



# BIRT Best Practices

Christian Brell

Orientation in Objects GmbH

Weinheimer Str. 68  
68309 Mannheim

www.oio.de  
info@oio.de



## Java, XML und Open Source seit 1998

# Java und XML

### ) Software Factory )

- Schlüsselfertige Realisierung von Java Software
- Individualsoftware
- Pilot- und Migrationsprojekte
- Sanierung von Software
- Software Wartung

### ) Object Rangers )

- Unterstützung laufender Java Projekte
- Perfect Match
- Rent-a-team
- Coaching on the project
- Inhouse Outsourcing

### ) Competence Center )

- Schulungen, Coaching, Weiterbildungsberatung, Train & Solve-Programme
- Methoden, Standards und Tools für die Entwicklung von offenen, unternehmensweiten Systemen

## Warum Best Practices?

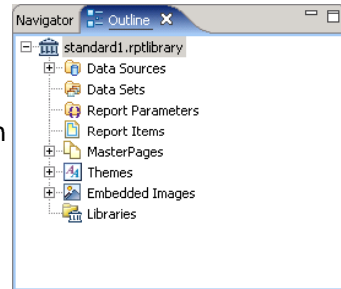
- Ein Bericht existiert selten allein
  - Copy And Paste ist schlecht
- Bericht baut auf Datenbank auf
  - Starke Abhängigkeit von Bericht zu Datenbank
- Verschiedene Ausgabeformate
  - Komplexe Logik um Bericht auf Ausgabeformate zu optimieren
- Entwickeln im Team
  - Wartung und Weiterentwicklung von ursprünglichem Entwickler geleistet?

## Gliederung

- **Arbeit mit Libraries**
- Templates
- BIRT JavaScript
- Event-Handler
- Datenbanken
- Security
- Checkliste für Reportprojekte
- Zusammenfassung

## Libraries

- „Container“ für wiederverwendbare Elemente
- Modularisierung und Wiederverwendung mittels Libraries von
  - Data Sources
  - Data Sets
  - Styles
  - Elemente
  - ...
- Bericht kann beliebig viele Libraries nutzen
  - Drag&Drop von Elementen in Bericht
- Änderungen im Lib führt zu Änderung in den abhängigen Berichten



## Styles

- "Styles" zur Wiederverwendung von Formatierung
- Hybrid aus CSS und MS Word Styles
- Styles erhalten Namen
- Styles können einzelnen Report Elementen zugewiesen werden
- Beispiel
  - Style mit Namen ColumnHeading erstellen
  - Allen Spaltenüberschriften zuweisen
- Vordefinierte Styles erleichtern Formatierung
  - Bsp: table-header entspricht allen Tabellenüberschriften

- Gruppierung von Styles zu Themes in Library
- Pro Report kann ein Theme gewählt werden
  - Nur Styles aus gewähltem Theme nutzbar
- Themes sind voneinander unabhängig
  - Können verschiedene Styles beinhalten

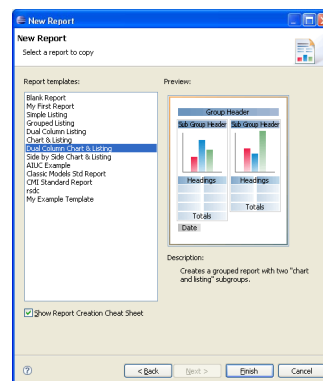
- Mehrfache Libraries empfohlen
  - Library für Datenquellen: Datenbankverbindungen, Abfragen,...
  - Library für Themes: Hinterlegen von Styles und Master Pages
  - Library für Elemente: Wiederverwendbare Grids, Label, Tabellen, ...
- Nur wiederverwendbare Elemente gehören in Library
  - Lohnt wenn gleiches Element **mehr** als 1x auftaucht
- Elemente in Library einfach halten
  - Komplexe Elemente häufig nicht wiederverwendbar
  - Features schwer zu warten

## Gliederung

- Arbeit mit Libraries
- **Templates**
- BIRT JavaScript
- Event-Handler
- Datenbanken
- Security
- Checkliste für Reportprojekte
- Zusammenfassung

