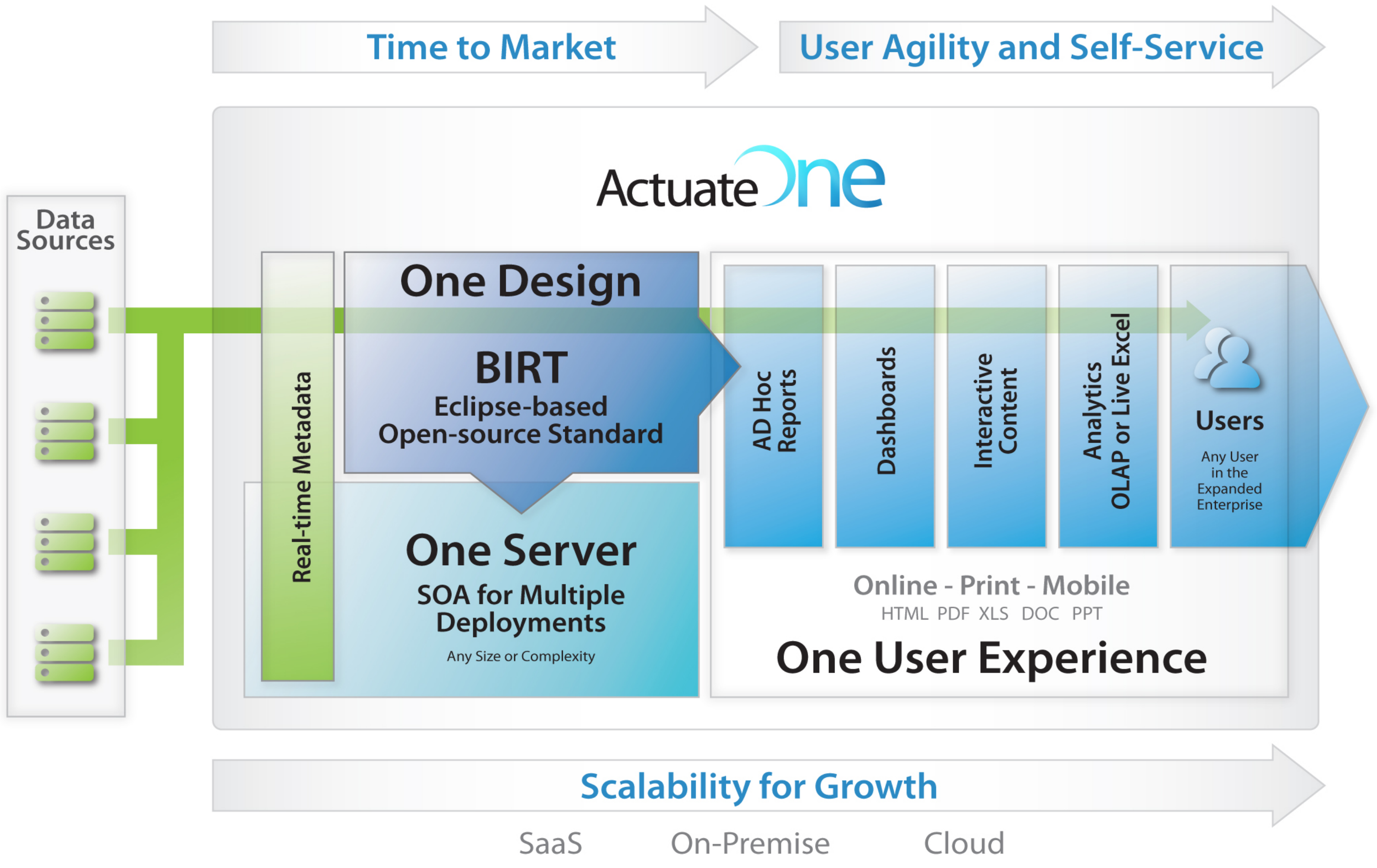
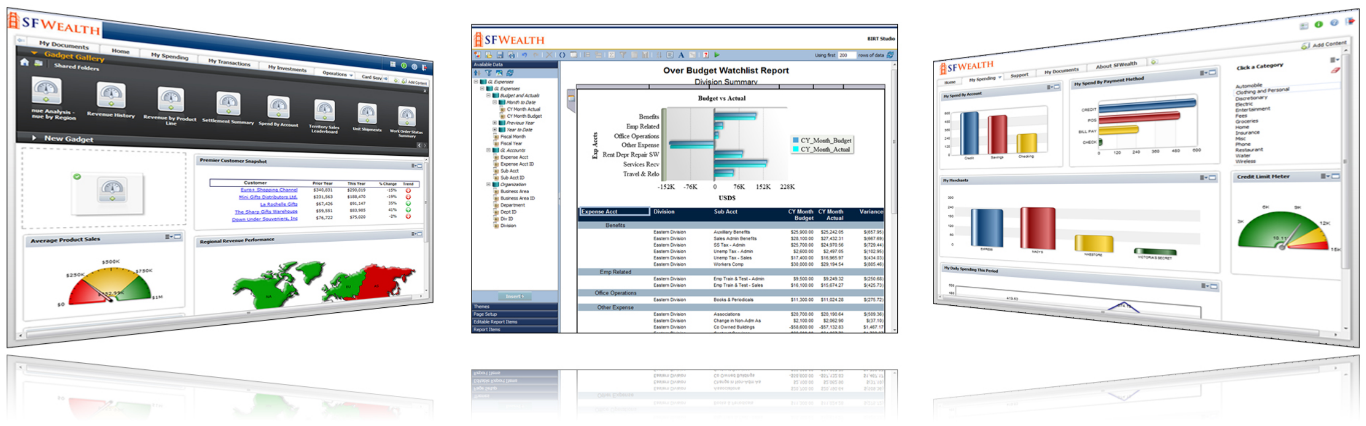
ActuateOne®: Low Risk, Rapid Development, High-Speed Delivery of BIRT Content



