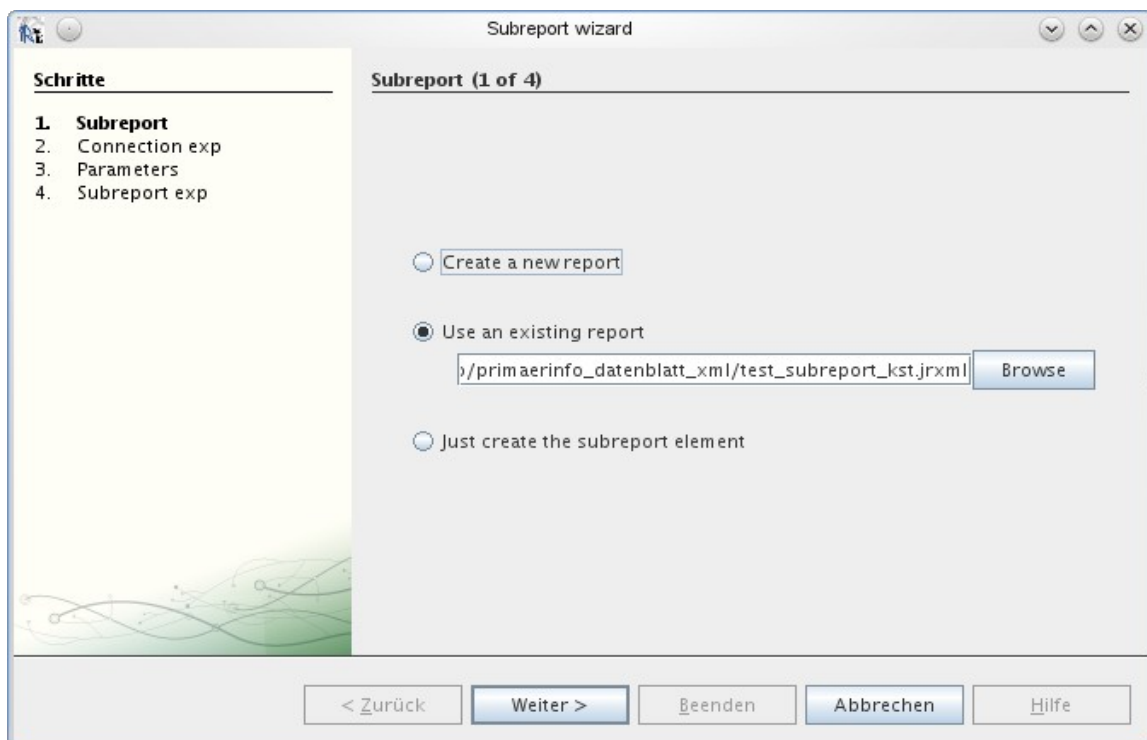
2.5.1 Konfiguration des Hauptberichts

Erzeugen Sie für den Unterbericht mit Hilfe des [Template-Generators](#) in Edustore einen leeren Bericht, und speichern Sie diesen am besten im gleichen Verzeichnis wie den Hauptbericht.

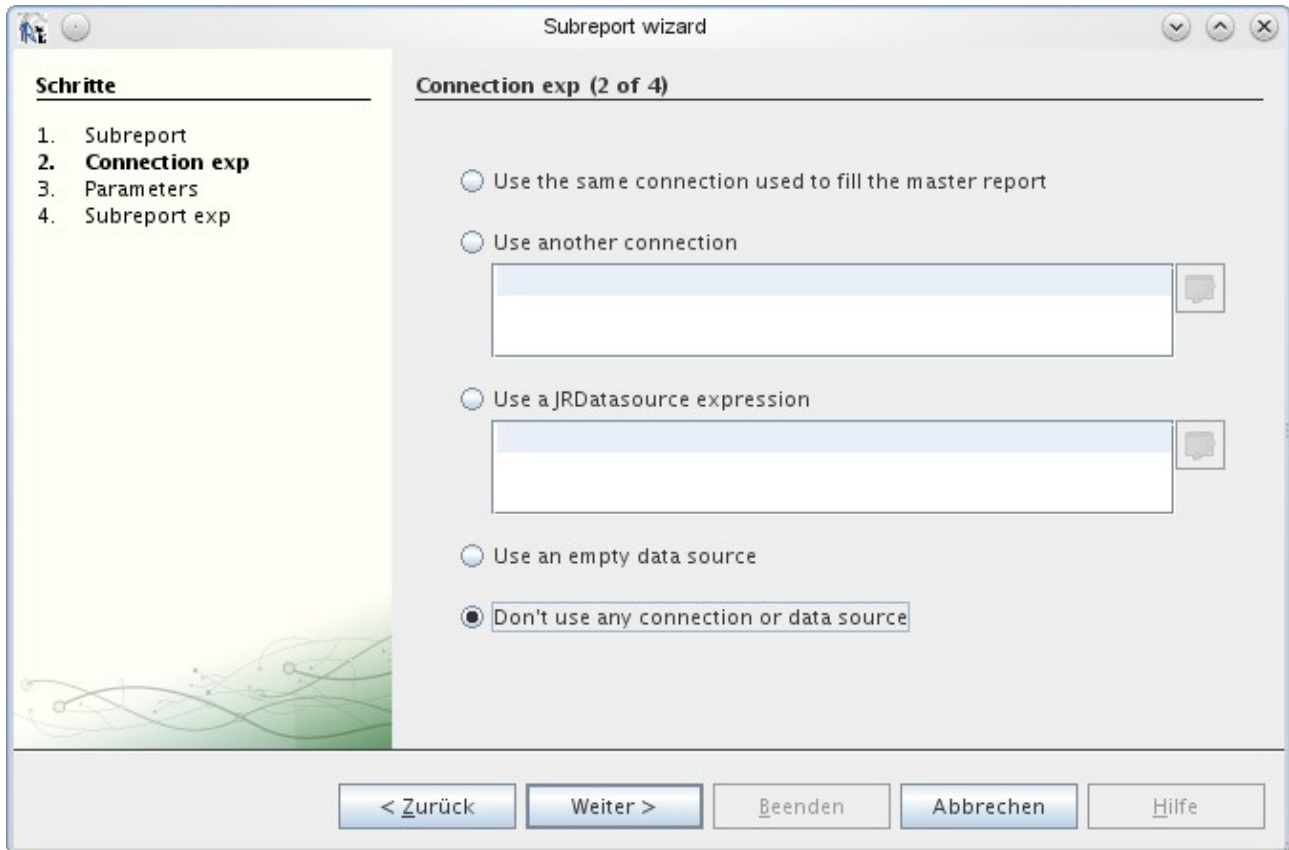
Ziehen Sie dann das Subreport-Werkzeug in den Entwurfsbereich:



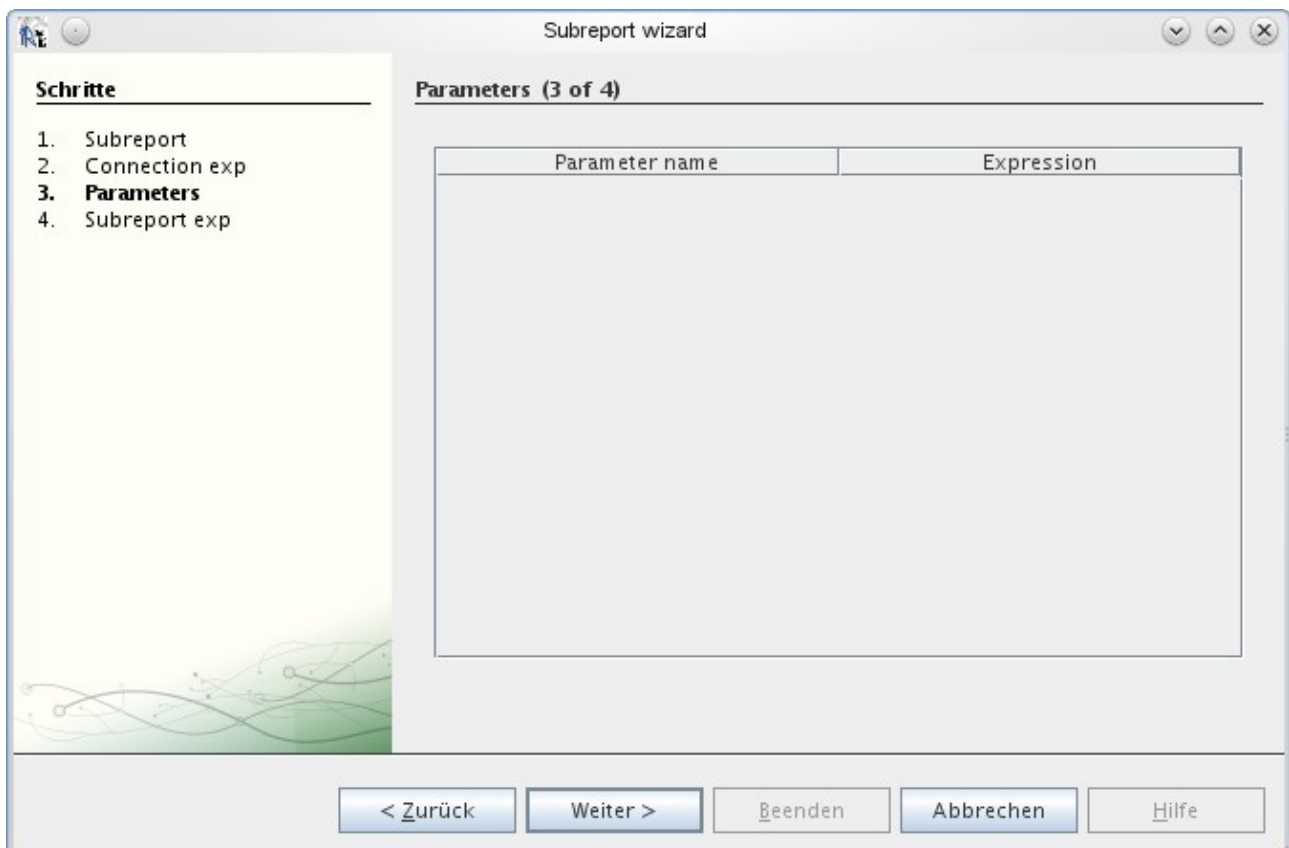
Es startet der Subreport-Assistent. Hier geben Sie zunächst den Dateinamen oben an:



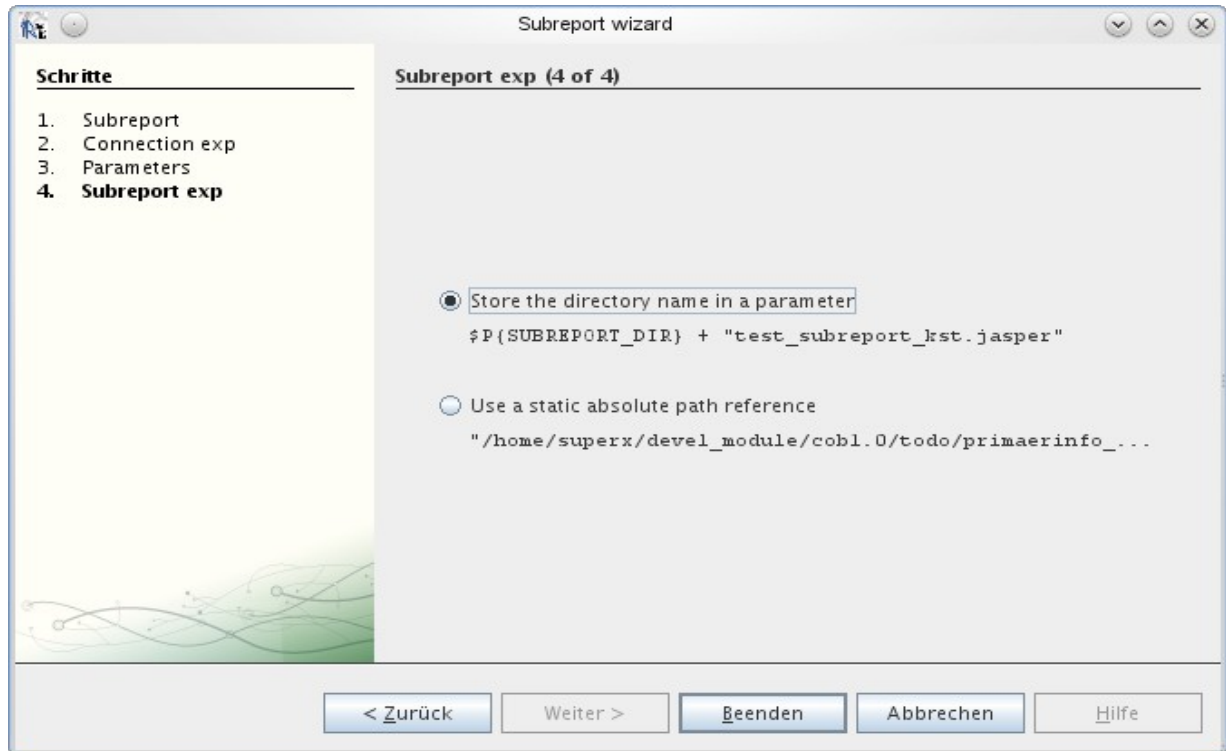
Im nächsten Schritt geben Sie eine Datenbank-Verbindung an. Hier wählen Sie "Don't use any connection...":



Im nächsten Schritt werden Parameter angezeigt, zu diesem Zeitpunkt gibt es aber noch keine.

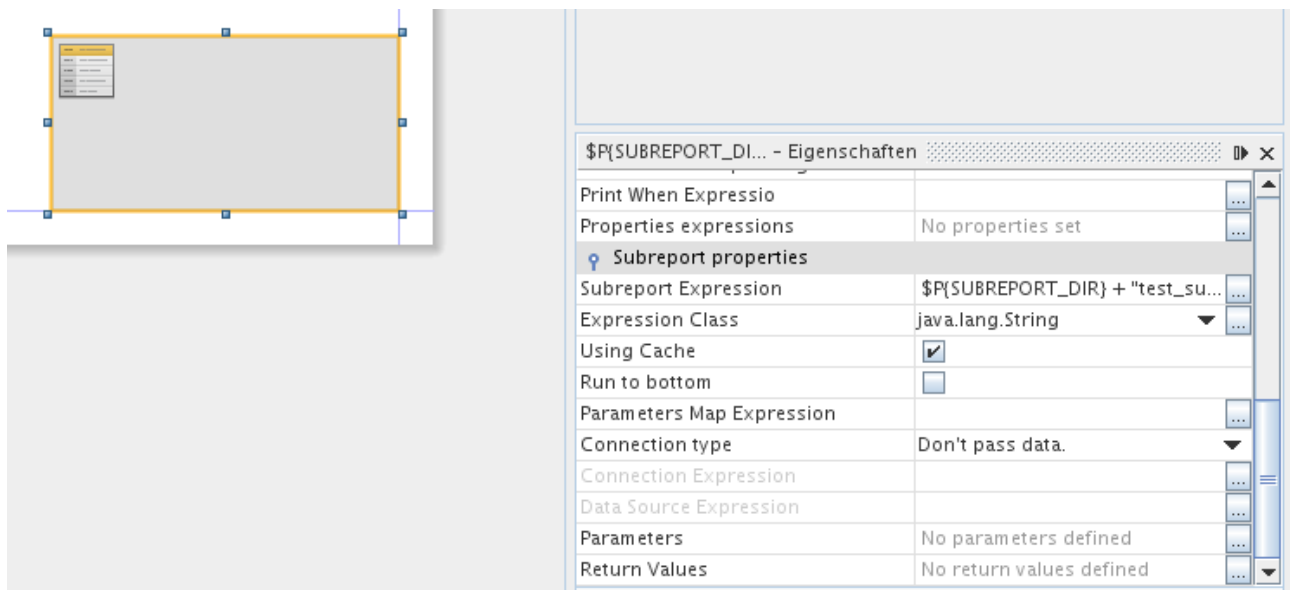


Klicken Sie einfach auf "Weiter". Sie werden dann aufgefordert, einen Dateipfad für den Unterbericht zu vergeben. IReport legt diesen standardmäßig in das gleiche Verzeichnis wie den Hauptbericht. Für Edustore sollten Sie dies auch tun.

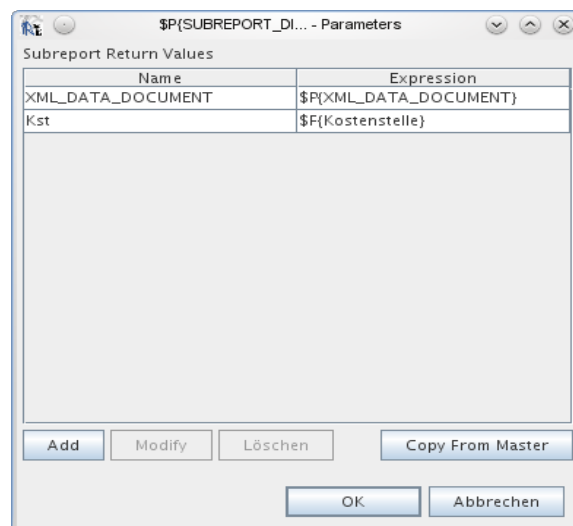


Danach sehen Sie einen Platzhalter für den Unterbericht im Entwurfsbereich. Beachten Sie bitte daß die Größe des Platzhalters unwichtig ist: die Ausdehnung des Unterberichts wird im Unterbericht definiert. Der Platzhalter ist lediglich wichtig, um die linke obere Ecke des Unterberichts festzulegen.

Rechts neben dem Platzhalter sehen Sie die Eigenschaften des Unterberichts:

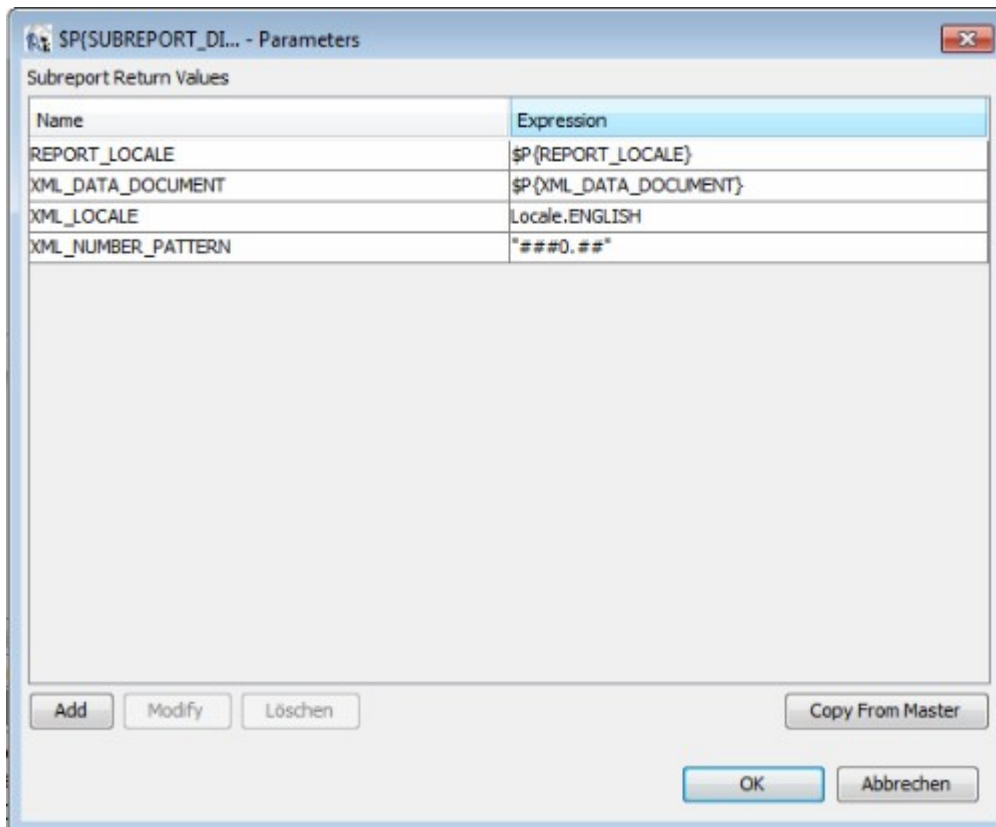


Ein Unterbericht arbeitet meist mit Übergabe von Parametern, z.B. will man vom Hauptbericht den Inhalt des Feldes "Kostenstelle" an den Unterbericht übergeben. Dies müssen wir in den Eigenschaften des Unterberichts "deklarieren". In der Zeile "Parameters" steht nach Bedienung des Assistenten noch kein Wert. Für die Datenquelle XML gibt es außerdem einen Parameter XML_DATA_DOCUMENT, der immer übergeben werden muss: Die XML-Datenquelle des Hauptberichts. Hier eine Übersicht, welche Parameter übergeben werden:



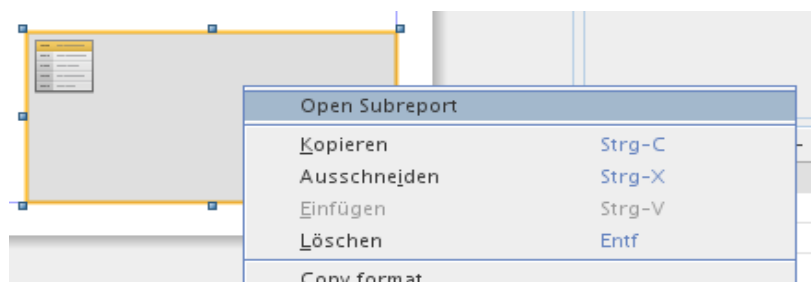
Die Zeilen mit Großschreibung sind die vorgegebenen Werte, der Parameter "Kst" speist sich aus dem JR-Feld "Kostenstelle". Das Beispiel zeigt, daß der Name für den Parameter beliebig ist.

Hinweis für Unterberichte mit Dezimalzahlen: Wenn der Unterbericht mit Zahlen mit Nachkommastellen arbeitet, und Sie in iReport testen, dann müssen Sie noch die [Parameter](#) XML_LOCALE und XML_NUMBER_PATTERN wie folgt übergeben:

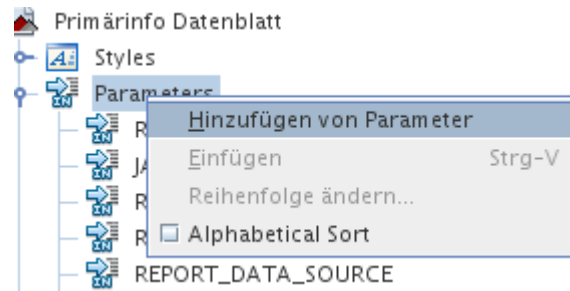


2.5.2 Konfiguration des Unterberichts

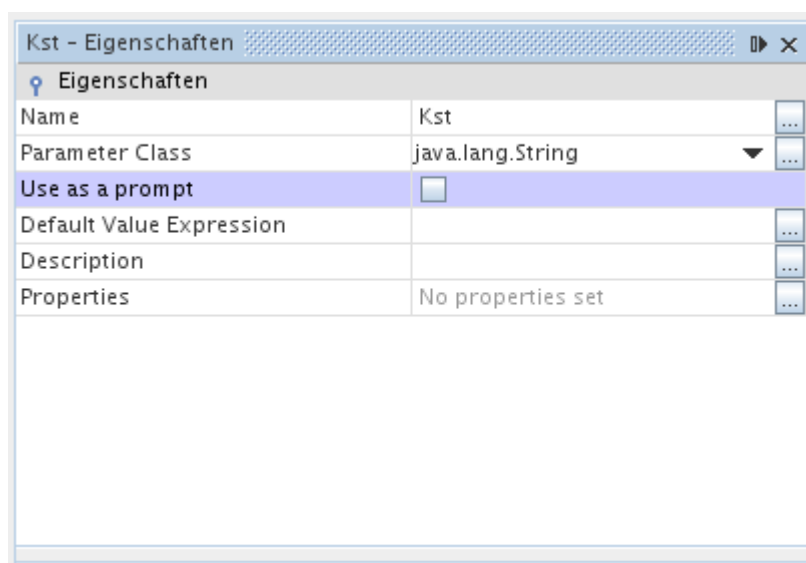
Nun können wir den Unterbericht öffnen, indem wir über dem Platzhalter die rechte Maustaste betätigen und "Open Subreport" wählen.




