

# Reporting mit Application Express jenseits von BI Publisher

Michael Pergande  
PROMATIS software GmbH  
Ettlingen

## Schlüsselworte:

Application Express, Reporting, BI Publisher, Eclipse BIRT, Apache FOP, Cocoon

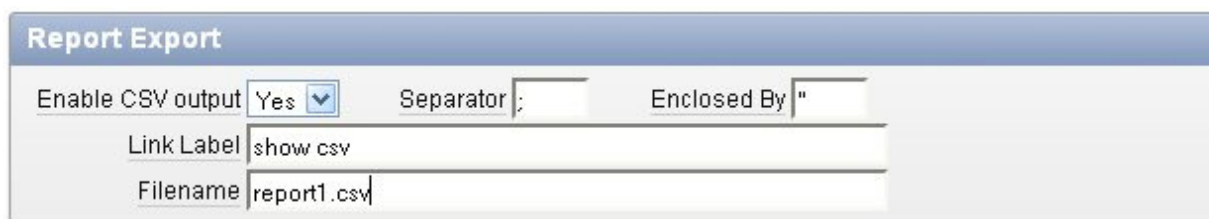
## Einleitung

Application Express (Apex) bietet eine Reihe von integrierten Möglichkeiten, Reporting-Ergebnisse in externe Formate zu exportieren. Aufgrund von funktionalen oder lizenzierungstechnischen Beschränkungen kann es aber nötig werden, weitere externe Werkzeuge für solche Aufgaben einzubinden. Nach einer kurzen Gegenüberstellung der Techniken csv-Export, Apache FOP und BI Publisher Integration wird auf das Werkzeug BIRT (Business Intelligence and Reporting Tools) aus dem Eclipse Umfeld eingegangen. Es wird aufgezeigt, wie mit diesem Werkzeug Reports definiert werden und welche Funktionen es bietet. Anschließend wird dargestellt, wie diese Technik in bestehende Application Express Umgebungen integriert werden kann. Ein Vergleich der Techniken mit ihren Vor- und Nachteilen rundet den Vortrag ab.

## Übersicht über Reporting-Optionen

In Apex gibt es eine ganze Reihe von Techniken, die bereits in das Werkzeug integriert sind, teilweise ohne Konfiguration von externen Komponenten, teilweise mit weiterer externer Konfiguration, sowie die Möglichkeit, existierende Reporting-Komponenten lose einzubinden. Diese werden im Folgenden in einer Übersicht beschrieben.

*Klassischer csv-Export:* Diese Technik kommt gänzlich ohne sonstige Konfiguration aus. Man setzt einfach in der Definition der Report Region das entsprechende Häkchen und legt fest, wie die Felder zu trennen sind sowie mit welchem Zeichen die einzelnen Zelleninhalte zu umschließen sind, und gibt den Dateinamen für die resultierende Datei an. Abbildung 1 zeigt dies. Dies führt dazu, dass unter dem Report ein Download Link erscheint (bzw. unter Download csv ausgewählt werden kann im Falle von Interactive Reports). Ist Excel richtig konfiguriert, so öffnet es sich beim Klick direkt und zeigt das Sheet an, welches mit den dortigen Mitteln weiterbearbeitet werden kann.



Report Export		
Enable CSV output	Yes	Separator ;
		Enclosed By "
Link Label	show csv	
Filename	report1.csv	

Abb. 1: Einstellungen für csv-Export

*Apache FOP/Cocoon:* Für diese Technik stellt Apex bereits eine enge Integration bereit. Zwar muss separat eine Apache FOP-Umgebung bereitgestellt werden, die dann über Apex global bekannt

gemacht werden kann. Alles Weitere kann dann aber direkt in den Report-Eigenschaften definiert werden. Sogar einfache Layoutänderungen (Farben, Schriftarten) können direkt von Apex ausgehend gesteuert werden (siehe Abbildung 2). Die einfachste Möglichkeit, eine solche Umgebung bereitzustellen, ergibt sich mittels eines Tomcat Servers mit Cocoon Deployment. Hier sind diverse Ausgabeformate, neben PDF auch Office-Formate, möglich. Insbesondere ist es sogar möglich, eigene Stylesheets zu erstellen, mit Hilfe derer beliebige Berichtslayouts realisiert werden können. Dies erhöht allerdings die Komplexität der Entwicklung beträchtlich, und es steht keine freie komfortable Entwicklungsumgebung zur Verfügung.

The screenshot displays the configuration interface for an Apex Report, specifically the 'Print Attributes' section. The breadcrumb trail at the top indicates the path: Home > Application Builder > Application 115 > Page 105 > Print Attributes. The page number 'Page 105' is also visible in the top right corner.

