

Inhaltsverzeichnis	1
Berichtsdesign mit JasperReports	3
JasperReports installieren und konfigurieren	3
Voraussetzungen	3
JasperReports und Jaspersoft Studio	3
Jaspersoft Studio	3
Jaspersoft Studio Einstellungen	3
Jaspersoft Studio Entwurfsbereich	3
Jaspersoft Studio Sortierung und Filterung der Datenquelle	4
Jaspersoft Studio Berichtsdesign und Vorschau	4
Jaspersoft Studio Berichtsstruktur	4
Jaspersoft Studio Elemente bearbeiten	4
iReport	4
iReport Einstellungen	5
iReport Entwurfsbereich	5
Unterschiede iReport und JasperSoft Studio	5
Handhabung	5
Preview	5
Palette	5
Objekt-Eigenschaften	6
JasperReports als Kommandozeilen-Tool	6
Datenquellen	6
Datenquelle Datenbank	6
Datenquelle Excel-Datei	7
Berichtsdesign	7
Sortierung und Filtern	8
Listen	8
Gruppierung	8
Gruppen definieren	8
Seitenumbrüche	9
Gruppensummen	9
Gruppierung ohne eigenes Band	9
Gruppierung mit virtuellen Dimensionen	10
Variablen und Gruppierung	10
Berechnungen auf Gruppenebene	10
Tabellengestaltung	11
Summen, Seitenzahlen	11
Seitenumbrüche steuern	11
Zeilen alternierend einfärben	12
Rahmenlinien vorausschauend setzen	13
Unterberichte	13
Konfiguration des Hauptberichts	13
Unterbericht mit Header und Footer vom Hauptbericht	14
Konfiguration des Unterberichts	14
Einhängen des Berichts in Edustore	15
Kreuztabellen	16
Bedienung des Assistenten für Kreuztabellen	16
Nachbearbeitung des Kreuztabellen-Assistenten	16
Kreuztabellen im Gruppenfuß	17
Kreuztabellen mit virtuellen Dimensionen	17
Sortierung in Kreuztabellen	17
Parameter für Kreuztabellen	17
Manuelle Kreuztabelle	18
Layout von Kreuztabellen bearbeiten	18
Total einer 'Row Group' entfernen	18
Header von 'Row Groups' für jede Zeile wiederholen	18
'Static Field' und 'Text Field' frei gestalten/bewegen	19
Reihenfolge der 'Row Groups' ändern	19
Diagramme	19
Einfache Diagramme	19
Einfache Diagramme in JasperSoft Studio	19
Einfache Diagramme in iReport	20
Diagramme nach Tabelle ausgeben	21
Diagramme mit berechneten Werten	21
Chart Customizer	23
Balken Labels	23
Booklets	23
Hauptbericht im Booklet	24
Aufbau des Hauptberichts	24
Übergabe von Layouts vom Hauptbericht an Unterberichte	24
Booklet Deckblatt	24
Booklet Inhaltsverzeichnis	25
Booklet erster Detailbericht	25
Das fertige Booklet	26
Schriften	26
Allgemeines zu Schriften	26
Schriften in iReport einbinden	26
Schriften in JasperSoft Studio einbinden	27
Default-Schriftart in iReport	27
Fonts auf dem Server	27
Font Extension	28
TextField-Formatierung	28
Ausgabe von NULL als String verhindern	28
Einzelne Wörter unterstreichen oder fett drucken	29
Zahlen- und Datumsformat	29
Pattern Expression	29

Effiziente Formatierung	29
Conditional styles – Bedingte Formatierung	30
Beispiel in JasperSoft Studio – Schrift fett machen	30
Beispiel in iReport – Schrift fett machen	31
Textfeld vs. Conditional Style – Priorisierung	31
Default Style	32
Expression Editor	32
Bedingungen	32
Umgang mit Zeichenketten	32
Umgang mit Datumsfeldern	33
Datumswerte aggregieren	33
Zeichenketten zum Datumstyp konvertieren	33
Besonderheiten / Workarounds	33
Häufig benötigte Transformationen	33
Umrechnung Semester-tid in Namen	34
Umrechnung Semester in akad. Jahr	34
Pattern Expression	34
Fortgeschrittene Techniken bei Feldern	34
Einrücken von Feldern	35
Berechnungen	35
Prozentrechnung	35
Rundung	36
Hyperlinks	36
Externe Links	36
Interne Links/Anker	37
Berichts-Properties setzen	37
Seitenlayout	37
Seitenumbrüche	37
Die Höhe von Textfeldern dynamisch anpassen	38
Bericht ausgeben, ohne Dateninput	38
Besonderheiten beim Export nach Excel	38
Excel-Formeln	38
Zellennummern dynamisch berechnen (Ein Beispiel)	39
WENN-/IF-Formel – Ein Fallstrick	39
SUMIF – Summen mit Bedingungen (Bsp.: Zwischensummen ignorieren)	40
Statt 0-Werten eine leere Zelle ausgeben	40
OpenOffice vs Excel	40
„Stretch with Overflow“ in Excel korrekt verarbeiten	40
Pagination in Excel-Dokumenten	41
Excel-Export in einzelne Tabellenblätter	41
Einzelne Berichte	41
Haupt- und Unterberichte	42
Individuelle Gestaltung bei Booklets	42
Das Drucklayout in Excel konfigurieren	42
Seitenränder und Kopfzeile im Drucklayout formatieren	42
Vordefinierte Zeilen im Drucklayout auf jeder Seite wiederholen	42
Seitenumbrüche im Drucklayout verhindern	43
Zellen verbinden	43
Zellschutz aktivieren	43
Zellen fixieren	43
Hintergrund- oder Textfarben beim Excel-Export	44
Seitenkopf und Seitenfuß bei Excel unterdrücken	44
Dimension Tausend-Euro	44
Nutzung in Edustore	45
Berichtsentwicklung bei Grunddaten und Basisberichten	45
Allgemein	45
Datenquelle erzeugen	45
Datenquelle in JasperSoft Studio	45
Datenquelle in iReport	46
Template erzeugen	46
Template einbinden	47
Reportdesign verlinken	47
Nutzung des Upload Servlets	47
Probleme beim Deployment im Server	47
Feldnamen im Berichtsdesign weichen vom Datenbank-Feldnamen ab	48
Korrektur der Feldnamen im Berichtsdesign	48
Zuordnung von Default-Feldbelegungen zu Stylesheets	48
Kurzanleitung Direktaufruf von Jasper-Berichten von der Maske	49
Spezielle Variablen	49
Makroberichte als Datenquelle für JasperReports	50
Kompatibilität von iReport mit JasperReports	51
iReport Grafiken in JasperReports	51
Weitere Bibliotheken von iReport	51
Troubleshooting	52
Report Problem: Class not set for bucket: <<Variable>>	52
Die jasper Datei	52
Im Excel Export fehlen Felder	52
Hinweis für HisInOne 7.0 Kompatibilität	52
HTML Export keinen whitespace	53
UUID's	53
Weitere Dokus	53

Berichtsdesign mit JasperReports

Berichtsdesign mit JasperReports

von Daniel Quathamer & Andre Knieschewski

JasperReports installieren und konfigurieren

Voraussetzungen

- Jasperreports ist eine Laufzeitumgebung fürs Reporting, die in java-basierte Reporting Software integriert werden kann. Der Download ist hier: <https://sourceforge.net/projects/jasperreports/>. Dieses Paket enthält kommandozeilenbasierte Demos und die benötigten Libraries. Die Demos sind sehr nützlich und als Referenz gut nutzbar.
- Die Entwicklungsumgebung für JasperReports heißt "Jaspersoft Studio" und kann bezogen werden von <https://community.jaspersoft.com/project/jaspersoft-studio>. Es enthält auch obige Laufzeitumgebung.

Da es ein Eclipse Plugin ist, können Sie es auch direkt in Ihrem Eclipse installieren (Menü Help → Eclipse Marketplace → Suchwort "Jaspersoft"). Dort können Sie aber nur die jeweils aktuelle Version installieren.

Da der Download bei Jaspersoft nun nur für registrierte Nutzer zur Verfügung steht, hier noch ein Link für den Download: <https://sourceforge.net/projects/jasperstudio/files/>

- Für alte Projekte wurde auch iReport verwendet. 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 Apache ANT <http://ant.apache.org/>
- Java 1.8 JRE oder höher

JasperReports und Jaspersoft Studio

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

Jaspersoft Studio

Nach dem Download benötigen Sie Java 1.8 oder höher zum Ausführen. Nach dem Entpacken können Sie die Anwendung direkt starten.

Bei JasperSoft Studio wird als Grundlage Eclipse verwendet. Wenn man JasperSoft Studio herunter lädt, erhält man automatisch Eclipse mit dabei. JasperSoft Studio kann [hier](#) heruntergeladen werden.

Wenn lieber ein bereits vorhandenes Eclipse verwendet werden soll und Jaspersoft Studio dort als Plugin installiert werden soll, gibt es 2 Wege:

1. Unter Help → Eclipse Marketplace kann nach „Jaspersoft“ gesucht werden und das Plugin installiert werden

[JasperReports-Handbuch jaspersoftstudio plugin install1.png](#)

2. Man kann auch über die Marketplace Webseite von Eclipse das Plugin Jaspersoft Studio suchen und das Icon per Drag&Drop nach Eclipse ziehen. Dann wird das Plugin automatisch installiert. [Hier](#) die Webseite. (Bei mir hat das Drag&Drop leider nicht funktioniert. Das kann an Linux liegen. Unter Windows wird es sicherlich funktionieren.)

Jaspersoft Studio Einstellungen

Unter Windows - > Preferences → Jaspersoft Studio können Sie verschiedene Einstellungen vornehmen. Wichtig in der Menüpunkt "Compatibility": wählen Sie dort die JR-Version, die auch Ihre Ziel-Plattform enthält.

Im Dateimanager von JasperSoft Studio gibt es Eclipse-typisch ein "virtuelles" Dateisystem mit dem Standardnamen "MyReports", zumindest beim Speichern von Dateien. Dieses Dateisystem liegt physisch im Ordner <<Home-Verzeichnis>>/JaspersoftWorkspace/MyReports.

Beim Öffnen von Dateien können Sie direkt mit "Open File" aufs normale Dateisystem zugreifen.

Jaspersoft Studio Entwurfsbereich

Hier eine Übersicht des Entwurfsbereichs, in rot finden Sie Erläuterungen:

[JasperReports-Handbuch jaspersoft designer.png](#)

Die Unterfenster können Sie Eclipse-typisch beliebig anordnen. Oben ein Beispiel.

Jaspersoft Studio Sortierung und Filterung der Datenquelle

Wenn die Datenquelle definiert ist, kann /sollte der Bericht die Sortierung und Filterung der Daten festlegen. Dies ist etwas versteckt im Designer mittig oberhalb des Entwurfsbereichs:

[JasperReports-Handbuch jaspersoft design query.png](#)

Im "Dataset and Query Dialog" sind die Felder der Datenquelle aufgeführt. Unten finden Sie die Reiter "Sorting" und "Filter Expression". Dort pflegen Sie beides.

Jaspersoft Studio Berichtsdesign und Vorschau

Im Designmodus können Sie den Bericht entwerfen, oder den XML-Quellcode. Beim Klick auf den Reiter "Preview" bekommen Sie eine Vorschau des Berichts:

[JasperReports-Handbuch jaspersoft design preview.png](#)

Oberhalb der Vorschau können Sie die Datenquelle (d.h. den "Data Adapter") angeben, und rechts daneben die Anwendung, die den Bericht zur Anzeige bringt ("Viewer"). Voreingestellt ist hier der Java-interne Viewer von JasperReports, Sie können den aber umstellen z.B: auf PDF oder Excel.

Mit Klick auf das grüne Dreieck führen Sie den Bericht aus und erhalten die Vorschau.

Achtung: Es ist aufgefallen, dass beispielsweise Fettdruck in der PDF-Vorschau nicht angezeigt wird. Im Java-Preview und aus der Weboberfläche geladen, ist der Fettdruck ersichtlich.

Jaspersoft Studio Berichtsstruktur

Im "Outline View" sehen Sie die Struktur des Berichts, d.h. aktive "Bänder" und Metadaten:

[JasperReports-Handbuch jaspersoft design outline.png](#)

Die Elemente unter "Fields" werden bei Nutzung in [Edustore](#) automatisch gefüllt. Alles weitere definieren Sie.

Jaspersoft Studio Elemente bearbeiten

Wenn Sie im Design Modus ein Element markieren, wird es automatisch im Outline und in dem Reiter "Properties" selektiert:

[JasperReports-Handbuch jaspersoft design properties.png](#)

Im Outline zeigt es lediglich, wo es sich befindet, und im Reiter "Properties" können Sie Formatierung, Anzeige etc. bearbeiten.

iReport

[JasperReports-Handbuch_welcome.png](#) Das Paket enthält bereits die Laufzeitumgebung JasperReports, diese muss also nicht separat installiert werden. Nichtsdestotrotz empfiehlt sich der Download, denn in dem JasperReports-F

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-<<Versionsnr.>>/ireport/libs kopieren. Danach müssen Sie iReport neu starten.

Start unter Windows: iReport-<<Versionsnr.>>/bin/ireport.exe

Start unter Linux: iReport-<<Versionsnr.>>/bin/ireport

iReport 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:

[JasperReports-Handbuch ireport viewers.png](#)

iReport 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:

[JasperReports-Handbuch designer ueberblick.png](#)

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.

Unterschiede iReport und JasperSoft Studio

Mit der Version 5.5 wurde die Entwicklung von iReport eingestellt, die Firma JasperSoft empfiehlt seit dem die Benutzung von JasperSoft Studio. Nach unseren Erfahrungen ist die Arbeit mit iReport 4.1.3 (die letzte stabile Version) hier und da "leichtgewichtiger" und einfacher, ganz zu schweigen vom Overhead, den JasperSoft Studio mit Eclipse einbringt. Besonders nervig empfanden wir es, dass JasperSoft Studio anders als iReport verschiedene Objekteigenschaften in Reitern organisiert, und bei jedem Wechsel der Markierung eines Berichtselements muss man in JasperSoft Studio wieder zum richtigen Reiter wechseln.