## Templates

- Report in irgendeinem Stadium
  - Startpunkt für neuen Report
- Gleiche Möglichkeiten wie Reports
  - Einbinden von Libraries
  - Einfügen von Datenquellen, Abfragen
  - Tabellen
  - Strukturelemente
- Bei anlegen eines neuen Reports kann Template ausgewählt werden



## Unterschiede Libraries und Templates

- Reports verlieren ihre Verbindung zu verwendetem Template
  - Änderungen am Template gelten nur für zukünftige Reports
- Reports behalten ihre Verbindung zu Libraries
  - Änderung an Element aus Library spiegelt sich in allen Reports wieder die dieses Element verwenden
- Templates sind „unfertige“ Reports
- Libraries sind keine Reports

## Templates – Best Practices

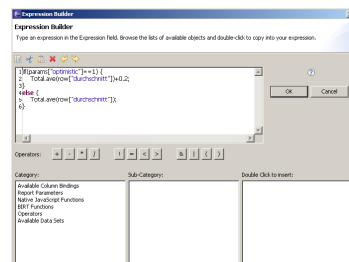
- Einsatz von Templates pro Reportprojekt
  - Unterstützung der Reportdesigner durch definierten Ausgangspunkt
- Template verlinkt Report bereits mit notwendigen Libraries
  - Importiert über Reports hinweg verwendete Elemente
    - **Master-Pages**
    - **Data-Sources**
    - **Stukturelemente**
  - Definiert evtl. bereits verwendeten Theme

## Gliederung

- Arbeit mit Libraries
- Templates
- **BIRT JavaScript**
- Event-Handler
- Datenbanken
- Security
- Checkliste für Reportprojekte
- Zusammenfassung

## BIRT Scripting

- Basiert auf Mozilla Rhino Script
- Expressions (*Ausdruckserstellung*)
  - Ausdrücke die einen einzelnen Werte zurück geben
  - Rückgabewert ist implizit letzte Zeile die vor Beendigung erreicht wird
    - **Kein return statement erlaubt!**
  - Anwendung mittels Expression Builder
- Eventhandler (Method Scripting)
  - Später!



- Häufig verwendete Funktionen pro Bericht nachprogrammiert
    - Copy&Paste
    - Reimplementierung pro Bericht
  - Berichte mit Geschäftslogik überfrachtet
    - Teilweise nur schwer von Berichtslogik abzugrenzen
  - Wenig Unterstützung durch Entwicklungsumgebung
    - Kein Syntaxhighlighting, keine Code-Completion
- Funktionen schwer wartbar  
→ Kein Überblick über verwendete Funktionen

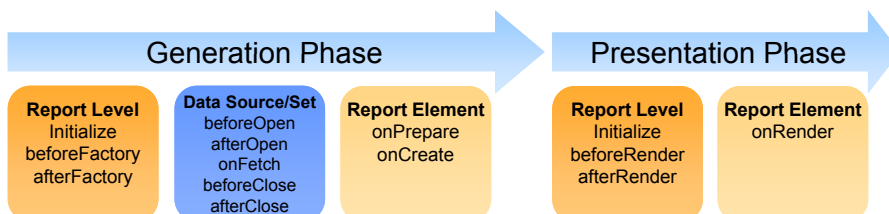


- Auslagern oft verwendeter Skripte als Funktion in eigene Datei
    - .js Datei enthält RhinoScript Funktionen
    - deklarierte Funktionen in Report nutzbar
      - **Trennung von Geschäftslogik und Berichtsdesign**
    - Funktionen sind isoliert wart- und testbar
      - **Junit?**
  - Komplexe Logik aufwändig umzusetzen
    - Wenig Unterstützung über vorhandene Tools
      - **Nur begrenztes Syntaxhighlighting**
      - **Keine Code-completion**
- Logik in Java programmieren und über Rhino Script einbinden