The interface is divided into several sections:

- Region Definition:** Shows 'Region Name: Attributes' with 'Cancel' and 'Apply Changes' buttons.
- Navigation:** A row of tabs includes 'Show All', 'Printing', 'Page Attributes', 'Page Header', 'Report Column Headings', 'Report Columns', and 'Page Footer'. 'Printing' is currently selected.
- Printing Section:**
  - 'Enable Report Printing' is set to 'No'.
  - 'Link Label' is 'Print'.
  - 'Response Header' is 'Report Settings'.
  - 'View File As' is 'Attachment'.
  - 'Output Format' is 'PDF'.
  - 'Report Layout' is 'Default Report Layout'.
  - 'Print Server Overwrite' is empty.
  - 'Print URL' is 'f?p=&APP\_ID.:105.&SESSION:FLOW\_XMLP\_OUTPUT\_R204561209258050'.
- Page Attributes Section:**
  - 'Paper Size' is 'Letter'.
  - 'Orientation' is 'Landscape'.
  - 'Units' is 'Inches'.
  - 'Width' is '11'.
  - 'Height' is '8.5'.
  - 'Border Width' is '5'.
  - 'Border Color' is selected as 'Green'.
- Page Header Section:**
  - 'Font' is 'Helvetica'.
  - 'Font Weight' is 'Normal'.
  - 'Font Size' is '12'.
  - 'Font Color' is '#000000' (Black).
  - 'Alignment' is 'center'.
  - 'Page Header' text area is empty.

Abb. 2: Apex Report Druckeinstellungen für FOP

**BI Publisher:** Der Oracle BI Publisher ist ein flexibles Werkzeug, um Berichte, Dokumente, Serienbriefe und vieles mehr für beliebige Datenquellen zu erstellen, wobei die Layoutgestaltung mit gängigen Desktop-Werkzeugen wie MS Word oder Adobe Acrobat, aber auch im Browser vorgenommen werden kann. Es wird eine konsequente Trennung von Daten, Layout und Sprache betrieben. Die Bestandteile werden erst zur Laufzeit zusammengefügt und ermöglichen so einen hohen Grad an Flexibilität mit gleichzeitig guten Verwaltungsmöglichkeiten. Analog zu Apache FOP gibt es auch für BI Publisher schon eine vorgegebene Integration. Für beide externen Werkzeuge werden global für die Apex Instanz die Daten der externen Reporting-Komponente angegeben, wie in Abbildung 3 gezeigt.

Home > Manage Instance > Instance Settings

Cancel Apply Changes

Show All Self Service Email Provisioning Storage Email Wallet Report Printing New Workspace Request Size Workspace Change Request Size

**Report Printing**

Print Server:  Standard  Advanced (requires Oracle BI Publisher)

Print Server Protocol  HTTP  HTTPS

Print Server Host Address

Print Server Port

Print Server Script

Abb. 3: Apex globale Report Server-Einstellungen

*Integration mit Eclipse BIRT:* Business Intelligence and Reporting Tools (BIRT) ist ein Open-Source-Projekt, welches Berichtswesen- und Business Intelligence-Funktionalität für Rich Clients und Web-Applikationen zur Verfügung stellt. BIRT besteht aus zwei Hauptkomponenten, einem graphischen Berichte-Editor innerhalb der Eclipse IDE, um BIRT-Berichte zu designen, und einer Laufzeitkomponente für die Erzeugung von Berichten, die in jeder Java-Umgebung eingesetzt werden kann. Weiterhin existiert der sogenannte BIRT RCP Designer, der ein eigenes Programm darstellt. Durch diesen ist keine Installation mehr von Eclipse IDE nötig. Die Komponente BIRT-Viewer ermöglicht es, Berichte in einer Web-Umgebung anzuzeigen, zum Beispiel in Form einer JEE-Web-Anwendung mit Apache Tomcat. BIRT-Bericht-Designs werden als XML-Dateien gespeichert und können auf zahlreiche Datenquellen zugreifen, darunter SQL-Datenbanken (damit insbesondere auch alle Oracle Datenbanken), JDO-Datastores, JFire-Scripting-Objekte, POJOs, XML und Web Services. Aus Sicht von Oracle Application Express stellen sich jegliche BIRT Reports als URLs, denen Parameter übergeben werden, dar. Sie können insofern als Verzweigung bzw. als URL hinter einem Button aufgerufen werden. Die Kopplung ist somit komplett lose.