Andererseits ist JasperSoft Studio kompatibel mit neueren JR-Möglichkeiten. Insbesondere bei dem Umgang mit Diagrammen gibt es in JasperSoft Studio erheblich mehr Möglichkeiten, z.B. Chart Customization. Außerdem ist es vorteilhaft, wenn Sie ohnehin Eclipse nutzen. Sie sollten für Ihre Arbeit also abwägen ob Sie die neueren Möglichkeiten im jeweiligen Bericht überhaupt brauchen, ggf. sind Sie mit iReport besser bedient. Aber das ist Geschmackssache.

Beachten Sie aber bitte: sobald Sie einen Bericht mit JasperSoft bearbeitet und gespeichert haben, ist er mit iReport 4.1.3 nicht mehr bearbeitbar, bzw. nur wenn Sie die [UUIDs](#) entfernt haben.

Beide Produkte befinden sich noch im Einsatz, daher hier nur ein paar Kapitel zur Handhabung der jew. Software.

Handhabung

Preview

Bei iReport waren die Buttons für Designer und Preview immer oben.

[JasperReports-Handbuch jss-ir preview1.png](#)

Bei Jaspersoft Studio liegen die Buttons unten:

[JasperReports-Handbuch jss-ir preview2.png](#)

Palette

Die Palette ist bei iReport und Jaspersoft Studio an gleicher Position zu finden.

iReport:

[JasperReports-Handbuch jss-ir palette1.png](#)

Jaspersoft Studio:

[JasperReports-Handbuch jss-ir palette2.png](#)

Objekt-Eigenschaften

Die Objekt-Eigenschaften liegen auch an der gleichen Stelle, jedoch sind bei Jaspersoft Studio die Gruppierungen der Eigenschaften in Karteireiter unterteilt. Bei iReport ist es einfach eine lange Liste mit Unterteilungen.

iReport:

[JasperReports-Handbuch jss-ir properties2.png](#)

Jaspersoft Studio:

[JasperReports-Handbuch jss-ir properties1.png](#)

JasperReports als Kommandozeilen-Tool

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

Beispiel xmldatasource:

```
cd demo/samples
cd xmldatasource
```

Die Ausgabe ist:

[JasperReports-Handbuch ant xmldatasource.png](#)

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).

Datenquellen

Allgemein unterstützt JasperReports verschiedene Datenquellen, z.B. relationale Datenbanken, Excel, CSV oder XML. Auch Webservices sind möglich. Speziell für SuperX/ HISinOne-BI gibt es spezielle Datenquellen-Vorgaben, damit Sie die Daten auch im Server-Betrieb nutzen können. Hier also nur die allgemeinen, also SuperX/BI-unabhängigen Quellen, wenn Sie JasperReports am lokalen Arbeitsplatz nutzen.

Datenquelle Datenbank

Eine relationale Datenbank ist in JasperSoft Studio als "Data Adapter" einzurichten. Oben links befindet sich der Dialog zum Erzeugen (1) des Data Adapter, und dann darunter die Zeile "Database Jdbc Connection" (2):

[JasperReports-Handbuch data adapter1.png](#)

Dort geben Sie dann die jdbc-spezifischen Daten an (Datenbanktyp, Name etc.).

Achtung: die Nutzung setzt eine direkte Datenbankverbindung zum DB-Server voraus, an vielen Unternehmenanwendungen wird dies nicht mehr erlaubt. Ggf. fragen Sie bei Ihren Systemadministrator/innen nach.

Datenquelle Excel-Datei

Eine Excel Datei ist in JasperSoft Studio als "Data Adapter" einzurichten. Hier ein Beispiel:

[JasperReports-Handbuch datenquelle excel 1.png](#)

Die Excel Datei hat vier Spalten und eine führende Zeile mit den Spaltenüberschriften. Achten Sie darauf, dass die Datentypen der Spalten in Excel dem Inhalt entsprechen. Hier die Spalten:

- Frage Nr.
- Text
- Mit_ohne_schule
- Summe

Aus der Excel-Datei erzeugen Sie einen "Data Adapter". Der Data Adapter vom Typ "Excel File" hat ein paar Eigenschaften:

[JasperReports-Handbuch datenquelle excel 2.png](#)

Danach können Sie in einem Berichtsdesign die Datenquelle definieren:

[JasperReports-Handbuch datenquelle excel 5.png](#)

Sie wählen dort oben links zunächst den "Data Adapter" aus (1), und klicken dann rechts auf "Read Fields" (2):

[JasperReports-Handbuch datenquelle excel 3.png](#)

Aus der Datenquelle werden dann die Datenfelder (3) und Inhalte (Reiter "Data preview") automatisch übernommen.

Berichtsdesign

Das Erstellen eines Berichtes erleichtert JasperReports 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.

JasperReports 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. Es wird also eine Schleife über die Daten (Tabelle) ausgeführt und für jeden Datensatz ein mal das Detail Band ausgegeben. Die linke Spalte könnte z.B so aussehen:

[JasperReports-Handbuch iReport-iReport Leiste Neu.jpg](#)

Hier sieht man, das 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.

[JasperReports-Handbuch iReport-Feld Name.jpg](#)

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:

[JasperReports-Handbuch iReport-Beispiel 1 Namen.jpg](#)

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

[JasperReports-Handbuch iReport-Designer Preview.jpg](#)

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

[JasperReports-Handbuch iReport-Beispiel 1 Namen Ergebnis.jpg](#)

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 „Title“ wird nur einmal auf der ersten Seite abgebildet, Band „Page Header/Footer“ auf jeder Seite am Anfang/Ende.

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:

[JasperReports-Handbuch ireport sort1.png](#)

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

[JasperReports-Handbuch ireport sort2.png](#)

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

[JasperReports-Handbuch ireport sort3.png](#)

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

[JasperReports-Handbuch ireport filter1.png](#)

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](#).

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:

[JasperReports-Handbuch listen stretch.png](#)

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](#).

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 JasperReports im [Berichtsentwurf](#).

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.

[JasperReports-Handbuch designer add group.png](#)

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:

[JasperReports-Handbuch iReport-Beispiel Monat Namen.jpg](#)

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

[JasperReports-Handbuch iReport-Beispiel Monat Namen Ergebnis.jpg](#)

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:

[JasperReports-Handbuch designer add group new page.png](#)

Mit "Start on a new page" wird eine neue Seite bei jedem Gruppenwechsel begonnen. Speziell beim Export nach Excel kann dies in Tandem mit Arbeitsblättern nutzen, siehe [unten](#).

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":

[JasperReports-Handbuch designer add group foot1.png](#)

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

[JasperReports-Handbuch designer add group foot2.png](#)

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:

[JasperReports-Handbuch gruppierung spalten1 vorlage.png](#)

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.

[JasperReports-Handbuch gruppierung spalten3 band.png](#)

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

[JasperReports-Handbuch gruppierung spalten2 variable 1fs.png](#)

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-Kurzform.

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:

[JasperReports-Handbuch gruppierung spalten4 band.png](#)

Und das Ergebnis sieht so aus:

[JasperReports-Handbuch gruppierung spalten5 band.png](#)

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

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.

Wichtig: Wenn die Gruppierungsvariable nicht im Summary Band genutzt wird, muss der resettype der Variable auf „None“ gesetzt werden.

Im Abschnitt zu [Kreuztabellen](#) wird z.B. gezeigt, wie die Fächer Mathe, Physik und Biologie in einen neuen Wert "Naturwissenschaften" transformiert werden.

Variablen und Gruppierung

Beim Anlegen von Gruppen-Bändern und beim Ziehen von Feldern in das jew. Gruppenband legt JR automatisch eine Variable an, mit der jew. Aggregationsfunktion. Diese Variable hat folgende Eigenschaften:

Expression: das Feld, aus dem die Variable gespeist wird

Calculation: Die Aggregatfunktion, z.B. "sum"

Reset Type: Die Gruppierungsebene

Wenn Sie die Variable in Textfeldern des Gruppenbandes nutzen, gibt sie jeweils den richtigen Wert an, egal ob Gruppenkopf oder Fuß.

Wenn Sie aber mit "Print-when-Expressions" im Gruppenband arbeiten und auf die Variablen zugreifen, funktionieren diese standardmäßig nur im Gruppenfuß, weil erst am Ende der Gruppe der Wert bekannt ist. In diesem Fall müssen Sie der Variable auch eine "Initial Value Expression" mit dem jew. Feld geben.

Berechnungen auf Gruppenebene

Wenn Sie auf der Gruppenebene Berechnungen, z.B. Mittelwerte, durchführen wollen, gibt es einige Fallstricke:

- Die in JR eingebauten Funktionen wie "avg" arbeiten auf Gruppenebene nicht immer exakt.
- Komplexere Formeln auf Gruppenebene in Variablen abzubilden und diese dann als Quelle für Textfelder zu nehmen hat bei uns nicht geklappt.

Hier unser Vorgehen:

- Definieren Sie auf der Ebene der jew. Gruppe eine Variable mit der Gesamtsumme, und eine mit der Summe der zu mitteln Werte. Diese Variablen haben als "reset Type" die jew. Gruppierungsebene.
- Im Textfeld, das Sie ausgeben, verwenden Sie dann die Berechnungsformel, z.B. einen Quotienten. Verwenden Sie für die Berechnung nicht eine neue Variable, sondern fügen Sie die Formel direkt in das Textfeld ein.

Im folgenden ein Beispiel: Wir haben eine Absolventenliste pro Studiengang und Semester, darin werden dann Mittelwerte von Abschlussnoten gebildet. Die Datenquelle hat summierte Zeilen pro Note:

[JasperReports-Handbuch notenschnitt daten.png](#)

Im obigen Beispiel haben wir in der Zeile 5 eine Absolventin mit der Note 1,1. In der letzten Zeile haben wir zwei Absolventinnen mit der Note 1,2.

Würden wir den Notenschnitt manuell berechnen, müßten wir die Notensumme (Spalte H*K) ermitteln, hier wäre der Wert 32,3. Die Summe der Absolventen in Spalte K wäre 16, d.h. der Notenschnitt wäre $32,3/16 = 2,02$.

In JasperReports wäre der Entwurf wie folgt:

[JasperReports-Handbuch notenschnitt design1.png](#)

Wir haben eine Variable "Summe_1" auf der Ebene des Semester-Bandes, die im obigen Excel der Summe der Spalte "Summe" entspricht.

Außerdem haben wir eine Variable "Notensumme" mit der Formel $\$F\{Note\} * \$F\{Summe\}$, auf der Ebene der Gruppe "Semester". Achten Sie drauf, dass Sie als Typ "java.lang.Double" nehmen, damit Sie keine Rundungsfehler bekommen.

Der Quotient wird dann im Textfeld gebildet, das den Notenschnitt auf Semesterebene enthält, und auf zwei Nachkommastellen gerundet:

[JasperReports-Handbuch notenschnitt design2.png](#)

Das Textfeld selbst muss bei "Evaluation time" den Wert "Group" haben, und bei "Evaluation group" dann den Wert der Gruppe "Semester". Und hier das Ergebnis in der Vorschau:

[JasperReports-Handbuch notenschnitt vorschau.png](#)

Der Wert 2,02 stimmt mit unserer manuellen Berechnung überein.

Tabellengestaltung

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.

[JasperReports-Handbuch iReport-Page Number.jpg](#)

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“.

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:

[JasperReports-Handbuch iReport-Beispiel 2 Monat Namen.jpg](#)

Beispiel im Preview:

[JasperReports-Handbuch iReport-Beispiel 2 1 Monat Namen Ergebnis.jpg](#)

[JasperReports-Handbuch iReport-Beispiel 2 2 Monat Namen Ergebnis.jpg](#)

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

[JasperReports-Handbuch iReport-Even Odd 2.jpg](#)

und passt es der Größe des Bandes an.

[JasperReports-Handbuch iReport-Even Odd 1.jpg](#)

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 „==“.

[JasperReports-Handbuch iReport-Even Odd 3.jpg](#)

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

[JasperReports-Handbuch iReport-Even Odd 4.jpg](#)

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:

[JasperReports-Handbuch iReport-Beispiel 3 Uma.jpg](#)

Und im Preview:

[JasperReports-Handbuch iReport-Beispiel 3 Uma Ergebnis 1.jpg](#)

[JasperReports-Handbuch iReport-Beispiel 3 Uma Ergebnis 2.jpg](#)

Rahmenlinien vorausschauend setzen

Sowohl bei Text- als auch bei Static-Fields findet sich ein eigener Reiter zum setzen von Rahmeneigenschaften. Dort wird der Abstand des Inhaltes zum Rahmen des jeweiligen Feldes definiert, sowie Rahmenlinien gesetzt.

Beim Setzen von Rahmenlinien wird empfohlen, dass bei nebeneinanderliegenden Feldern, welche durch eine Rahmenlinie getrennt werden, jeweils in beiden Feldern Rahmenlinien gesetzt werden und nicht nur in einem der beiden Felder. Denn beim Export in das PDF-Format ist ersichtlich ob nur eines der Felder oder beide Felder Rahmenlinien besitzen. Falls dies also nicht einheitlich umgesetzt wird, erscheinen Rahmenlinien an entsprechenden Stellen dünner beziehungsweise dicker (bei Excel-Dokumenten tritt dieser Effekt nicht auf):

[JasperReports-Handbuch borders2.png](#)

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 JasperReports Version genutzt werden wie im Server.

Im folgenden eine Anleitung zur Erstellung eines Unterberichts:

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:

Ggf.

[JasperReports-Handbuch subreport0 tool.png](#)

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

[JasperReports-Handbuch subreport1 wizard1.png](#)

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

[JasperReports-Handbuch subreport1 wizard2.png](#)

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

[JasperReports-Handbuch subreport1 wizard3.png](#)

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.

Bitte beachten Sie dass Sie beim Unterbericht nicht die *.jrxml-Datei verlinken, sondern die *.jasper-Datei, d.h. den kompilierten Report.

