

classix Software GmbH

Performance Ver- besserung der BIRT Berichte

**Tipps zur Performance Verbesserung der Berichte unabhängig
von der Engine**

classix

a class of its own

1 Inhalt

2	Allgemeine Aussagen.....	2
3	Performance Verbesserung von BIRT-Berichten.....	2
4	Übergreifende Verbesserungstipps	3
5	Performance Messen	3
6	Bindings	3
7	Aggregation.....	4
8	Filtern und Sortieren.....	4
8.1	Filtern.....	4
8.2	Sortierung	4
9	Verbesserungsmöglichkeiten beim Output	4
10	Literaturverzeichnis.....	4

2 Allgemeine Aussagen

Generell liegt das Performance-Problem meistens nicht an den BIRT-Berichten selbst, sondern an den Zugriffen auf die Data-Source.

Folgende Aussagen gibt es zu xml-Data-Source:

- Die in BIRT bereits vorhandene XML Schnittstelle ist sehr langsam, die Schnelligkeit kann aber durch einen Adapter, der selbst geschrieben werden muss verbessert werden.
- Java API für XML Binding (JAXB) ist eine spezialisierte API für Java, die genutzt werden kann um effizient und schnell ein festes Schema XML Daten-File zu parsen
- Vorteil: Ungefähr 10 mal so schnell wie die ODA XML Schnittstelle, die BIRT mitliefert
- Nachteil: Allerdings muss falls sich das XML Schema ändert, die JAXB Klassen neu gebildet werden.
- Nachteil: es gibt keine UI um Data Sets zu erstellen, stattdessen müssen JAXB-Klassen innerhalb einer scripted data Source verwendet werden
- Alle obigen Aussagen gelten auch für die WEB Service ODA

Allgemein wird empfohlen nicht den Umweg über XML zu gehen, sondern direkt von BIRT auf ClassiX-Daten zuzugreifen. Allerdings bezieht sich diese Aussage auf die automatisch vorhandene XML Schnittstelle.

3 Performance Verbesserung von BIRT-Berichten

Report Name	Size (MB)	Average (in milliseconds)	Diff	Compare Report
Single Table	4.30	1,954.40		
Single Table Formatted	4.30	1,931.80	-1.16%	Single Table
Single Table Double Pages	4.35	1,996.00	2.13%	Single Table
Group By City (4 instances)	4.51	2,201.20	10.28%	Single Table Double Pages
Group By Customer (400 instances)	4.69	2,353.20	17.90%	Single Table Double Pages
Group By Customer Sorted	4.69	2,207.60	10.60%	Single Table Double Pages
Group By City Aggregates (2 per group)	4.71	2,377.60	8.01%	Group By City
Group By Customer Aggregates (2 per group)	4.89	2,542.60	8.05%	Group By Customer
Group By City Two Pass	4.81	2,366.40	7.50%	Group By City
Group By Customer Two Pass	5.00	2,397.60	1.89%	Group By Customer
Single Chart	5.92	2,607.40	33.41%	Single Table
Group By City Chart	6.25	2,888.80	31.24%	Group By City
Group By Customer Chart	21.6	5,704.80	142.43%	Group By Customer

Nach dieser Tabelle folgt:

- Die Geschwindigkeit bei der Erstellung eines Berichts hängt ab von:
 - Anzahl von Berichtsobjekten
 - Das Vorhandensein von Aggregations (Summierungen)
 - Anzahl der Gruppen
 - Sortierung der Daten
 - Das Vorhandensein von Charts
- Zeit pro Seite hängt vom Output-Format ab
- Seiten pro Sekunde hängt vom Layout ab
 - Eine Verminderung der Page-Break-Anzahl „verdoppelt“ die Performance (Master-Page → Properties → Bericht)

4 Übergreifende Verbesserungstipps

- Reduzierung der Anzahl von Berichtsobjekten → Unnötige Grid und Table Elemente sollten entfernt werden → Grid z.B. in ein Databinding mit ...+...umwandeln
- Werte Zusammenfassen, wenn es Sinn macht:
 - Vorname + Nachname
- Vermeidung von Data-Bindings, wenn diese nicht genutzt werden
- Crosstab sind gut in ihrer Performance und sollten wenn möglich genutzt werden.