- Personalized	- Self-Service	- Easy Integration	- Manageable
- Interactive	- Extensible	- Open Architecture	- Predictable
- Localized	- High-Fidelity	- High-Performance	- Quality



Download 45-day Evaluation of ActuateOne at http://www.birt-exchange.com

Updated for BIRT 3.7!

BIRT 3.7 Report Design

Eclipse-Based BI and Big Data Visualization

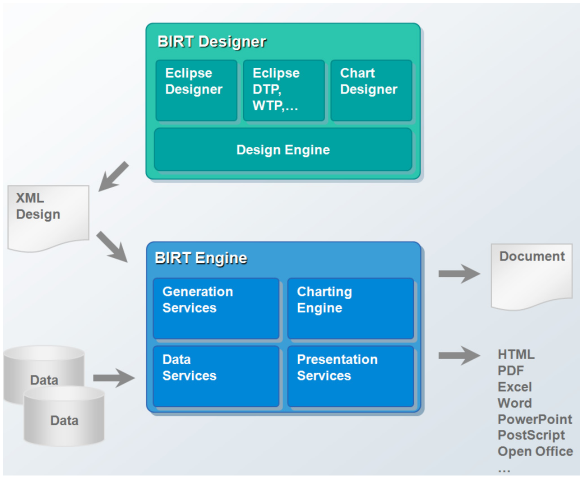
By Michael Williams

WHAT IS BIRT?

Eclipse Business Intelligence and Reporting Tools (BIRT) is an open-source, Eclipse-based reporting system that integrates with your Java EE application to produce compelling reports. BIRT is the only top-level Eclipse project focused on business intelligence. BIRT provides core reporting features such as report layout, data access, and scripting. This Refcard provides an overview of the BIRT components, focusing on a few key capabilities of the BIRT Designer, BIRT Runtime APIs, and BIRT Web Viewer. This information should be interesting to report designers as well as to developers or architects involved in integrating BIRT reports into applications.

Design and Runtime Components

BIRT has two main components: a report designer based on Eclipse and a runtime component that you can add to your application. The charting engine within BIRT can also be used by itself, allowing you to add charts to your application.



