BIRT Best Practices

John Ward
OEM BIRT Engineer/Expert
Background

- Author of several publications on BIRT
  - Practical Data Analysis with BIRT
  - BIRT 2.6 Reporting
  - IBM Developerworks Series on Extension Points

- Comes from several years consulting on commercial and open source BIRT deployments

- So many best practices, I’ve picked the most helpful
5 Areas We Will Focus On

- Development Environment
- Report Design
- Scripting
- Data Design
- Troubleshooting and Improving Performance
Further Information


http://www.amazon.com/BIRT-Data-Analysis-Reporting-ebook/dp/B005OY7RYO/ref=sr_1_8?ie=UTF8&qid=1383539021&sr=8-8&keywords=BIRT

http://www.youtube.com/user/BIRTdeveloper

*Yes, I totally stole Amazons image for my own book =D
Tips That Got Cut Due to Time – From Our BIRT Developers YouTube Channel

• Troubleshooting Someone Elses Reports Without Access to Their Data Source
  • http://www.youtube.com/watch?v=M5ezBPQIpUw&list=TLBJn_AGkP71l_qJvm1JWwVaL1QRvf7uVv

• Using the Report Examples View
  • http://www.youtube.com/watch?v=1hc6KlRL3Us

• Using Styles and Themes
  • http://www.youtube.com/watch?v=1exDwCCEnTY&list=TLBa_cQuZRAqiZxVz97f-hI9IzoktzyL14

• Reusable Components with Libraries
  • http://www.youtube.com/watch?v=HnSjj2N8xFc&list=TLIBhUdm3SqmKNJqVM5mhx-egNF1LA0iet

• BIRT Console Logging (not on YouTube Channel)
  • http://birtworld.blogspot.com/2010/01/quick-and-dirty-logging.html
Managing a Large Development Shop – A Common Problem

- Lots of Developers
- No Standardized Naming or Organization Strategy
- Lots of different file silos on individual machines with nothing centralized except deployment environments
- No way to head off problems when changes are made
- No documentation of changes made
- “Keys to the Kingdom”
Organizing Development Environment

• Projects
  • Use different folders for each project
  • Folder names should match up what is in your deployment environment

• Resources
  • Keep in single, separate project, not in each project
    • Javascript source
    • Libraries
    • Templates
    • Styles
File Naming

• Use full, descriptive names!
  • *Camel Case*
  • *If using the commercial BIRT use spaces*
  • *If using the open source BIRT, don’t use spaces*
  • *Don’t be afraid to be verbose.*

• Consider using language of names
  • Examples
    • *Prefix report names with department or report type*
  • Make sure this is consistent
• Set up the BIRT development environment to use a single Resource project

• Use the Report Example View, and look at a report designed to be used with Open Source BIRT
Source Repository

• Storing BIRT Reports in Source repository
  • CVS

• Subversion

• Github

• Google Code

• Fastest way to get new recruits up and going
• Import CustomersOrderFinal project into workspace

• Change a report, using Chart DaVinci, stylize the chart

• Submit a report to the SVN

• Look at the Change Log to see changes, open historical versions of report

• Show Old Report and New Report side by side
Continuous Integration and Unit Testing

- Continuous Integration Environment
  - Hudson/Jenkins

- Unit Testing
  - Junit
  - Homegrown BIRT Report
    - Persistent report list as parameter
    - Used as standalone, does not integrate with CI

- Should not be testing the integrity of data. Should be testing the business logic of the report
Demo

- Use of Junit tests to track particular functionality in a report
- See local testing inside of the development environment
- Submit a bad change to the SVN
- Use of a CI environment to immediately notify when a bad change has been submitted and regression testing fails
- Revert CustomerOrderReport from history in SVN
Report Design

- Use meaningful ID’s and Bookmark names in report designs
- Avoid using Nested Elements and Styles
- Make use of re-usable resources in report designs
- Try making designs as flexible so that you manage a single report design, not multiple designs for the same purpose
- Use the Outline when designing reports to avoid adding properties to write report elements
- Don’t be afraid to use plugins to extend capabilities
Scripting

- Use meaningful Variable Names and documentation in scripts.
- Make use of the reportContext and getAppContext.
- Store Javascript in .js files and use as resources.
- You always have access to the entire Java classpath in report designs.
- If business logic is too complex, type sensitive, use Java instead.
- Make use of vars[] and global variables.
- Remember that Text element <script> tags can make use of Server Side Javascript with <Value-Of> tag.
• Open Customer Orders report with newly applied Chart DaVinci script

• Show proper commenting in report

• Remove Initialize Script

• Use shared .JS file in Resource folder

• Open Scripting Example DaVinci examples, show initialize removed, using shared resource
Data Considerations

- Use Data stored in libraries whenever possible
- Make use of Bound Query Parameters instead of Filters whenever possible
- Validate parameters to prevent SQL Injection Attacks
  - Use Function Lib plugin
  - [https://code.google.com/a/eclipselabs.org/p/birt-functions-lib/](https://code.google.com/a/eclipselabs.org/p/birt-functions-lib/)
Troubleshooting

- Check your memory settings
  - Get rid of the dreaded PermGen error
- Identify Bottlenecks in phases
  - Run Phase
    - *Data Retrieval*
    - *Layout*
  - Render Phase
- Use Benchmarking tools
- Quick and Dirty Logging
- Minimize remote data calls
  - Use Information Objects or BIRT Data Objects
  - If using Open Source, use Report Document ODA
Locations for Tools Used in Presentation

• Function Lib
  • https://code.google.com/a/eclipselabs.org/p/birt-functions-lib/

• XML Emitter

• Chart DaVinci
  • http://www.birt-exchange.org/devshare/ /designing-birt-reports/birt-chart-davinci-plugin-in-r1482
Questions?
Evaluate This Session

1. Sign-in: [www.eclipsecon.org](http://www.eclipsecon.org)

2. Select session from schedule

3. Evaluate: [+1]  [0]  [-1]