5 Performance Messen

Nach „Le Birt Experts Guide“ Kapitel 18.3, aber abgewandelt:

```
<method name="initialize"><![CDATA[//importPackage(java.io);
var seconds = new Date().getTime() / 1000;
reportContext.setGlobalVariable("startSeconds",seconds);
logToDebugWindow("Starting initialize Method at: " + seconds]]></method>
```

```
    <method name="afterFactory"><![CDATA
var seconds = new Date().getTime() / 1000;
logToDebugWindow("Ending afterFactory Method at: " + seconds);
var totalTime = seconds - startSeconds
logToDebugWindow("Total time to run: " + totalTime);]]></method>
```

6 Bindings

- Seit der Version 2.1.3 gibt es Änderungen im Zusammenhang mit dem Data Set caching.
- Jedes Reportobjekt mit einem speziellen Data-Binding zwingt das Data Set sich neu auszuführen für jedes Binding.
- Bindings haben dann eine Auswirkung auf alle beinhalteten Reportobjekte (Data Bindings auf einer Tabelle haben Auswirkungen auf Objekte innerhalb einer Tabelle)
- Normalerweise reicht es aus nur ein Binding für ein Data Set zu spezifizieren
- Nur extrem komplizierte Berichte mit einem verwebten Data Set benötigten mehrere Bindings pro Data Set. In diesem Fall sollten besser Joint Data Sets verwendet werden um mehrere Bindings auf einem einzigen Data Set zu verhindern
- Es sollten keine Data Sets auf der Master Page gebunden werden.

7 Aggregation

Es gibt zwei Arten von Funktionen:

- Running: wird gemacht während die Tabelle erstellt wird
- Look-Ahead: Benötigt zweifaches durchlaufen über die Daten
 - o Es sollte seit der Version BIRT 2.2.2 nicht die "Total" Funktion verwendet werden. Es ist bekannt, dass diese Funktion schlechte Performance besonders mit Filtern hat.
 - o Tipps zur Performance-Verbesserung hier:
 - Erstellung einer gespeicherten Prozedur zur Berechnung
 - Benutzung einer separaten Query
 - Benutzung eines Datenfilters um Gesamtsummen in jede Spalte zu erhalten

8 Filtern und Sortieren

Das Filtern und Sortieren sollte von DataSource Seite geschehen, da dieser Vorgang im Memory durchgeführt wird und somit sehr lange dauert.

8.1 Filtern

- BIRT ermöglicht Filtern auf verschiedenen Ebenen, wie beispielsweise in einer Tabelle
- Man sollte wenn möglich das Filtern der Datenbank überlassen
- Darüber hinaus sollte die Größe des Result Sets reduziert werden
- Besonders wichtig ist dies im Zusammenhang mit Look-Ahead-Aggregation (siehe Punkt 7)

8.2 Sortierung

- Sobald eine Gruppierung hinzugefügt wird, sortiert BIRT automatisch das Data Set im Memory
- Es gibt keine Möglichkeit BIRT mitzuteilen, dass die Daten bereits sortiert sind.

9 Verbesserungsmöglichkeiten beim Output

PDF

Es sollte eine angemessene Seitengröße in der Master Page festgelegt werden → Setzen eines „Fixed Layouts“. Dadurch kann die dynamische Geometrie signifikant reduziert werden.

HTML

Es sollten Gruppierungsabschnitte mit vielen Einträgen vermieden werden, da dies in eine Lange Table-Of-Contents Liste resultiert.

10 Literaturverzeichnis

Birt Exchange. (2008). *Birt Exchange*. Von <http://www.birt-exchange.org/org/devshare/designing-birt-reports/371-designing-high-performance-birt-reports/> abgerufen

Mehi, D. (2010). *Le BIRT Expert's Guide to BIRT Best Practice*. Detroit, MI, USA.