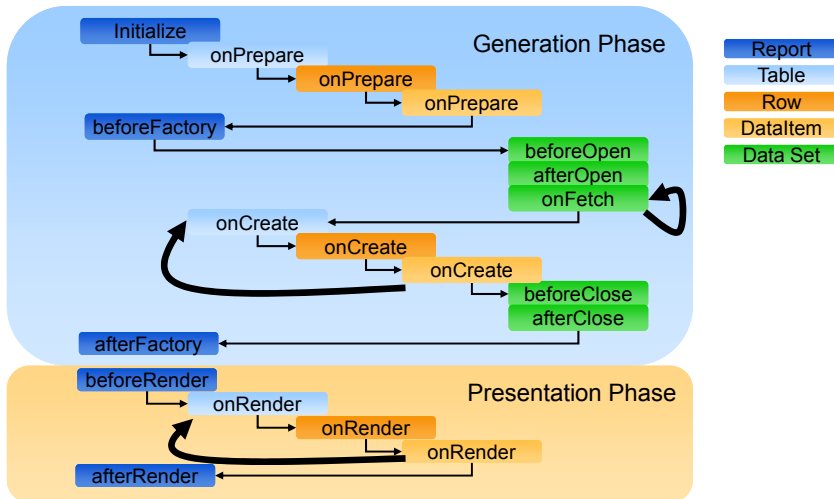


- Arbeit mit Libraries
- Templates
- BIRT JavaScript
- **Event-Handler**
- Datenbanken
- Security
- Checkliste für Reportprojekte
- Zusammenfassung

- Bei Berichtsgenerierung werden Hook-Methoden aufgerufen
- Beziehen sich auf einzelne Reportelemente
  - Bericht als ganzes
  - Datenquellen und Datasets
  - Elemente des Berichts
- Implementierung in Java oder JavaScript



## Events - Run then Render Pipeline (Web viewer)



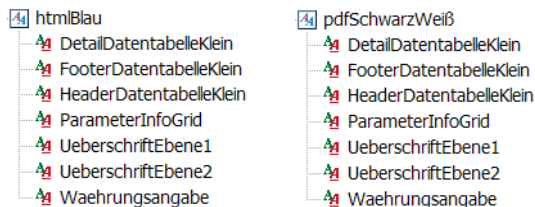
## Events – Best Practice

- initialize
  - Wird als erstes aufgerufen
  - globale Variablen initialisieren
  - Security Checks
- beforeFactory
  - Manipulieren der Reportstruktur
    - **Z.B. Löschen von Elementen**
- beforeRender
  - Logik in Abhängigkeit des Ausgabeformats
    - **Z.B. Theme Switching (später)**

## Events – Best Practice

- afterFactory, afterRender
  - Cleanup
- onCreate
  - Elemente in Abhängigkeit der Daten verändern
- onRender
  - Elemente in Abhängigkeit der Daten und/oder des Ausgabeformats ändern
    - **Größenanpassung für PDF**

## Events - Theme Switching



```
Script: beforeRender
if (reportContext.getOutputFormat ()=="pdf") {
    reportContext.getDesignHandle ().setThemeName ("styles.pdfSchwarzWeiß");
}
```

- Arbeit mit Libraries
- Templates
- BIRT JavaScript
- Event-Handler
- **Arbeit mit Datenbanken**
- Security
- Checkliste für Reportprojekte
- Zusammenfassung

- Datenbankverbindung immer in Library auslagern
- Öffnen einer Datenbankverbindung teuer
  - Verbindung nicht von BIRT pro Lauf herstellen lassen
    - **Besser Möglichkeiten der Laufzeitumgebung nutzen**
      - JNDI
      - Datenbankverbindung über ReportEngine API injizieren
- Datenbanksystem optimiert auf Datenverarbeitung
  - Filtern und Sortieren in Datenbank ist schneller!
- Starke Abhängigkeit zwischen Report und Tabellenstruktur
  - Entkoppelung durch Arbeit mit Views
  - Entkoppelung durch Verwenden von Stored Procedures
  - Delegieren an DB-Entwickler

- Arbeit mit Libraries
- Templates
- BIRT JavaScript
- Event-Handler
- Arbeit mit Datenbanken
- **Security**
- Checkliste für Reportprojekte
- Zusammenfassung

- Visibility meist kein Mittel um Daten zu verstecken
  - .rptdocument enthält Daten aller Reportelemente
  - Besser Element in beforeFactory Event aus Report löschen

```
table = reportContext.getDesignHandle().findElement("meine Tabelle");  
if (table != null) {  
    table.drop();  
}
```

