

JasperReports

A third party reporting tool



Reporting engines: key concepts

One key thing: page formatters vs reporting engines

- Page formatters (XSL-FO): generates an output page with XSL transformation (XML data + XSLT file => XSL-FO file => rendering engine
http://en.wikipedia.org/wiki/XSL_Formatting_Objects

Examples: Apache FOP, RenderX, Antenna House, ...

Problem: XML causes parsing overhead, large memory footprint, will be superseded by future CSS paged media spec:

<http://www.rockweb.co.uk/blog/2014/06/xsl-fo-is-dead,-css-paged-media-is-prime-suspect/>

- Reporting engines: designed in the core for reporting, engines optimised for running complex multi-page reports, very efficient database binding & querying

Problem: vendor specific – lock-in

Reporting engines: a lot of choices!

- Built-in reporting: Zen Reports (XSL-FO)
- TIBCO JasperSoft: JasperReports engine (opensource & commercial)
- Eclipse BIRT (opensource only)
- Crystal Reports
- Many more ...

Most products are Java-based, but you don't need to "learn" Java to create reports!

Why JasperReports?

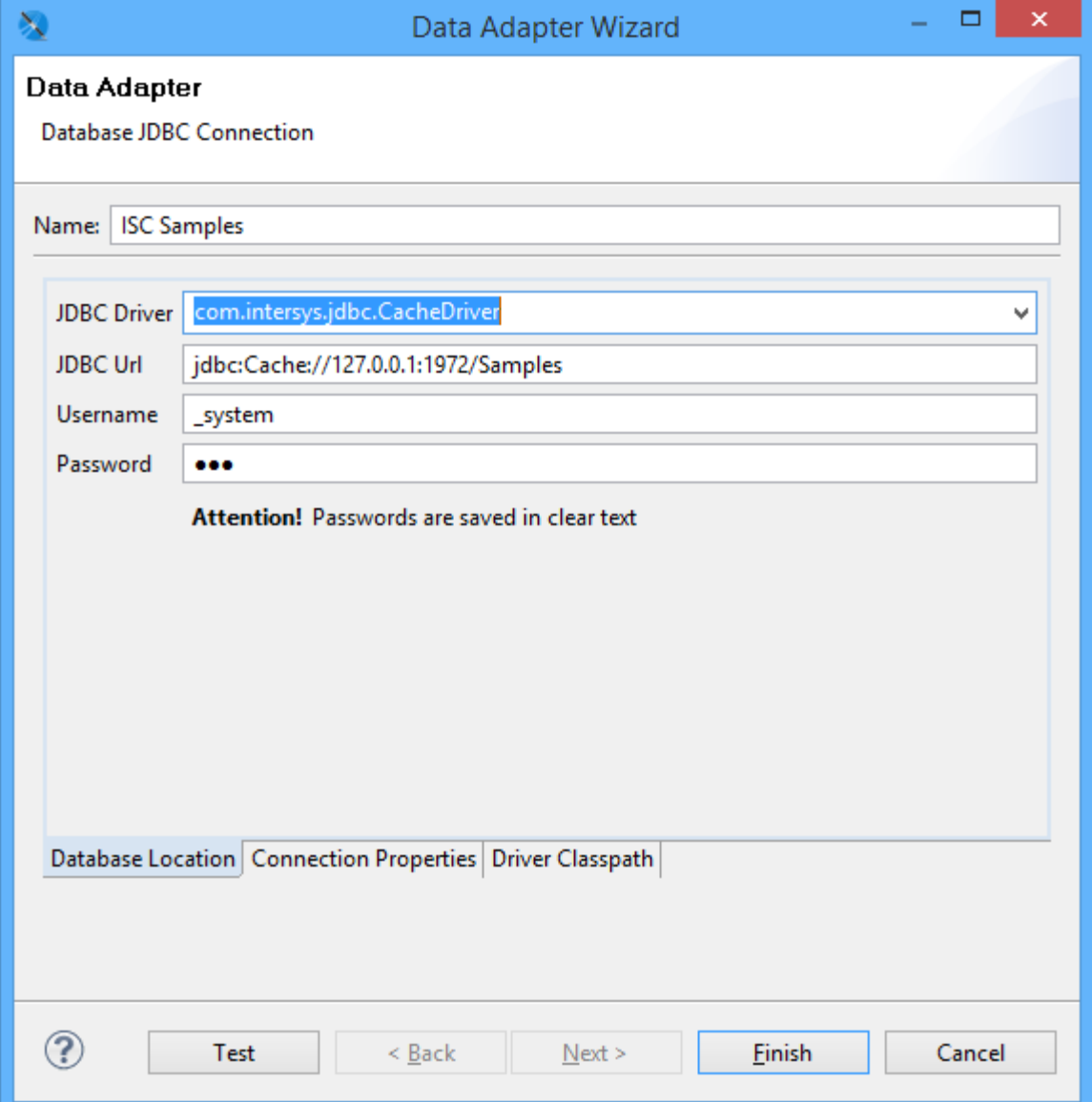
- Use of open standard Java libraries
- Long-term stability, the engine is the core of their commercial products
- Very high performance, efficient use of memory
- Easy to integrate in your own application using REST API
- Very extensive database connection possibilities
- Can output a lot of formats: PDF, XLS(X), DOC(X), HTML, ...