[JasperReports-Handbuch subreport1 wizard4.png](#)

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:

[JasperReports-Handbuch subreport2 params none.png](#)

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:

[JasperReports-Handbuch subreport10 parameters.png](#)

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 JasperReports testen, dann müssen Sie noch die [Parameter](#) XML_LOCALE und XML_NUMBER_PATTERN wie folgt übergeben:

[JasperReports-Handbuch subreport2 params number.png](#)

Unterbericht mit Header und Footer vom Hauptbericht

Wenn der Unterbericht in einem normalen Gruppen- oder Detail-Band liegt, wird der Header und Footer vom Hauptbericht automatisch mitbenutzt. In vielen Fällen möchte man aber den Unterbericht nur ein mal ausgeben und legt ihn daher in das Summary-Band. Hier wird Standardmäßig der Header und Footer nicht ausgegeben oder wiederholt. Möchte man aber z.B. die Seitenzahl die im Footer liegt weiterhin angezeigt bekommen, gibt es eine Eigenschaft vom Hauptbericht, die man nur aktivieren braucht. Diese Eigenschaft nennt sich: Summary with Page Header and Footer.

Also einfach auf den Hauptbericht auswählen und in Eigenschaften diesen Punkt aktivieren.

[JasperReports-Handbuch subreport header footer.png](#)

Konfiguration des Unterberichts

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

[JasperReports-Handbuch subreport3 open.png](#)

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:

[JasperReports-Handbuch subreport4 param.png](#)

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

[JasperReports-Handbuch subreport4 param2.png](#)

Weitere Änderungen sind nicht notwendig.

Wir ergänzen nun die "Report Query" [JasperReports-Handbuch report query.png](#) 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:

[JasperReports-Handbuch subreport4 query.png](#)

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

[JasperReports-Handbuch subreport5 group1.png](#)

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

[JasperReports-Handbuch subreport5 group2.png](#)

Danach ist die Gruppierung als Band verfügbar:

[JasperReports-Handbuch subreport5 group3.png](#)

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

[JasperReports-Handbuch subreport5 group4.png](#)

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

[JasperReports-Handbuch subreport5 group5.png](#)

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

[JasperReports-Handbuch subreport5 group6.png](#)

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 "Koste

[JasperReports-Handbuch subreport15 align.png](#)

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

[JasperReports-Handbuch subreport20 erg.png](#)

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.
- Bei Nutzung von Versionskontrollsystemen haben Sie die *.jasper-Dateien vermutlich ignoriert (gitignore / svnignore). Sie müssen die Dateien daher mit "-f" adden / committen.

Vorsicht beim Einsatz von JasperSoft Studio: Die Anwendung erzeugt bei der Einzelbearbeitung eines Unterberichts nicht automatisch eine entsprechende .jasper-Datei beim Previewen des Hauptberichts (iReport machte das). Man muss also in Jaspersoft Studio im Bearbeitungsfenster oben immer "Compile Report" anklicken, um eine neue .jasper Datei zu erzeugen.

Kreuztabellen

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:

[JasperReports-Handbuch designer crosstab1.png](#)

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

[JasperReports-Handbuch designer crosstab2.png](#)

Als nächsten definieren wir die Zeilendimension des Berichts:

[JasperReports-Handbuch designer crosstab3.png](#)

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

[JasperReports-Handbuch designer crosstab4.png](#)

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

[JasperReports-Handbuch designer crosstab5.png](#)

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

[JasperReports-Handbuch designer crosstab6.png](#)

Im Entwurfsbereich erscheint ein neues Fenster mit der Kreuztabelle.

[JasperReports-Handbuch designer crosstab7.png](#)

Hier können wir nun Spaltenbreiten und Überschriften ändern:

[JasperReports-Handbuch designer crosstab9.png](#)

Das Ergebnis sieht wie folgt aus:

[JasperReports-Handbuch designer crosstab8.png](#)

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.

[JasperReports-Handbuch crosstab new row.png](#)

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.

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 auf "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!

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:

[JasperReports-Handbuch gruppierung virtuell.png](#)

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

[JasperReports-Handbuch gruppierung virtuell2.png](#)

Sortierung in Kreuztabellen

Zeilen- oder Spaltendimensionen werden in Kreuztabellen immer alphabetisch sortiert, egal wie die Sortierung in der Datenquelle ist. Das ist in der Regel kein Problem, aber wenn z.B. eine Zeilendimension anders sortiert werden soll, geht das nicht ohne weiteres. Hier ein Problembeispiel:

[JasperReports-Handbuch kreuztab sortierung falsch.png](#)

Das Balkendiagramm übernimmt die Sortierung nach dem Schlüssel der Gehaltsgruppe aus der Datenquelle. In der Kreuztabelle wird die Zeilendimension "Bezeichnung der Gehaltsgruppe" angezeigt, und diese sortiert alphabetisch, was an dieser Stelle aber unsinnig ist: Der Eintrag "Kein Einkommen" sollte ganz oben stehen.

Wir schauen uns zunächst die Definition der Kreuztabelle an:

[JasperReports-Handbuch kreuztab sortierung richtig1.png](#)

Die Zeilendimension sortiert aufsteigend nach Bezeichnungstext. Man kann auch keine alternative Expression als Sortierkriterium angeben. Wir müssen also einen Trick anwenden: wir nehmen den Sortierschlüssel mit in die "Bucket Expression" der Zeilendimension auf, und setzen ihn als Präfix davor:

[JasperReports-Handbuch kreuztab sortierung richtig2.png](#)

Danach ist die Sortierung richtig:

[JasperReports-Handbuch kreuztab sortierung richtig3.png](#)

Aber optisch ist das natürlich unschön, der Sortierschlüssel sollte am besten gar nicht angezeigt werden. Wir lösen das, indem wir den Anzeigetext wiederum so abwandeln, dass der erste Buchstabe nicht sichtbar ist. Java bietet dazu die Funktion "substring()":

[JasperReports-Handbuch kreuztab sortierung richtig4.png](#)

Und siehe da: die Sortierung und Anzeige ist perfekt:

[JasperReports-Handbuch kreuztab sortierung richtig5.png](#)

Dies ist nur ein Beispiel, wie man mit Hilfe von Java Expressions die Anzeige von Kreuztabellen variieren kann.

Parameter für Kreuztabellen

In Kreuztabellen hat man keinen direkten Zugriff auf die Variablen und Felder des Hauptberichts. Man kann die Werte aber als Parameter übergeben. Dazu gehen Sie wie folgt vor:

Sie machen den Parameter im Crosstab-Element bekannt, hier z.B. der Parameter "RSZ" für Regelstudienzeit):

[JasperReports-Handbuch crosstab param1.png](#)

Danach übergeben Sie den Parameter im Hauptbericht. Mit Rechtsklick auf das Cosstab Element können Sie eine Parameters-Map Expression hinterlegen, das ist einfach eine Zuweisung von einer oder mehreren Variablen zu jeweils einem Wert in der Groovy-Syntax mit "[Variable:Wert,...]". In dem Beispiel unten weisen wir dem Parameter "RSZ" den Wert des Feldes "RSZ" zu:

[JasperReports-Handbuch crosstab param2.png](#)

Danach können wir den Parameter in der Kreuztabelle nutzen, z.B. um in einer Spaltenüberschrift nur dann eine Zahl mit Soll-Kreditpunkten anzuzeigen, wenn die Fachsemester innerhalb der Regelstudienzeit liegen.

[JasperReports-Handbuch crosstab param3.png](#)

Und hier das Ergebnis für einen Studiengang mit 6 Semestern Regelstudienzeit:

[JasperReports-Handbuch crosstab param5.png](#)

Manuelle Kreuztabelle

Wenn die gewünschte Konfiguration/Einstellungen in der Kreuztabelle nicht möglich ist, kann eine Kreuztabelle mit Einschränkungen auch selber erstellt werden. Im Beispiel mit Semestern, muss vorher abgeschätzt werden wie viele Semester voraussichtlich in den Spalten vorkommen werden. Für jedes Semester (z.B. für 10 Semester) wird dann eine Variable angelegt. Die erste bekommt das höchste Semester (wie im Screenshot zu sehen).

[JasperReports-Handbuch kreuztabelle manuell 1.jpg](#)

Das zweite Semester wir dann das höchste welches aber unterhalb von dem ersten liegen soll, usw.

[JasperReports-Handbuch kreuztabelle manuell 2.jpg](#)

Diese Semester können nun als Spalten verwendet werden mit der Option „blank when null“.

Layout von Kreuztabellen bearbeiten

Dieser Abschnitt befasst sich mit Themen zur Layoutanpassung.

Total einer ‚Row Group‘ entfernen

Wenn eine neue ‚Row Group‘ angelegt wird, wird automatisch ein dazugehöriges Total-Feld erzeugt und mit eingefügt:

[642x367px](#)

Das ist in diesem Falle nicht erwünscht. Um dieses Feld zu entfernen, wird in den Properties der ‚Row Group‘ die ‚Total Position‘ = ‚none‘ gesetzt:

[642x263px](#)

Header von ‚Row Groups‘ für jede Zeile wiederholen

Im folgenden Screenshot ist eine Kreuztabelle dargestellt, bei welcher die Header der ‚Row Groups‘ nur einmal dargestellt werden:

[434x315px](#)

Zwar ließe sich das Feld noch, so dass es bis zum nächsten ‚Row Header‘ reicht, aber in diesem Beispiel ist erwünscht, dass sich der Header für jede Zeile wiederholt:

[530x420px](#)

Dazu wird die Funktion ‚Merge Header Cells‘ in den Properties der ‚Row Group‘ deaktiviert. Es kann sein, dass die Funktion einmal aktiviert und dann wieder deaktiviert werden muss, damit sie als deaktiviert erkannt wird.

[642x275px](#)

Damit der Text in der ersten Spalte nicht abgeschnitten wird, wurde ‚Stretch with Overflow‘ in den Text-Field-Properties ausgewählt. Dadurch ist eine Lücke bei den anderen Spalten entstanden.

[509x404px](#)

Um diese Lücke zu schließen wird der ‚Stretch Type‘ in allen folgend eingerahmten Textfeldern auf ‚Relative To Band Height (deprecated)‘ gesetzt:

[642x322px](#)

‚Static Field‘ und ‚Text Field‘ frei gestalten/bewegen

Im Default sind ‚Static Field‘ und ‚Text Field‘ in einer Kreuztabelle nicht frei positionierbar/gestaltbar. Um das zu ändern, wird im jeweiligen Bereich das Layout auf ‚Free Layout‘ eingestellt.

[642x279px](#)

Reihenfolge der ‚Row Groups‘ ändern

Die Reihenfolge der ‚Row Groups‘ muss nachträglich im ‚Outline‘-Bereich geändert werden. Dies erfolgt per Drag and Drop im rot markierten Bereich:

[642x437px](#)

Die Textfelder im Design-Bereich zu verschieben, reicht nicht aus, denn die ‚Row Group‘ selbst bleibt wo sie ist und dann würde nur das Textfeld in eine andere Row Group verschoben. Im folgenden Negativbeispiel wurden nachträglich im Design-Bereich die Textfelder Intervall und Datenherkunft vertauscht. Dadurch sind sie in der jeweils falschen ‚Row Group‘ gelandet:

[642x387px](#)

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 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.

Einfache Diagramme

Nehmen wir an wir haben eine Tabelle, die die Studierendenzahl mehrerer Semester nach Geschlecht aufführt. Hier ein Screenshot der Datengrundlage:

[JasperReports-Handbuch jaspersoft design query3.png](#)

Die Tabelle enthält die Merkmale Geschlecht, Semester und Summe. Daraus erzeugen wir in Jaspersoft Studio (und weiter unten in iReport) ein Balkendiagramm.

Einfache Diagramme in JasperSoft Studio

Zunächst definieren wir die Sortierung der Daten. Im "Dataset and Query Dialog" sind die Felder der Datenquelle aufgeführt. Unten finden Sie den Reiter "Sorting":

[JasperReports-Handbuch jaspersoft design query2.png](#)

Damit die Grafik richtig funktioniert müssen wir nach beiden Merkmalen (Semester und Geschlecht) sortieren. Das Semester sollte nicht alphabetisch nach Namen sortiert sein, sondern nach dem Zahlenschlüssel. Sonst ständen alle Wintersemester untereinander, und davor alle Sommersemester.

Dann ziehen wir aus der Palette das Chart Element in das Summary Band. Es startet automatisch ein "Chart Wizard". Hier wählen wir "Stacked Bar" als Grafiktypen:

[JasperReports-Handbuch jaspersoft design chart wizard1.png](#)

Nun müssen wir ihm sagen welches Feld die Y-Achse darstellt (hier "Summe"), und welches die X-Achse (hier "Semester der Belegung"). Das Merkmal zur Unterscheidung der Balkenfarben ("Series") ist das Geschlecht:

[JasperReports-Handbuch jaspersoft design chart wizard2.png](#)

Danach klicken wir auf "Finish", und die Grafik wird als Platzhalter im Summary Band eingefügt. Dort definieren wir noch die Breite der Grafik:

[JasperReports-Handbuch jaspersoft design chart1.png](#)

Damit ist der Entwurf fertiggestellt. So sieht die Grafik aus:

[JasperReports-Handbuch jaspersoft design chart wizard3.png](#)

Es ist etwas ungünstig dass die Labels der X-Achse nicht ganz lesbar sind. Wir lösen dies mit Rotation der Labels um 45 Grad:

[JasperReports-Handbuch jaspersoft design chart wizard4.png](#)

So sieht es besser aus:

[JasperReports-Handbuch jaspersoft design chart wizard5.png](#)

Das Beispiel lädt zur weiteren Verfeinerung an.

Einfache Diagramme in iReport

Hier die Ansicht in iReports "Refresh Preview Data":

[JasperReports-Handbuch designer chart2.png](#)

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

[JasperReports-Handbuch grafik chart in summary ziehen.png](#)

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

[JasperReports-Handbuch grafik chart auswaehlen.png](#)

Nach dem Klick auf "OK", geht es weiter mit den Inhalten der Grafik.