### Integration mit Eclipse BIRT

Um diese Technik nutzen zu können, muss man zunächst die Installation serverseitig sowie auf den Entwicklungs-Clients durchführen. Man geht dabei im Einzelnen wie folgt vor:

- Installation von Apache Tomcat.
- Deployment der BIRT Runtime (durch einfaches Kopieren des entpackten Archivs in das webapps Verzeichnis der Apache Installation).
- Installation des DB Treibers durch Kopieren des Jar Files in das Verzeichnis WEB-INF/lib (im Falle einer Oracle DB oc4j14.jar)
- Bekanntmachen der DB Verbindungsinformationen durch Eintragen der Daten in die data-sources.xml.
- Installation des Entwicklungs-Clients wahlweise als Eclipse Plugin oder als Standalone Komponente mit einfachem Setup Programm.
- Installation DB Treiber und Hinterlegen der Verbindungsinformationen. Hier stehen dafür Konfigurationsmasken zur Verfügung.

- Erstellung eines einfachen Testreports mit dem Designer und anschließender Transfer der resultierenden rptdesign Datei auf den Server.
- Test der Report URL.
- Erstellung eines Buttons in Application Express, der die Report URL aufruft.

Mit dieser Basis beschränkt sich die weitere Entwicklungstätigkeit dann auf die Implementierung des eigentlichen Reports mit BIRT und die einfache Realisierung des parametrisierten Aufrufs in Application Express, was im Folgenden noch ausführlicher beschrieben wird. Es ergibt sich so eine Architektur des Systems wie in Abbildung 4 gezeigt.

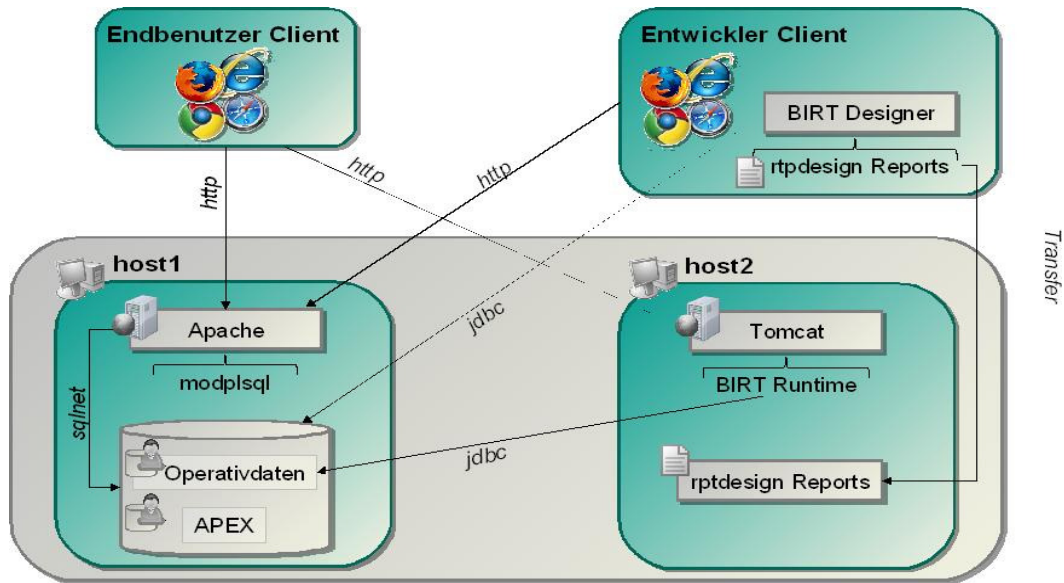


Abb. 4: Systemarchitektur BIRT Integration

Nachdem das System zur Verfügung steht, wird zunächst die Report Region wie gewohnt erstellt. Im Anschluss daran wird mit dem BIRT RCP Designer der Report erstellt und formatiert. Es kann von hier auch eine Preview des Ergebnisses mit den operativen Echtdateien erstellt werden. Abbildung 5 gibt einen Eindruck der Entwicklungsoberfläche. Das Ergebnis dieses Entwicklungsschritts ist die Reportdefinition in einer XML-Datei mit der Endung .rptdesign.