- Parameterinjektion
  - Keine Reportparameter über JavaScript in Abfrage „basteln“
    - **Nur über Parameter-Tab in Dataset verlinken**
- Reportdesign öffentlich?
  - Aktive Suche nach Schwachstellen möglich

- Arbeit mit Libraries
- Templates
- BIRT JavaScript
- Event-Handler
- Datenbanken
- Security
- **Checkliste für Reportprojekte**
- Zusammenfassung

- Requirement-Analyse für Reports durchführen
- Namenskonventionen festlegen
- Verzeichnisstruktur für Projekt festlegen
- Report- und Funktionsbibliotheken anlegen
- Laufzeitumgebung (Server) festlegen
- Datenzugriffe spezifizieren (SQL-Query vs. Stored Procedures, XML,...)

## Checkliste für Reportprojekte - 2

- Wiederverwendung evtl. bestehender Java Klassen analysieren
- Mehrsprachigkeit?
- Templates zur Unterstützung der Entwickler erstellen
- Templates zur Reportdokumentation anfertigen (Word, Wiki, ...)
- Testplan für fertige Reports entwickeln
- Projekt für Bugtracking Tool aufsetzen (Jira, Bugzilla, ...)

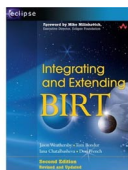
## Gliederung

- Arbeit mit Libraries
- Templates
- BIRT JavaScript
- Event-Handler
- Datenbanken
- Security
- Checkliste für Reportprojekte
- **Zusammenfassung**

- JavaScript sehr mächtiges Werkzeug
  - Entwicklung komplexer Logik besser in Java
- Einsatz von Libraries dringend empfohlen
  - Templates können Arbeitsalltag deutlich vereinfachen
- Erstellen von Report- und Funktionsbibliotheken iterativer Prozess
  - Regelmäßiger Austausch wichtig
  - Refactoring
- Komplexität wächst mit Größe des Projekts
  - Ebenso Wichtigkeit einer strukturierten Vorgehensweise
  - Häufig schnell mit 2 Berichten begonnen, dann ...?



- BIRT: A Field Guide to Reporting (Revised)
  - Broschiert: 794 Seiten
  - Addison Wesley
  - Erscheinungsdat: 28. August 2008
  - Sprache: Englisch
  - ISBN: 978-0321580276



- Integrating and Extending BIRT (Revised)
  - Broschiert: 724 Seiten
  - Addison Wesley
  - Erscheinungsdat: 7. August 2008
  - Sprache: Englisch
  - ISBN: 978-0321580306





Le BIRT Expert's Guide to ...

### BIRT Best Practices

How to Develop a Reporting Framework,  
Implement Best Practices and  
Manage Your BIRT Reporting Project

### • Le BIRT Expert's Guide to BIRT Best Practices

- David Mehi, eBook
- Erscheinungsdatum: 2009 (fortlaufend aktualisiert)
- Sprache: Englisch
- Bezugsquelle: <http://www.lebirtexpert.com>

## BIRT User Group Mannheim

- Interessiert an regelmäßigem Austausch zum Thema BIRT?
- Gründung der BIRT User Group Mannheim im Anschluss an Actuate Kundentag (heute, hier, 17:30 ☺)



- Schulung: Reporting mit Eclipse BIRT
  - <http://www.oio.de/seminar/open-source/eclipse-birt-reporting-schulung.htm>
- Vortrag Jax2009: Eclipse BIRT Day: BIRT within Java Enterprise
  - <http://www.oio.de/m/konf/jax2009/BIRT-Enterprise-Architektur.pdf>
- Vortrag W-Jax2006: Reporting mit Eclipse BIRT
  - <http://www.oio.de/m/konf/wjax2006/ReportingMitEclipseBIRT-final.pdf>



# Vielen Dank für ihre Aufmerksamkeit !

Orientation in Objects GmbH

Weinheimer Str. 68  
68309 Mannheim

[www.oio.de](http://www.oio.de)  
[info@oio.de](mailto:info@oio.de)



**Fragen ?**

**Orientation in Objects GmbH**

Weinheimer Str. 68  
68309 Mannheim

[www.oio.de](http://www.oio.de)  
[info@oio.de](mailto:info@oio.de)