Getting BIRT

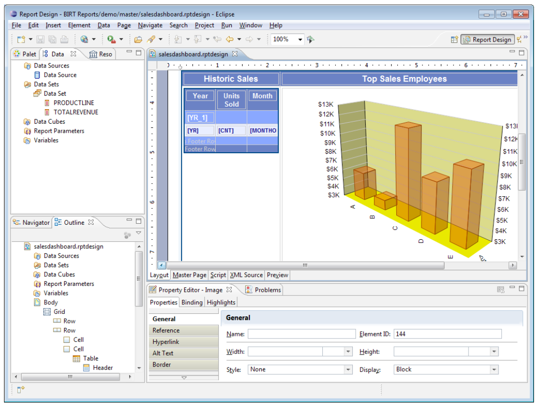
Open-source BIRT can be downloaded from http://download.eclipse.org/birt/downloads or http://www.birt-exchange.org. There are several different packages containing BIRT depending on your needs.

BIRT All-In-One Download	The fastest way to get started designing BIRT reports on Windows. Includes everything you need to start designing BIRT Reports, including the full Eclipse SDK.
BIRT Framework	This download allows you to add the BIRT plug-in to your existing Eclipse environment. (Make sure you check the dependencies and update those, too.)
BIRT POJO Runtime	Deployment components of the BIRT project including a command line example, API examples, and example web viewer.
BIRT Web Tools Integration	Contains the plug-ins required to use the BIRT Web Project Wizard and the BIRT Viewer JSP tag library.

Hot Tip

You can also get BIRT into your existing Eclipse environment through the Eclipse Update Manager (http://wiki.eclipse.org/BIRT_Update_Site_URL). Be sure to also select the Data Tools Project when using this approach.

BIRT REPORT DESIGNER



The BIRT report designer is an easy-to-use visual report development tool that meets a comprehensive range of reporting requirements. The report designer includes task-specific editors, builders, and wizards that make it easy to create reports that can be integrated into web applications. The BIRT report designer supports:

1. Component-based model for reuse
2. Ease of use features
3. Support for a wide range of reports, layouts, and formatting
4. Programmatic control
5. Data access across multiple data sources

BIRT File Types

Design File (*.rpt-design)	An XML file that contains the data connection information, report layout, and instructions. Created when making a report in the BIRT Designer.
Template File (*.rpt-template)	Ensures all reports you create start with some common elements, such as a company header or predefined styles. The starting point for a BIRT report.

Get the Most Out of BIRT

- Forums
- Technical Tips
- Sample Designs
- Plug-Ins

Join the Community at
http://www.birt-exchange.org

BIRT Exchange
sponsored by Actuate

Library File (*.rptlibrary)	Stores commonly used report elements, such as a company logo, so they are managed in one place for all reports.
Report Document (*.rptdocument)	The completed report including layout instructions and data. Can be transformed into final report output, such as HTML, PDF, and XLS.

Data Sources

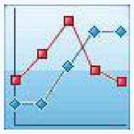
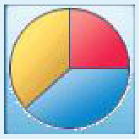
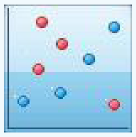
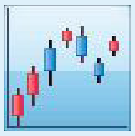
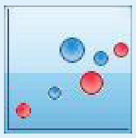
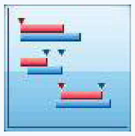
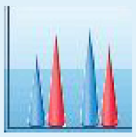
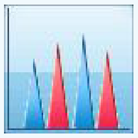
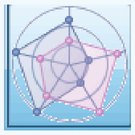
BIRT supports a variety of data sources and can be extended to support any data to which you have access. In addition to the list below, BIRT also ships with a connection to the Classic Models sample database and also includes a Joint Data Set, which allows you to join data across data sources.