Es erscheint ein neues Fenster mit einem leeren Entwurfsbereich. Sie könnten nun loslegen und im Detailbereich verschiedene Elemente gestalten. Aber man muss bedenken daß ein Unterbericht für jeden Datensatz im Hauptbericht ausgeführt und eingebettet wird. Daher wird man vermutlich in den meisten Fällen im Unterbericht auf einen Filter zurückgreifen. Hier kommt dann wieder der oben genannte Parameter, z.B. "Kst" ins Spiel: Als ersten deklarieren wir den Parameter im Unterbericht, indem wir Links bei den Parametern die rechte Maustaste drücken, und "Hinzufügen..." wählen:

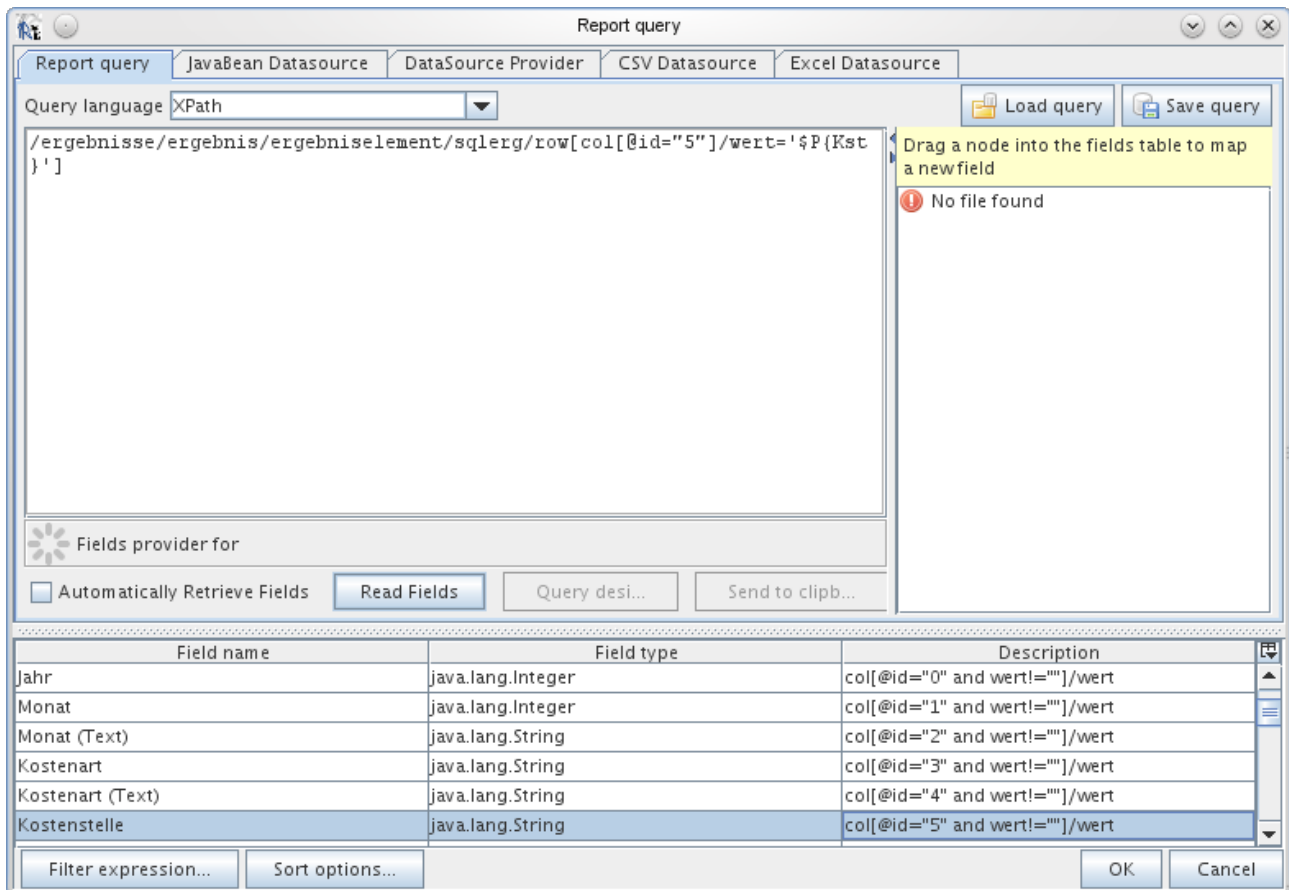


Standardmäßig nennt IReport den Parameter "parameter1". Rechts können wir in den Eigenschaften des Parameters den Namen ändern, und "Use as a prompt" abwählen:



Weitere Änderungen sind nicht notwendig.

Wir ergänzen nun die "Report Query"  des Unterberichts von `/ergebnisse/ergebnis/ergebniselement/sqlerg/row` um einen Filter, der mit der Abfragesprache XPATH formuliert wird. Im obigen Beispiel würden wir jetzt nur die Zeilen der Datenquelle auswählen, wo die Nummer der Kostenstelle mit der des Hauptberichts übereinstimmt. Wir erinnern uns: letztere steht im Parameter "Kst". Anhand der vorhandenen Report Query können wir sehen, dass die Kostenstelle in den Spalten ("col") mit der id=5 steht. Der Ausdruck wird also um den Filter "`[col[@id="5"]/wert=' $P{Kst}']`" ergänzt:

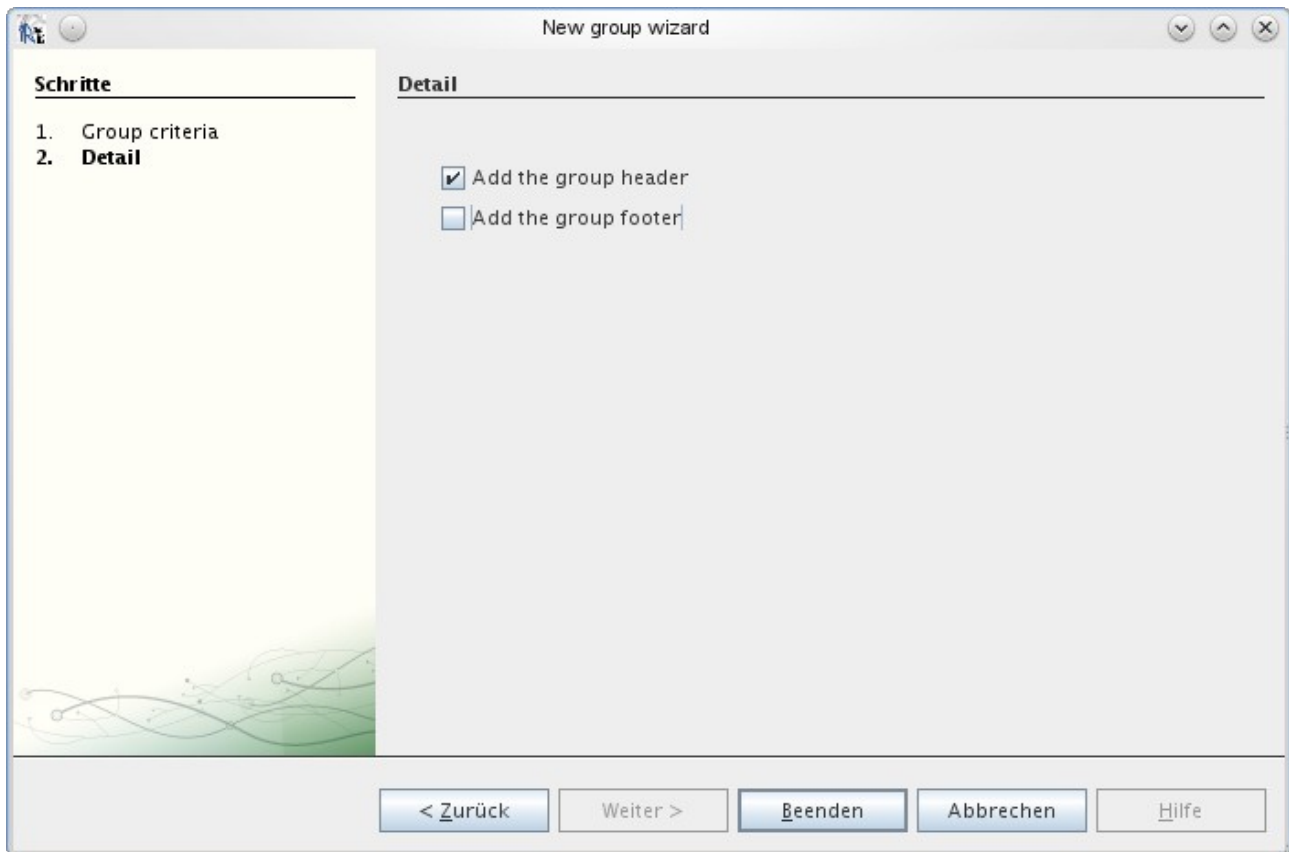


Nun ändern wir im Entwurfsbereich des Unterberichts das Layout: für das aktuelle Beispiel wollen wir im Unterbericht eine Summe der Beträge für jede Kostenstelle ausgeben. Dazu müssen wir zunächst eine Gruppierung nach Kostenstelle hinzufügen:

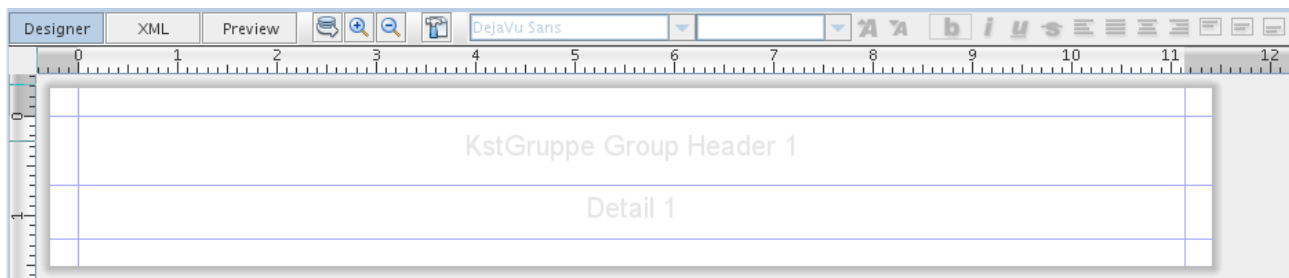
The image shows a 'New group wizard' dialog box with the following elements:

- Schritte:**
 1. Group criteria
 2. Detail
- Group criteria:**
 - Group name: KstGruppe
 - Group by the following report object: Kostenstelle Field String
 - Group by the following expression:
- Buttons:** < Zurück, Weiter >, Beenden, Abbrechen, Hilfe

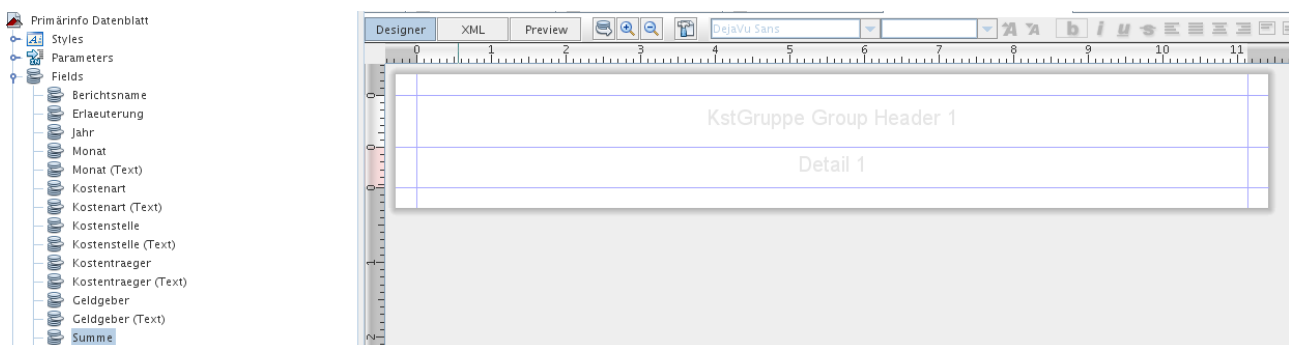
Im Gruppierungsassistenten wählen wir das Feld "Kostenstelle" als Gruppierungsmerkmal, und im nächsten Schritt sagen wir, daß ein Gruppenkopf hinzugefügt werden soll:



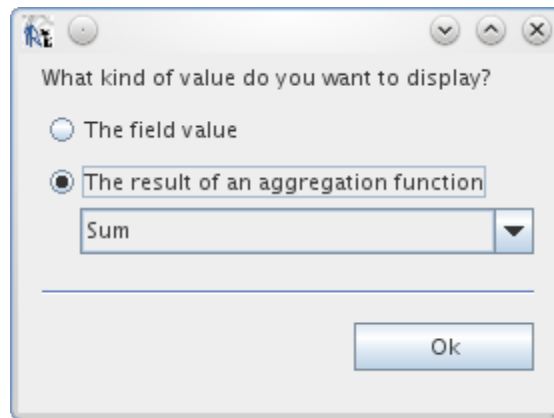
Danach ist die Gruppierung als Band verfügbar:



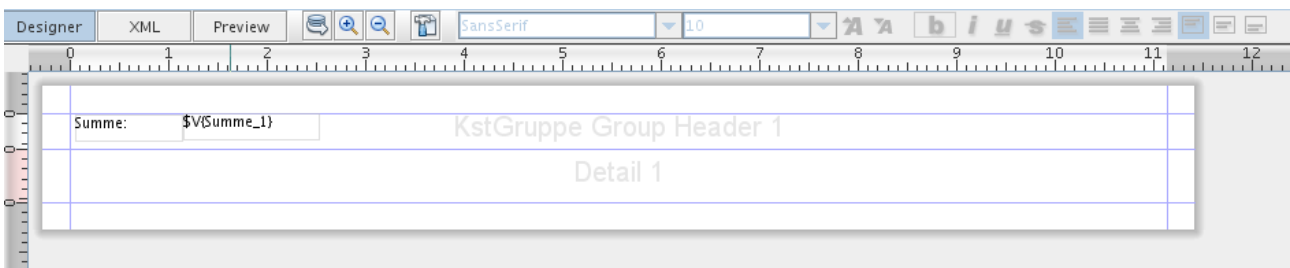
In die Gruppierung ziehen wir dann das Feld "Summe":



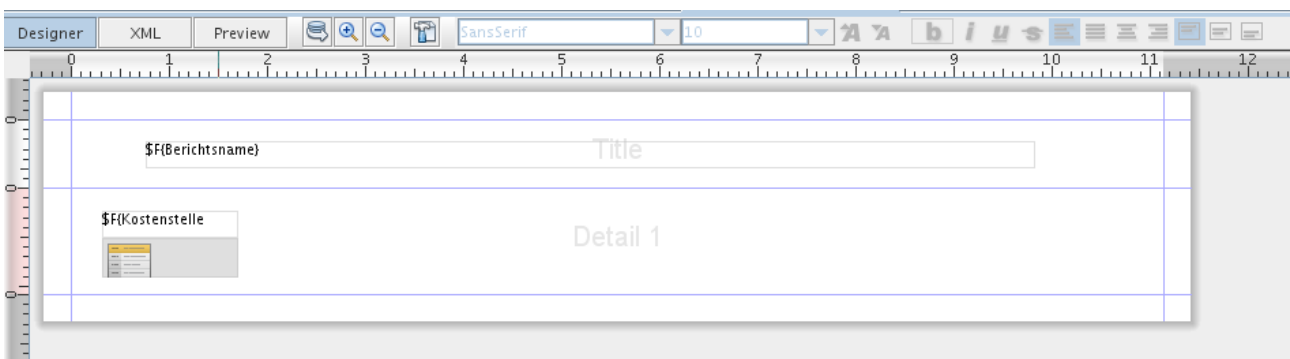
Beim "Loslassen" des Feldes werden wir gefragt, ob wir eine Aggregatfunktion nutzen wollen; wir wählen hier die Summierung:



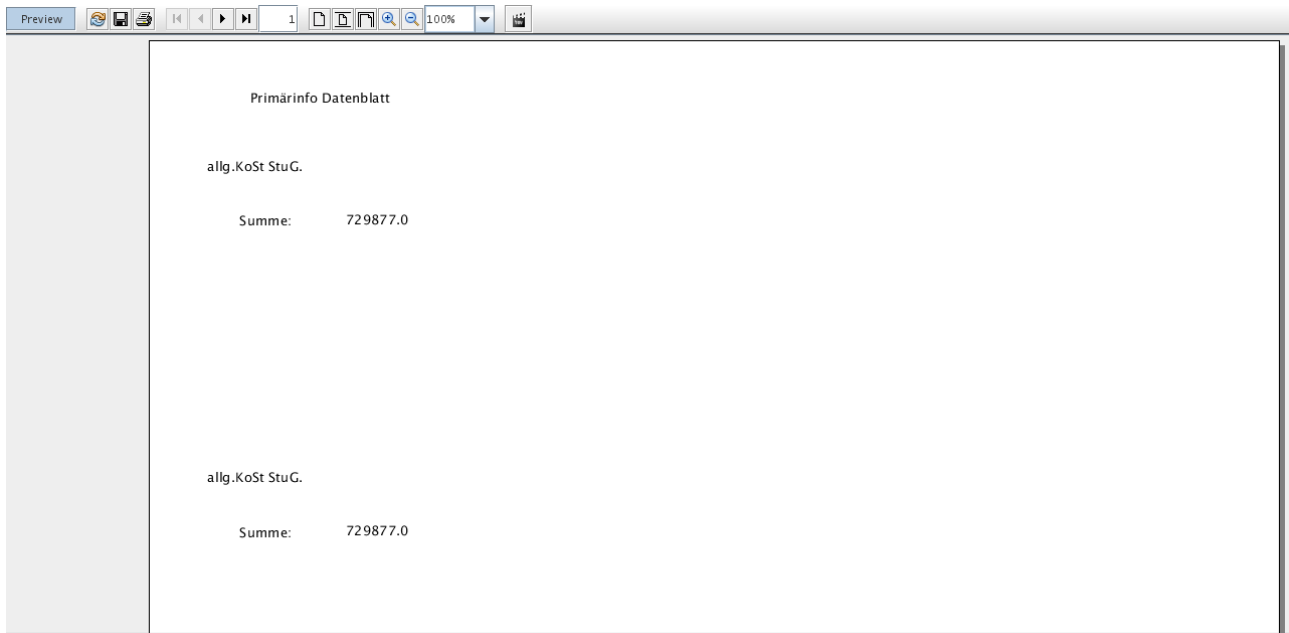
Danach ist das Feld verfügbar. Wir setzen noch das Label "Summe:" davor.



Den Detailbereich brauchen wir eigentlich nicht, wir können den ganz schmal machen. Danach speichern wir den Unterbericht, und gehen wieder zum Hauptbericht. Hier setzen wir den Unterbericht direkt unter das Feld "Kostenstelle":



Danach können wir den Bericht mit der jew. Datenquelle ausführen. Das Ergebnis läßt je nach Datenmenge ein bisschen auf sich warten:



2.5.3 Einhängen des Berichts in Edustore

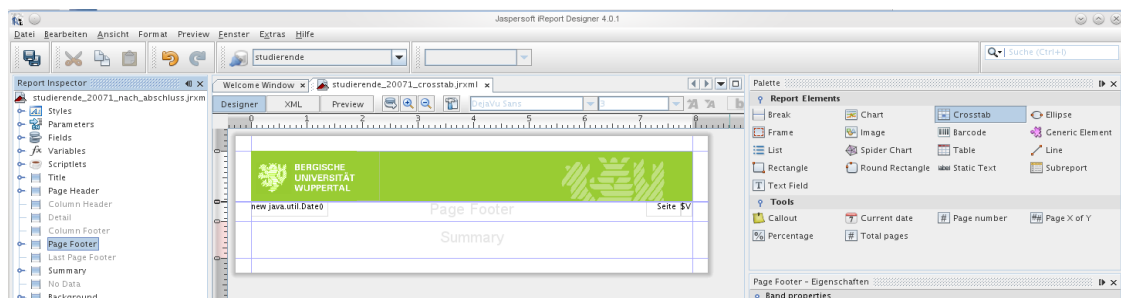
Damit der Bericht auch im Server nutzbar ist, muss noch folgendes eingestellt werden:

- Der Server sollte ausreichend RAM haben (Variable CATALINA_OPTS), weil hier XML als Datenquelle verarbeitet wird.
- Im Hauptbericht gibt es einen neuen Parameter SUBREPORT_DIR. Setzen Sie hier den Defaultwert ("Default value Expression" auf den Pfad, in dem die Dateien liegen (z.B. "/home/superx/webserver/tomcat/webapps/superx/WEB-INF/reports/").
- Kopieren Sie die .jasper Datei des Unterberichts in dieses Verzeichnis.