Als Dataset gibt es hier nur eine Auswahl.

[JasperReports-Handbuch grafik dataset auswaehlen.png](#)

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":

[JasperReports-Handbuch designer chart3.png](#)

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

[JasperReports-Handbuch designer chart4.png](#)

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:

[JasperReports-Handbuch designer chart5.png](#)

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.

[JasperReports-Handbuch designer chart6.png](#)

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)

Diagramme nach Tabelle ausgeben

Üblicherweise ähnelt ein Diagramm in JasperReports 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.

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:

[JasperReports-Handbuch grafik datenblatt ohne grafik.jpg](#)

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.

[JasperReports-Handbuch grafik how to add summary.png](#)

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

[JasperReports-Handbuch grafik chart in summary ziehen.png](#)

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

[JasperReports-Handbuch grafik chart auswaehlen.png](#)

Nach dem Klick auf "OK", geht es weiter mit den Inhalten der Grafik.

Als Dataset gibt es hier nur eine Auswahl.

[JasperReports-Handbuch grafik dataset auswaehlen.png](#)

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.

[JasperReports-Handbuch grafik series auswaehlen.png](#)

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.

[JasperReports-Handbuch grafik category auswaehlen1.png](#)

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

[JasperReports-Handbuch grafik category auswaehlen2.png](#)

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.

[JasperReports-Handbuch grafik category auswaehlen3.png](#)

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 JasperReports addiert werden können.

[JasperReports-Handbuch grafik value auswaehlen1.png](#)

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

[JasperReports-Handbuch grafik value auswaehlen2.png](#)

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.

[JasperReports-Handbuch grafik erstes ergebnis.png](#)

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.

[JasperReports-Handbuch grafik chart details.png](#)

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

[JasperReports-Handbuch grafik chart m std 1fs hinz.png](#)

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.

[JasperReports-Handbuch grafik chart m std 1fs hinz 1.png](#)

Nun erhalten wir als Ergebnis folgende Tabelle:

[JasperReports-Handbuch grafik zweites ergebnis.png](#)

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.

[JasperReports-Handbuch grafik zweites ergebnis1.png](#)

Chart Customizer

Mit JasperSoft Studio können die über sog. "Chart Customizer" Details bzgl. des Layouts vor Diagrammen steuern. Im folgenden ein Beispiel:

Wir haben ein einfaches Balkendiagramm, in dem wir eine (automatisch) ermittelte Y-Achse mit Werten von 0-15 haben:

[JasperReports-Handbuch jr chart customizer range1.png](#)

Nun wollen wir den Bereich auf 12 (in 2er-Schritten ändern). Man könnte dies ganz leicht über die Eigenschaften ändern:

[JasperReports-Handbuch jr chart customizer range2a.png](#)

Hier kann man aber nicht die Schrittweite ändern. Wir legen dazu einen Chart Customizer vom Typ "Value Axis Range and Tick" an:

[JasperReports-Handbuch jr chart customizer range2.png](#)

und damit sieht das Diagramm so aus:

[JasperReports-Handbuch jr chart customizer range3.png](#)

Die Y-Achse zählt nun in 2er Schritten von 0 bis 12.

Noch ein Hinweis: Bei Y-Achsen mit kleinen Wertebereichen kann es vorkommen, dass JR die Y-Achse automatisch mit Intervallen mit Nachkommastellen versieht, also z.B. 1 - 1,5 - 2 - 2,5 etc. Bei manchen Maßen macht die Nachkommastelle keinen Sinn (z.B. wenn Studierende gezählt werden). Um das zu ändern gehen Sie in der Eigenschaften der Y-Achse (Chart → Value axis) und setzen bei der Eigenschaft "Value axis tick label mask" den Wert "#" (Pattern für ganzzahlig) ein. Damit wird nur noch ganzzahlig hochgezählt.

Balken Labels

In Balkendiagrammen können die exakten Zahlen nicht genau ermittelt werden. Aus diesem Grund wird oft gewünscht, die Zahlen noch mal schriftlich darzustellen.

[JasperReports-Handbuch balken labels 1.png](#)

Um die Zahlen anzeigen zu lassen muss einfach in den Chart Properties unter Advanced „Common Chart Properties → Plot → Show Labels“ auf true gesetzt werden.

Booklets

Ab JR 6.2 oder höher (d.h. SuperX-Kernmodul 4.8 oder HISinOne-BI 2018.12) gibt es die Möglichkeit, Berichts-"Bücher" mit Deckblatt, Inhaltsverzeichnis und Schlußseite zu erstellen. Die Erstellung ist übersichtlicher als eine Kombination aus Haupt- und Unterbericht, und bietet beim Seitenlayout mehr Optionen, z.B. kann man in einem Booklet Hoch- und Querformat kombinieren.

Englische Anleitungen siehe

<https://community.jaspersoft.com/documentation/tibco-jaspersoft-studio-user-guide/v60/report-books>

Wg. der Version ist diese Möglichkeit in iReport nicht mehr verfügbar, Sie müssen JasperSoft Studio nutzen. Wenn Sie dort mit File -> New einen neuen Report erzeugen, können Sie bei Report type den Wert "Book report" wählen. Der Assistent fragt Sie dann, ob Sie Deckblatt, Inhaltsverzeichnis und die Schlußseite benötigen, kreuzen Sie die jew. Position an. Sie erhalten danach ein Entwurfsfenster mit dem Hauptbericht, der drei Abschnitte enthält: Das Deckblatt und Inhaltsverzeichnis ("Cover and table of contents"), den Buchinhalt ("Content") und die Schlußseite ("Backcover"):

[JasperReports-Handbuch booklet main1.png](#)

Wenn Sie die Datei speichern, werden die jew. Unter-Dateien automatisch mit angelegt, mit dem Zusatz "_cover.jrxml", "_toc.jrxml" und "_backcover.jrxml".

Hauptbericht im Booklet

Aufbau des Hauptberichts

Startpunkt beim Booklet ist der Hauptbericht. Dieser hat keine "Bänder" im Sinne einer Ergebnisausgabe, insbesondere gibt es kein Detailband. Trotzdem arbeitet der Hauptbericht mit einer Datenquelle im Sinne eines Detailbandes, hier heißt das Band "Content". Für Makroberichte in SuperX-/BI müssen wir für alle Unterberichte dieselbe XML-Quelle verwenden, d.h. wir müssen im Hauptbericht genau eine Zeile der XML-Quelle filtern. Wir machen das mit XPATH mit dem Ausdruck

```
/ergebnisse/ergebnis[  
  @ordnr=0]/ergebniselement/sql
```

Dies liefert die erste Zeile der ersten Tabelle des Makroberichts im Ergebnis.

Klicken Sie also oben auf die Datenquelle des Hauptberichts, und fügen Sie den Filter ein:

[JasperReports-Handbuch booklet main2.png](#)

Die Funktion „Read Fields“ funktioniert in diesem Fall nicht. Die Felder für den Hauptbericht werden manuell in den Sourcecode eingefügt. Dazu erstellen Sie zunächst, wie in [Abschnitt 4.5](#) beschrieben, ein JR-Template für den Makrobericht. Aus diesem Template kopieren Sie aus dem Sourcecode die Felder und fügen diese im Booklet ein. Fügen Sie alle Felder sowohl im Haupt- als auch in den Unterberichten des Booklets ein.

Im Content Band können Sie dann mit der rechten Maustaste die jew. Unterberichte verlinken. Die Technik ist identisch mit den normalen Haupt-/Unterberichten, nur dass das Seitenlayout nicht vom Hauptbericht zum Unterbericht vererbt wird.

Im Beispiel haben wir einen Unterbericht verlinkt:

[JasperReports-Handbuch booklet main3.png](#)

Im Feld "Component Expression" geben Sie den Unterbericht an. Im Feld "Tab name expression" können Sie den Namen des TABs in Excel definieren. Standardmäßig wird bei einem Booklet beim Export nach Excel also ein Arbeitsblatt mit mehreren Sheets.

Der Unterbericht wird mit den gleichen Parametern aufgerufen wie bei klassischen [Unterberichten](#). Der erste Unterbericht bekommt die ordnr 0, der zweite Unterbericht die ordnr 1 usw.:

[JasperReports-Handbuch booklet main4.png](#)

Übergabe von Layouts vom Hauptbericht an Unterberichte

Generell werden im Booklet keine Layoutvorgaben vom Hauptbericht an den jew. Unterbericht übergeben. Ausnahme: Die Property "net.sf.jasperreports.export.xls.detect.cell.type", die beim Excel Export als Zahl formatierte Zellen auch in Excel als Zahl ausgibt, wird im Hauptbericht gesetzt und dann an die Unterberichte übergeben.

Achtung: Je nach JasperSoft Studio Version kann man diese Property nicht in der graphischen Entwicklungsumgebung setzen, sondern nur im *.jrxml-Quellcode.

Booklet Deckblatt

Das Deckblatt ist ein normaler JasperReport mit Elementen im Titel-Band. Beachten Sie dass im Unterbericht auch Seitenränder und Seitenformate definiert sein müssen. Sie können auch vom Hauptbericht Parameter übergeben, hier z.B. das Semester:

[JasperReports-Handbuch booklet cover1.png](#)

Im Hauptbericht wiederum wird dieser Parameter dann übergeben, hier z.B. aus der Legende der Maske:

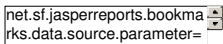
[JasperReports-Handbuch booklet cover2.png](#)

Booklet Inhaltsverzeichnis

Das Inhaltsverzeichnis ist ein Bericht ohne Datenquelle, und bekommt über die Evaluation Time="Report" die Anweisung, die Inhaltsdokumente nach Lesezeichen (HTML-Anchors) zu durchsuchen.

[JasperReports-Handbuch booklet toc1a.png](#)

Die Datenquelle ist leer, und über die Property



wird als Datenquelle der Content angegeben.

[JasperReports-Handbuch booklet toc1b.png](#)

Das Berichtsdesign ist vom Assistenten vorgegeben und relativ einfach. Sie können bis zu drei Ebenen für das Inhaltsverzeichnis angeben, jede Ebene benötigt ein eigenes Detailband.

[JasperReports-Handbuch booklet toc2.png](#)

Die Feldnamen im Bericht sollten nicht geändert werden. Sie können aber das Layout anpassen.

Booklet erster Detailbericht

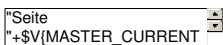
Der Detailbericht ist ein normaler Unterbericht, d.h. er hat den Parameter "ordnr":

[JasperReports-Handbuch booklet content1.png](#)

Die Variable "ordnr" wird dann bei der Datenquelle des Unterberichts ausgewertet:

[JasperReports-Handbuch booklet content2.png](#)

Der Unterbericht soll sicherlich auch Seitenzahlen anzeigen, dazu fügen Sie noch ein "Page footer" Band hinzu, und erzeugen dort ein Textfeld mit dem Inhalt z.B.



Das Textfeld muss bei Evaluation time den Wert "Master" haben.

[JasperReports-Handbuch booklet content3.png](#)

Danach wird die Seitenzahl jedes Unterberichts an den Hauptbericht angepaßt.

Für das Inhaltsverzeichnis müssen Sie im jew. Detailbericht Lesezeichen (HTML-Anchors) und jeweils eine Ebene angeben. Wir können z.B. für das Textfeld der Überschrift einen Anker der Ebene 1 festlegen:

[JasperReports-Handbuch booklet content4.png](#)

Danach legen wir für das Gruppenband für das Textfeld mit dem Label der Gruppe (Fakultätsname) einen Anker der Ebene 2:

[JasperReports-Handbuch booklet content6.png](#)

Das fertige Booklet

So sieht das Booklet im Preview im JasperSoft Studio aus: Zuerst das Deckblatt...

[JasperReports-Handbuch booklet preview1.png](#)

... und dann auf Seite 2 das Inhaltsverzeichnis:

[JasperReports-Handbuch booklet preview1b.png](#)

und dann auf Seite 3 der erste Unterbericht:

[JasperReports-Handbuch booklet preview2.png](#)

Schriften

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 JasperReports 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 Wahrscheinlichk

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

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:

[JasperReports-Handbuch font import step1.png](#)

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:

[JasperReports-Handbuch font import step2.png](#)

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:

[JasperReports-Handbuch font import.png](#)

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

[JasperReports-Handbuch font import1.png](#)

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

Schriften in JasperSoft Studio einbinden

Unter „Window“ → „Preferences“ können Sie Fonts einbinden:

[JasperReports-Handbuch font import4.png](#)

Nach einem Klick auf „Add“ öffnet sich ein neues Fenster. Dort geben Sie dem Font einen Namen und wählen die zugehörige Datei für den normale Ausführung des Fonts. Es gibt weitere Reiter für Bold, Italic und Bold Italic:

[JasperReports-Handbuch font import5.png](#)

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:

[JasperReports-Handbuch font import6.png](#)

Achtung: Beachten Sie lizenzrechtliche Fragen, die wenigsten Fonts sind ohne weiteres kopierbar. Wir empfehlen den Liberation-Font, der steht unter der OpenSource Lizenz GPL.

Default-Schriftart in iReport

Sie können die Default Schriftart in iReport einstellen:

[JasperReports-Handbuch ireport default font.png](#)

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

Fonts auf dem Server

Um Schriftarten auf einen Tomcat-Server zu installieren, müssen die vier ttf-Dateien (normal, fett, kursiv, fett-kursiv) in den Ordner \$JRE_HOME/lib/fonts kopiert werden. Für alle Java Anwendungen wie den Tomcat und JasperReports stehen dann die in dem verwendeten Java zur Verfügung.