Flat File Data Source	Supports tab, comma, semicolon, and pipe delimited data
JDBC Data Source	Supports connections to relational databases
Scripted Data Source	Allows you to implement custom logic, communicate with Java objects, or get access to data within your application.
Web Services Data Source	Supports connections to a web service. A wizard helps you point at a service through a WSDL and select the data
XML Data Source	Supports data from XML
Hive/Hadoop Data Source	Allows access to Hadoop data through Hive using Hive Query Language (HQL)
Additional Data Sources	BIRT has been extended by both the open source community and within commercial products allowing additional data connections to POJOs, Amazon RDS, LDAP, LinkedIn, Facebook, Excel, MongoDB, GitHub, and Spring Beans

Palette of Report Items

Label	Use to include static (or localized) text within a report. Typically for report titles, column headers, or any other report text.
Text	Use to include richly formatted text to your report, including the ability to integrate HTML formatting with your dynamic data.
Dynamic Text	Use to integrate your static text with dynamic or conditional data.
Data	Use to include data from your connection in the report.
Image	Use to include images from various embedded sources or dynamic locations.
Grid	Use to define the layout of a report. Can be nested within other grids to support complex layouts.
List	Use to display Data elements from your data source that repeat and creates a new report row for each data set row. Can contain multiple levels of grouping.
Table	Use to display repeating data elements within your report and has support for multiple columns and multiple levels of grouping.
Chart	Use to add rich interactive charting to your BIRT report.
Cross Tab	Use to display grouped and dynamic data by both the row and column level.
Aggregation	Use to build totals for tables and groups. Includes over 30 built-in functions like COUNT, SUM, MAX, MIN, AVE, RUNNINGSUM, COUNTDISTINCT, and RANK.
Additional Report Items	BIRT has been extended by both the open source community and within commercial products providing additional report items such as Google Translate item, HTML5 charts, Flash gadgets/objects, and interactive HTML buttons.

Chart Types

Bar	**Sub-Types:** Side-by-Side Stacked Percent Stacked **Dimensions:** 2D, 2D w/ Depth. and 3D	Line 	**Sub-Types:** Overlay Stacked Percent Stacked **Dimensions:** 2D and 3D
Area	**Sub-Types:** Overlay Stacked Percent Stacked **Dimensions:** 2D, 2D w. Depth, and 3D	Pie 	**Sub-Types:** Standard **Dimensions:** 2D and 2D w. Depth
Meter	**Sub-Types:** Standard Superimposed **Dimensions:** 2D	Scatter 	**Sub-Types:** Standard **Dimensions:** 2D
Stock 	**Sub-Types:** Candlestick Bar-Stick **Dimensions:** 2D	Bubble 	**Sub-Types:** Standard **Dimensions:** 2D
Difference	**Sub-Types:** Standard **Dimensions:** 2D	Gantt 	**Sub-Types:** Standard **Dimensions:** 2D
Tube	**Sub-Types:** Side-by-Side Stacked **Dimensions:** 2D, 2D w. Depth, and 3D	Cone 	**Sub-Types:** Side-by-Side Stacked Percent Stacked **Dimensions:** 2D, 2D w. Depth, and 3D
Pyramid 	**Sub-Types:** Side-by-Side Stacked Percent Stacked **Dimensions:** 2D, 2D w. Depth, and 3D	Radar 	**Sub-Types:** Standard Spider Bullseye **Dimensions:** 2D
	Additional Chart TYpes	BIRT has been extended by both the open source community and within commercial products providing additional chart types such as heat maps, segment charts, HTML 5 charts, Flash Charts, and Flash Maps.	

Hot Tip

Creating your First Report

1. Create a new Report Project from the category of Business Intelligence of Reporting Tools. Change to the Report Design perspective.
2. File -> New ->Report. Select the template called My First Report, which launches a cheat sheet containing a step-by-step tutorial assisting you with connecting to data sources, creating data sets, and laying out your report.

Localization

BIRT supports internationalization of report data including support for bidirectional text. BIRT also supports the localization of static report elements within a report allowing you to replace report labels, table headers, and chart titles with localized text. BIRT allows the use of multiple resource files with name/value pairs and a *.properties file extension. For example, a file called MyLocalizedText_de.properties can include a line that says "welcomeMessage=Willkommen". To use these files within a BIRT report:

Assign resource files to entire report	Report -> Properties -> Resources -> Resource Files
Assign individual keys to a label	Label -> Properties -> Localization -> Text key

Styles

Reports designed with the BIRT report designer can be richly formatted with styles that match your existing web application.