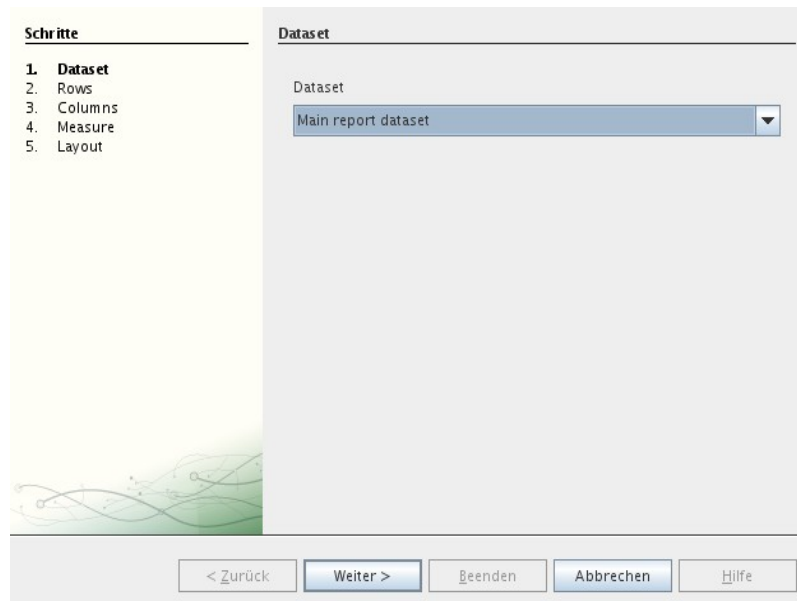
2.6 Kreuztabellen

2.6.1 Bedienung des Assistenten für Kreuztabellen

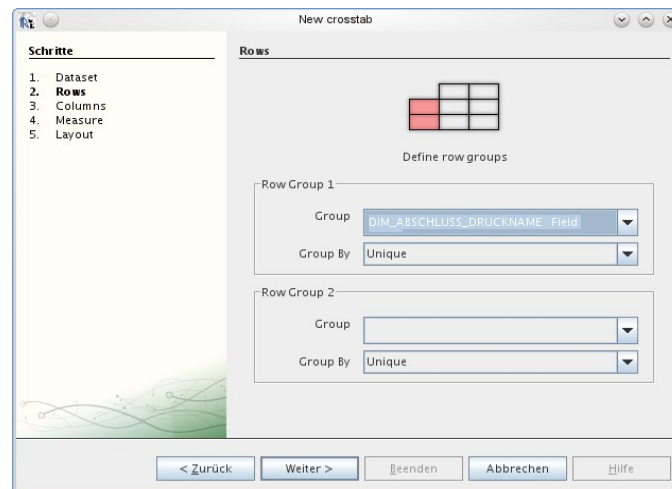
Für die Erstellung von Kreuztabellen gibt es einen eigenen Assistenten. Vorab sollte man aber zunächst ein "Summary"-Band erzeugen. Im folgenden Beispiel haben wir alle anderen Gruppierungen und Detail-Bands entfernt:



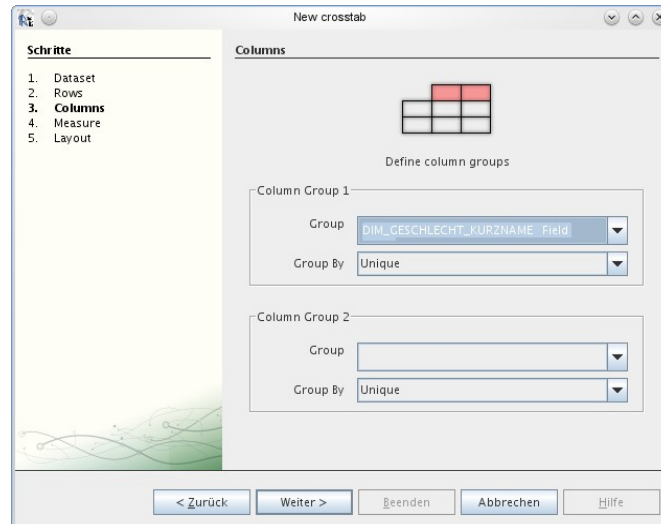
Dann ziehen wir das Crosstab-Element in den Summary Band. Es erscheint der Assistent. Als Datenquelle nutzen wir die Abfrage des Berichts.



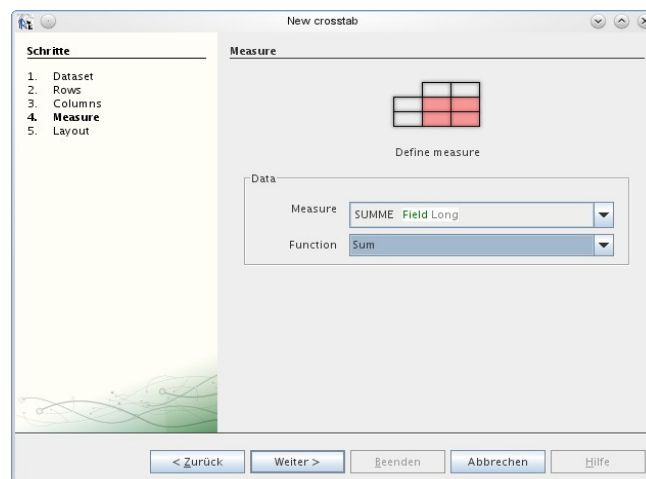
Als nächsten definieren wir die Zeilendimension des Berichts:



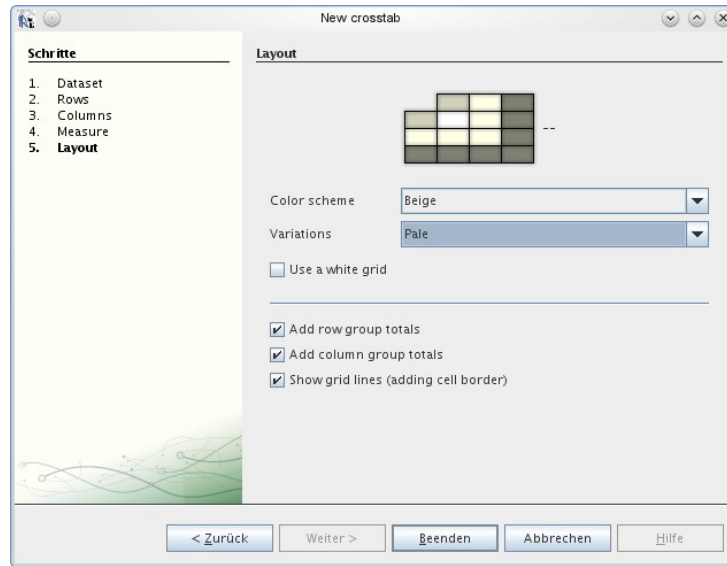
Wir wählen z.B. die Abschlüsse. Danach definieren wir die Spalten-Dimension:



Wir wählen das Merkmal "Geschlecht". Danach definieren wir das Merkmal, das in den Zellen ausgegeben werden soll. Dies ist die Summe der Studierenden:



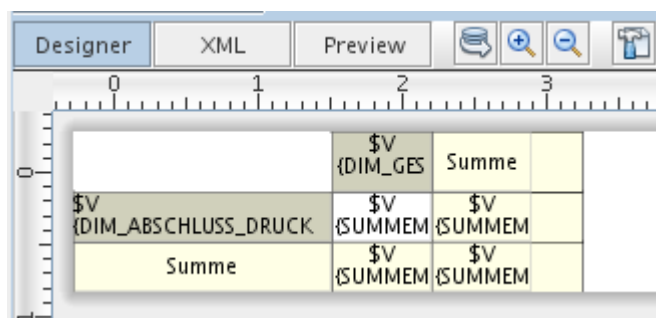
Zum Schluß kann man ein Layout festlegen, und Zeilen- und Spaltensummen erzeugen:



Im Entwurfsbereich erscheint ein neues Fenster mit der Kreuztabelle.



Hier können wir nun Spaltenbreiten und Überschriften ändern:

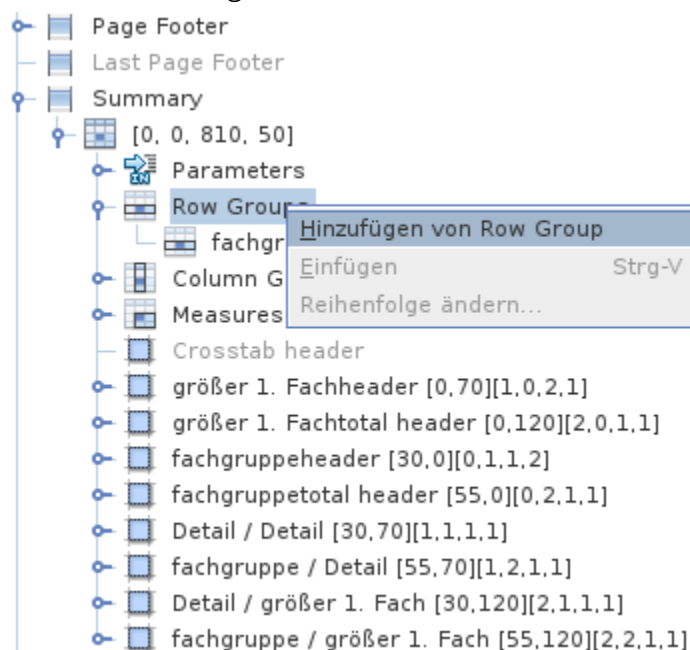


Das Ergebnis sieht wie folgt aus:

	männlich	weiblich	Summe
Bachelor IM	118	197	315
Bachelor VM	376	1102	1478
Diplom	44	241	285
Diplom International	148	134	282
EP LA Berufliche Sch.	1	1	2
EP LA Grund- u. Hauptsch.	0	12	12
EP LA Realschulen	6	9	15
EP LA Sonderschulen	0	3	3
Kontaktstudium	1	8	9
LA Berufliche Schulen	238	16	254
LA Grund- u. Hauptschulen	232	1373	1605
LA an Realschulen	568	905	1473
LA an Sonderschulen	128	465	593
Magister	3	29	32

2.6.2 Nachbearbeitung des Kreuztabellen-Assistenten

Der Assistent liefert nur einfache Kreuztabellen, ggf. wollen Sie diese im Nachhinein anpassen. Sie können z.B. mehr als zweistufige Hierarchien in Kreuztabellen einrichten, indem Sie im Report Inspector manuell neue Merkmale anlegen.



Auch bei "Measures" können Sie neue "abhängige Variablen" anlegen, z.B. wenn Sie in einer Tabelle mit den Zeilen "Ausgaben" und "Einnahmen" die Ausgaben mit negativem Vorzeichen versehen wollen und dann saldieren wollen: Sie würden ein neues "Measure" anlegen, mit der Calculation="Sum", Typ "java.lang.Float", und der folgenden Expression bekommen alle Ausgaben ein negat. Vorzeichen:

```
($F{Kostenart}=="Ausgaben")?-$F{Betrag}:$F{Betrag}
```

Das neue Measure können Sie dann in die Zeilen- oder Spaltensummen ziehen.

2.6.3 Kreuztabellen im Gruppenfuß

Kreuztabellen lassen sich nicht nur im Summary Band nutzen, sondern auch in Gruppenfuß/Gruppenkopf-Bändern. Wenn Sie die Kreuztabelle z.B. in einen Groupfooter legen, müssen Sie den "reset type" der Gruppe anpassen. Das machen Sie indem Sie ein rechts klick auf die Kreuztabelle machen und dann drauf "Crosstab data" klicken. Dort stellen Sie den "Reset type" auf "Group" und die "Reset group" auf die Gruppe in der die Kreuztabelle gelegt wurde. Wenn Sie das nicht machen funktioniert der Report nicht!

2.6.4 Kreuztabellen mit virtuellen Dimensionen

Sie können auch virtuelle Dimensionen in Kreuztabellen nutzen, indem Sie ähnlich wie bei der [Gruppierung ohne eigenes Band](#) mit Variablen arbeiten.

Um z.B. alle Studienfächer im Feld "1. Fach" auszugeben, und nur die Fächer Mathematik, Physik und Biologie zu einem Fach "Naturwissenschaften" zusammenzufassen, legen Sie eine neue Variable "fachgruppe" an, mit der Expression:

The screenshot displays the 'fachgruppe - Variable Expression' dialog box. The main text area contains the following expression:

```
($F{1. Fach}=="Biologie" ||
$F{1. Fach}=="Physik" ||
$F{1. Fach}=="Mathematik")? "Naturwissenschaft":$F{1. Fach}
```

The left sidebar shows the report structure with 'fachgruppe' selected under 'Variables'. The bottom panel shows a list of parameters and fields, including 'Berichtsname', 'Erläuterung', and various report heading fields.

In der Kreuztabelle können Sie nun z.B. in den Zeilen die Variable "fachgruppe" auswählen, und erhalten eine Tabelle mit der "künstlichen" Zeile "Naturwissenschaften":

Memtext University

Fächerkombinationen Datenblatt
Stand:17.04.2013

	Biologie	Chemie	Erziehungswissenschaften	Ev. Theologie	Französisch	Mathematik	Sport, Sportwiss	null	Total größer 1.
Agrarwissenschaft	0	0	0	0	0	0	0	2	2
Anglistik	1	0	1	1	1	0	0	0	4
Geowissenschaften	0	0	0	0	0	0	0	3	3
Intelligente Eingebettete	0	0	0	0	0	0	0	1	1
Maschinenbau	0	0	0	0	0	0	0	3	3
Medizin	0	0	0	0	0	0	0	2	2
Naturwissenschaft	6	1	49	0	0	42	1	3	102
Soziologie	0	0	0	0	0	0	0	2	2
Total fachgruppe	7	1	50	1	1	42	1	16	119

2.7 Diagramme

Um einen Bericht ansprechend zu gestalten dürfen Diagramme wie Torten- oder Balkendiagramme nicht fehlen. Diese in einen Bericht einzubringen ist auch nicht schwer.

Vorab sei aber darauf hingewiesen: Grafiken in JasperReports arbeiten nicht analog zu Kreuztabellen, in denen Daten nach Zeilen- oder Spaltenmerkmalen automatisch gruppiert und summiert werden. Für Grafiken sollten Sie gewährleisten, dass die Datenbasis für die Grafik **bereits gruppiert und ggf. summiert** sind, Sie können also keine Einzeldaten für Grafiken verwenden. Bei Datenblatt-Berichten in Edustore ist dies gewährleistet, wenn Sie im Feld "Felder" nur die Felder selektieren, nach denen in der Grafik gruppiert werden soll.

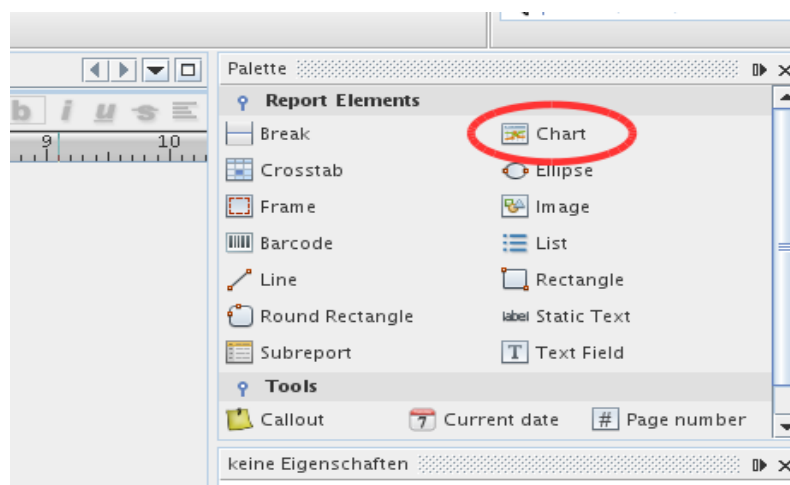
2.7.1 Einfache Diagramme

Nehmen wir an wir haben eine Tabelle, die die Studierendenzahl mehrerer Semester nach Geschlecht aufführt. Hier die Ansicht in iReports "Refresh Preview Data":

Field name	Field type	Description
SUMME	java.lang.Long	
DIM_GESCHLECHT_KURZNAME	java.lang.String	
DIM_SEMESTER_DRUCKNAME	java.lang.String	
SORTNR	java.lang.String	

SORTNR	DIM_SEMESTER_DRUCKNAME	SUMME	DIM_GESCHLECHT_KURZNAME
20051	SS 2005	1562	männlich
20051	SS 2005	3448	weiblich
20052	WS 2005/2006	2164	männlich
20052	WS 2005/2006	4825	weiblich
20061	SS 2006	2174	männlich
20061	SS 2006	4815	weiblich
20062	WS 2006/2007	2490	männlich
20062	WS 2006/2007	5540	weiblich
20071	SS 2007	1959	männlich
20071	SS 2007	4653	weiblich

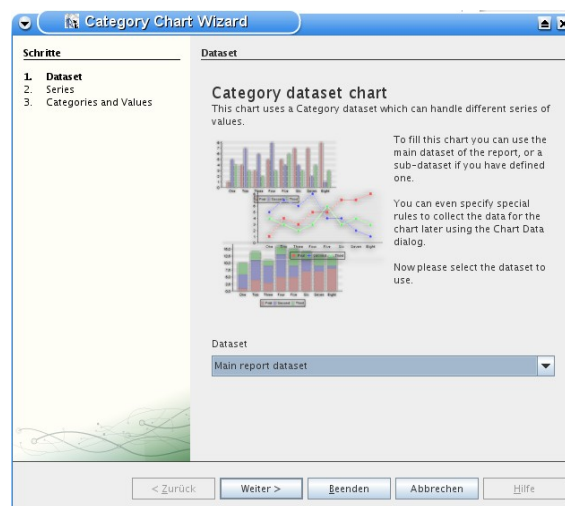
Die Spalten der Tabelle sind selbsterklärend. Nun ziehen wir aus dem "Report Elements" die "Chart" in das Band "Summary".



Dabei öffnet sich ein Fenster in dem Sie sich die Art der Grafik auswählen können. In diesem Beispiel wählen wir "Stacked Bar 3D".

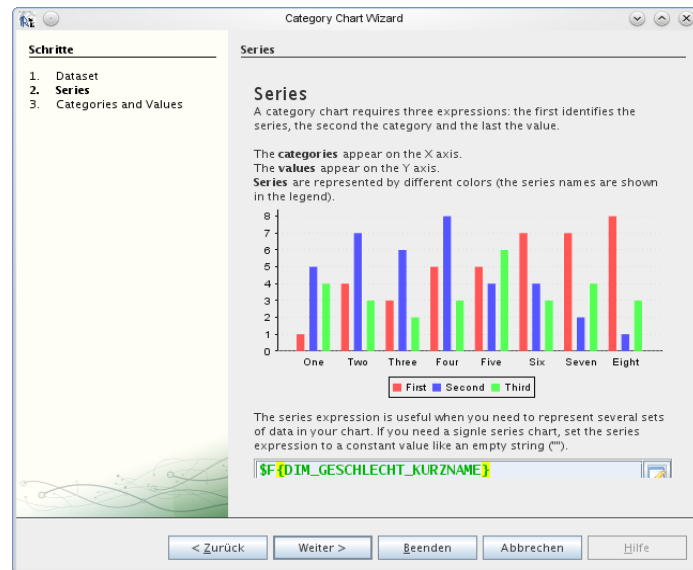


Nach dem Klick auf "OK", geht es weiter mit den Inhalten der Grafik.
Als Dataset gibt es hier nur eine Auswahl.

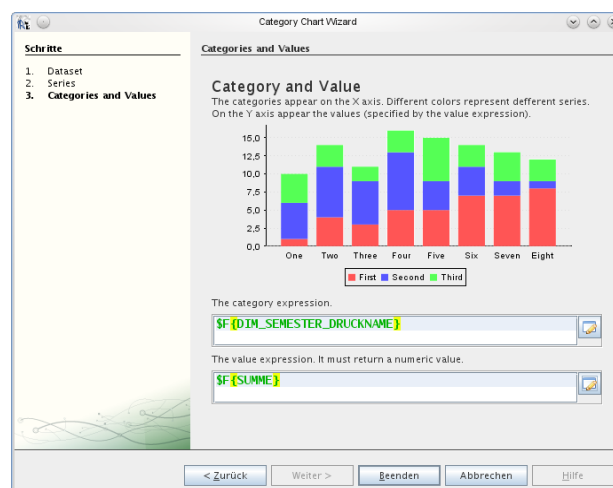


Daher akzeptieren wir dies mit einem Klick auf "Weiter".

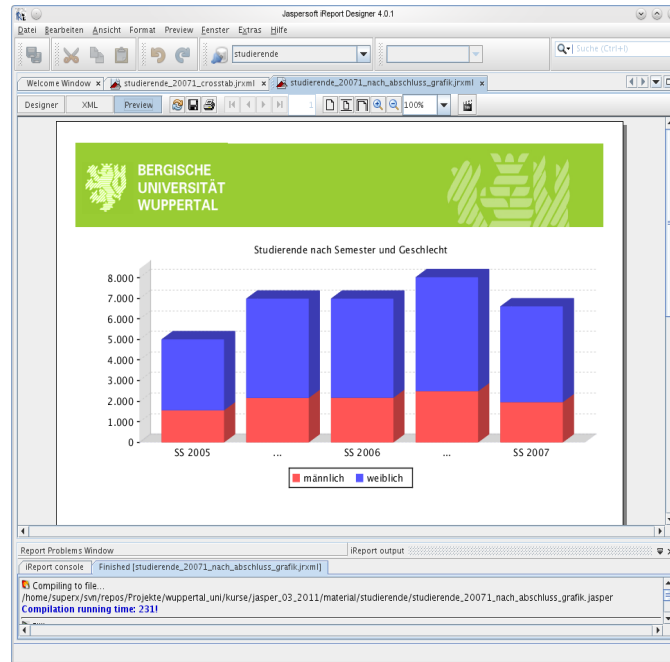
Die "Series" sind die verschiedenen Unterteilungen eines Balkens. Die "Series" sind die verschiedenen Unterteilungen eines Balkens. Wir wählen hier das Merkmal "Geschlecht":



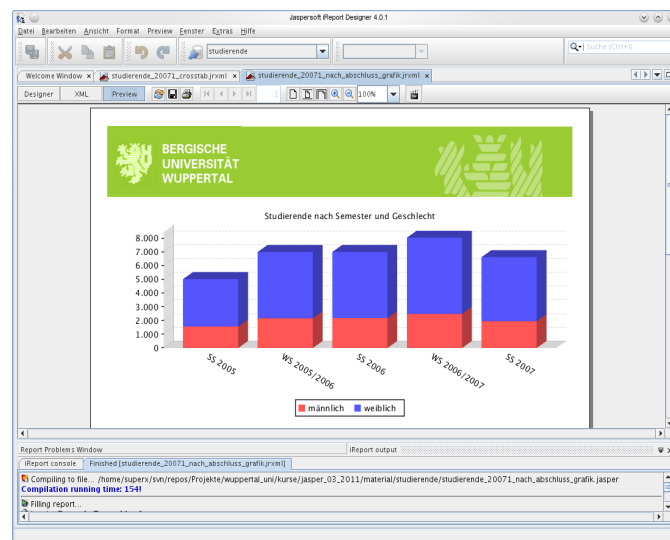
Bei der X-Achse (in iReport heißt das "Category Axis") nehmen wir das Merkmal Semester, und bei der "Value expression" die Summe.



Damit ist die Definition beendet. Wir können dann in den Eigenschaften der Grafik noch eine "Title Expression" festlegen, z.B. "Studierende nach Semester und Geschlecht". In der Vorschau sieht die Grafik so aus:



Die Wintersemester werden noch nicht abgedruckt, dazu ändern wie die "Label rotation" auf 30°. Dies lässt die Schrift unter den Balken rotieren, damit längere Schriftzüge möglich sind.



Wichtig für die Grafik war die Sortierung in der Datenquelle nach "dim_semester.quell_id". Mit "Series Colors" kann man die Farbe der Balken ändern etc.

Wenn sie jeden Balken bzw. Teil des Balkens zusätzlich mit den Werten ausgeben wollen, müssen sie in den Chart-Eigenschaften unter den "Bar*Properties" das Ankreuzfeld "Show labels" ankreuzen. Wenn es bereits angekreuzt ist, sollten Sie das Kreuz einmal entfernen und wieder setzen (Bug in iReport seit Version 4.x)

2.7.2 Diagramme nach Tabelle ausgeben

Üblicherweise ähnelt ein Diagramm in iReport einer Kreuztabelle. Wenn nun ein Diagramm nach Vorgabe einer Tabelle ausgegeben werden soll, muss ein wenig getrickst werden.

Erläuterung:

In der Series expression werden die Zeilen angegeben.

In der Category expression werden die Spalten angegeben.

In der Value expression wird der Wert angegeben.

Um ein Diagramm nun zu erstellen muss pro Series + Category eine Kategorie series angelegt werden und das entsprechende Feld angegeben werden.