Tip: need real XLSX output in your application? Just create a report!

- Large interactive community, excellent support
- Platform independent: Windows, Linux, OS X

How to use with Caché?

- **Install JasperReports Server (runtime engine):**
<http://community.jaspersoft.com/project/jasperreports-server>
- **Install Jaspersoft Studio (GUI report designer):**
<http://community.jaspersoft.com/project/jaspersoft-studio>
- **Start Jaspersoft Studio and define your JDBC connection**
Don't forget to add the cachejdbc.jar in the "Driver Classpath" tab! Default location is:
C:\InterSystems\Cache\lib\cachejdbc.jar
- **Test the connection!**



The screenshot shows the "Data Adapter Wizard" dialog box, titled "Data Adapter Wizard". The main title is "Data Adapter" and the subtitle is "Database JDBC Connection".

The "Name:" field contains "ISC Samples".

The "JDBC Driver" dropdown menu is set to "com.intersys.jdbc.CacheDriver".

The "JDBC Url" field contains "jdbc:Cache://127.0.0.1:1972/Samples".

The "Username" field contains "_system".

The "Password" field contains three dots (masked).

Below the fields, there is an "Attention!" message: "Passwords are saved in clear text".

At the bottom of the dialog, there are three tabs: "Database Location", "Connection Properties", and "Driver Classpath". The "Driver Classpath" tab is currently selected.

At the very bottom, there are several buttons: a help icon (?), "Test", "< Back", "Next >", "Finish", and "Cancel".

What's next?

- Create a new report: define your Caché query for the report
- Layout your report visually, add fields, ...
- Preview report live in the Studio
- Save & publish to the JasperReports server
- Run it interactively on the server
- **Demo time!**

How to integrate in your application?

Several possibilities:

- Using \$ZF(...) using an opensource commandline tool:
<http://jasperstarter.sourceforge.net/>
Pro's: easy for testing, limited memory footprint
Con's: no official support, limited data source types
- Using a HTTP Java servlet: basic JasperReports library integration, using HTML form POST requests, servlet runs on Apache Tomcat
Pro's & con's: see commandline tool
- Recommended: use REST API of Jasperserver
https://community.jaspersoft.com/sites/default/files/docs/jasperreports-server-web-services-guide_0.pdf
Pro's: report/user management, no font/image dir hassle, officially supported API, report scheduling, ...
Con's: slightly bigger memory footprint

Deployment on a production server?

- Install Jasperserver & deploy your reports, run them using the REST API from your application
- Two options for output:
 - Preview: using Acrobat Reader on screen
 - Printing in background: use the VeryPDF PDFPrint commandline printing tool using \$ZF(...), works very stable on a multi-user/session server

Allows setting scaling, margins, paper type, printer and output bin as options on the commandline!

<http://www.verypdf.com/app/pdf-print-cmd/index.html>

Demo!



Thank you!

Ward De Backer

 [@wdbacker](https://twitter.com/wdbacker)