Built-In Styles	Built-in styles can be shared in a report library for managing style across multiple reports.
CSS style sheet	BIRT can import CSS files at design time or reference existing CSS files at run time.

Below are some examples of CSS styles:

```
.table-header {
            background : #93BE95;
            border-bottom : double;
            border-top : solid;
            border-top-width : thin;
            border-color : #483D8B;
            font-family : sans-serif;
            font-size : x-small;
            font-weight : bold;
            color : #FFFFE0;
}

.table-detail {
            background : #DFECDF;
            font-family : sans-serif;
            font-size : x-small;
            color : #426E44;
}

.table-footer {
            background : #93BE95;
            border-top : double;
            border-bottom : solid;
            border-bottom-width : thin;
            border-color : #483D8B;
            font-family : sans-serif;
            font-size : x-small;
            font-weight : bold;
            color : #FFFFE0;
}

.crosstab-detail {
            background : #DFECDF;
            font-family : sans-serif;
            font-size : x-small;
            color : #426E44;
}

.crosstab-header {
            background : #5B975B;
            font-family : sans-serif;
            font-size : small;
            font-weight : bold;
            color : #FFFFE0;
}

.crosstab-cell {
            border-top : solid;
            border-top-width : thin;
            border-bottom : solid;
            border-bottom-width : thin;
            border-left : solid;
            border-left-width : thin;
            border-right : solid;
            border-right-width : thin;
            border-color : #294429;
}
```

To see style examples, visit http://www.birt-exchange.org/org/devshare and enter keyword "style".

Customization with Expressions, Scripting and Events

BIRT includes out-of-the-box functionality that is available through drag-and-drop or by setting some properties,. BIRT also supports more advanced customizations through expressions, scripting, and events. The expression builder in BIRT allows you to do conditional report processing just about anywhere you need to instead of hard coding values. For example, the expression below will display the shipped date for orders that have already shipped; otherwise, it will display the order date.

```
if (dataSetRow["STATUS"] == "Shipped") {
  dataSetRow["SHIPPEDDATE"];
} else {
  dataSetRow["ORDERDATE"];
}
```

Scripting of a BIRT report can be done in either JavaScript or Java depending on your skill set and needs. Scripting allows you to circumvent the traditional event processing of the BIRT report. You can add scripting to report object, data source, and data element event types. Each of these event types has several events that you can overwrite.

For example, you can use scripting to navigate your Java objects and add them to a BIRT Data Set.

```
favoritesClass = new Packages.SimpleClass();
favorites = favoritesClass.readData();
…
var favrow = favorites.get(currentrow);

var Customer = favrow[0];
var Favorite = favrow[1];
var Color = favrow[2];

row["Customer"]=Customer;
row["Favorite"]=Favorite;
row["Color"]=Color;
```

Use scripting to change bar colors on a chart based on plotted data.

```
if (dph.getOrthogonalValue() < 1000) {
  fill.set(255,0,0); //red
} else if (dph.getOrthogonalValue() < 5000) {
  fill.set(255,255,0); //yellow
} else {
  fill.set(0,255,0); //green
}
```

Use scripting to add or drop a report table based on a user parameter.

```
if (params["showOrders"] == false){
reportContext.getDesignHandle().findElement("ordersTable").drop();
}
```

Or use scripting to include dynamic images that are based on the report data.

```
if (row["CREDITLIMIT"] <= 0) {
"down.jpg"
} else {
"up.jpg"
}
```

You can also use scripting within a text box using the <value-of> tag for generation time evaluation or with the <viewtime-value-of> tag for render time evaluation.

```
<value-of>
if (row["myField"] > 0) {
"positive"
} else {
"negative"
}
</value-of>

<viewtime-value-of>
vars["Group_Page"]
</viewtime-value-of>
```

Or use html <script> tags to create client-side script, like creating a function to hide a certain table that will be called by an html button.

```
<script>
function hidetable( tblbtn,tblname){
var mytable=document.getElementById(tblname);
var hide=true;
if(mytable.style.display == 'none'){
            hide=false;
}

if( hide ){
document.getElementById(tblbtn).value="+";
            mytable.style.display='none';
}else{
document.getElementById(tblbtn).value="-";
            mytable.style.display='';
}
}
</script>
```

For more scripting examples, visit http://www.birt-exchange.org/org/devshare and enter keyword "scripting".