2.7.3 Diagramme mit berechneten Werten

Wenn die einem Diagramm zugrunde liegende Tabelle noch Berechnungen erfordert, muss man mit Variablen arbeiten. Als Beispiel wird hier die Maske "Studierende und Studienanfänger (Zeitreihe)" benutzt. Das Ergebnis wird eine Tabelle mit Balkendiagramm der "Studierende und Studienanfänger im 1. Fachsemester (Zeitreihe)" sein.

Ein bereits fertiges Datenblatt könnte folgendermaßen aussehen:

Studierende und Studienanfänger im 1. Fachsemester (Zeitreihe)			
Semester	1. FS Frauen	1. FS Männer	Summe
#{Semester}	#{1. FS Frauen}	#{1. FS Männer}	#{1. FS Summe}
Summary			

Hier fehlt jetzt die Grafik, die in dem noch leeren Band "Summary" eingefügt werden soll. Wenn das Band noch nicht eingefügt wurde, einfach mit Rechtsklick auf Summary (im "Report Inspector") und dann auf "Add Band" um es anzuzeigen.

The screenshot shows the iReport 3.7.1 application window. The title bar reads "iReport 3.7.1". The menu bar includes "Datei", "Bearbeiten", "Ansicht", "Format", "Preview", "Fenster", "Extras", and "Hilfe". The toolbar contains icons for print, undo, redo, and a search icon. A dropdown menu shows "Studierende".

The "Report Inspector" panel on the left lists report elements for "Studierende und Studienanfänger (Zeitreihe)":

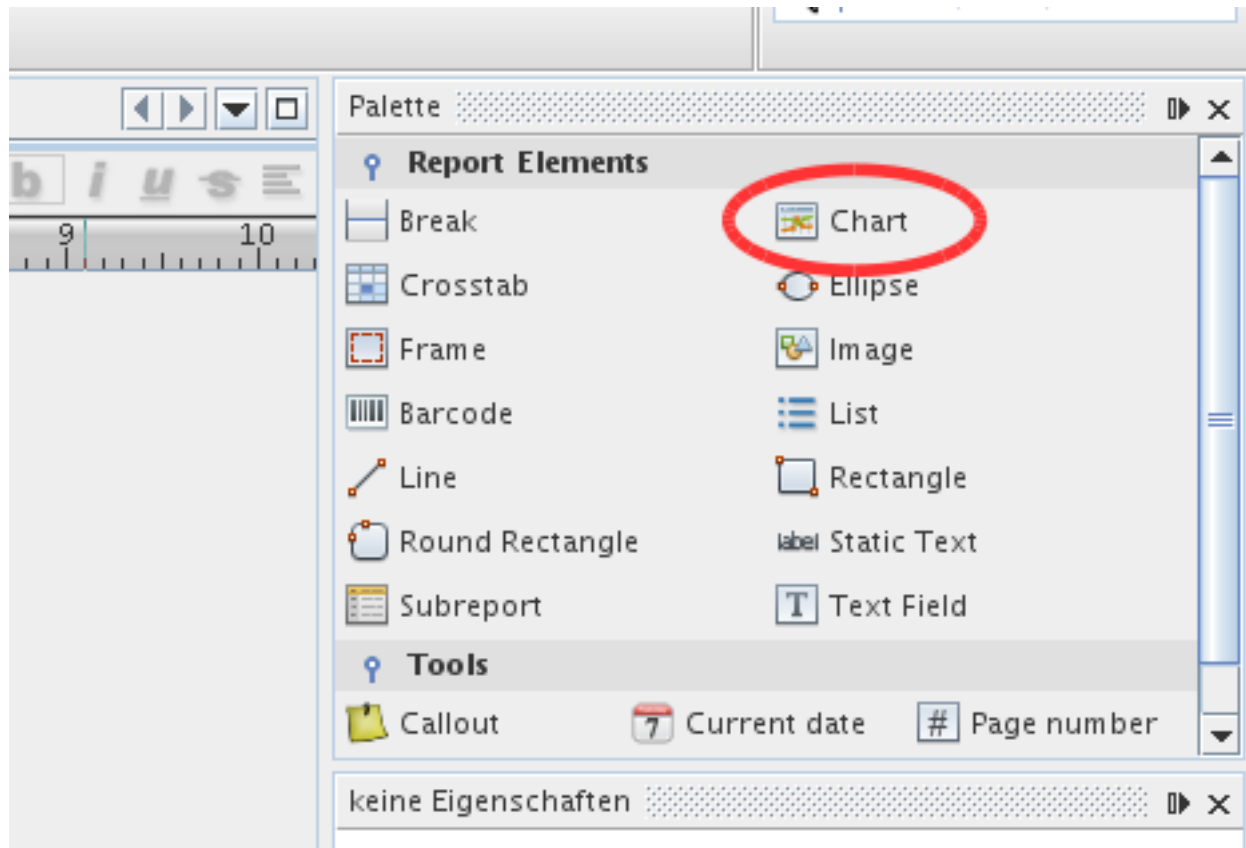
- Styles
- Parameters
- Fields
- Variables
- Scriptlets
- Title
- Page Header
- Column Header
- Detail 1
- Column Footer
- Page Footer
- Last Page Footer
- Summary** (circled in red)
- No Data
- Background

The "Designer" view on the right shows a table with the following structure:

Studierende und Studienanfänger (Zeitreihe)	
Semester	1. F
<code>#{Semester}</code>	<code>#{1. F</code>

The "Welcome Window" tab is active, and the "tabelle_16" report is open. The "Designer" view is selected, and a ruler is visible at the top of the design area.

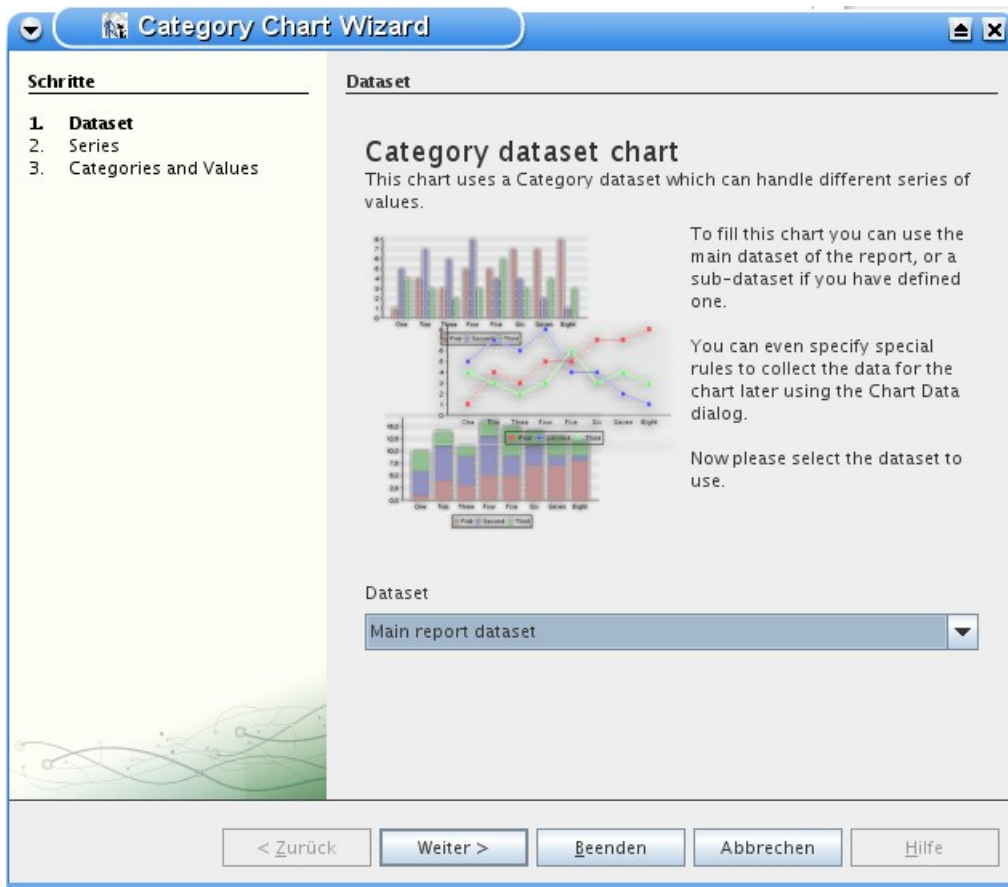
Nun ziehen wir aus dem "Report Elements" die "Chart" in das Band "Summary".



Dabei öffnet sich ein Fenster in dem Sie sich die Art der Grafik auswählen können. In diesem Beispiel wählen wir "Stacked Bar 3D".



Nach dem Klick auf "OK", geht es weiter mit den Inhalten der Grafik. Als Dataset gibt es hier nur eine Auswahl.



Daher akzeptieren wir dies mit einem Klick auf "Weiter".

Die "Series" sind die verschiedenen Unterteilungen eines Balkens. Da es in diesem Beispiel dafür keine Variable gibt, wird später noch per Hand eine Weitere hinzugefügt. Wenn es bei Ihnen möglich ist, klicken Sie rechts neben dem Feld auf den Button und fügen dort eine Variable ein. In diesem Beispiel wird als fester String "Frauen im 1. FS" eingegeben.

Category Chart Wizard

Schritte

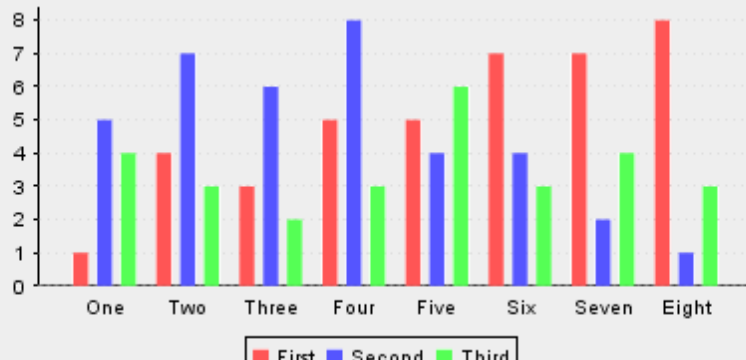
1. Dataset
2. **Series**
3. Categories and Values

Series

Series


A category chart requires three expressions: the first identifies the series, the second the category and the last the value.

The **categories** appear on the X axis.
The **values** appear on the Y axis.
Series are represented by different colors (the series names are shown in the legend).



Category	First	Second	Third
One	1	5	4
Two	4	7	3
Three	3	6	2
Four	5	8	3
Five	5	4	6
Six	7	4	3
Seven	7	2	4
Eight	8	1	3

The series expression is useful when you need to represent several sets of data in your chart. If you need a single series chart, set the series expression to a constant value like an empty string ("").

"Frauen im 1. FS" 

< Zurück Weiter > Beenden Abbrechen Hilfe

Und dann wieder auf "Weiter".

Auf der X-Achse soll in diesem Beispiel die Semester erscheinen und auf der Y-Achse die Anzahl der Weiblichen Studierenden im 1. Fachsemester. Um dies zu erhalten wählt man als "category expression" die Variable "Semester" aus.

Category Chart Wizard

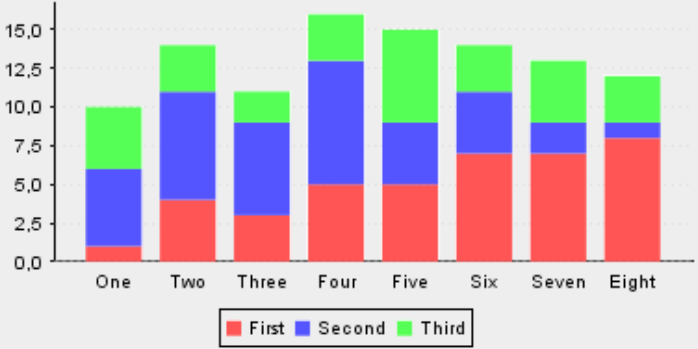
Schritte

1. Dataset
2. Series
3. **Categories and Values**

Categories and Values


Category and Value

The categories appear on the X axis. Different colors represent different series. On the Y axis appear the values (specified by the value expression).




Category	First	Second	Third
One	1.0	4.0	3.0
Two	4.0	6.0	3.0
Three	3.0	5.0	2.0
Four	5.0	8.0	2.0
Five	5.0	4.0	6.0
Six	7.0	4.0	2.0
Seven	7.0	2.0	3.0
Eight	8.0	1.0	2.0

The category expression.



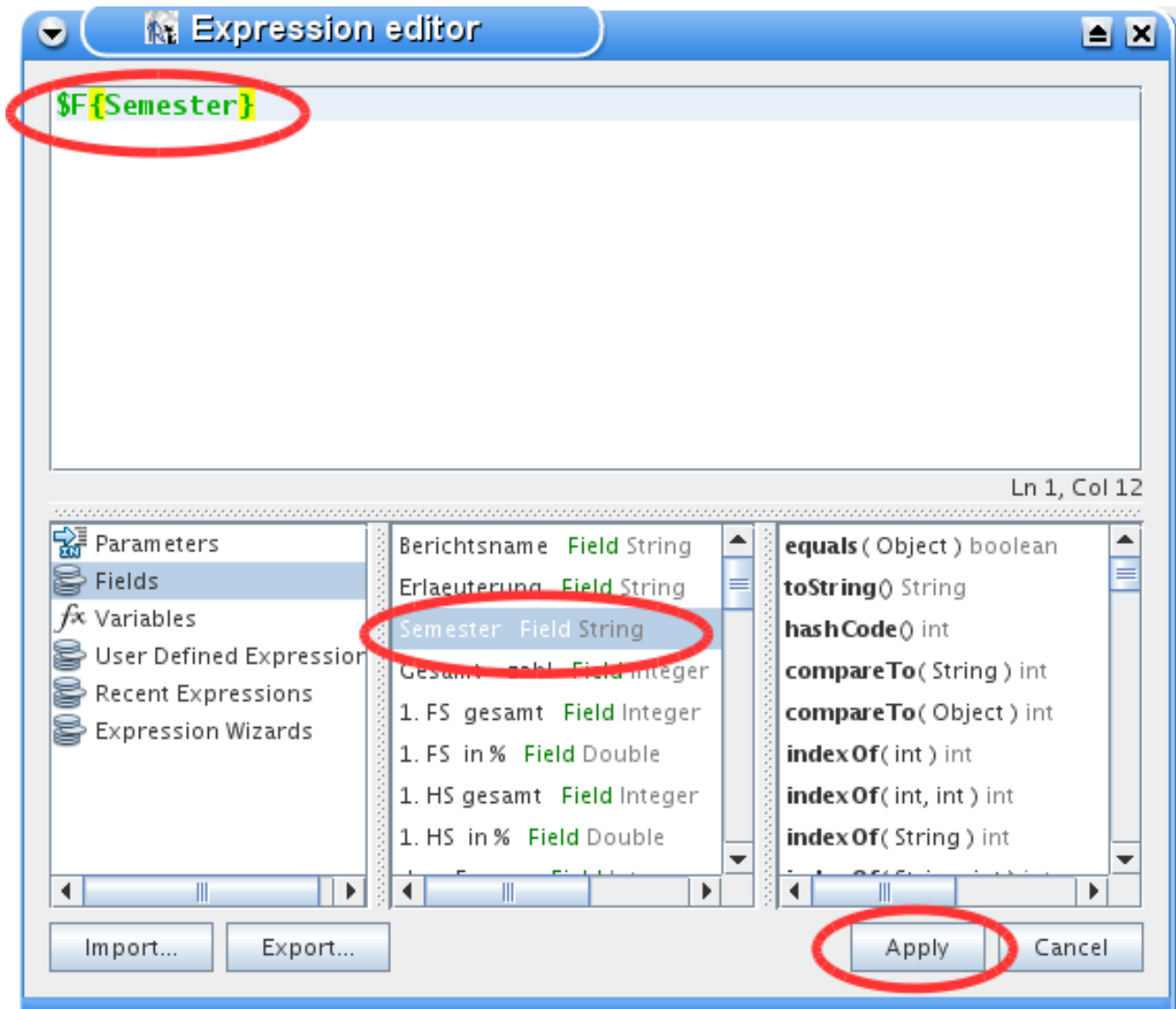
The value expression. It must return a numeric value.



✖ Please specify a valid Category expression

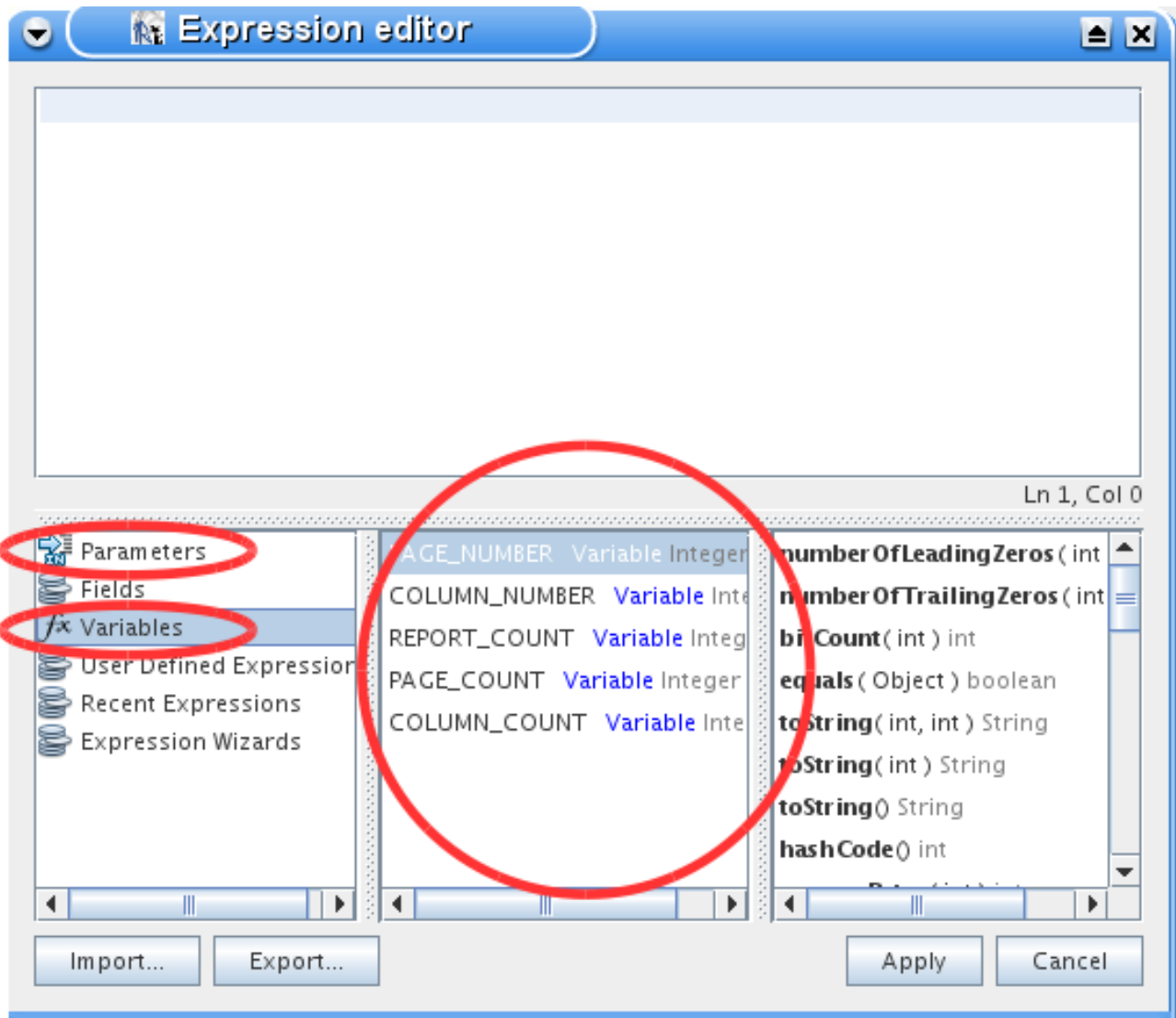
< Zurück Weiter > Beenden Abbrechen Hilfe

Dies geschieht, indem man auf den Button rechts neben dem Feld klickt,



dann doppelt auf das Feld "Semester" klickt und dann auf "Apply".

In den meisten Fällen werden hier Felder benutzt. Wenn Sie jedoch z.B. eine Variable benutzen wollen, müssen Sie nach dem Klick rechts auf den Button in dem neuen Fenster links unten z.B. auf "Variables" oder "Parameter" klicken. Dabei ändert sich der Inhalt von dem mittleren Feld.



Hier können Sie nun auch eine andere Auswahl treffen.

Das selbe gilt auch für "value expression". Der Unterschied ist hierbei, die "category expression" ist die Bezeichnung und in "value expression" dürfen nur Zahlen vorkommen, welche von Jasper-Reports addiert werden können.

Category Chart Wizard

Schritte

1. Dataset
2. Series
3. **Categories and Values**

Categories and Values

Category and Value

The categories appear on the X axis. Different colors represent different series. On the Y axis appear the values (specified by the value expression).

Category	First	Second	Third
One	1.0	4.0	4.0
Two	4.0	6.0	4.0
Three	3.0	5.0	3.0
Four	5.0	8.0	3.0
Five	5.0	4.0	6.0
Six	7.0	4.0	3.0
Seven	7.0	2.0	3.0
Eight	8.0	1.0	3.0

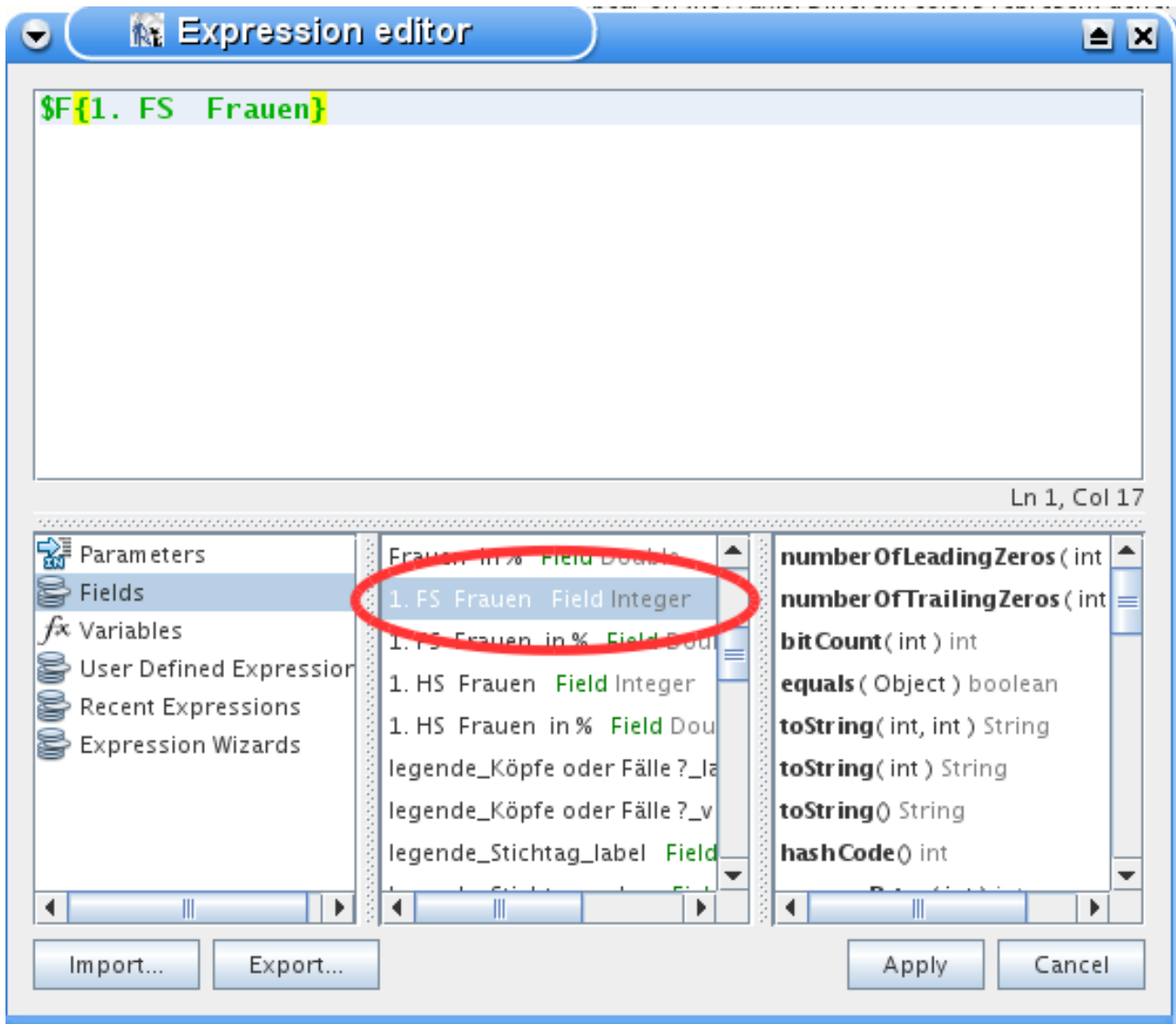
The category expression:

The value expression. It must return a numeric value.