Beispiel Arial unter Ubuntu 18.04:

Prüfen Sie mit dem Kommando

```
ps fax | grep tomc
```

welcher Tomcat läuft. Sie bekommen eine Ausgabe z.B.

```
47769 ? Sl 36:25 /usr/lib/jvm/java-8-openjdk-amd64/bin/java -Djava.util.logging.config.file=/var/lib/tomcat8/conf/logging.properties -
Djava.util.logging.manager=org.apache.juli.ClassLoaderLogManager -Djdk.tls.ephemeralDHKeySize=2048 -Djava.protocol.handler.pkgs=org.apache.catalina.webresources -Xmx3500M -
XX:MaxPermSize=400m -Djava.awt.headless=true -Dfile.encoding=UTF-8 -DSuperX-DB-PROPERTIES-SET=true -Djava.endorsed.dirs=/var/lib/tomcat8/endorsed -classpath
/var/lib/tomcat8/bin/bootstrap.jar:/var/lib/tomcat8/bin/tomcat-juli.jar -Dcatalina.base=/var/lib/tomcat8 -Dcatalina.home=/var/lib/tomcat8 -Djava.io.tmpdir=/var/lib/tomcat8/temp
org.apache.catalina.startup.Bootstrap start
```

Der erste Befehl zeigt den Pfad zu JRE_HOME, also /usr/lib/jvm/java-8-openjdk-amd64. Sie kopieren also die Dateien z.B.

arial.ttf

arialbd.ttf

arialbi.ttf

ariali.ttf

von

C:\Windows\winsxs\amd64-
microsoft-windows-font-

nach

/usr/lib/jvm/java-8-openjdk-
amd64/lib/fonts

und starten dann Tomcat neu.

Achtung: Sorgen Sie dafür dass Sie eine Lizenz für die jew. Fonts haben, weil diese bei PDF in die ausgelieferten Dateien eingebettet werden.

Font Extension

Eine Alternative zu dem Einbinden der Schriftart über das Java Verzeichnisse Font Extension. Diese hat aber nicht immer funktioniert. Um auf diesem Wege 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:

[JasperReports-Handbuch font import2.png](#)

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":

[JasperReports-Handbuch font import3.png](#)

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

TextField-Formatierung

In diesem Abschnitt werden verschiedene Einstellungen und Möglichkeiten der Formatierung in TextFields vorgestellt und erläutert.

Ausgabe von NULL als String verhindern

Falls NULL in einem TextField als String ausgeschrieben wird, kann dies verhindert werden, indem die Einstellung „Blank When NULL“ aktiviert wird. Das Feld bleibt dann leer:

[JasperReports-Handbuch text field formatting.png](#)

Einzelne Wörter unterstreichen oder fett drucken

Wenn in einem TextField einzelne Wörter unterstrichen oder fett gedruckt werden sollen, einfach in den „Text properties“ unter „Markup“ „styled“ auswählen. In dem TextField dann einfach um das zu unterstreichende Wort „<u> ... </u>“ schreiben oder „ ... “ um es fett gedruckt auszugeben.

Bitte beachten, dass die PDF-Preview aus Jaspersoft Studio heraus keinen Fettdruck darstellt.

Zahlen- und Datumsformat

Die Formatierung der Werte kann mit Hilfe des Attributes "Pattern" gesetzt

werden. Hier ein Beispiel:

[JasperReports-Handbuch textfield pattern1.png](#)

Das Textfield bekommt das Pattern "0", d.h. es wird eine Ganzzahl inkl. 0 angezeigt. Der Platzhalter "#" steht für Zahlenbestandteile, hier wird eine 0 allerdings nicht angezeigt. Weitere Zahlen- und Datumsformate sind im Assistenten (linker Teil des Screenshots) sicher selbsterklärend.

Pattern Expression

Hat man pro Spalte / Zelle unterschiedliche Werte z.B. Datumsangaben und Währungseinheiten

oder Ganzzahlen und Kommazahlen und möchte diese unterschiedlich formatieren, so kann dies

mit dem Attribut "Pattern Expression" dargestellt werden.

Hier ein Beispiel für die Ausgabe von Ganzzahlen und Kommazahlen:

Hier der Pattern Expression Editor:

[JasperReports-Handbuch ireport pattern expression editor.png](#)

Hier das Ergebnis:

[JasperReports-Handbuch ireport pattern expression ergebnis.png](#)

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.

[JasperReports-Handbuch copy format.png](#)

Für Hochschulen ist ein einheitliches Layout der Berichte wichtig, und so bietet es sich an, zentrale Layouts zu definieren und dann wiederzuverwenden. In JasperReports 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.

[JasperReports-Handbuch styles library.png](#)

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":

[JasperReports-Handbuch style def folie.png](#)

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:

[JasperReports-Handbuch styleref design folie.png](#)

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

Conditional styles – Bedingte Formatierung

Wenn man in einzelne Textfelder "bedingte Formatierung" nutzen will (z.B. farbiger Hintergrund, Fettdruck), hat man früher mehrere Textfelder einzeln angelegt, mit dem gleichen Inhalt und einer jeweiligen "Print-when expression". Das geht seit JR 3.x eleganter mit den sog. "Conditional styles".

Hierfür wird ein Style angelegt und diesem diverse „Conditional Styles“ untergeordnet. Die „Conditional Styles“ enthalten Bedingungen und Formatierungseinstellungen. Der Style kann beispielsweise einem Textfeld übergeben werden. Sobald die Bedingung eines „Conditional Styles“ erfüllt wird, werden die im „Conditional Style“ festgelegten Formatierungseinstellungen auf das Textfeld angewandt. Die Einstellungen im Textfeld selbst werden allerdings höher priorisiert. Eine detaillierte Erläuterung der Priorisierung anhand eines Beispiels findet sich in Abschnitt „Textfeld vs. Conditional Style – Priorisierung“

Beispiel in JasperSoft Studio – Schrift fett machen

Zunächst legen Sie die Formatierungen an, indem Sie unter Styles die rechte Maustaste drücken und über „Create Style“ einen Style erstellen, beispielsweise **Daten_angepasst**.

[JasperReports-Handbuch cond style3.png](#)

Durch Rechtsklick auf **Daten_angepasst** wird über „Create Conditional Style“ ein neuer Conditional Style hinzugefügt. Es erscheint ein neuer Unterknoten <No condition set>. Über die Properties können anschließend die zu erfüllende Bedingung als Expression, beispielsweise

```
#{Name}.equals("Summe  
Aktiva")||#{Name}.equals("...
```

festgelegt und die Formatierungseinstellungen vorgenommen werden. In diesem Falle ist Fettschrift (bold) gewählt :

[JasperReports-Handbuch cond style4.png](#)

Um den Style inklusive seiner Conditional Styles auf beispielsweise ein Textfeld anzuwenden, werden die Properties des entsprechenden Feldes geöffnet. Im Reiter Appearance wird der gewünscht Style, in diesem Falle **Daten_angepasst**, dem Textfeld zugeordnet:

[JasperReports-Handbuch cond style5.png](#)

Bei mehreren conditional styles wird die Reihenfolge ausgewertet, alle zutreffende conditional styles werden genommen. Die Reihenfolge wird direkt im XML oder per Drag&Drop geändert.

Es kam in der Vergangenheit bereits zu dem Phänomen, dass im Conditional Style Fettschrift gewählt war, aber nicht ins Textfeld übernommen wurde, während beispielsweise Kursivschrift problemlos und wie erwartet funktionierte. Eine genauere Beschreibung und Lösung findet sich im Abschnitt „Textfeld vs. Conditional Style – Priorisierung“

Beispiel in iReport – Schrift fett machen

Um z.B. einzelne Zeilen nach einer Bedingung fett zu machen oder nicht kann man so vorgehen: Zunächst legen Sie die Formatierungen an, indem Sie im Report Inspector unter Styles die rechte Maustaste drücken und ein Style hinzufügen, z.B. **zw_maybe_bold** für einen Style der Fett sein soll, wenn im Feld **zw** bestimmte Werte stehen.

[JasperReports-Handbuch styles.png](#)

Danach erzeugen Sie mit Rechtsklick auf **zw_maybe_bold** eine neue Bedingung mit „Hinzufügen von Conditional Style“. Bei dem neuen Unterknoten <No condition set> klicken Sie auf die rechte Maustaste, dann Edit Condition. Hier können Sie wie gewohnt eine Bedingung in Form einer Expression eingeben, z.B.

```
$F{Zweckbestimmung}.equals("Gesamtausgaben")||$F
```

Dann müssen Sie den neuen Conditional Style anklicken

[JasperReports-Handbuch cond style.png](#)

und in den Eigenschaften bold auf true setzen.

Dann wählt man im Report designer die Felder aus, die ggfs. fett dargestellt werden sollen, und bei den Eigenschaften als Style **zw_maybe_bold**.

[JasperReports-Handbuch cond style2.png](#)

Wenn man mit Padding arbeiten will, kann man für den Style padding angeben, muss aber anscheinend manuell im jrxml leftpadding entfernen

```
<box topPadding="0" leftPadding="0"
```

Bei mehreren conditional styles wird die Reihenfolge ausgewertet, alle zutreffende conditional styles werden genommen. Es gibt zwar einen Menüpunkt „Reihenfolge ändern“ bei rechtem Mausklick, der ist jedoch deaktiviert, man kann die Reihenfolge aber direkt im XML ändern oder per Drag&Drop.

Leider kann man nicht einen conditionalstyle auf ein leere Zelle anwenden (z.B. Unterstrich), wenn man Excelexport macht, wird daraus einfach eine leere Zelle ohne Formatierung selbst wenn man Leerzeichen in der Text Expression hat.

Werden Variablen im conditional style verwendet, muss die evaluation time aktiviert werden über die property:

```
<property name="net.sf.jasperreports.
```

Siehe <https://community.jaspersoft.com/blog/tip-data-set-variable-usage-conditional-style-table-component-jasper-reports>

Ein Style kann auch mit einem anderen Style verknüpft werden, indem in dem entsprechenden Style unter dem Reiter „Style“ in dem Feld „Style“ das entsprechende Style ausgewählt wird. Somit können bestimmte Wiederholungen gespart werden.

Textfeld vs. Conditional Style – Priorisierung

In diesem Abschnitt wird anhand eines Beispiels aufgezeigt, dass Textfelder höher priorisiert sind als Conditional Styles und welche Fallstricke dadurch entstehen können. Das Beispiel:

Einem Textfeld wird ein Style zugeordnet, der einen Conditional Style enthält, welcher bei erfüllter Bedingung den Text fett und kursiv darstellen soll:

[JasperReports-Handbuch cond style6.png](#)

Im Textfeld selbst sind beide Einstellungen deaktiviert:

[JasperReports-Handbuch cond style7.png](#)

Das Ergebnis ist, dass im Bericht der Text dieses Feldes zwar kursiv, aber nicht fett dargestellt wird. Zum Testen werden weitere Formatierungseinstellungen im Conditional Style hinzugefügt, beispielsweise Unter- und Durchstreichen und auch dies wird umgesetzt. Die Bedingung des Conditional Styles ist also erfüllt.

Der Grund für dieses Verhalten ist nur mit Blick auf den Source-Code ersichtlich:

[JasperReports-Handbuch cond style8.png](#)

Dort findet sich in den Einstellungen des Textfeldes `isBold="false"`. Dieses Fragment ist dadurch entstanden, dass in der Designoberfläche von JasperSoft Studio für das Textfeld die Einstellung **bold** einmal aktiviert und dann wieder deaktiviert wurde. Im Source-Code wird nach dem ersten Aktivieren `isBold="true"` hinzugefügt und beim Deaktivieren nicht wieder komplett entfernt, sondern auf `false` gesetzt. Da die Einstellungen im Textfeld selbst höher priorisiert werden, als jene der Conditional Styles, wird der Fettdruck des Conditional Styles nicht umgesetzt. Löscht man im Source-Code das Fragment `isBold="false"`, dann wird der im Conditional Style definierte Fettdruck umgesetzt.

Default Style

Es ist möglich einen Style als Default Style zu verwenden. Hierfür wird beim Style, nicht beim Conditional Style, die entsprechende Einstellung gesetzt:

[JasperReports-Handbuch cond style9.png](#)

Dieser Style wird nun auf jedes Feld angewandt, dem kein anderer Style zugeordnet ist.

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.

[JasperReports-Handbuch ireport filter1.png](#)

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_val

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.

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

Bedingungen:

String1 == String2
String1 != String2

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.

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/7/docs/api/java/lang/String.html>

Umgang mit Datumsfeldern

Datumswerte aggregieren

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":

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

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.

wenn man Text "Erzeugungsdatum" und das Datum in einem Feld haben will geht so:

```
"Erzeugungsdatum:"+new SimpleDateFormat("dd.MM.YYYY", Locale.GERMAN).format(new java.util.Date())
```

Zeichenketten zum Datumstyp konvertieren

Wenn Sie ein Datum als String-Feld geliefert bekommen, können Sie es mit Groovy zum Typen "java.util.Date" konvertieren, z.B. um danach zu sortieren. Wenn z.B. das Feld „Beginn“ den Typ String hat und gefüllt ist mit dem Wert "18.11.2020 18:05:00", dann können Sie eine Variable vom Typ `java.util.Date` erzeugen, mit der Expression:

```
Date.parse("dd.MM.yyyy  
hh:mm:ss", $F{Beginn})
```

Besonderheiten / Workarounds

- Bei Feldern mit Länge 1 klappte einfaches equals nicht, mit einer substring Anweisung geht es dann:

```
$F{legende_Art des  
Berichts_value}.substring(0,
```

- Folgejahr berechnen: Selbst es das Feld Jahr gibt und es als Integer definiert ist in der Felderinfo, klappt der Bericht nur in JasperReports nicht in SuperX nicht wenn man schreibt:

```
$F{legende_Jahr_value}.intValue()+1
```

geht nicht, statt dessen:

```
new  
Integer($F{legende_Jahr_v
```

- 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
- Zeilenbruch in Feld

Markup auf html setzen und dann z.B.

```
"Plan<br>"+$F{legende_Ja  
hr_value}
```

Häufig benötigte Transformationen

Umrechnung Semester-tid in Namen

```
$V(Semester  
(Schlüssel)).toString().ends
```

Umrechnung Semester in akad. Jahr

Die folgende Formel formt die Semesterzahl aus dem Studierenden-Datenblatt in ein akad. Jahr um nach der Formel "Akad. Jahr=WS + darauf folgendes SS", z.B. WS 2009/2010 und SS 2010 werden zum akad. Jahr 2009:

```
$F(Semester der Belegung  
(Schlüssel)).toString().ends
```

Die Formel nimmt beim WS die ersten vier Stellen des Semester-Schlüssels, und beim SS werden die ersten vier Stellen als Ganzzahl codiert, und dann wird 1 subtrahiert.