REPORT DEPLOYMENT OPTIONS

Once you create your report design, there are several different ways to generate the report output. Obviously, you can run these reports directly from the BIRT Designer, but you can also run BIRT reports from the command line, generate BIRT reports from your Java application using the BIRT APIs, integrate and customize the example web viewer, or deploy your reports within commercial business intelligence servers.

APIs

BIRT supplies several APIs and an example Java EE application for generating and viewing reports. The major APIs are the Design Engine API(DE API), Report Engine API(RE API) and the Chart Engine API (CE API). In addition to the APIs, BIRT supports scripting using either Java or JavaScript within report designs.

Design Engine API(DE API)	Use the Design Engine API (DE API) to create a custom report designer tool, or to explore or modify BIRT report designs. The BIRT Designer uses this API. You can call this API within a BIRT script to modify the currently running report design.
Report Engine API(RE API)	Use the Report Engine API to run BIRT reports directly from Java code or to create a custom web application front end for BIRT.
Chart Engine API (CE API)	Use the Chart Engine API to create and render charts apart from BIRT

To see API examples, visit http://www.birt-exchange.org/org/devshare and enter keyword "API".

BIRT Report Engine Tasks

There are several tasks supplied by the Report Engine API that can be used to generate report output. A few key tasks are listed below.

IRunAndRenderTask	Use this task to run a report and create the output directly to one of the supported output formats. This task does not create a report document.
IRunTask	Use this task to run a report and generate a report document, which is saved to disk.
IRenderTask	Use this task to render a report document created in the IRunTask to a specific output (eg, HTML, PDF, etc.) This class renders the report based on the supplied page range, page number or all if no page is specified.
IGetParameterDefinitionTask	Use this task to create your own parameter GUI and to obtain information about parameters, including their default values.
IDataExtractionTask	Use this task to extract data from a report document. The BIRT viewer uses this class to extract report data into CSV format.

BIRT Report Engine Example

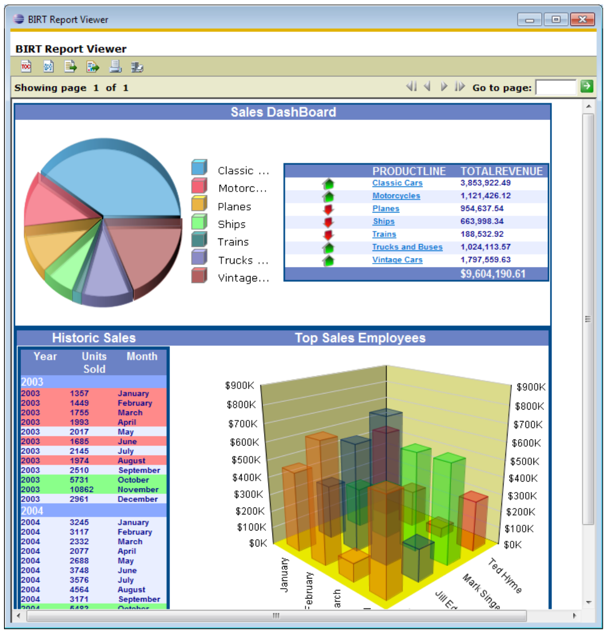
```
static void executeReport() throws EngineException
{
            IReportEngine engine=null;
            EngineConfig config = null;

try{
            // start up Platform
            config = new EngineConfig( );
            config.setLogConfig("C:\\BIRT_version\\logs", java.util.logging.Level.
FINEST);
            Platform.startup( config );

            // create new Report Engine
            IReportEngineFactory factory = (IReportEngineFactory) Platform.
createFactoryObject( IReportEngineFactory.EXTENSION_REPORT_ENGINE_FACTORY );
            engine = factory.createReportEngine( config );

            // open the report design
            IReportRunnable design = null;
            design = engine.openReportDesign("C:\\BIRT_version\\designs\\param.
rptdesign");

            // create RunandRender Task
            IRunAndRenderTask task = engine.createRunAndRenderTask(design);

            // pass necessary parameters
            task.setParameterValue("ordParam", (new Integer(10101)));
            task.validateParameters();

            // set render options including output type
            PDFRenderOption options = new PDFRenderOption();
            options.setOutputFileName("my_report.pdf");
            options.setOutputFormat("pdf");

            task.setRenderOption(options);

            // run task
```

```
            task.run();
            task.close();
            engine.destroy();
}
catch( Exception ex){
            ex.printStackTrace();
}
finally
{
        Platform.shutdown( );
}
```