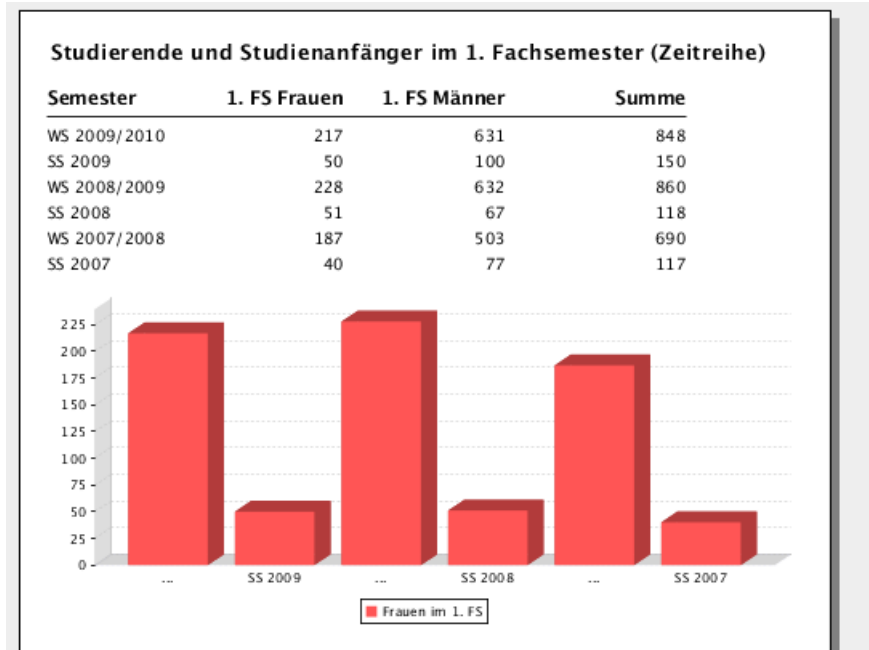
Please specify a valid Value expression

< Zurück Weiter > Beenden Abbrechen Hilfe

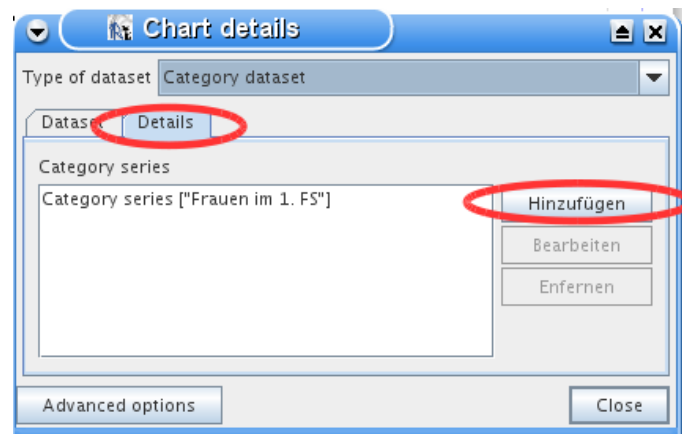
Hier wählen wir nun z.B. das Feld "1. FS Frauen" aus.



Nun noch abschließend auf Beenden drücken und wir erhalten das erste Ergebnis. Über "Preview" wird nun das Diagramm mit den Frauen aus dem 1. Fachsemester über die Semester angezeigt.



Um nun die männlichen Studenten aus dem 1. FS noch hinzuzufügen oder Änderungen an der gerade getroffenen Auswahl vorzunehmen klickt man mit der rechten Maustaste auf die Chart und dann auf "Chart Data". In dem Reiter "Details" kann man nun die eben angelegte Category bearbeiten/entfernen oder auch eine neue anlegen.



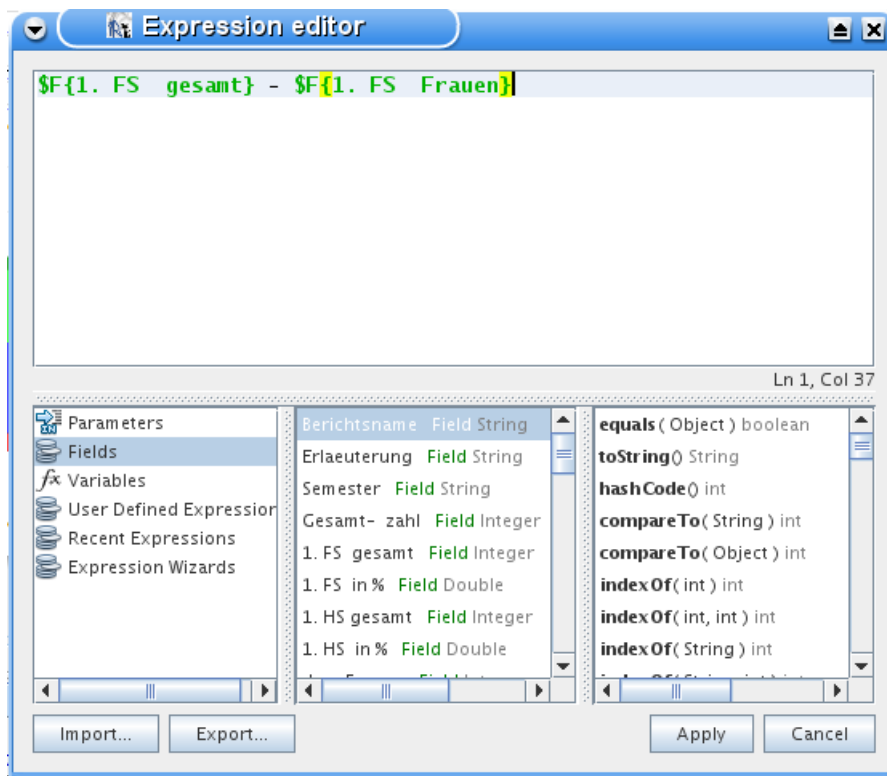
Hier wollen wir jetzt noch die männlichen Studierende aus dem 1. FS noch Hinzufügen. Also klickt man auf "Hinzufügen".

The image shows a dialog box with a blue title bar and standard window controls. It has two tabs: "Data" (selected) and "Item Hyperlink". The dialog is divided into four main sections, each with a text input field and a small edit icon (pencil) on the right:

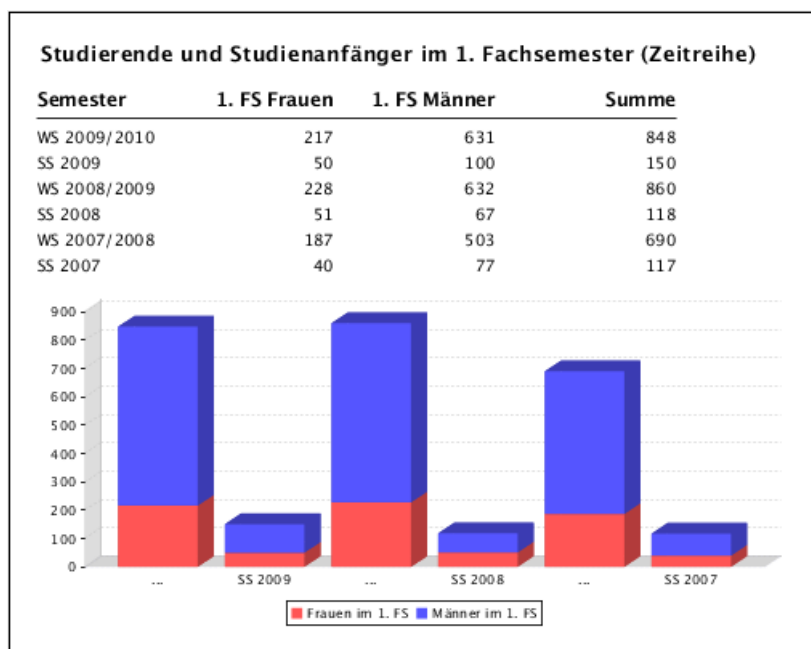
- Series expression (required):** The text field contains the string `"Männer im 1. FS"`.
- Category expression (required):** The text field contains the expression `#{Semester}`.
- Value expression (required):** The text field contains the expression `#{1. FS gesamt} - #{1. FS Frauen}`.
- Label Expression:** The text field is currently empty.

At the bottom right of the dialog, there are two buttons: "OK" and "Abbruch".

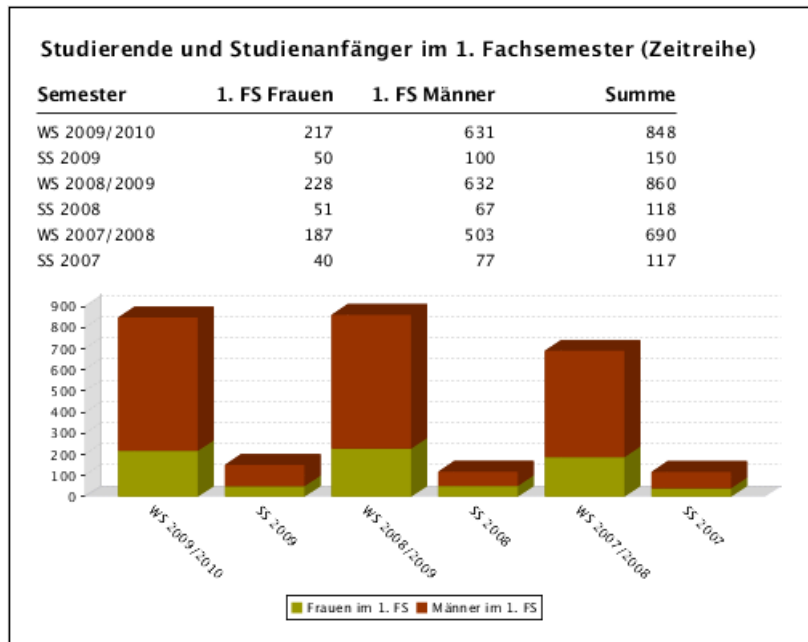
Die Eingabe funktioniert hier genau wie in dem Assistenten bei der Erstellung einer neuen Chart. Nur taucht jetzt das Problem auf, dass es kein Feld mit den Männlichen Studierenden gibt, sondern nur gesamt. Die Lösung ist aber ganz einfach. Zunächst geht man wieder in den "Expression editor" der "Value expression" und wählt das Feld "1. FS gesamt" aus, setzt dahinter ein Minuszeichen und danach wählt man noch das Feld "1. FS Frauen" aus. Jasper berechnet dann jeweils das Ergebnis und benutzt dieses als Value.



Nun erhalten wir als Ergebnis folgende Tabelle:



Es gibt noch viele Änderungsmöglichkeiten unter Einstellungen wie z.B. "Label rotation" lässt die Schrift unter den Balken rotieren, damit längere Schriftzüge möglich sind und "Series Colors" um die Farbe der Balken zu ändern.



2.8 Schriften

2.8.1 Allgemeines zu Schriften

Für den Entwurf von Berichten ist es wichtig, daß die jew. Schriftart auch beim Server-System und beim Client verfügbar sind.

Ein häufiger Fallstrick ist hier: unter Windows wird mit iReport entworfen, und dort z.B. die Schriftart "Arial" gewählt. Der Report soll auf dem Server unter Linux laufen, dort gibt es die Schriftart nicht. Resultat: der Bericht ist nicht kompilierbar.

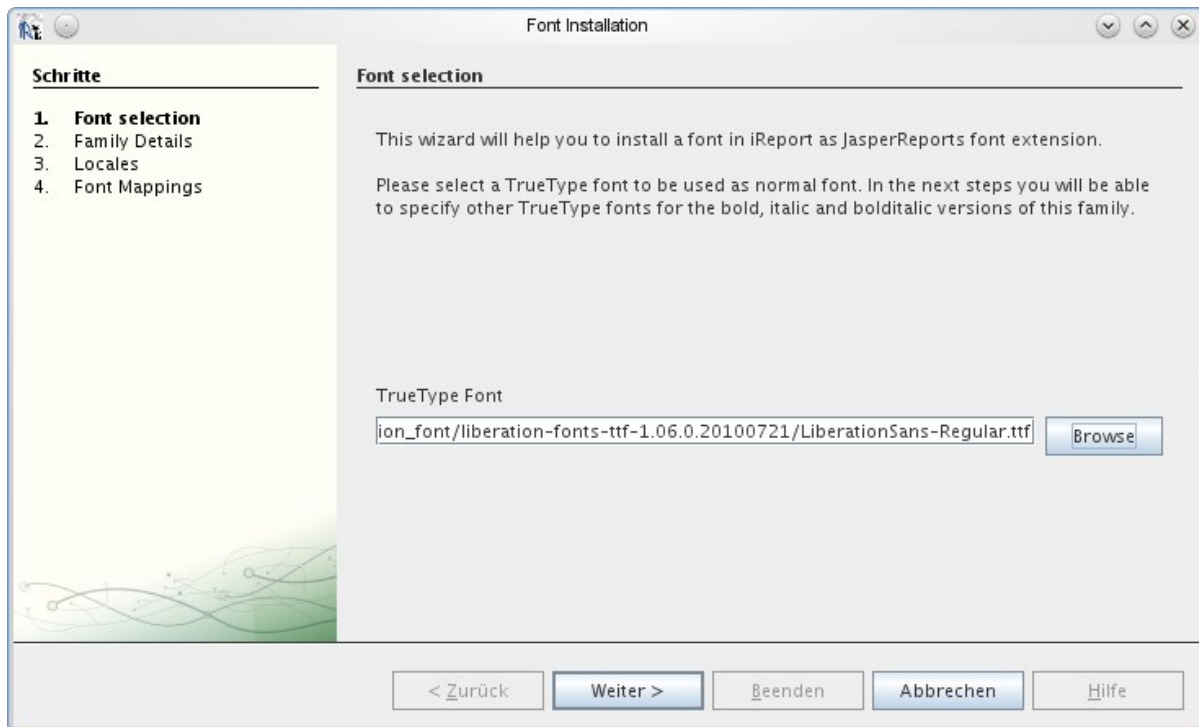
Bei PDF-Dateien lassen sich Fonts ggf. auch in die Datei einbetten, bei Exportformaten wie OpenOffice oder Word ist dies nicht möglich. Sie sollten daher Schriftarten wählen, die auch beim Client mit größter Wahrscheinlichkeit installiert sind. Die sicherste Wahl ist es, Font-Familien anzugeben, z.B. "Serif".

Im Zweifelsfall installieren Sie die Schriftart auf allen beteiligten Systemen. TrueType Fonts lassen sich in iReport importieren und nutzen, sie können sogar in PDF-Dateien standardmäßig eingebettet werden. Im folgenden wird die Schriftart "Liberation Sans" installiert, die man auf <https://fedorahosted.org/liberation-fonts/> herunterladen kann. Entpacken Sie das Download Paket in einem beliebigen Verzeichnis.

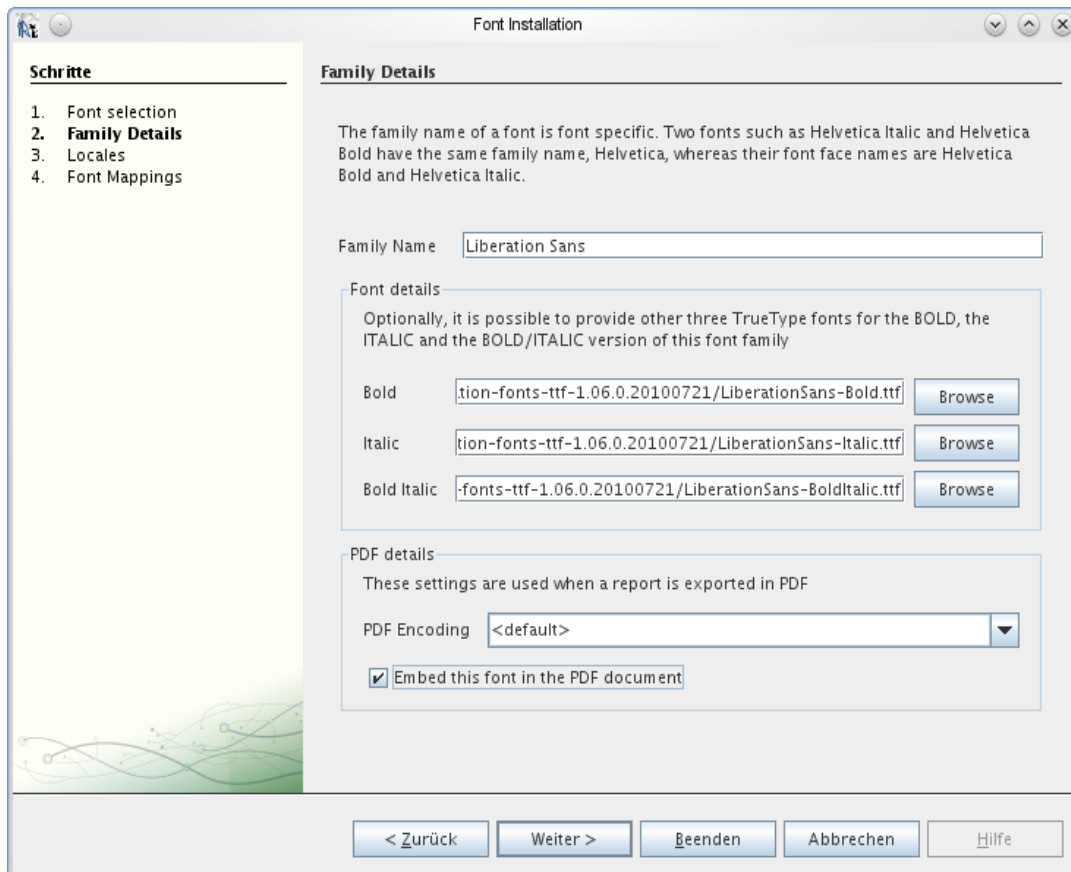
2.8.2 Schriften in iReport einbinden

Achtung: wenn Sie iReport unter Windows nutzen, müssen Sie für die folgenden Schritte iReport als Admin starten:

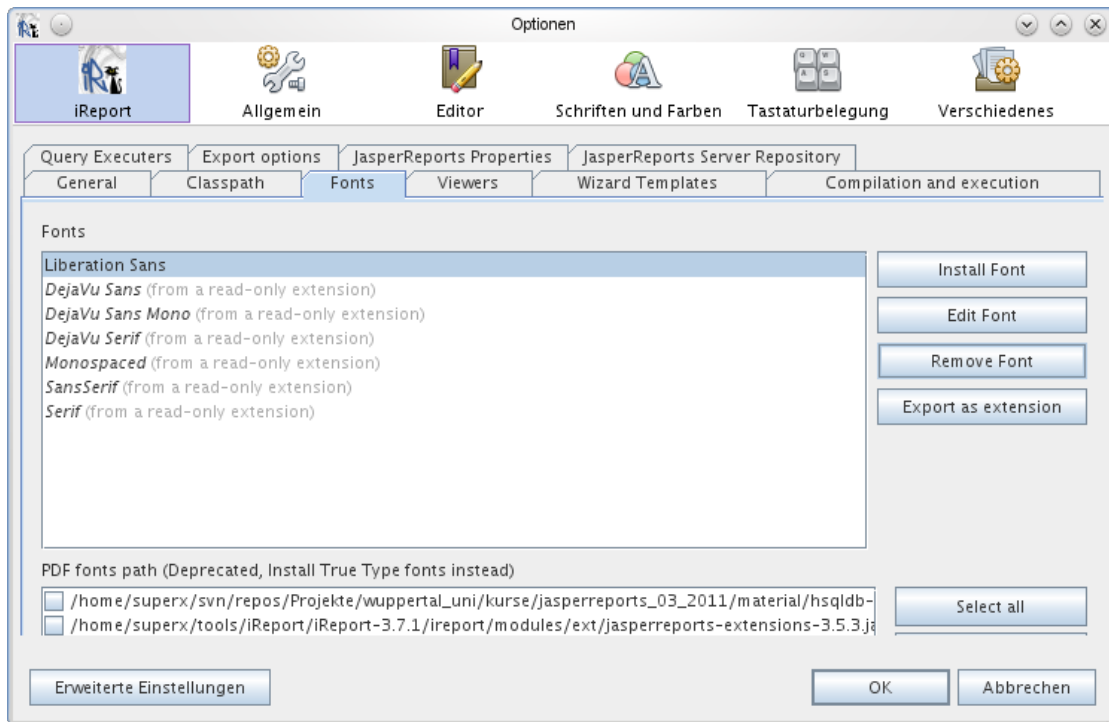
Im Menü "Extras"->"Optionen" können Sie den Font importieren:



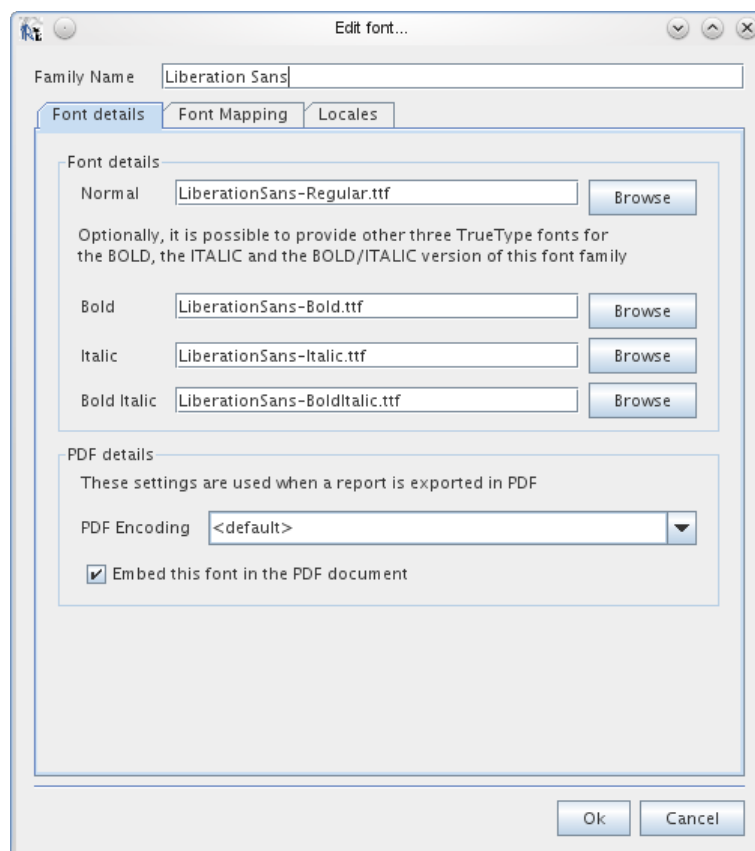
Wählen Sie aus der Dateiliste zunächst den "Regular"-Font aus, z.B. die Sans-Serif-Variante. Im nächsten Schritt geben Sie dem Font einen Namen, und Sie wählen die zugehörigen Dateien für Fettdruck etc:



Außerdem kreuzen Sie an, daß der Font in die jew. PDF-Datei eingebettet wird. Die nächsten zwei Dialoge (Locales, Font Mappings) lassen wir der Einfachheit halber erstmal weg. Wenn Sie dann OK drücken, ist der Font in iReport installiert. Sie können die Einstellungen jederzeit ändern, indem Sie auf "Edit Font" klicken:



Hier die Ansicht, wenn Sie auf "Edit Font" klicken:

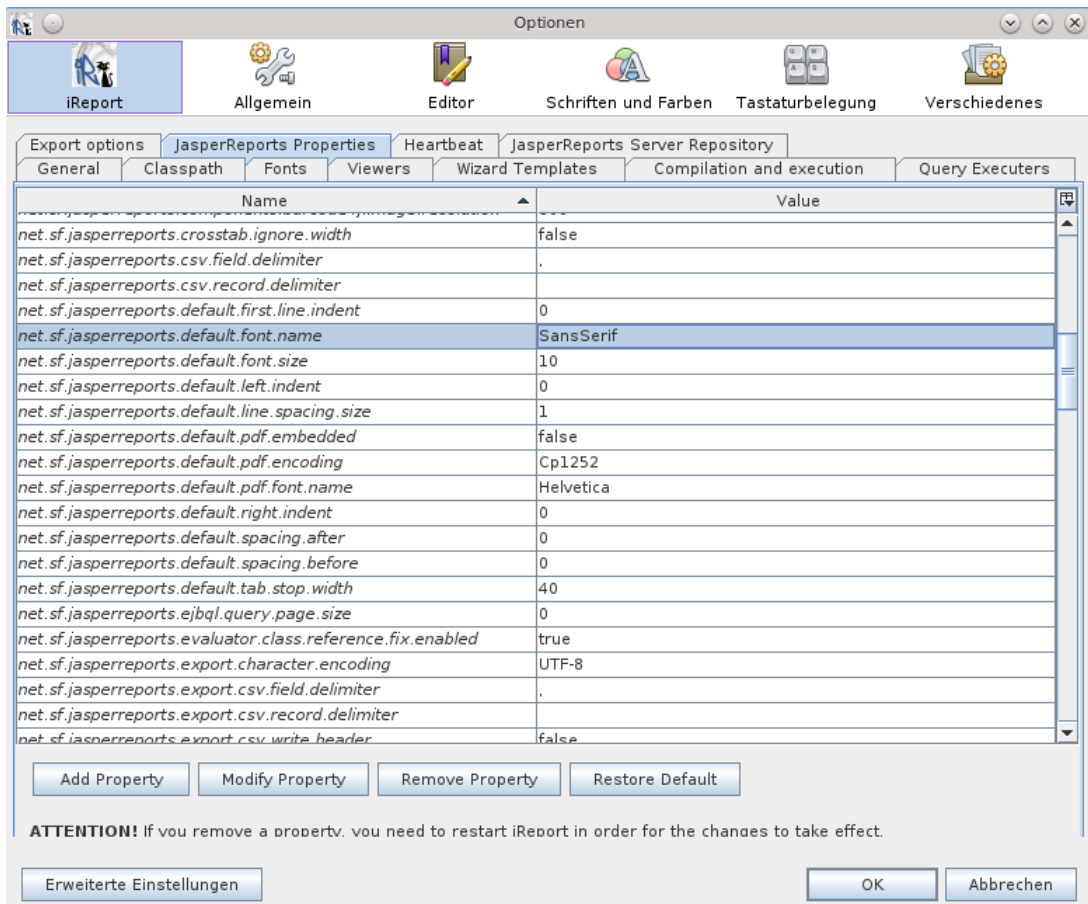


Am Ende des Formulars können Sie noch PDF-Exportoptionen wählen, z.B. ob die Schriftart in PDF-Dateien eingebunden werden soll. Achtung: beachten Sie lizenzrechtliche Fragen, die wenig-

ten Fonts sind ohne weiteres kopierbar. Wir empfehlen den Liberation-Font, der steht unter der OpenSource Lizenz GPL.

2.8.3 Default-Schriftart in iReport

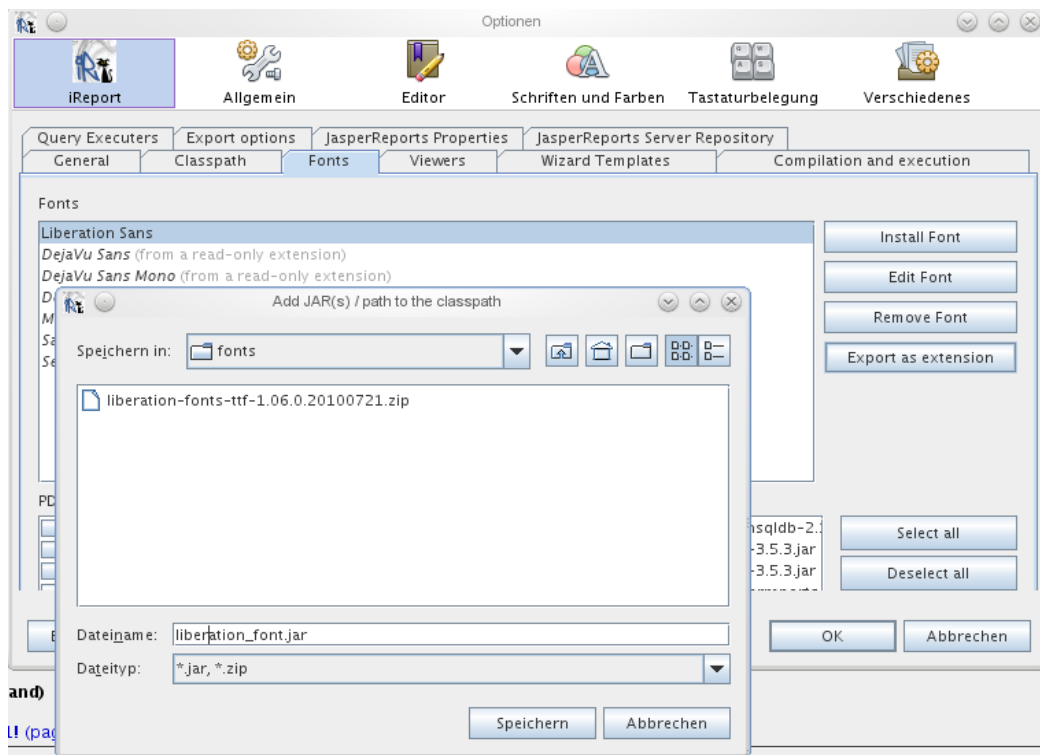
Sie können die Default Schriftart in iReport einstellen:



Stellen Sie den Default-Font auf "Liberation Sans". Danach müssen Sie iReport neu starten.

2.8.4 Fonts auf dem Server

Um die Fonts auf einem Ziel-Server einzubinden, kann man die Schriftart als Font Extension exportieren, die dann wiederum in den CLASSPATH des Servers (bei Webanwendungen WEB-INF/lib) kopiert werden müssen:



Nutzen Sie dazu den obigen Button "Export as extension", und wählen Sie einen Dateinamen mit der Endung ".jar".

Hinweis: Die Einbettung einer Font Extension in einer JasperReports-Laufzeitumgebung klappt erst mit JasperReports 3.7.2 oder höher.

Wenn Sie in iReport auch PDF als Vorschau wählen, erzeugt iReport im Verzeichnis der jrxml-Datei auch eine entsprechende PDF-Datei. Damit diese die Fonts enthält, müssen Sie auch in iReport im CLASSPATH die Font-Extension einbinden. In diesem Falle ist es vorteilhaft, die Endung ".jar" (nicht ".zip") zu wählen, weil iReport hier nur jar-Dateien einbindet.

Noch ein Hinweis zu den Attributen "PDF Font name" und "PDF Encoding":

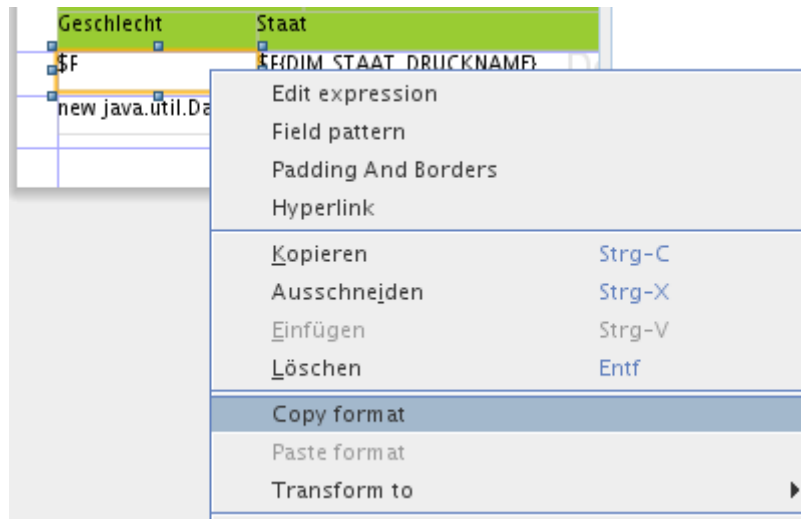
Text properties	
Font name	Liberation Sans ▼
Size	12 ▼
Bold	<input type="checkbox"/>
Italic	<input type="checkbox"/>
Underline	<input type="checkbox"/>
Strike Through	<input type="checkbox"/>
Pdf Font name	Courier ▼
Pdf Embedded	<input type="checkbox"/>
Pdf Encoding	CP1252 (Western E... ▼

Diese Attribute werden nicht mehr ausgewertet, wenn Sie mit Font Extensions arbeiten.

2.9 Effiziente Formatierung

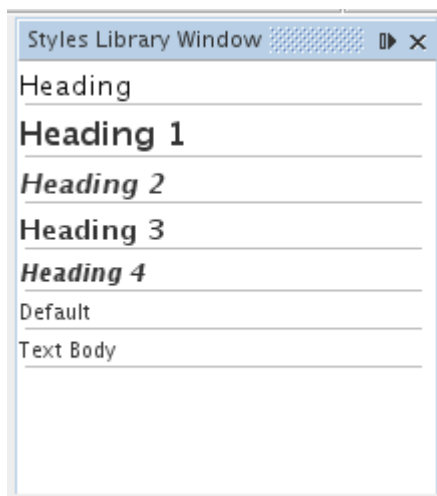
Wenn Sie keine Lust haben, jedes Textelement eines Berichts einzeln zu formatieren, empfehlen wir als ersten Schritt, Formatierungen zu kopieren.

Im Kontextmenü jedes Elements gibt es die Menüpunkte "Copy Format" und "Paste Format", die jeweils Formate kopieren und einfügen.



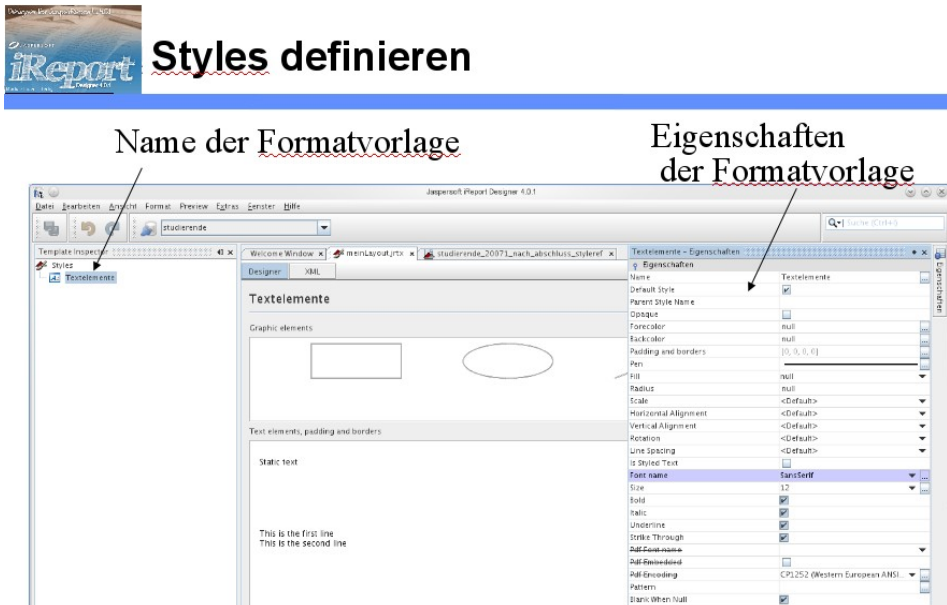
Für Hochschulen ist ein einheitliches Layout der Berichte wichtig, und so bietet es sich an, zentrale Layouts zu definieren und dann wiederzuverwenden. In iReport heißen Layouts auch "Styles" bzw. Formatvorlagen. Ähnlich wie in OpenOffice Writer oder MS Word lassen sich Formatvorlagen definieren und zuweisen. Bei Änderungen der Formatvorlage werden diese Änderungen auf alle Elemente automatisch übertragen - auch nachträglich.

In der "Styles Library" werden z.B. einige häufig verwandte Layoutvorlagen angeboten.



Sie können die Formate anwenden, indem Sie einfach die jew. Vorlage von der "Styles Library" per Drag and Drop auf das Ziel-Textfeld im Report Design ziehen und somit anwenden.

Sie können auch eigene Formatvorlagen zentral definieren. Im Menü Datei->New->Style geht es los: Wir erzeugen z.B. eine neue Style-Datei mit dem Namen "meinLayout.jrtx":



In der jrtx-Datei definieren sie einen oder mehrere Formatvorlagen. Wir definieren z.B. die Formatvorlage "Textelemente", und geben ihr die Eigenschaft "SansSerif, Schriftgröße 12".

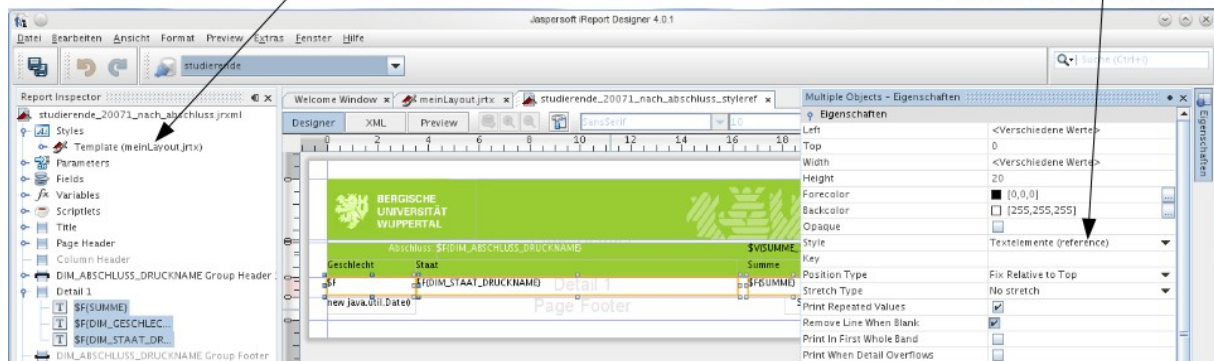
Danach können sie die Datei in der jrxml-Datei im Report Inspector unter "Styles" einbinden. Damit sind die Formatvorlagen in dieser Datei verfügbar, und können bei den Elementeigenschaften im Attribut "Style" zugewiesen werden:



Styles referenzieren

Referenz zur Style-Datei

Name der Formatvorlage



Wichtig:
kein absoluter Pfad

```

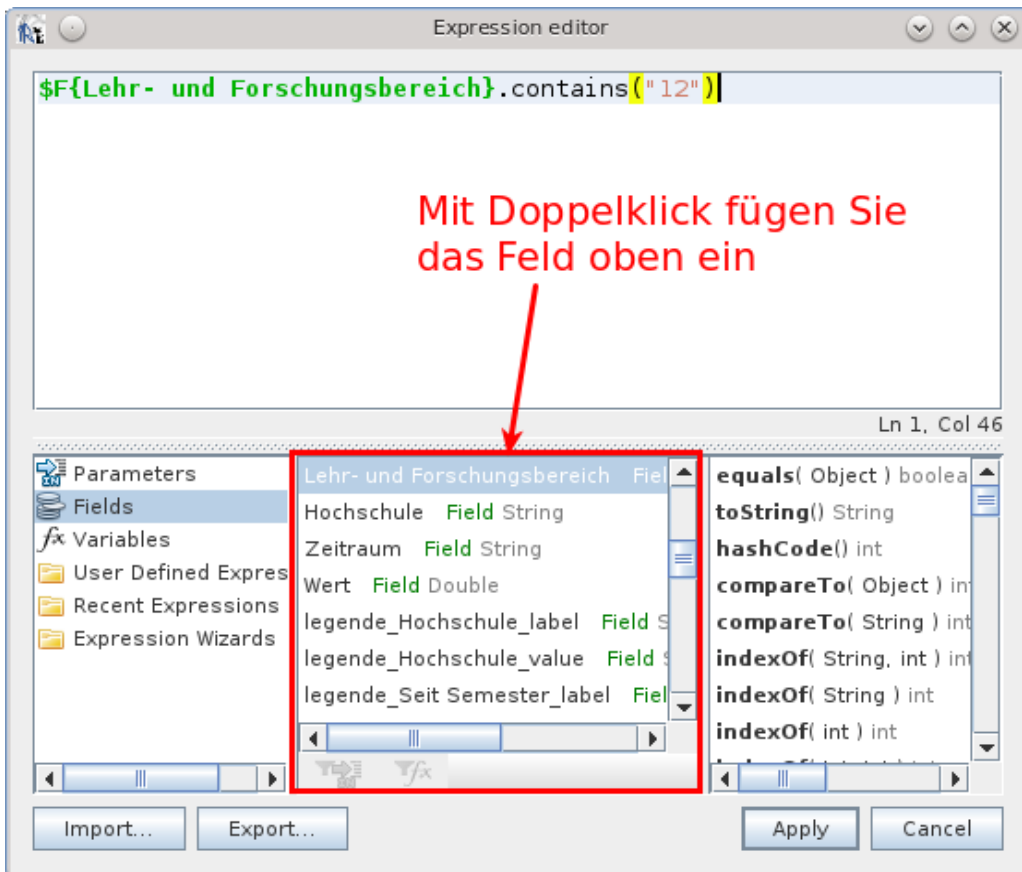
<?xml version="1.0" encoding="UTF-8"?>
<jasperReport xmlns="http://jasperreports.sourceforge.net/jasperreports" xmlns:ireport="http://jasperreports.sourceforge.net/ireport">
  <property name="ireport.zoom" value="1.0"/>
  <property name="ireport.x" value="0"/>
  <property name="ireport.y" value="0"/>
  <template><![CDATA["meinLayout.jrtx"]]></template>
  <queryString>
    <![CDATA[SELECT
      sum(FT_STUDIERENDE."SUMME") AS Summe,
      DIM_ABSCHLUSS."DRUCKNAME" AS DIM_ABSCHLUSS_DRUCKNAME,
      DIM_SEMESTER."DRUCKNAME" AS DIM_SEMESTER_DRUCKNAME,
    ]]></queryString>
  </jasperReport>

```

iReport nimmt standardmäßig den absoluten Pfad der jrtx-Datei, in der XML-Quelle sollten Sie diesen Pfad auf relativ umstellen.

2.10 Expression Editor

Im Expression Editor werden die für die Ausgabe bestimmten Werte angegeben. Zahlen werden ohne Zusatz eingetragen. Feste String Werte werden zwischen Anführungszeichen gesetzt. Felder und Variablen können über ein Doppelklick auf den jeweiligen Namen unterhalb des Eingabefeldes eingefügt werden.



Zu den Feldern und Variablen gibt es auch noch Funktionen in dem 3. unterem Kästchen.

Ein Beispiel für Textausgabe:

Es soll ausgegeben werden „Auswertungszeitraum 1.1.<<Jahr>> bis 31.12.<<Jahr>>“.

Anstatt zwei statische Textfelder und zwei dynamische zu machen, legt man eine Expression an:

"Auswertungszeitraum: 1.1."+\$F{legende_Jahr_value}+" bis 31.12."+\$F{legende_Jahr_value}

Der Clou an JasperReports ist: da das Programm auf der Programmiersprache Java beruht, kann man in Expressions echten Java Code einfügen bzw. um eigenen Java Code erweitern. Wenn Sie z.B. wissen dass das benutzte Feld "\$F{Lehr- und Forschungsbereich}" vom Typ String ist, können Sie alle in Java eingebauten String Funktionen benutzen, z. B. Substring, contains, indexOf, toLowerCase etc. Analog auch für andere Datentypen.

2.10.1 Bedingungen

Bedingungen sind in dem Expression Editor auch möglich. Wenn z.B. eine Variable nur Ausgegeben werden soll, wenn der Wert größer, kleiner oder gleich einem anderem vorgegebenen Wert oder Variable sein soll, wird folgende Syntax verwendet:

```
<<Bedingung>> ? <<Ausgabe wenn Bedingung wahr>> : <<Ausgabe wenn Bedingung falsch>>
```

Bedingungen:

```
String1 == String2
String1 != String2
Nummer1 > Nummer2
String1.contains( String2 )
String1 == null
usw.
```

Wie oben angegeben: Wenn die Bedingung wahr ist, wird der Wert nach dem Fragezeichen genommen. Wenn die Bedingung falsch ist, wird der Wert nach dem Doppelpunkt genommen.

2.10.2 Umgang mit Zeichenketten

Häufig werden für Berichtsausgaben Zeichenketten manipuliert. Dadurch, daß JasperReports innerhalb der Java-Runtime läuft, stehen uns mächtige Methodenaufrufe zur Verfügung, die in Java schon eingebaut sind.

Für Zeichenketten hier eine Übersicht:

<http://docs.oracle.com/javase/1.4.2/docs/api/java/lang/String.html>

2.10.3 Umgang mit Datumsfeldern

Fürs Berichtsdesign auf aggregierter Ebene ist es oft praktisch, Datumswerte auf "grobere" Kategorien zu aggregieren, z.B. Monate, Quartale, oder Jahre. Um solche Werte auszugeben bzw. danach zu gruppieren können Sie eigene Variablen anlegen, die dann im Expression Editor mit Hilfe von Java berechnet werden. Hier ein Beispiel für eine Transformation vom Datum zur Einheit "Jahr-Monat":

```
(( ${Datum} .getYear() + 1900 ) * 100 ) + ( ${Datum} .getMonth() + 1 )
```

Wenn wir annehmen, dass im Feld "Datum" ein Datums- oder Zeitstempel-Wert steht, dann wird über die o.g. Expression daraus ein Zahlenwert nach dem Muster Jahr (vierstellig) + Monat (zweistellig), z.B. für den 13.5.2013 würde daraus der Wert 201305. Nach diesem ganzzahligen Wert kann man dann sortieren und gruppieren.