Da es diverse Definitionen von Akad. Jahr / Studienjahr / Prüfungsjahr gibt, kann man das obige Beispiel leicht für eigene Bedürfnisse abwandeln.

Pattern Expression

Die Formatierung der Werte kann mit Hilfe des Attributes "Pattern" gesetzt

werden. Hat man allerdings unterschiedliche Werte z.B. Datumsangaben und Währungseinheiten

oder Ganzzahlen und Kommazahlen und möchte diese unterschiedlich formatieren, so kann dies

mit dem Attribut "Pattern Expression" dargestellt werden.

Hier ein Beispiel für die Ausgabe von Ganzzahlen und Kommazahlen:

```
$V(valueMeasure).doubleValue()  
alue() ==
```

Hier der Pattern Expression Editor:

[JasperReports-Handbuch ireport pattern expression editor.png](#)

Hier das Ergebnis:

[JasperReports-Handbuch ireport pattern expression ergebnis.png](#)

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 CLASSPATH 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:

[JasperReports-Handbuch expression stringbuilder.png](#)

Im Ergebnis sieht das so aus:

[JasperReports-Handbuch expression stringbuilder1.png](#)

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.

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	xxxxxxxxx

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

```
$(Einruecken).replace('x',' ')+$(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:

```
||| |
| $(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.

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.

Prozentrechnung

Für die Prozentrechnung muss 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

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

Bei Prozentberechnungen in Grafiken muss in manchen Fällen der Wert auf 0 überprüft werden. Das kann man z.B. mit folgendem Code machen:

`($V{zahl1} == 0 ||
$V{zahl2} == 0 ? 0 : (`

Damit ist dann sichergestellt, dass es keine Probleme bei der Berechnung gibt, auch wenn die Zahlen mal leer sind.

Rundung

Wenn man runden möchte den Datentyp am besten auf BigDecimal stellen. Ansonsten wird schon mal aus 2,45 gerundet 2,4.

Hyperlinks

Externe Links

Ein Textfeld kann auch als Hyperlink befüllt und ausgegeben werden. Vorab bemerkt: Hyperlinks funktionieren in internen Viewer von JasperSoft Studio sowie beim Export nach Excel nicht bzw. nur eingeschränkt, zumindestens wenn man POI als Excel Engine nutzt. Selbst wenn Hyperlinks ausgegeben werden, ist in den Sicherheitseinstellungen von Excel oder OpenOffice eine Änderung vorzunehmen, damit Hyperlinks auch aktiv sind. Da es hier komplexe Unterschiede bei MS Excel-/OpenOffice-Versionen gibt, haben wir das Thema nicht weiter verfolgt und raten davon ab.

In den anderen üblichen Ausgabeformaten (PDF, HTML) funktionieren die Hyperlinks.

Dabei gibt es zwei Varianten:

Das Textfeld hat als Markup Typ den Wert "html", und die Hyperlinks werden in HTML-Notation eingefügt. Beispiel:

[JasperReports-Handbuch hyperlink pfd3.png](#)

Hier der Quellcode der textField-Expression zum Kopieren:

`$V{Gross_Forsch_Proj_Spr
ech).replaceAll("n","<br`

Im Ergebnis sieht das dann so aus:

[JasperReports-Handbuch hyperlink pfd2.png](#)

Bei Variante 2 erhält ein ReportElement im Quellcode die entsprechende Eigenschaft. In der von uns genutzten Version von JasperSoft Studio (6.5.1 final) war das noch nicht im GUI bearbeitbar, daher kann man das nur im Quelltext machen. Immerhin "zerstört" JasperSoft Studio dies beim Speichern nicht.

Die Eigenschaft des textField heißt hyperlinkType="Reference" und wird wie folgt belegt:

`<textField pattern=""
isBlankWhenNull="true"`

Am Ende des textField wird dann im Element hyperlinkReferenceExpression das Ziel des Hyperlinks definiert. Da dies eine Expression ist, kann man die Hyperlinks auch dynamisch aufbauen. In der Ausgabe dies das dann so aus:

[JasperReports-Handbuch hyperlink pdf.png](#)

Je nach PDF Viewer muss man auch hier in den Sicherheitseinstellungen Hyperlinks aktivieren.

Interne Links/Anker

Auf einem Text Field kann ein Link eingetragen werden. Dafür muss man einfach mit der rechten Maustaste auf das Textfeld klicken und dann auf Hyperlink klicken.

Das erste Feld oben „Anchor Name Expression“ kann einen Text enthalten, auf den von einer anderen Stelle verwiesen werden kann. Es ist also eine Markierung.

Wenn ich einen Hyperlink Type wähle, kann ich auch Verweise in den generierten Bericht erzeugen. Z.B. kann ich hier den Type „LocalAnchor“ wählen.

Um z.B. ein Inhaltsverzeichnis zu erzeugen, welches im PDF bei Klick auf ein Element dort hin springen soll, gehe ich zunächst zu der Stelle an der die Sprungmarke gesetzt werden soll und erzeuge dort ein Text Field mit der „Anchor Name Expression“ z.B. „Überschrift1“. Das Inhaltsverzeichnis muss dann auch aus Text Fields erzeugt werden und dort wähle ich dann bei Hyperlink type „LocalAnchor“. Nun habe ich unter der Auswahl von dem Hyperlink type ein neues Eingabefeld „Hyperlink Anchor Expression“. Dort trage ich jetzt exakt den gleichen Text „Überschrift1“ ein und im generierten PDF kann ich dort drauf klicken um an die Markierte Stelle zu kommen.

Berichts-Properties setzen

Es gibt auf Berichtsebene die Möglichkeit, die Ausgabe der Berichte über sog. "Properties" zu steuern, teilweise auch formatspezifisch. So können Sie z.B. steuern, daß bei der Ausgabe nach Excel Zahlenfelder als Zahlenfelder ausgegeben werden, nicht als Textfelder.

Um die Berichts-Properties zu setzen, können Sie dies entweder für einzelne Berichte, oder auch Server-weit einstellen.

Bei einzelnen Berichten gehen Sie im Report Inspector auf das oberste Element, und dort auf "Eigenschaften".

[JasperReports-Handbuch report properties1.png](#)

Sie erhalten dann eine Liste der Eigenschaften:

[JasperReports-Handbuch report properties2.png](#)

Dort können Sie Zeilen hinzufügen, bearbeiten und löschen. Eine Erläuterung der jew. Property erhalten Sie in dem jew. Dialog.

Um eine Property systemweit zu speichern, legen Sie eine Textdatei mit dem Namen "jasperreports.properties" an, und legen diese in den CLASSPATH. Im Kontext von Tomcat ist dies das Verzeichnis webapps/superx/WEB-INF/classes.

Seitenlayout

Seitenumbrüche

Es gibt drei Typen von Seitenumbrüchen:

- Wenn der gesamte Bericht paginiert ist (Report-Eigenschaft "Ignore Pagination"="false"), berechnet JR aufgrund des Seitenformats die Seitenumbrüche automatisch, und fügt auch, sofern vorhanden, die Bänder "page header" und "page footer" hinzu.
- Bei eingeschalteten Eigenschaften des jew. Bandes "Start on a new page=true" werden Seitenumbrüche ebenfalls aufgeführt
- Es gibt für manuelle Seitenumbrüche in der Palette das Element "Break", das an beliebiger Stelle einen manuellen Seitenumbruch einfügt. Nun ist die Frage: wenn gesamte Bericht nicht paginiert ist ("Ignore Pagination"="true"), was dann bei einem Break passieren soll. In älteren JR Versionen (vor 4.8) wurde der Break trotzdem gesetzt. In Versionen ab 4.8 passiert das nicht mehr, man kann aber ab Version 5.1 über die Berichts-Property

`net.sf.jasperreports.page.break.no.pagination=apply`

das alte Verhalten nachbilden.

In JasperSoft Studio oder iReport kann man zusätzlich noch beim Rendern des Berichts über den Parameter JRParameter.IS_IGNORE_PAGINATION steuern, ob Seitenumbrüche ausgeführt werden. Dies überlagert die Berichte

Leider kann man diesen Parameter nicht im Servlet setzen. Wenn Sie Seitenkopf/Fuß bei PDF wünschen, bei Excel aber nicht, nutzen Sie die Anleitung [unten](#).

Es gibt verschiedene Möglichkeiten, wie sich das Detailband bei einem Seitenumbruch verhalten soll. Diese werden über den ‚Split Type‘ des Bandes gesteuert. Beispiel:

Bei einem Seitenumbruch wird die aktuelle Zeile des Detailbandes aufgeteilt und geht auf der nächsten Seite weiter. Der ‚Split Type‘ steht auf Stretch.

[642x170px](#)

Falls erwünscht ist, dass jede Zeile ganz dargestellt wird, kann der ‚Split Type‘ auf Prevent gestellt werden:

[300x281px](#) Das Ergebnis sieht wie folgt aus:

[642x223px](#)

Die Höhe von Textfeldern dynamisch anpassen

[JasperReports-Handbuch layout1.png](#)

Textfelder besitzen die Eigenschaft „Stretch With Overflow“. Wird diese aktiviert, passt das Textfeld dynamisch seine Größe, falls der Inhalt nicht in das Feld passt. Der Text bricht um und die Höhe des Feldes wird vergrößert. Damit auch alle anderen Felder derselben Zeile ihre Größe entsprechend anpassen wird der „Stretch Type“ aller Felder dieser Zeile auf „Relative To Tallest Object (deprecated)“ gesetzt. Die Felder passen ihre Höhe dann dem höchsten Feld dieser Zeile an. Im Detailband genügt dies. Für Bänder mit festgelegter Zeilenanzahl, wie beispielsweise dem Titelband ist eine weitere Einstellung vorzunehmen.

Um im Titelband die Funktion „Stretch With Overflow“ nutzen zu können, ist es notwendig allen Feldern in darunterliegenden Zeilen den „Position Type“ „Float“ zuzuweisen. Die Position der Felder verändert sich dann entsprechend der Höhe darüberliegender Felder.

[JasperReports-Handbuch layout2.png](#)

Bericht ausgeben, ohne Dateninput

Wenn der Bericht bzw. das Layout mit Titelband etc. ausgegeben werden soll, obwohl es keine Daten gibt, dann wird die Funktion ‚When No Data Type‘ auf ‚All Sections No Detail‘ gestellt.

[336x374px](#)

Besonderheiten beim Export nach Excel

Für jedes Exportformat gibt es auf Berichtsebene bzw. darunter spezielle Properties, die Feinheiten des Zielformates festlegen. Dies wird im folgenden anhand des Excel-Exports von Formeln und Namen von Arbeitsblättern (Sheet-Names) gezeigt.

Bitte beachten Sie dass diese Hinweise für den "alten" Excel Export ("xls") gelten. Für die Exportformate xlsx, JExcel-API oder OpenOffice-Spreadsheets funktionieren diese teilweise nicht.

Um leere Zeilen im Excel-Export zu vermeiden, im jrxml-Kopf:

```
<property
name="net.sf.jasperreports.
```

wenn man Zahlenanpassung möchte

```
<property
name="net.sf.jasperreports.
```

→ font wird ggfs. verkleinert

```
<property
name="net.sf.jasperreports.
```

Excel-Formeln

Es gibt auch die Möglichkeit Zahlen nicht nur vorab zu berechnen, sondern beim Export nach Excel als Excel-Formel zu übergeben.

Um diese Funktion zu aktivieren muss im JRXML die Property mit folgender Zeile eingefügt werden:

```
<property
name="net.sf.jasperreports."
```

Oder über die Report Properties:

[JasperReports-Handbuch excel export detect cell type1.png](#)

Danach können in jedem Text Field in der Property expression:

[JasperReports-Handbuch excel export detect cell type2.png](#)

folgende Einträge ergänzt werden:

[JasperReports-Handbuch excel export detect cell type3.png](#)

Rechts neben net.sf.jasperreports.export.xls.formula wird die Formel für Excel eingetragen, allerdings ohne = am Anfang. Außerdem werden nicht alle Excel-Formeln unterstützt, insbesondere nicht deren deutsche Lokalisierung (z.B. oben SUMME(D119...)). Details finden Sie in der [POI](#)-Dokumentation. Um zu erfahren wie die Felder lauten, welche berechnet werden sollen, am besten den Bericht einmal ausführen und in Excel anzeigen lassen. Achtung: Wenn der Bericht noch bearbeitet wird, können sich die Felder noch verändern.

Bei net.sf.jasperreports.export.xls.pattern können speziell für Excel Formatangaben gemacht werden. In dem obigen Beispiel "#,##0.00;-#,##0.00" gibt es 2 Nachkommastellen und ein Tausender-Trennzeichen. Wenn z.B. Tausende von Euro angezeigt werden sollen, nimmt man als Pattern #.

Fallstrick: in der von uns getesteten Version (4.1.3) wurde die Property net.sf.jasperreports.export.xls.formula nur dann ausgewertet, wenn das das jew. Textfeld eine korrekte "Text field expression" hat, sie darf also nicht leer sein. Ob das ein Bug ist oder nicht ist uns derzeit unklar.

Beispiel für eine komplexe Summenformel SUM(D7:D20,D28,D33,D40)

Man kann die Formel auch als Expression formulieren, z.B. um eine Division durch 0 zu verhindern. Setzen Sie dazu das Häkchen bei "Use an expression":

[JasperReports-Handbuch excel formula expression.png](#)

Das obige Beispiel führt die Formel "D91/D149" nur aus, wenn die Variable \$V{Haush_Professoren_1} (die Quelle für die Zelle D149) einen Wert größer 0 hat. Wenn nicht, wird keine Formel reingeschrieben, sondern der Wert 0.

Diese Funktion bietet natürlich noch viel mehr Möglichkeiten, z.B. könnte man damit Excel-Zellennummern dynamisch berechnen etc. Beispiele dazu finden Sie in den JasperReport Demos in demo/samples/xlsformula/reports/XlsFormulaReport.jrxml.