Hot Tip Platform startup and shutdown should only occur at the beginning and the end of the application, respectively. Also, be sure to get the jars from the reportengine/lib directory in the runtime download added to the classpath/buildpath.

Web Viewer

The BIRT WebViewer is an example application that illustrates generating and rendering BIRT report output in a web application. This viewer demonstrates report pagination, an integrated table of contents, report export to several formats, and printing to local and server-side printers.



	PRODUCTLINE	TOTALREVENUE
▲	Classic Cars	3,853,922.49
▲	Motorcycles	1,121,426.12
▼	Planes	954,637.54
▼	Ships	663,998.34
▼	Trains	188,532.92
▲	Trucks and Buses	1,024,113.57
▲	Vintage Cars	1,797,559.63
		$9,604,190.61

Year	Units Sold	Month
2003		
2003	1357	January
2003	1449	February
2003	1755	March
2003	1993	April
2003	2017	May
2003	1685	June
2003	2145	July
2003	1974	August
2003	2510	September
2003	5731	October
2003	10862	November
2003	2961	December
2004		
2004	3245	January
2004	3117	February
2004	2332	March
2004	2077	April
2004	2688	May
2004	3748	June
2004	3576	July
2004	4564	August
2004	3171	September

The BIRT Web Viewer can be used in a variety of ways:

Stand-alone (Example Viewer)	Use as a pre-built web application, for running and viewing static reports, that doesn't require security.
Modify Viewer source	Use as a starter web application that you can customize to your fit within your environment.
RCP application	Use as a plug-in for your existing RCP application.
Integrated with existing web application	The viewer can be integrated with URLs or BIRT JSP tag library.

The BIRT Web Viewer consists of two main Servlets, the ViewerServlet and the BirtEngineServlet. These Servlets handle three mappings: (/frameset, /run, and /preview).

/frameset	Renders the report in the full AJAX viewer, complete with toolbar, navigation bar, and table of contents features. This mapping also generates an intermediate report document from the report design file to support the AJAX-based features. For example, http://localhost:8080/viewer/frameset?__report=myreport.rptdesign&parm1=value.
/run	Runs and renders the report but does not create a report document. This mapping does not supply HTML pagination, TOC or toolbar features, but does use the AJAX framework to collect parameters, support report cancelling and retrieve the report output in HTML format. For example, http://localhost:8080/viewer/run?__report=myreport.rptdesign&parm1=value.

/preview	Runs and renders the report but does not generate a report document, although an existing report document can be used; in this case, just the render operation occurs. The output from the run and render operation is sent directly to the browser. For example, http://localhost:8080/viewer/preview?__report=myreport.rptdesign&parm1=value.

Viewer URL Parameters

Below are a few of the key URL parameters available for the viewer. These parameters can be used along with the Servlet mappings, such as run, frameset, and preview, listed in the Web Viewer section.