Anmerkung: die o.g. Methoden `getYear()` und `getMonth()` sind seit einigen Jahren in Java "deprecated", d.h. sie könnten in zukünftigen Java Versionen entfallen. Aber da dies in den letzten 7 Jahren nicht passiert ist, können wir damit noch arbeiten und optimistisch sein. Falls die Methoden wirklich entfernt werden, muss man eigene statische Methoden mit dem in Java eingebauten Calendar-Objekt erstellen, Codebeispiele dazu gibt es viele im WWW.

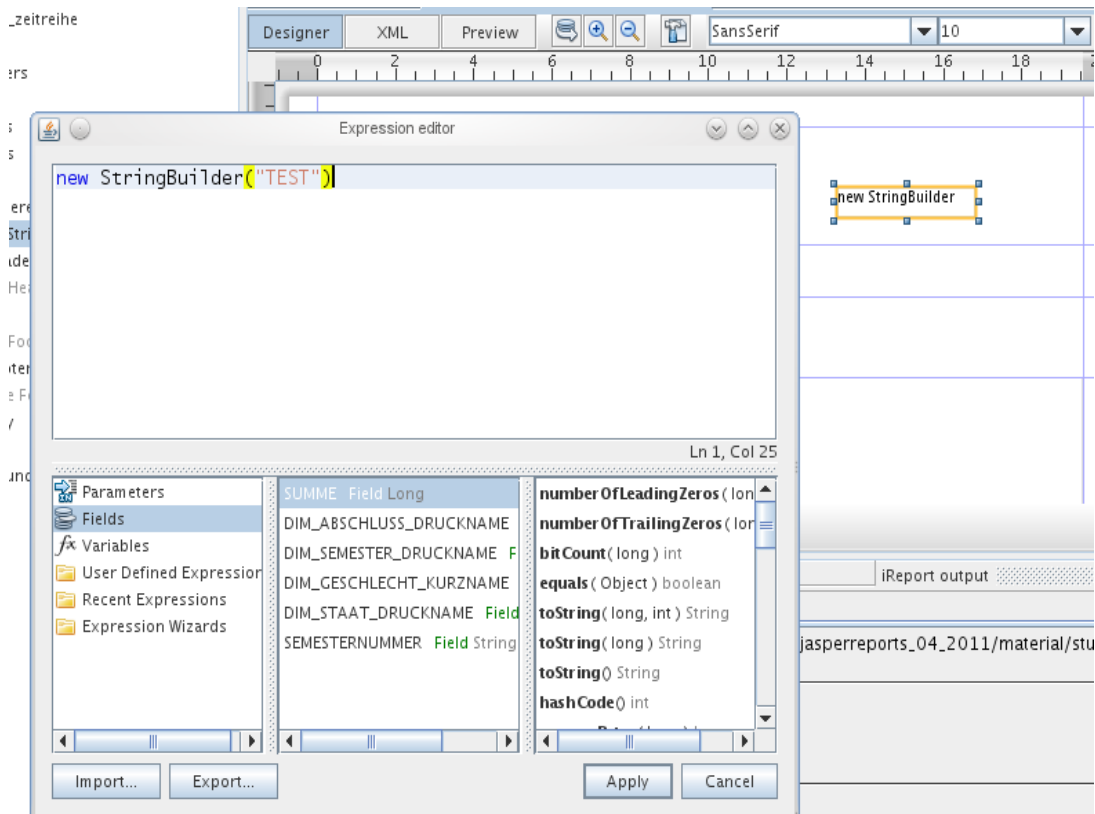
2.10.4 Besonderheiten / Workarounds

- Bei Feldern mit Länge 1 klappte einfaches equals nicht, mit einer substring Anweisung geht es dann:
`#{legende_Art des Berichts_value}.substring(0,1).equals("B")`
- Folgejahr berechnen: Selbst es das Feld Jahr gibt und es als Integer definiert ist in der Felderinfo, klappt der Bericht nur in iReport nicht in SuperX nicht wenn man schreibt:
`#{legende_Jahr_value}.intValue()+1`
 geht nicht, statt dessen:
`new Integer(#{legende_Jahr_value}).intValue()+1`
- Statt das Element table zu nutzen kann man auch manuell eine Tabelle erzeugen, in dem man per Rechtsklick auf ein Element Padding und Borders bearbeitet. Siehe <http://www.youtube.com/watch?v=LbU6qTj3zQM>
- Achten muss man dann darauf, dass Felder die Null sein können nicht „remove when blank“ haben und als TextFieldExpression besser nehmen: `#{Betrag}==null?" " :#{Betrag}`
- außerdem, wenn die erste Spalte einen stretch with overflow hat auch bei „stretch type“ „relative to tallest object“ einstellen
- Zeilenumbruch in Feld
 Markup auf html setzen und dann z.B. "Plan
"+#{legende_Jahr_value}

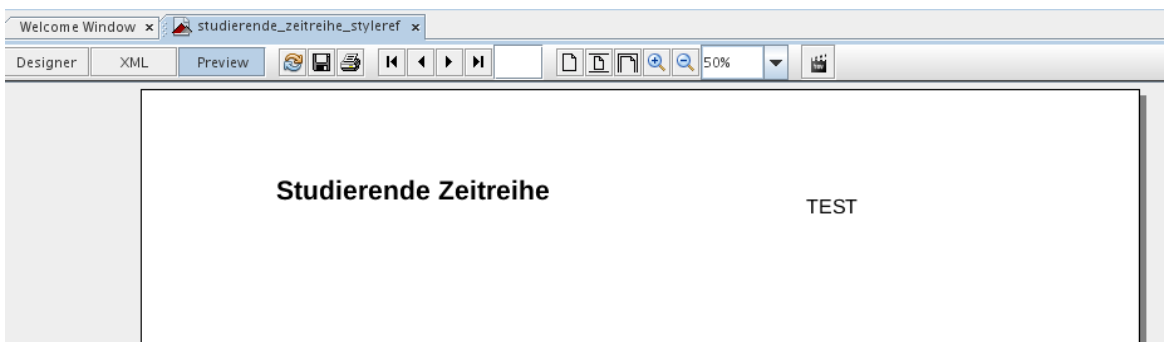
2.11 Fortgeschrittene Techniken bei Feldern

Textfelder enthalten standardmäßig nur den Feldnamen, aber die zugrunde liegenden [Ausdrücke](#) lassen sich erweitern. Dabei kann auch alle Java Klassen zugegriffen werden, die sich im CLASS-PATH befinden, z.B. [java.lang.String](#).

Hier ist ein Beispiel, in dem die Klasse "StringBuilder" (wird mit Java ausgeliefert , java.lang.StringBuilder) genutzt wird, um einen Text auszugeben:



Im Ergebnis sieht das so aus:



Das Beispiel verdeutlicht: Man könnte da auch `de.meineKlasse.erzeugeText("mein Text")` schreiben, und diese Klasse dann in den CLASSPATH bei Jasper Start legen, und voila! Durch diese Funktionalität lassen sich also beliebige Java Klassen einbinden. Das gleiche geht übrigens auch mit Groovy.

2.12 Einrücken von Feldern

Das Einrücken von Feldern macht man am besten, in dem man mehrere Textfelder definiert und jeweils eine `printWhenExpression` auf die Ebene einbaut.

Alternativer Versuch war folgendes:

Wobei Nachteil bei sehr langen Texten, bei denen stretch with overflows zur Geltung kommt, dann die zweite Zeile nicht eingerückt wird.

Das Einrücken von Felder kann man für Textfelder (Strings) machen, indem der SuperX-Bericht eine Spalte einruecken definiert, der je nach gewünschter Einrückung eine Anzahl von x enthält, also z.B.

Name	Betrag	Einrücken
1. Überschrift	30000	X (mindestens 1 x nötig)
1.1 Unterüberschrift	15000	Xxxxx
1.3. UnterUnterüberschrift	3000	xxxxxxxxxx

Dann kann man ein Textfeld erzeugen und bei TextFieldExpression einfach eintragen:

```
$F{Einruecken}.replace('x','')+${Name}
```

Für Zahlfelder (z.B. Double) geht das so nicht so gut, weil dann das Pattern der Darstellung ###0,00 nicht mehr greift.

Man definiert drei Felder nebeneinander, Beispiel:

\${Name}	\${Betrag} PrintWhenExpression \${Hierarchie_Ebene}.intValue() >=6	\${Betrag} PrintWhenExpression \${Hierarchie_Ebene}.intValue() <6
----------	---	--

Bei Ebene 1-5 wird ganz rechts ausgegeben, bei Ebene >=6 wird mittig ausgegeben.

2.13 Berechnungen

Bei Berechnungen muss immer auch auf (ich nenne es mal) die Gültigkeit der Variable geachtet werden. Wenn eine Variable eine Summe bildet kann diese für z.B. „jetzt“ oder für eine Gruppe oder den ganzen Report gelten. Je nach Gültigkeit gilt z.B. die Summe nur für die aktuelle Gruppe (bei Gruppenbändern).

Achtung: Es gibt nicht nur eine Gültigkeit bei der Variable selber (Reset type) sondern auch bei dem Element in dem Bericht. Wenn eine Variable in den Bericht gezogen wird, besitzt das entstandene Element eigene Eigenschaften, bei dem dann die „Evaluation Time“ eingestellt werden kann.

2.13.1 Prozentrechnung

Für die Prozentrechnung musst zuerst eine Variable z.B. mit dem Namen "Gesamt" anlegen mit den Eigenschaften:

```
Calculation = Sum  
Reset type = Report
```

Dannach kann ein Textfeld mit dem Inhalt:

```
$F{Summe} / $V{Gesamt}
```


erstellt werden mit den Eigenschaften:

```
Expression Class = Double
```

```
Pattern = #.##0.00 % (kannst du unter Prozent auswählen)
```

```
Evaluation Time = Auto
```

In der Variable „Gesamt“ wird die Summe aus allen Werten des Feldes Summe gezogen.

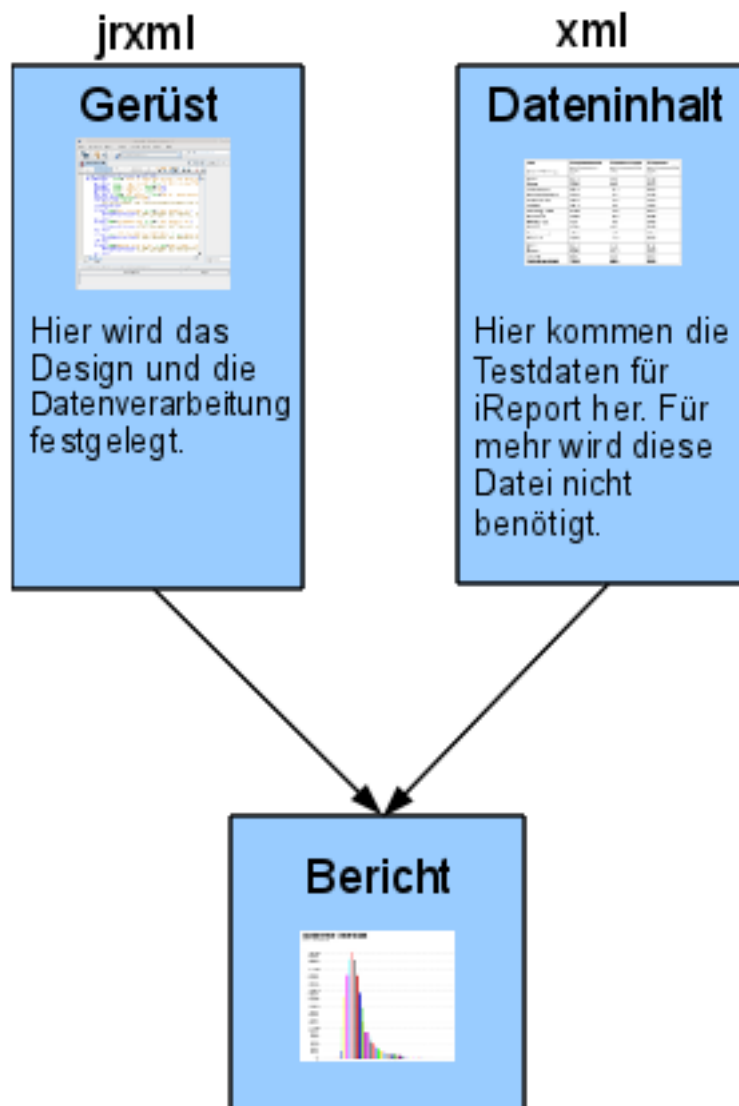
3 Nutzung in Edustore

3.1 Berichtsentwicklung bei Grunddaten und Basisberichten

3.1.1 Allgemein

In den Grunddaten- und Basisberichten kann man beliebige Ergebnistabellen lokal als XML speichern, und dann als Datenquelle in iReport einbinden. Dann kann man den Bericht lokal gestalten, und, wenn er fertig ist, in die Weboberfläche einhängen. Die XML Datei wird nur für die Testausgabe bei iReport verwendet und dient als Ersatz der Datenbank. Wenn der Bericht in der Weboberfläche eingehängt ist, wird dieser direkt mit frischen Daten versorgt.

Für die Folgenden 2 Punkte „Datenquelle erzeugen“ und „Template erzeugen“ gibt es [hier](#) einen Kurzfilm.



3.1.2 Datenquelle erzeugen

Wenn die Ergebnistabelle eines Berichtes vorliegt, erhält man über den XML Export von SuperX die Datenquelle für den Jasper Report.



Diese Datei speichert man lokal ab: Wenn das Ergebnis im Browser geöffnet wurde muss man sich den Quelltext der Seite anzeigen lassen, den Quelltext komplett markieren und kopieren (Tastenkombination Strg + a und dann Strg + c), in einen UTF-8-fähigen Texteditor einfügen (Strg + v) und dann als eine xml-Datei abspeichern.

Falls der Tomcat in ISO Kodierung läuft, muss für iReport in der ersten Zeile bei encoding ein UTF-8 eingetragen werden.

Beispiel:

```
<?xml version="1.0" encoding="ISO-8859-1"?>
```

wird ersetzt mit:

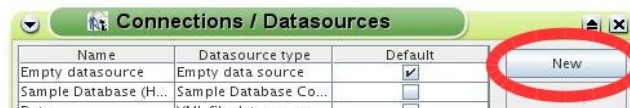
```
<?xml version="1.0" encoding="UTF-8"?>
```

In iReport sind dann folgende Schritte zu tun, um die neue Datenquelle einzubinden:

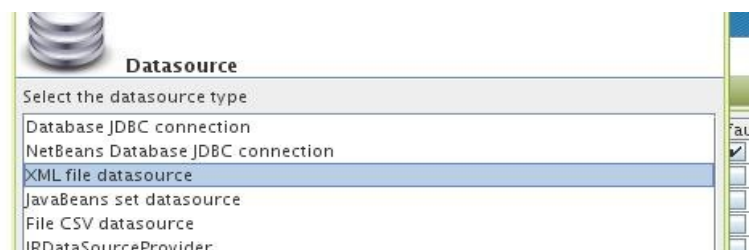
Klick auf „Report Datasources“. Es öffnet sich ein Fenster, in dem die vorhandenen Datenquellen angezeigt werden.



In diesem Fenster klickt man auf „New“ um eine neue Datenquelle anzulegen.



Nun wählt man „XML file datasource“ aus und klickt auf „Weiter“.



Im letzten Schritt wählt man einen Eindeutigen Namen aus, der auch später in dem Dropdown Menü erscheint, gibt den Dateipfad an und ändert die restlichen Einstellungen wie im Screenshot.

The screenshot shows the 'XML file datasource' configuration window. It includes fields for Name, XML File, Date pattern, Number pattern, and Locale/Time zone. Red arrows highlight the Name field, the XML File field, the selected radio button for XPath expressions, the Date pattern field, the Number pattern field, the Locale dropdown, and the Time zone dropdown.

Die Formate für Datum und Nummer müssen angegeben werden, da sonst die amerikanischen angenommen werden. Bei Locale müssen Sie "Englisch" angeben, weil Dezimalzahlen in den Edustore-XML-Quellen den "." als Dezimaltrenner haben.

Hinweis: in iReport 3.7.1 werden trotz des korrekten Zahlenformats die Dezimalstellen von Beträgen immer mit ",00" wiedergegeben. Dies liegt an der "Locale", man muss diese auf "Englisch" umstellen. Das Problem ist allerdings: sobald man das macht, kann man die XML-Datenquelle in der Datenquellenverwaltung nicht mehr ändern, d.h. man muss sie bei Änderungen löschen und neu einfügen.

Nach dem Speichern erscheint die Datenquelle im dropdown Menü und kann ausgewählt werden. Damit man die Datenquelle nutzen kann, braucht man auch noch ein Template.

3.1.3 Template erzeugen

Um ein leeres Template zu erzeugen, geht man wieder in die Ergebnistabelle der selben Maske, deren Datenquelle erstellt wurde, wählt im Dropdown Menü neben „Bericht entwerfen“ „Leerer Bericht“ aus und klickt auf das Icon daneben „Export als JasperReport-Template“.

Bericht - Anzeige

Sie sind hier: [Grunddaten und Basisberichte](#) > [Personal Stellen](#) > [Personal](#) > [Personal Buchungen Datenblatt/Bericht erstellen](#) > Datensätze/Personal Buchungen Datenblatt

Bericht entwerfen: **Leerer Bericht**

Personal Buchungen Datenblatt

Legende
 Monat: **August** ; Jahr: **2009** ; User: superx Stand: 18.03.2010

Monat	Jahr	Kapitel	Titel	Buchungsstelle	Kostenstelle	Kostenstelle Text	Kostenart	Vertragsart	Personal Nr.	Name	Brutto	Sozialversicherung	Zusätzliche Versicherung	Summe
Kein Satz gefunden														

Diese Datei speichert man Lokal ab. Wenn das Ergebnis im Browser geöffnet wurde muss man sich den Quelltext der Seite anzeigen lassen, den Quelltext komplett markieren und kopieren (Tastenkombination Strg + a und dann Strg + c), in einen UTF-8-fähigen Texteditor einfügen (Strg + v) und dann als eine jrxml Datei abspeichern. JR in der Endung jrxml steht für JasperReports. Damit können die Datenquellen und die Templates unterschieden werden und sogar den selben Dateinamen tragen (nur unterschiedliche Dateierweiterung).

Sie sollten es sich nicht zumuten und versuchen über iReport einen neuen Bericht zu entwerfen. Nehmen Sie immer den "leeren" Bericht aus der entsprechende Maske, da dort sehr viele Kleinigkeiten voreingestellt sind. Sie sparen sich damit immens viel Arbeit.

3.1.4 Template einbinden

3.1.4.1 Reportdesign verlinken

Um das erstellte Template einzubinden, muss zunächst die jrxml Datei in das Verzeichnis webapps/superx/WEB-INF/reports kopiert werden. Danach muss das Stylesheet im System bekannt gegeben werden. In Grunddaten und Basisberichte geht man dazu in „Administration“ → „Masken“ → „Stylesheet suchen“ und schickt die Maske so ab. Danach klickt man bei einem beliebigen Stylesheet auf den „bearbeiten“ Button ganz rechts in der Tabelle.

Stylesheet suchen

Legende
User: superx Stand: 15.01.2010

Datensatz 1 - 30 von insgesamt 35 Sätzen ▶ ❏

Stylesheet Nr	Dateiname	Beschreibung	Bearbeiten
85	cob/tabelle_html_12000.xml	Kurze Zusammenstellung von Kennzahlen aus der Kostenrechnung im Layout eines Kanzlerbriefbogens	
47	huel_pdf_allg.xml	Hül-PDF	
57	kostentraeger_lehre_fo.xml	PDF-Ausgabe Kostenträger Lehre	

Klickt dann auf „Neuen Datensatz erzeugen“.

Stylesheets verwalten. Erläuterung

Dateiname	huel_pdf_allg.xsl
Beschriftung	HÜL-PDF
Beschreibung	HÜL-PDF
Tabelle oder Maske	Stylesheet für eine Ergebnistabelle ▾
Dateityp	application/pdf

In Dateiname kommt der Dateiname der jrxml Datei rein, welche sich in dem reports Ordner befindet. Beschriftung und Beschreibung sollte eindeutig dem Stylesheet zuzuordnen sein. Für Jasper-Report wählt man bei „Tabelle oder Maske“ „Stylesheet für eine Ergebnistabelle“ aus. Der Dateityp stellt das default Ausgabeformat fest.

Um dieses Stylesheet nun einer Maske zuzuordnen, geht man in die Maske „Administration“ → „Masken“ → „Maske suchen“, sucht in dem Feld „Maske“ die gewünschte Maske aus und klickt dann auf „Abschicken“. Nun klickt man auf den Button Stylesheets neben der Maske, der das neue Stylesheet zugeordnet werden soll.

Maske suchen

Legende

Maske: **19400 - Personal Buchungen Datenblatt** ; User: superx Stand: 15.01.2010

Maske Nr	Name	Erläuterung	Sachgebiet	Bearbeiten	Sachgebiete	Stylesheets
19.400	Personal Buchungen Datenblatt	Auflistung aller an einer Institution angestellten Personen	Personal			

Datensatz 1 - 1 von insgesamt 1 Satz.

In dem neuen Fenster wählt man das neu erstellte Stylesheet aus und speichert diese Einstellung ab.



Damit ist der Bericht der Maske zugeordnet.

Normal wird im Servlet direkt vom Berichtsergebnis nach JasperReports übergeben, d.h. ohne den Zwischenweg XML. Wenn aber im Dateinamen *_xmlsource.jrxml steht, wird Datenquelle XML erzwungen (für Makroberichte, Unterberichte, Subdatasets oder spezielle XPATH-Selektionen wichtig).

3.1.4.2 Probleme beim Deployment im Server

Es kann u.U. beim Deployment eines Berichts im Server das Problem auftreten, daß ein Bericht im lokalen iReport mit der XML-Datenquelle funktioniert, aber im Server für einzelne Merkmale kein Ergebnis geliefert wird. Dies hat folgende Ursache:

3.1.4.2.1 Feldnamen im Berichtsdesign weichen vom Datenbank-Feldnamen ab

Bei der lokalen Entwicklung in iReport werden die Berichte mit der Datenquelle XML erzeugt. In den Muster-JRXML-Dateien, die man über das Feld "Bericht erzeugen" generiert, sind die Felder über die folgende Notation deklariert:

```
<field name="Fakultät/Fachbereich (Schlüssel)" class="java.lang.String">
  <fieldDescription><![CDATA[col[@id="4" and wert!
=" " ]/wert]]></fieldDescription>
</field>
```

Dies bedeutet: Das Feld "Fakultät/Fachbereich (Schlüssel)" hat den Typ "String" und befindet sich im XML-Datenstrom in der vierten Spalte (col@id=4). Dabei befindet sich in Datenblattberichten im Spaltennamen immer ein "\n" vor der Zeichenkette "(Schlüssel)", um einen Zeilenumbruch im generischen Standardlayout zu erzeugen. Dieses "\n" wird dann für iReport und JasperReports automatisch durch ein Leerzeichen ersetzt.