Zellennummern dynamisch berechnen (Ein Beispiel)

Will man dass in der gesamten Exceldatei immer Spalte F=D+E ist, braucht man nicht für jede Zeile feste Formeln eintragen.

Als Erstes schaut man in welcher Zeile die eigentlichen Daten anfangen, also z.B. nach den zwei Headerzeilen ganz am Anfang der Tabelle und den 3 Spaltenüberschriften, die pro Seite wiederholt werden.

Hier ein Beispiel

```
($F{Zweckbestimmung}.isE
mpty())
```

Elegant geht es auch so

```
($F{Zweckbestimmung}.isE
mpty())
```

Das hat nur den Nachteil, dass die Formel in Excel nicht so schön ist.

WENN-/IF-Formel – Ein Fallstrick

Ein Fallstrick, der uns einige Stunden Arbeitszeit gekostet hat: die IF-Formel wird in OpenOffice und MS Excel mit dem Trennzeichen ";" für die THEN- und ELSE-Bedingung formuliert. JasperReports nutzt als Trennzeichen aber ",". Interessanterweise wird dann im fertigen Excel das "," als ";" angezeigt.

SUMIF – Summen mit Bedingungen (Bsp.: Zwischensummen ignorieren)

Wenn die Summe über einen großen Bereich gezogen werden soll, in dem auch Zwischensummen vorkommen, gibt es die folgende Formel:

SUMIF(A" + \$V{zeile_start}.i
ntValue() + ":" + A" + \$V{zeile_en

Wenn die Zwischensummen mit addiert werden würden, wäre das Ergebnis doppelt so hoch wie es sein sollte. Die Funktion SUMIF beinhaltet 3 Bereiche. Zuerst wird der Bereich eingegeben, auf den die Bedingung angewendet wird:

```
A" + $V{zeile_start}.intValue(
) + ":A" + $V{zeile_ende}.intVa
```

Hier ist das z.B. von A1 bis A10.

Danach kommt die Bedingung:

"<>Summe*"

Das bedeutet, es werden alle Zeilen genommen, die **NICHT** mit „Summe“ anfangen.

Als letztes kommt der Bereich, der summiert werden soll:

```
B" + $V{zeile_start}.intValue(
) + ":B" + $V{zeile_ende}.intVa
```

Hier ist das z.B. von B1 bis B10.

In diesem Beispiel werden alle Zahlen zwischen B1 und B10 aufsummiert, wenn in der entsprechenden Zeile in der Spalte A der Text nicht mit „Summe“ beginnt.

Statt 0-Werten eine leere Zelle ausgeben

Im Standardfall wird eine 0 ausgegeben, falls eine Formel in Excel auf leere Zellen oder Zellen mit einer 0 verweist. Um stattdessen eine leere Zelle auszugeben wird die Property `Property net.sf.jasperreports.export.xls.pattern` genutzt. Das obige Beispiel wird erweitert:

###0.00;-###0.00;;@

Die jeweiligen Zellen weisen nun 2 Nachkommastellen aus und zudem werden 0-Werte nicht mehr angezeigt.

OpenOffice vs Excel

OpenOffice kommt mit folgender Berechnung klar

"INDIRECT(\"E\" &
ROW())+INDIRECT(\"F\" &

Excel macht hier Ärger, wenn eine der Zellen Null ist, wird als Gesamtergebnis #WERT angezeigt, entweder Sum-Formel benutzen, oder noch mal mit IF prüfen, ob Zelle gefüllt ist

Struktur ist (IF(Zelle leer,0,sonst Zellenwert))

```
"IF(INDIRECT("\E" &
```

„Stretch with Overflow“ in Excel korrekt verarbeiten

Wenn Sie eine Zelle haben mit besonders langem Text, z.B. „Unterstützungen auf Grund der Unterstützungsgrundsätze, Fürsorgemaßnahmen sowie Kosten nach dem Arbeitssicherheitsgesetz“, hilft es, nur bei der Zelle, Line Spacing auf double zu setzen, dann sieht es in Excel besser aus.

Es wird das Stretch richtig verarbeitet.

```
<property
name="net.sf.jasperreports."
```

→ verkleinert fonts bei bedarf

```
<property
name="net.sf.jasperreports."
```

Pagination in Excel-Dokumenten

In Excel-Dokumenten in die Seitengröße naturgemäß irrelevant. Trotzdem wird das im jrxml gesetzte Seitenformat beim Excel-Export beachtet. Das stellt sich so dar, dass im Excel-Dokument der „Rest der Seite“ durch einzelne breitere beziehungsweise höhere Zellen gefüllt wird. Das sieht beispielsweise folgendermaßen aus:

[JasperReports-Handbuch excel export pagination1.png](#)

Unter dem Tabellenschluss ist die Zeile 84 so hoch, dass die im jrxml eingestellte Seitenhöhe erreicht ist. Um das zu verhindern wird für den gesamten Bericht `isIgnorePagination="true"` gesetzt. Die Einstellung wird im source-Code ganz oben im Tag `<jasperReport isIgnorePagination="true">` gesetzt oder über die grafische Oberfläche:

[JasperReports-Handbuch excel export pagination2.png](#)

Excel-Export in einzelne Tabellenblätter

Normalerweise werden beim Export nach Excel die Berichtsinhalte in das erste Tabellenblatt geschrieben. Je nachdem wie Sie auf Berichtsebene die Property `net.sf.jasperreports.export.xls.one.page.per.sheet` setzen, wird der Seitenumbruch gesteuert. Setzen Sie den Wert auf "true", dann führt jeder Seitenumbruch zu einem neuen Sheet bzw. Arbeitsblatt in der Excel-Datei. Standardmäßig werden diese dann "Page 1", "Page 2" etc. benannt.

Wenn Sie die Namen der Sheets explizit vergeben wollen, können Sie dies wie folgt machen: Wenn Sie z.B. die Berichtsproperty `net.sf.jasperreports.export.xls.sheet.names.all` auf den Wert

"Hauptseite/wichtige Seite/Unwichtige Seite" setzen, bekommen Sie drei Sheets mit den jew. Namen, d.h. der "/" fungiert als Trennzeichen. Dementsprechend sollte im Namen des Sheets kein "/" auftauchen.

Seit JR 4.1 gibt es auch die Möglichkeit, die Sheetnamen dynamisch zu vergeben. Auch hier werden Properties mit Expressions genutzt.

Es gibt verschiedene Möglichkeiten Berichtsinhalte in verschiedene Excel-Tabellenblätter zu verteilen. Im Folgenden werden zunächst Möglichkeiten betrachtet, welche bei Einzelberichten in Frage kommen. Danach geht es um die klassische Haupt- und Unterberichtsstruktur und zuletzt um Booklets, welche die in den meisten Fällen wohl elegantere Alternative zu klassischen Haupt- und Unterberichten darstellen.

Einzelne Berichte

Nehmen wir an wir haben einen Bericht mit der Gruppierung auf die Spalte "LFB", im Gruppenkopf befindet sich ein Text Field mit der Expression `$F{LFB}`.

Sie können diesem Textfeld eine Property `net.sf.jasperreports.export.xls.sheet.name` geben:

[JasperReports-Handbuch excel export sheets names.png](#)

Das Textfeld erhält somit beim Füllen des Berichts den jew. Wert, z.B. "Anglistik, Amerikanistik", und diese Überschrift steht auch im Gruppenkopf. Wenn der Gruppenkopf das Attribut "Start on a new page" hat, dann beginnt der Gruppenkopf auf einer neuen Seite und, dank der Berichts-Property `net.sf.jasperreports.export.xls.one.page.per.sheet`, erzeugt Excel / OpenOffice dann ein neues Sheet mit dem Namen Anglistik, Amerikanistik. Beim Sheetnamen soll nur sichergestellt werden, dass kein "/" im Text vorkommt, daher der replace-Befehl zur Sicherheit. Hier ein Screenshot in OpenOffice:

[JasperReports-Handbuch excel export sheets names2.png](#)

Man kann das gesamte Verhalten auch deaktivieren, indem man auf Berichtsebene das Attribut "Ignore Pagination" auf "true" setzen. In diesem Falle werden alle Seitenumbrüche ignoriert.

Hinweis für SuperX/Edustore Nutzer: leider haben wir in älteren Kernmodul-Versionen dieses Attribut "Ignore Pagination" im Servlet fest auf "true" gesetzt, so dass Sie diese Funktion im Servlet erst ab Version SuperX 4.5 bzw. HISinOne-BI 7.0 nutzen können.

Haupt- und Unterberichte

Das vorherige Kapitel hat auf gezeigt, wie innerhalb eines einzelnen Einzelberichtes mit gezieltem setzen von Seitenumbrüchen ein solcher Berichte auf verschiedene Tabellenblätter eines Excel-Dokumentes aufgeteilt wird.

Um mehrere Einzelberichte in einem Excel-Dokument gesondert auf verschiedene Tabellenblätter aufzuteilen eignen sich klassische Haupt- und Unterberichtsstrukturen (vgl. Kapitel 3.5). Die simpelste Methode wäre das Seitenlayout eines Berichtes so groß zu wählen, dass der gesamte Bericht auf eine Seite passt.

Falls Haupt- und Unterberichte verwendet werden, darauf achten, dass der Hauptbericht direkt unter dem letzten Unterberichtselement endet. Ansonsten entsteht ein weiteres leeres Tabellenblatt. Der folgende Screenshot zeigt, wie es nicht sein sollte. Hier muss das Summary-Band verkleinert werden, sodass es mit dem Unterberichtselement endet.

[JasperReports-Handbuch excel export main- subreport.png](#)

Individuelle Gestaltung bei Booklets

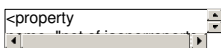
Booklets bieten diverse Vorteile beim Gestalten der einzelnen Seiten beziehungsweise Sheets. Denn jeder Unterbericht kann hier individuell formatiert werden. Dies betrifft beispielsweise die Definition von Sheetnamen, da hierfür die Eigenschaft "Tab name expression" im "Buch"-[Bericht](#) vorgesehen ist. Auch das Problem, dass ein Sheet in diesem Falle nicht mehrere Seite umfassen darf (und man somit die Seitengröße unrealistisch groß machen muss) ist damit behoben.

Eine Ausnahme stellen jedoch die Properties dar. Diese werden im Hauptbericht gesetzt und gelten für alle Unterberichte.

Das Drucklayout in Excel konfigurieren

Seitenränder und Kopfzeile im Drucklayout formatieren

Zum Anlegen eines Textes in der Kopfzeile gibt es die Properties



Je nach gewünschter Position wird der jeweiligen Property ein Wert übergeben. Ein Beispiel:



In diesem Beispiel wird zentral der Text „**Meldung der Ist-Besetzung**“ übergeben. Der Text ist in der Schriftart Liberation Sans und fettgedruckt formatiert (&"Liberation Sans,Bold"), sowie in Schriftgröße 10 (&10).

Rechts wird der Text „**Anlage 1** (zu VwV HwiF 3.2.1)“ verwandt. Der erste Teile „**Anlage 1**“ ist wie der zentrale Text formatiert (&"Liberation Sans,Bold"&10). Für den zweiten Teil „(zu VwV HwiF 3.2.1)“ wird von Fett- auf Regulärdruck gewechselt (&"Liberation Sans,Regular"&10). Weitere Möglichkeiten zur Formatierung finden sich auf <https://poi.apache.org/apidocs/dev/org/apache/poi/xssf/usermodel/extensions/XSSFHeaderFooter.html>.

Damit die Kopfzeile im Drucklayout sichtbar ist, ist es notwendig den Seitenrand groß genug zu gestalten. Dafür werden folgende Properties genutzt:



Vordefinierte Zeilen im Drucklayout auf jeder Seite wiederholen

Um im Drucklayout beispielsweise einen Tabellenkopf auf jeder Seite zu wiederholen, wird die Property

```
<property
name="net.sf.jasperreports."
```

genutzt (dafür eigene Klasse `de.superx.servlet.SxJRXIsExporter` geschaffen und `de.superx.servlet.JasperCreator` angepasst – aktuell in NHS-Pack für 7.1 enthalten). Der Wert gibt an, welche Zeilen des Excel-Dokumentes im Drucklayout auf jeder Seite wiederholt werden. Im Beispiel (`value="3:7"`) sind die Zeilen 3 bis 7 betroffen.

Seitenumbrüche im Drucklayout verhindern

Teilweise ist es wünschenswert Seitenumbrüche im Drucklayout zu verhindern, sowohl in der Breite als auch in der Höhe. Hierfür werden die Properties

```
<property
name="net.sf.jasperreports."
```

genutzt. Also value kommen hier 1 und 0 für entweder true oder false infrage. Setzt man der Wert true, dann wird nicht umgebrochen. Das bedeutet für dieses Beispiel, dass der gesamte Inhalt des Excel-Dokuments auf eine Seite gedruckt würde.

Zellen verbinden

```
<property
name="net.sf.jasperreports."
```

1,1,4,1 - will merge from B2 to E2. Remember it is zero based indexing.

Zellschutz aktivieren

```
<property
name="net.sf.jasperreports."
```

→ Alle Zellen sind mit Passwort geschützt.

Wenn bestimmte Spalten/Zeilen nicht geschützt sein sollen:

Property `net.sf.jasperreports.export.xls.cell.locked` false.

Dies kann man auch über eine komplexe PropertyExpression steuern, z.B.

```
(
($F{Zeile}!=null&&
$F{Zeile}.indexOf("A")>-1||
))?"true":"false"
```

Zellen fixieren

In Jaspersoft Studio können Sie festlegen, dass beim Excel-Export bestimmte Zeilen und Spalten fixiert werden, indem Sie spezielle Eigenschaften für den Export konfigurieren.

Um Zeilen und Spalten in Excel zu fixieren, müssen Sie die Eigenschaften

```
net.sf.jasperreports.export.xls
net.sf.jasperreports.export.xls
```

und

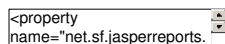


konfigurieren.