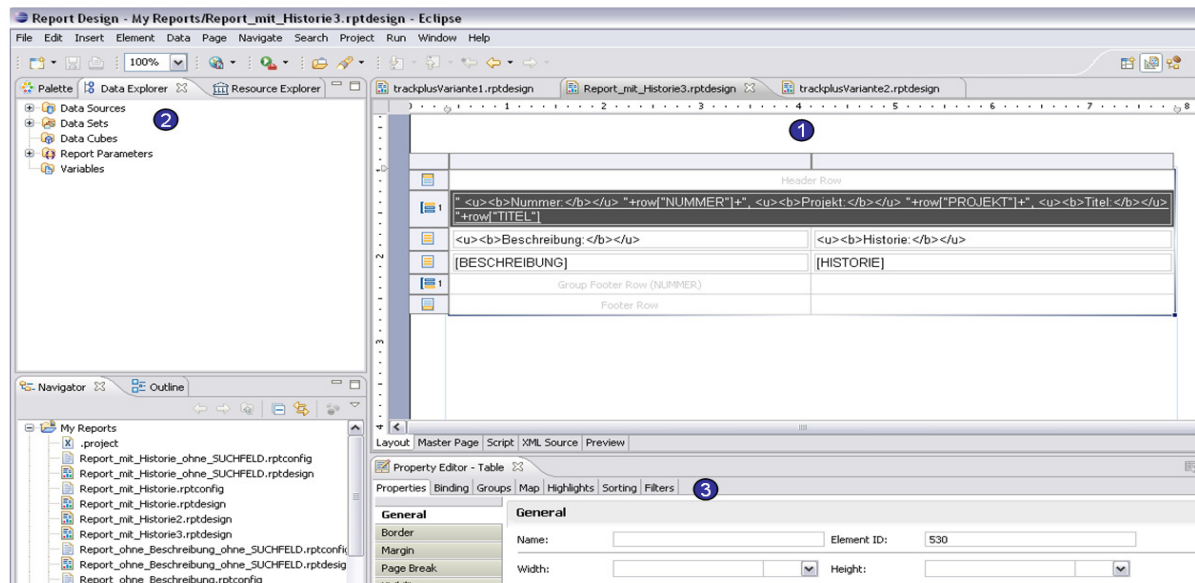


Abb. 5: BIRT Entwicklungs-Client

Diese wird nun auf den Server transferiert. Jetzt ist der Report über eine URL aufrufbar. Diese URL muss jetzt in der Apex Report Region über einen Button oder Link ebenfalls angesprochen werden. Dabei ist zu betrachten, welche Parameter benötigt und wie sie übergeben werden. Ferner muss mit reservierten Zeichen korrekt umgegangen werden. So wird z.B. das Prozent Zeichen (%) in einer URL falsch interpretiert und muss deshalb durch %25 ersetzt werden.

Noch nicht betrachtet wurde bei dieser Vorgehensweise das Thema Authentisierung. Hierfür gibt es verschiedene Ansätze. Natürlich können beide Komponenten in eine Single Sign On-Umgebung integriert werden. Damit löst sich das Problem automatisch. Aufgrund der hohen Lizenzierungskosten wird dieser Weg allerdings häufig nicht gangbar sein. Ein pragmatischer Ansatz ist der Schutz von BIRT durch einen dort angebotenen Authentisierungsmechanismus, in diesem Fall muss der Aufrufer allerdings seine Zugangsdaten zweimal (einmal für Apex, einmal für BIRT) angeben. Ein weiterer pragmatischer, kostenfreier, hinreichend sicherer und für den Endbenutzer komfortabler Ansatz ist, die Apex Session ID an BIRT zu übergeben und von dort zu überprüfen, ob diese aktuell im Apex Repository existiert.

## Vergleich und Fazit

Application Express stellt eine Vielzahl von Alternativen für die Ausgabe von Report-Ergebnissen bereit, alle mit unterschiedlichen Stärken und Einsatzbereichen. Reicht einfache Textausgabe, die nachfolgend in Office Produkten weiterbearbeitet wird, so ist der direkte csv-Export ohne Installation, Konfiguration oder Lizenzkosten unschlagbar. Sollen zwar binäre Ausgabeformate bereitgestellt werden, sind die Anforderungen an die Komplexität des Layouts der Ausgabe aber gering, so bietet sich der Einsatz von Apache FOP an, der geringe Installationsaufwände, null Lizenzkosten und weitgehende Steuerungsmöglichkeiten direkt aus der Apex Entwicklungsoberfläche bietet. Sind die Anforderungen komplexer, so bleibt nur BI Publisher oder die Einbindung einer externen Reporting-Komponente, wobei gegen BI Publisher einzig die Lizenzkosten sprechen. Im vorliegenden Artikel wurde Eclipse BIRT als eine mögliche externe Komponente vorgestellt. Der Installationsaufwand ist mit diesem Weg überschaubar, Lizenzkosten fallen keine an, beliebige Komplexität in der Ausgabe kann erreicht werden, und die Unterstützung bei der Entwicklung ist komfortabel. Die Schwierigkeit besteht darin, die Parameterübergabe sowie die Authentisierung elegant zu lösen, wofür hier Ansätze vorgestellt wurden.

**Kontaktadresse:**

**Michael Pergande**  
PROMATIS software GmbH  
Pforzheimer Str. 160  
D-76275 Ettlingen

Telefon: +49 (0) 7243-2179 0  
Fax: +49 (0) 7243-2179 99  
E-Mail: [michael.pergande@promatis.de](mailto:michael.pergande@promatis.de)  
Internet: [www.promatis.de](http://www.promatis.de)