Wenn die XML-Datenquelle nicht verändert wird, dann ist das Feld immer mit dem gleichen Inhalt gefüllt.

Wenn man den Bericht dann auf den Server kopiert und einbindet, ohne im Dateinamen "_xmlsource" zu verwenden, dann wird das Feld nicht über die Spaltennummer identifiziert, sondern über den Feldnamen, hier z.B. "Fakultät/Fachbereich (Schlüssel)". Das bedeutet, daß der Server genau diesen Feldnamen zurückliefern muss. Das kann u.U. abweichen:

- Der Server bzw. die zugrunde liegende Maske wurde verändert, so daß der Feldname geringfügig anders ist. Das sollte eigentlich nicht passieren, kann aber vorkommen.
- Beim Betrieb in HISinOne gibt es in allen Versionen bis 3.02 noch eine kleine Veränderung, wenn das Modul des zugrunde liegenden Datenblattberichts bzw. der Maske direkt per Tomcat-Start installiert wurde, und nicht in dem Edustore-Installation/Upgrade-Menü. Wenn die Module beim Tomcat Start in eine leere Datenbank installiert werden, dann werden Zeichenketten mit dem Feldinhalt "\n" verfremdet nach "n" (also der "\" wird entfernt). Das führt dann dazu, daß das Feld ohne den "\" in der jrxml-Datei deklariert wird, also z.B.

```
<field name="Fakultät/Fachbereichn(Schlüssel)" class="java.lang.String">
  <fieldDescription><![CDATA[col[@id="4" and wert!
=" "]/wert]]></fieldDescription>
</field>
```

Wenn man danach dann einen Upgrade der jew. Komponente ausführt, dann wird die Spalte in der Maske umbenannt nach "Fakultät/Fachbereich\n(Schlüssel)" und der Feldname korrekt nach "Fakultät/Fachbereich (Schlüssel)" umgesetzt. Durch das "n" vor der Zeichenkette "(Schlüssel)" paßt der Feldname in der jrxml-Datei nicht mehr zum Ergebnis aus der Datenbank, und das Feld ist leer. So kann es passieren, daß ein Bericht in iReport funktioniert, im Server aber nicht. Im folgenden die Lösung.

3.1.4.2.2 Korrektur der Feldnamen im Berichtsdesign

Um sicherzustellen, dass der Feldname der jrxml-Datei mit der jeweils aktuell benutzten Datenbank-Version übereinstimmt, geht man wie folgt vor:

- Erzeugen Sie in Edustore eine HTML-Ausgabe der zugrunde liegenden Abfrage, bei Datenblattberichten ist dies das "generische Standardlayout".
- Klicken Sie dann oben im Berichtskopf auf den Button "Bericht entwerfen" -> "leerer Bericht"
- Im Quellcode sehen Sie die Zeilen `<field name="Berichtsname" class="java.lang.String"> ...` Hier ist jedes einzelne Feld aus der Datenbank deklariert.
- Markieren Sie alle Zeilen bis zum letzten "`<field...>`"-Element, i.d.R. ist dies das Feld `<field name="user"...`, und kopieren Sie dies in die Zwischenablage
- Öffnen Sie die jrxml-Datei mit einem UTF-8-fähigen Editor, z.B. Jedit.
- Ersetzen Sie genau die "`<field...>`"-Passage aus der Zwischenablage.
- Speichern Sie den Bericht, und testen Sie die Vorschau in iReport. Wahrscheinlich kommt eine Fehlermeldung "Field ... does not exist" o.ä., dann müssen Sie das Feld mit Suchen->Ersetzen umbenennen.
- Das machen Sie so lange, bis die Vorschau funktioniert. Danach können Sie die jrxml-Datei wieder auf den Server kopieren. Die Felddeklaration paßt dann zum Server.

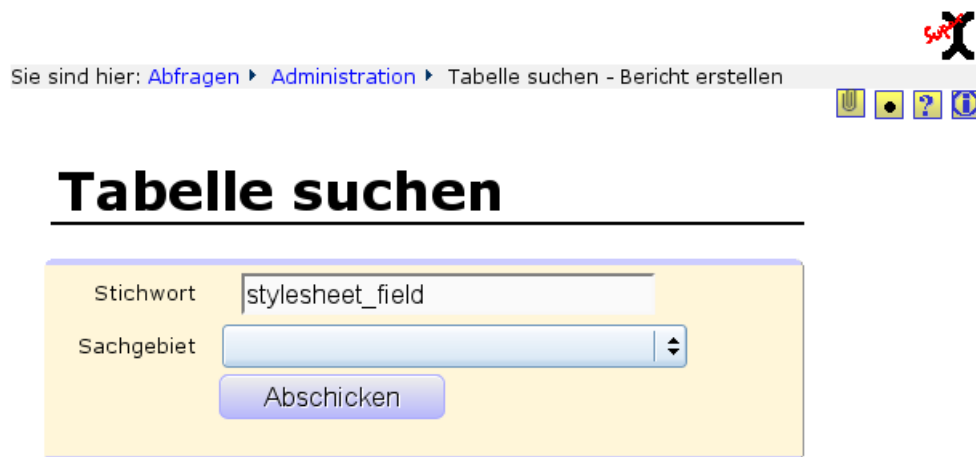
3.2 Zuordnung von Default-Feldbelegungen zu Stylesheets

Es können für Stylesheets Default-Feldbelegungen für Stylesheets hinterlegt werden. Hintergrund dieser Funktion ist, dass bei speziellen Stylesheets standardmäßig beliebig viele Spalten vorbelegt werden können. In Datenblattberichten gibt es z.B. das Feld "Felder", in dem die auszugebenden Spalten angegeben werden. Aus Performancegründen sollten nur die Spalten ausgegeben werden, die auch benötigt werden. So kann dann z.B. für JasperReport Stylesheets hinterlegt werden kön-

nen, welche Spalten auf jeden Fall benötigt werden. Der Anwender weiß nicht welche Spalten benötigt werden und würde im Normalfall alle Spalten ausgeben lassen um keine Fehlermeldung zu erhalten. Dadurch würde aber wiederum der Bericht unnötig lange dauern, da alle Spalten berechnet werden müssen, auch die, welche nicht benötigt werden.

Wenn Sie ein Stylesheet erstellt haben und dieses in SuperX eingebunden haben gehen Sie wie folgt vor:

Sie gehen in die Maske "Administration -> Tabelle suchen", geben dort bei Stichwort "stylesheet_field" ein und schicken die Maske ab.



Sie sind hier: [Abfragen](#) > [Administration](#) > Tabelle suchen - Bericht erstellen

Tabelle suchen

Stichwort

Sachgebiet

In der folgenden Ergebnistabelle klicken Sie bei "stylesheet_field_list" auf den "Bearbeiten" Button.



Export: **Druckversion** XML PDF XLS **Export als JasperReport-Template**

Sie sind hier: [Abfragen](#) ▶ [Administration](#) ▶ [Tabelle suchen/Bericht erstellen](#) ▶ Datensätze/Tabelle suchen

Bericht entwerfen: Leerer Bericht

Tabelle suchen

Stichwort: **stylesheet_field** ; User: superx Stand: 14.04.2011

Name	Tabelle	Beschriftung	Bearbeiten
stylesheet_field_edit	stylesheet_field	Stylesheets und zugehörige Felder	
stylesheet_field_list	stylesheet_field	Stylesheets und zugehörige Felder	

Datensatz 1 - 2 von insgesamt 2 Sätzen.

[Erläuterung](#)

In dem nun geöffneten Fenster sehen Sie alle Zuordnungen, welche bereits eingetragen sind. Hier wird aber lediglich das Stylesheet ausgegeben und nicht die Tabellen und Federzugehörigkeit. Dies hätte wegen der vielen Auswahlmöglichkeiten jeden Browser gekillt. Um Einzelheiten zu erfahren müssen Sie daher auf den Details-Button klicken. Auch wenn Sie eine neue Zuordnung erstellen möchten klicken Sie zunächst auf den Detail-Button eines beliebigen Stylesheets.

Stylesheets und zugehörige Felder.




Tid	Stylesheet	Details
2	Semester nach Geschlecht (tabelle_16650_semester_geschlecht.jrxml)	
3	Semester nach Geschlecht (tabelle_16650_semester_geschlecht.jrxml)	
4	Semester nach Geschlecht (tabelle_16650_semester_geschlecht.jrxml)	

Jetzt können Sie die Zugehörigkeit des Stylesheets zu Tabelle und Feld ändern oder eine neue Zuordnung erstellen.

Achtung: Wenn ein Tabellename ausgewählt und abgespeichert wurde, können nur Spalten dieser Tabelle ausgewählt werden. Wenn Sie die Tabelle ändern wollen, müssten Sie zunächst die Ta-

belle ändern, abspeichern und dieses Formular erneut aufrufen. Erst dann würden die Spalten der ausgewählten Tabelle in der Listbox Fieldname erscheinen.

Stylesheets und zugehörige Felder.

Tid	2
Stylesheet	Semester nach Geschlecht (tabelle_16650_semester_geschlecht.jrxml)
Tablename	sos_stg_aggr
Fieldname	sos_stg_aggr.geschlecht

Um eine neue Verknüpfung zu erstellen klicken Sie zuerst auf den + Button oben rechts. Danach erscheint das gleiche Formular nur ohne ausgefüllte Felder. Hier können Sie das Stylesheet und den Tabellennamen auswählen und die dazu passende Spalte der Tabelle die Sie zuvor ausgewählt haben. Achten Sie darauf die Tabelle zu nehmen, welche im Datenblatt verwendet wird. Danach nur noch über das Diskettensymbol oben rechts abspeichern und eine Spalten-Stylesheet-Verknüpfung ist fertig. Diesen Schritt wiederholen Sie für jede Spalte, die Sie in dem JasperReport Bericht benötigen.

Ab Kernmodul 4.2 bzw. HISinOne 5.1 geht es noch komfortabler, Sie können hier auf einzelne Stylesheets filtern : Administration → Masken → Stylesheet suchen

Wählen Sie zunächst den Bericht aus, und klicken Sie dann rechts beim Bearbeiten-Button in der Spalte Felder-Vorbelegungen bearbeiten:

Bericht - Anzeige

Sie sind hier: [Startseite](#) > [Grunddaten und Basisberichte](#) > [Administration](#) > [Masken](#) > [Stylesheet suchen/Bericht erstellen](#) > [Datensätze/Stylesheet suchen](#)

Bericht entwerfen: Leerer Bericht

Stylesheet suchen


Legende
Dateiname (Stichwort): **gesch** ; User: superx Stand: 26.03.2013










Stylesheet Nr	Dateiname	Beschreibung	Bearbeiten	Felder-Vorbelegung bearbeiten
299	tabelle_16650_semester_geschlecht.jrxml	Semester nach Geschlecht		

Datensatz 1 - 1 von insgesamt 1 Satz.
Datenschutz beachten!!!

Maske Stylesheets und zugehörige Felder - Mozilla Firefox

localhost:8080/superx/edit/kern/stylesheet_field_edit.jsp;jsessionid=4A6BAE17A3EAFB51F98AE

Stylesheets und zugehörige Felder. 

Tid	111
Stylesheet	Semester nach Geschlecht (tabelle_16650_semester_geschlecht.jrxml)
Tablename	sos_stg_aggr
Fieldname	sos_stg_aggr.geschlecht

In dem Bearbeitungsformular können Sie Datensätze erzeugen, löschen etc.

3.3 Kurzanleitung Direktaufruf von Jasper-Berichten von der Maske

Nach dem Abschicken der Maske kann man auch direkt auf den entsprechenden Jasper-Bericht umgeleitet werden. Es muss dazu in der Maske ein Feld geben mit dem Namen

tablestylesheet
und der relation

```
<<SQL>> select distinct filename,X.caption from sx_stylesheetsX,  
sx_mask_style S where X.tid=S.stylesheet_id and S.maskeninfo_id=23880  
und zum Beispiel für Default der Maske 23880
```

```
<<SQL>> select distinct filename,X.caption from sx_stylesheets X,sx_mask_style S where  
X.tid=S.stylesheet_id and S.maskeninfo_id=23880 and S.ord=2
```

Für die Anzeige des Feldes tablestylesheet im Browser wird eine Beschriftung caption hinterlegt, damit auf der Maske "Bericht" erscheint. Im neuen Kern-Modul passiert dies automatisch.

Es ist möglich einen manuellen Eintrag in die Tabelle sx_captions einzufügen (siehe Maske Administration -> Maske -> Beschriftung suchen).

tid	table_name	field_name	locale	contents_short
<<wird automat. vergeben>>	felderinfo	tablestylesheet	de	Bericht

Außerdem muss man die Maske um ein Feld Ausgabeformat ergänzen, das die Art=1 hat und folgende relation:

```
select element_value,description from menu_element where element='Ausgabeformat' and  
nature::smallint<100 order by nature::smallint
```

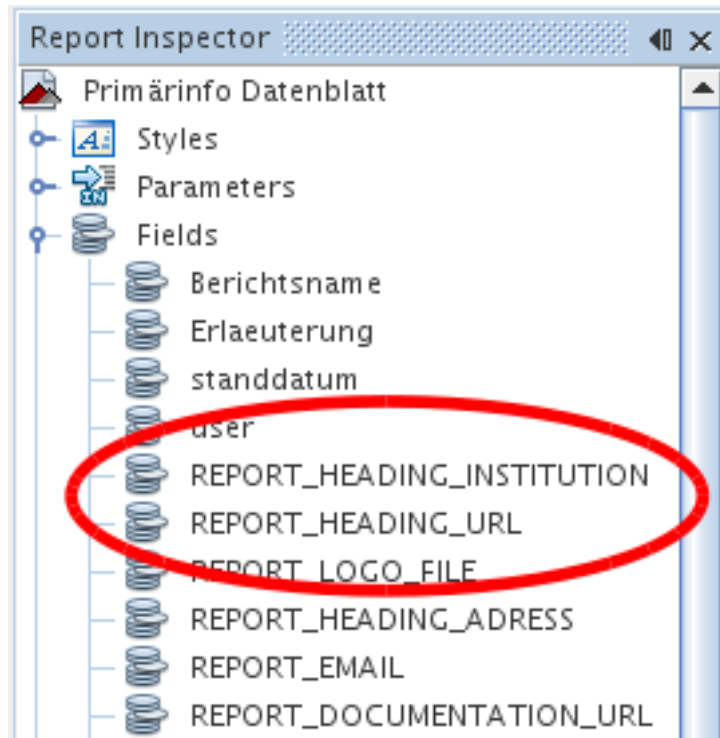
Beispiel für Default Excel:

```
<<SQL>> select element_value,description from menu_element where element='Ausgabeformat' and  
description='Excel'
```

Achtung: Hier kommen erst Ergebnisse, wenn Sie das Kernmodul 4.1 / HISinOne 3.0 oder höher installieren.

3.4 Spezielle Variablen

Im Berichtsdesign können neben den Feldern der Ergebnistabelle auch z.B. der Berichtsname, Erleuterung, user und standdatum benutzt werden. Zu finden sind diese im "Report Inspector" unter Fields. Um auch Hochschulspezifische Informationen wie Hochschulname oder Hochschul-Webadresse in einen Bericht importieren zu können wurden unter anderem die sx_captions Variablen "REPORT_HEADING_INSTITUTION" und "REPORT_HEADING_URL" integriert.



Diese können genau wie die anderen Fields in den Bericht per Drag&Drop gezogen werden.

3.5 Kompatibilität von iReport mit JasperReports

Wenn Sie mit iReport Berichte erzeugen, sollte wie gesagt die iReport-Versionsnr. mit der JasperReports-Version übereinstimmen, die der jew. Server nutzt. In Edustore ist dies derzeit die Version 5.1.0. Wenn die Versionsnummern nicht übereinstimmen, dürfen Sie keinesfalls die *.jasper-Dateien von dem lokalen Arbeitsplatz zum Tomcat Server kopieren.

Darüber hinaus kann es auch weitere Abhängigkeiten geben, iReport bringt ein paar mehr Bibliotheken mit, als JasperReports. Wenn Sie dann eine jrxml-Datei auf den Server kopieren, könnte es eine Fehlermeldung geben in dem Stil

```
java.lang.NoSuchMethodError:...
```

Das Vorgehen ist hier wie folgt:

- Prüfen Sie zunächst, ob Sie die jeweilige Bibliothek wirklich brauchen.
- Wenn ja, dann ermitteln Sie aus dem Namen der fehlenden Klasse den Namen der *.jar-Datei. Meist reicht es, wenn Sie den Klassennamen in einer Suchmaschine eingeben, oder in einer speziellen Java-Suchmaschine wie <http://www.jarfinder.com/>

- Wenn Sie die jar-Datei identifiziert haben, können Sie diese von iReport-xxx/ireport/modules/ext/ in das webapps/superx/WEB-INF/lib Verzeichns kopieren.
- Danach müssen Sie Tomcat neu starten.

Achtung: gehen Sie beim Kopieren in den WEB-INF/lib Ordner möglichst **sparsam** um: manchmal gibt es zwischen einzelnen Bibliotheken unerwünschte und unvorhersehbare Querabhängigkeiten, z.B. nutzt eine iReport-Bibliothek xy.jar eine andere Bibliothek vz.jar, welche bereits im Tomcat Server vorhanden ist – aber mit einer anderen Versionsnummer. In diesem Falle kann es zu seltsamen Phänomenen kommen, die sehr schwer zu diagnostizieren sind, bis hin zu einer Instabilität des Servers. Man muss solche Kopiermaßnahmen auf jeden Fall gründlich testen!

Unter uns: Wir nennen Probleme, die dadurch entstehen können, die "jar-Hölle"...

Im folgenden sammeln wir ein paar Beispiele für gefahrlose Kopieraktionen.

3.5.1.1 iReport Grafiken in JasperReports

Wenn Sie mit iReport Grafiken erzeugen, müssen Sie folgende jar-Dateien von iReport-xxx/ireport/modules/ext/ in das webapps/superx/WEB-INF/lib Verzeichns kopieren:

- jcommon-*.jar
- jfreechart-*.jar

Bitte vermeiden Sie, iReport-spezifische "Themes" zu benutzen (z.B. "aegean"), denn dafür würden weitere Bibliotheken benötigt. Wenn Sie z.B. die folgende Fehlermeldung bekommen, wurde ein Theme benutzt:

HTTP Status 500 -

type Exception report

message

description The server encountered an internal error () that prevented it from fulfilling this request.

exception

```
net.sf.jasperreports.engine.JRRuntimeException: Chart theme 'aegean' not found.
net.sf.jasperreports.charts.util.ChartUtil.getChartTheme(ChartUtil.java:165)
net.sf.jasperreports.engine.fill.JRFillChart.evaluateChart(JRFillChart.java:1279)
net.sf.jasperreports.engine.fill.JRFillChart.evaluateRenderer(JRFillChart.java:1254)
net.sf.jasperreports.engine.fill.JRFillChart.evaluate(JRFillChart.java:1243)
net.sf.jasperreports.engine.fill.JRFillElementContainer.evaluate(JRFillElementContainer.java:256)
net.sf.jasperreports.engine.fill.JRFillBand.evaluate(JRFillBand.java:482)
net.sf.jasperreports.engine.fill.JRVerticalFiller.fillSummaryNoLastFooterSamePage(JRVerticalFiller.java:1060)
net.sf.jasperreports.engine.fill.JRVerticalFiller.fillSummary(JRVerticalFiller.java:1025)
net.sf.jasperreports.engine.fill.JRVerticalFiller.fillReportEnd(JRVerticalFiller.java:283)
net.sf.jasperreports.engine.fill.JRVerticalFiller.fillReport(JRVerticalFiller.java:117)
net.sf.jasperreports.engine.fill.JRBaseFiller.fill(JRBaseFiller.java:923)
net.sf.jasperreports.engine.fill.JRFiller.fillReport(JRFiller.java:119)
net.sf.jasperreports.engine.JasperFillManager.fillReport(JasperFillManager.java:435)
net.sf.jasperreports.engine.JasperFillManager.fillReport(JasperFillManager.java:271)
de.superx.servlet.SuperXmlTabelle$JasperCreator.runExport(Unknown Source)
de.superx.servlet.SuperXmlTabelle$JasperCreator.perform(Unknown Source)
de.superx.servlet.SuperXmlTabelle$TableCreator.classicRun(Unknown Source)
de.superx.servlet.SuperXmlTabelle$TableCreator.perform(Unknown Source)
de.superx.servlet.SuperXServletHelper.run(Unknown Source)
de.superx.servlet.SuperXmlTabelle.doPost(Unknown Source)
javax.servlet.http.HttpServlet.service(HttpServlet.java:709)
javax.servlet.http.HttpServlet.service(HttpServlet.java:802)
```

note The full stack trace of the root cause is available in the Apache Tomcat/5.5.20 logs.

Apache Tomcat/5.5.20

Dann sollten Sie die Einstellung für das Objekt (hier das Theme der Grafik) wieder auf default setzen.

3.5.1.2 Weitere Bibliotheken von iReport

Bei der Meldung

```
java.lang.NoSuchMethodError: org.objectweb.asm.ClassWriter
```

sollten Sie die Bibliotheken

```
iReport-xxx/iReport/modules/ext/asm.jar
```

und zur Sicherheit auch

```
asm-attrs.jar
```

in den WEB-INF/lib-Ordner kopieren.

4 Troubleshooting

4.1 Report Problem: Class not set for bucket: <<Variable>>

Das Problem tritt in iReport 4.1.1 u.a. bei Kreuztabellen auf. Bei diesem Fehler fehlt in der jrxml-datei im Element <bucketExpression> Z.B.: class="java.lang.String". Diese einfach im Editor ergänzen und es funktioniert. Beispiel:

Aus :

```
<bucketExpression>
```

wird:

```
<bucketExpression class="java.lang.String">
```

4.2 Die jasper Datei

Die jasper Datei ist die kompilierte jrxml. Wenn ein Bericht übertragen werden soll, z.B. von dem Entwickler auf den Server oder von dem Development System auch das Produktivsystem immer nur die jrxml Datei übertragen, da es bei Versionsunterschieden in JasperReport zu Fehlern kommt, wenn die Kompilierte Version zur Verfügung steht. Sobald der Bericht ausgeführt wird, erstellt (kompilliert) JasperReports automatisch die jasper Datei.

Wenn die jasper Datei zur Verfügung steht, wird auch nur noch diese von JasperReports verwendet und nicht mehr neu kompiliert. Falls Sie also einen Bericht weiter entwickelt haben oder eine neue Version einspielen, löschen Sie direkt die jasper Datei. Ansonsten werden die Änderungen nicht wirksam.

5 Weitere Dokus

Ahammad, S. (2010). iReport 3.7. Learn how to use iReport to create, format, and export reports. Birmingham: Packt

Haneke, U. et al. (2010). Open Source Business Intelligence. Möglichkeiten, Chancen und Risiken quelloffener BI-Lösungen. Wien: Hanser

Toffoli, G. (2007). The definitive guide to iReport. Berkeley, CA: Apress

Noch mehr Infos (nur Englisch) erhält man unter:

http://jasperforge.org/plugins/project/project_home.php?projectname=ireport

und natürlich unsere Website:

http://www.superx-projekt.de/doku_devel/entwicklung/iReport-Handbuch/main.htm

und unser Forum:

<http://jasperforum.superx-projekt.de>