Wenn Sie möchten, dass die erste Zeile und die erste Spalte fixiert werden:



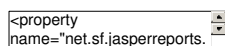
Nach dem Export werden in Excel die erste Zeile und die erste Spalte fixiert, und der Rest des Arbeitsblatts bleibt scrollbar.



Hintergrund- oder Textfarben beim Excel-Export

Beim Excel Export werden Zellen, die ein Rechteck mit einer Print-When-Expression zugewiesen bekommen haben, u.U. nicht richtig wiedergegeben, während es beim PDF Export klappt. Um dies zu beheben müssen Sie

1. den Farbraum beim Excel Export erweitern: Excel unterstützt bei Hintergrund- oder Vordergrundfarben nur wenige [Farbcodes](#). Für spezielle Farben nach RGB-Wert muss die Berichts-property



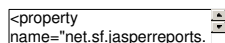
gesetzt werden (benötigt Bibliothek org-netbeans-core.jar im Classpath der Anwendung)

2. Statt farbiger Textfelder oder Rechtecke mit "Print-When-Expression" müssen Sie [Conditional Styles](#) nutzen.
3. Wenn Sie mit Hintergrundfarben arbeiten, müssen die Textfelder und die jew. Styles die Eigenschaft "Opaque"="true" haben.
4. Beim Export nach Excel müssen Sie das JExcelApi nutzen. Dies ist bei SuperX seit Kernmodul 4.5 der Standard, bei HISinOne-BI seit Version 7.1.

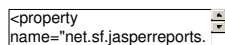
Seitenkopf und Seitenfuß bei Excel unterdrücken

Wenn Ihr Bericht normalerweise ein Band "Page Header" bzw. "Page Footer" hat, können Sie dies beim Exportformat XLS unterdrücken (z.B. wenn Sie dort keine Seitenzahlen wünschen, bei PDF aber schon).

Setzen Sie dazu die Berichts-Property



Beim Ausgabeformat XLSX heißen die Properties:



Dies klappt ab JR Version 4.5, sie können diese Properties aber lediglich im XML-Quellcode hinzufügen.

Dimension Tausend-Euro

In Excel als Dimension Tausend-Euro ausgeben, mit zwei Nachkommastellen

In Deutschem Excel bei manueller Angabe benutzerdefiniertes Format #.##0,00.

In Jasper muss Amerikanische Vorlage sein:

-> #,##0.00

Bei Excel kann man als Property Expression `net.sf.jasperreports.export.xls.pattern` z.B. nehmen

1. , → zeigt nur Tausender Werte an

`#,##0.00,`

→ zeigt Tausender Werte, mit zwei Nachkommastellen an

der Werte selber ist einfach `$F{Soll}`

Bei PDF/HTML muss man den Wert durch 1000 teilen `$F{Soll}.doubleValue()/1000`

und dann als Pattern z.B. `#,##0.00;-#,##0.00`

Die Rundung macht das Pattern.

Nutzung in Edustore

Berichtsentwicklung bei Grunddaten und Basisberichten

Allgemein

In den Grunddaten- und Basisberichten kann man beliebige Ergebnistabellen lokal als XML speichern, und dann als Datenquelle in JasperReports 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 JasperReports 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.

[JasperReports-Handbuch vergleich jrxml xml zusammen.png](#)

Datenquelle erzeugen

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

[JasperReports-Handbuch iReport-Datasource erstellen.jpg](#)

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 JasperReports 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 JasperReports sind dann folgende Schritte zu tun, um die neue Datenquelle einzubinden.

Datenquelle in JasperSoft Studio

Die Datenquellen heißen hier "Data Adapters" und werden im Repository hinzugefügt:

[JasperReports-Handbuch jaspersoft_datasource1.png](#)

Achtung: wenn Sie JasperSoft in Ihrem vorhandenen Eclipse als Plugin installiert haben, sehen Sie den Reiter "Repository" nicht; in diesem Fall liegen die Data Adapter im jew. Projekt.

Im Assistenten geben Sie XML-Dokument als Datentyp an, und im nächsten Reiter spezifizieren Sie das Format:

[JasperReports-Handbuch jaspersoft_datasource2.png](#)

Das Datumsformat ist bei Edustore das deutsche Schema "dd.MM.yyyy", und das Zahlenformat ist eine Zahl mit 6 Nachkommastellen und "." als Dezimaltrenner, also "#0.#####". Bei Locale müssen Sie "Englisch" angeben, weil Dezimalzahlen in den Edustore-XML-Quellen den "." als Dezimaltrenner haben.

Wichtig ist auch das Ankreuzfeld "Use the report Xpath expression", dies müssen Sie immer ankreuzen.

Datenquelle in iReport

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

[JasperReports-Handbuch iReport-Datasource 1.jpg](#)

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

[JasperReports-Handbuch iReport-Datasource 2.jpg](#)

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

[JasperReports-Handbuch iReport-Datasource 3.jpg](#)

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.

[JasperReports-Handbuch iReport-Datasource 4.jpg](#)

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 JasperReports 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.

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“.

[JasperReports-Handbuch iReport-leeren Bericht erstellen.jpg](#)

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 JasperReports 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.

Template einbinden

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.

[JasperReports-Handbuch iReport-Stylesheet 1.jpg](#)

Klickt dann auf „Neuen Datensatz erzeugen“.

[JasperReports-Handbuch iReport-Stylesheet 2.jpg](#)

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 JasperReport 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.

[JasperReports-Handbuch iReport-Stylesheet 3.jpg](#)

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

[JasperReports-Handbuch iReport-Stylesheet 4.jpg](#)

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).

Nutzung des Upload Servlets

Wenn Sie keinen Schreibzugriff auf den Ordner *WEB-INF/reports* haben, können Sie auch Templates über das [Upload Servlet](#) direkt im Browser hochladen. Dazu müssen Sie in der web.xml die Dateieindungen .jrxml und (sofern benötigt) .jasper erlauben, siehe obiger Link.

Danach können Sie die Report Templates hochladen:

[JasperReports-Handbuch jr upload1.png](#)

Nach dem Upload können Sie das Template zunächst registrieren:

[JasperReports-Handbuch jr upload2.png](#)

und dann der jew. Maske zuordnen. Beachten Sie, dass der Pfadname relativ zum Ordner WEB-INF/reports angegeben wird, also in der Regel

../default/custom

bzw. beim Mandantenbetrieb

../Mandant-ID/custom

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:

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"
```

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 HiSiOne 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/Fachbereic"
```

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.

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 Datenblatt-Berichten 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 JasperReports. 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.

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 können, 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.

[JasperReports-Handbuch_stylesheet_field_maske.png](#)

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

[JasperReports-Handbuch_stylesheet_field_tabelle.png](#)

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.

[JasperReports-Handbuch_stylesheet_field_list.png](#)

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

[JasperReports-Handbuch stylesheet field edit.png](#)

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:

[JasperReports-Handbuch stylesheet field.png](#)

In dem Bearbeitungsformular können Sie Datensätze erzeugen, löschen etc.

Kurzanleitung Direktaufruf von Jasper-Berichten von der Maske

Nach dem Abschieken 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
```

und zum Beispiel für Default der Maske 23880

```
<<SQL>> select distinct
filename,X.caption from
```

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
```

Beispiel für Default Excel:

```
<<SQL>> select
element_value,description
```

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

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.

[JasperReports-Handbuch sx captions markiert.png](#)

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

Makroberichte als Datenquelle für JasperReports

Sie können als Datenquelle für JR nicht nur einzelne Masken nehmen, sondern auch die Ausgabe von Makro-Masken. Dadurch können die verschiedenartigsten Informationen aus den normalen Masken für übergreifende Berichte genutzt werden.

Das folgende Beispiel zeigt, wie im KLR-Bericht "Berichtsblatt: Kennzahlen aus der Kostenrechnung" Teile verwendet werden, um einen eigenen JR zu erzeugen, der wiederum Elemente des Makroberichts als Unterbericht nutzt.

Zunächst rufen Sie den Makrobericht auf:

[JasperReports-Handbuch makro.html.png](#)

Das Bild zeigt die ersten drei Elemente des Makros, sie werden mit einer Ordnungsnummer durchnummeriert.

Wir laden dazu die XML-Datenquelle herunter, die XML-Datenquelle binden wir wie [gewohnt](#) in JasperReports ein:

[JasperReports-Handbuch makro datenquelle.png](#)

Dann klicken wir im Browser über den Tabellen den Button rechts von "Bericht entwerfen" → Leerer Bericht. Die JRXML-Datei speichern wir wie [gewohnt](#).

Im Designer sehen wir nun bei den Feldern eine erweiterte Liste:

[JasperReports-Handbuch makro jrxml.png](#)

Wir sehen also, daß die Ergebnisspalten mit einem Präfix für die Tabellennummer versehen sind, z.B. "tab_0". Dadurch können wir die Tabellen und somit auch die jew. Spalten eindeutig identifizieren. Dies ist gleichzeitig der Hauptbericht. Das Detail-Band löschen wir (ist nicht nutzbar), stattdessen aktivieren wir ein Summary Band.

Dann erzeugen wir den [Unterbericht](#) für Studierenden-VZÄ. Dazu speichern wir die obige jrxml-Datei z.B. mit dem Namen makro_beispiel_ub_stud.jrxml und entfernen das Titel-Band und Seitenfuß-Band. Dann fügen wir im Designer einen Parameter "ordnr" hinzu:

[JasperReports-Handbuch makro ub param.png](#)

Der Unterbericht bekommt dann in der Report Query folgende Änderung:

[JasperReports-Handbuch makro ub report query.png](#)

Die Ordnungsnummer wird vom Hauptbericht übergeben und dient dazu, in der XML-Datenquelle auf die einzelne Tabelle zu filtern.

Zusätzlich geben wir noch einen weiteren Filter ein, damit die Summenzeilen aus der Originaltabelle nicht in die Kreuztabelle aufgenommen werden:

[JasperReports-Handbuch makro ub filter.png](#)

Damit der Unterbericht auch "stand-alone" ausführbar ist, können Sie dem Parameter "ordnr" auch einen Standardwert zuweisen, hier z.B. "2":

[JasperReports-Handbuch makro ub ordnr default.png](#)

Nun können wir die [Kreuztabelle](#) mit dem Assistenten erzeugen. Für unser Beispiel enthält die Kreuztabelle Studierenden-VZÄ nach Studiengang und Fachkennzeichen:

[JasperReports-Handbuch makro ub crosstab.png](#)

Wir speichern den Unterbericht und binden ihn in den Hauptbericht im Summary Band ein. Dabei vergeben wir wie gewohnt die Parameter XML_LOCALE und XML_NUMBER_PATTERN. Außerdem erzeugen wir den Parameter "ordnr", und weisen diesem den Wert 2 zu, weil die Studierenden-Tabelle die Ordnungsnummer 2 hat (s.o.):

[JasperReports-Handbuch makro hauptbericht ub params.png](#)

Damit ist die Konfiguration beendet, der Bericht sieht so aus:

[JasperReports-Handbuch makro hb erg.png](#)

Weitere Unterberichte lassen sich analog hinzufügen.

Hinweis zum Einbinden im Servlet: Unterberichte arbeiten nur mit der Datenquelle XML, daher müssen Sie die Datei des Hauptberichts mit dem Suffix "_xmlsource" versehen.

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.NoSuchMethodErr
or:....
```

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.

iReport Grafiken in JasperReports

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:

[JasperReports-Handbuch grafik theme error.png](#)

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

Weitere Bibliotheken von iReport

Bei der Meldung

```
java.lang.NoSuchMethodErr
or:
```

sollten Sie die Bibliotheken

```
iReport-
xxx/ireport/modules/ext/asm
```

und zur Sicherheit auch

```
asm-attrs.jar
```

in den WEB-INF/lib-Ordner kopieren.

Troubleshooting

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">
```

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 (kompiliert) 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.

Im Excel Export fehlen Felder

In Excel werden überlappende Elemente nicht angezeigt. Wenn also 2 Elemente übereinander liegen wird in Excel nur das oberste (welches im Vordergrund liegt) angezeigt. Hier ist besonders drauf zu achten, dass visuell 1px Abstand zwischen den Elementen Platz ist. Wenn von beiden Elementen jeweils eine Linie zu sehen ist, ist es korrekt. Wenn nicht, zählt das schon als Überlappung. Dafür im iReport Designer am besten ganz nah heran zoomen um diesen Abstand zu prüfen.

Hier ein Beispiel. Die beiden oberen Elemente überlappen, die unteren nicht:

[JasperReports-Handbuch ireport excel.png](#)

Hinweis für HisInOne 7.0 Kompatibilität

Jrxml-Dateien, die unter iReport 4.1.3 entwickelt wurden und in HisInOne 7.1 funktionieren, können in 7.0. zur Fehlermeldung führen:

```
java.lang.NoClassDefFound
```

In der jasperreports5.1.0 (hisinone 7.0) ist die Klasse GroovyEvaluator nicht enthalten, für 7.1 ist die jasperreports-6.1.0.jar enthalten, die enthält den GroovyEvaluator.

Workaround:

```
language="groovy"
```

ändern in

```
language="java"
```

HTML Export keinen whitespace

Normal macht JR bei HTML-Exporten für den „Seitenumbruch“ eine Reihe von Leerzeilen. Um dies zu vermeiden, setzen Sie die Property `IgnorePagination=true`:



```
isIgnorePagination="true">
```

UUID's

In JasperSoft Studio wird an jedes Element eine lange UUID angehängt, in aktuellen Versionen von SuperX/ Edustore macht das keine Probleme. In älteren Versionen bzw. in iReport 4.* schon. Damit der JasperReport abwärtskompatibel ist, muss diese gelöscht werden. Bei großen Berichten taucht diese aber unter Umständen mehrere hundert Mal auf. Hier kann in JasperSoft Studio in dem "Source"-Reiter des Berichtsdesigners nach folgender Regular Expression gesucht werden und diese mit einem Leerzeichen ersetzt werden:

```
uuid="[0-9a-z-]{36}"
```

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/entwicklung/iReport-Handbuch/main.htm>