Attribute	Description
__id	Unique identifier for the viewer.
__title	Sets the report title.
__showtitle	Determines if the report title is shown in the frameset viewer. Defaults to true. Valid values are true and false.
__toolbar	Determines if the report toolbar is shown in the frameset viewer. Defaults to true. Valid values are true and false.
__navigationbar	Determines if the navigation bar is shown in the frameset viewer. Defaults to true. Valid values are true and false.
__parameterpage	Determines if the parameter page is displayed. By default, the frameset, run, and preview mappings automatically determines if the parameter page is required. This setting overrides this behavior. Valid values are true and false.
__report	Sets the name of the report design to process. This setting can be an absolute path or relative to the working folder.
__document	Sets the name for the rptdocument. The document is created when the report engine separates run and render tasks, and is used to support features like table of contents and pagination. This setting can be an absolute path or relative to the working folder.
__format	Specifies the desired output format, such as pdf, html, doc, ppt, or xls.
__locale	Specifies the locale for the specific operation. Note that this setting overrides the default locale.
__page	Specifies page to render.
__pagerange	Specifies page range to render such as 1-4,7.
__bookmark	Specifies a bookmark in the report to load. The viewer automatically loads the appropriate page.

Viewer Web.xml settings

The BIRT Web Viewer has several configuration options. These settings can be configured by modifying the web.xml file located in the WebViewerExample/WEB-INF folder. Below are a few of the key settings available for the viewer.

Attribute	Description
BIRT_VIEWER_LOCALE	This sets the default locale for the Web Viewer.
BIRT_VIEWER_WORKING_FOLDER	This is the default location for report designs. If the report design specified in a URL parameter is relative, this path is prepended to the report name.
BIRT_VIEWER_DOCUMENT_FOLDER	If the __document parameter is not used, a report document is generated in this location. If this setting is left blank, the default value, webapp/documents, is used. If the__document URL parameter is used and the value is relative, the report document is created in the working folder.
BIRT_VIEWER_IMAGE_DIR	Specifies the default location to store temporary images generated by the report engine. If this setting is left blank, the default location of webapp/report/images is used.
BIRT_VIEWER_LOG_DIR	Specifies the default location to store report engine log files. If this setting is left blank, the default location of webapp/logs is used.
BIRT_VIEWER_LOG_LEVEL	Sets the report engine log level. Valid values are:
OFF, SEVERE, WARNING, INFO, CONFIG, FINE, FINER, and FINEST.	Sets the name of the report design to process. This setting can be an absolute path or relative to the working folder.
URL_REPORT_PATH_POLICY	Allows the developer to control what value can be used in the __report URL parameter. Valid values are all, domain, and none.
BIRT_RESOURCE_PATH	This setting specifies the resource path used by report engine. The resource path is used to search for libraries, images, and properties files used by a report. If this setting is left blank, resources are searched for in the same directory as the report.
BIRT_VIEWER_MAX_ROWS	Specifies the maximum number of rows to retrieve from a dataset.
BIRT_VIEWER_PRINT_SERVERSIDE	This setting specifies whether server side printing is supported. If set to OFF, the toolbar icon used for server side printing is removed automatically. Valid values are ON and OFF.
__pagerange	Specifies page range to render such as 1-4,7.
__bookmark	Specifies a bookmark in the report to load. The viewer automatically loads the appropriate page.

Viewer JSP Tag Library

The BIRT Web Viewer includes a set of tags to make it easy to integrate BIRT reports into browser pages. These tags are available from the BIRT Web Tools Integration download. Below are a few of the key JSP tags and a description of their usage.

Attribute	Description
BIRT_VIEWER_LOCALE	This sets the default locale for the Web Viewer.
BIRT_VIEWER_WORKING_FOLDER	This is the default location for report designs. If the report design specified in a URL parameter is relative, this path is prepended to the report name.
BIRT_VIEWER_DOCUMENT_FOLDER	If the __document parameter is not used, a report document is generated in this location. If this setting is left blank, the default value, webapp/documents, is used. If the__document URL parameter is used and the value is relative, the report document is created in the working folder.
BIRT_VIEWER_IMAGE_DIR	Specifies the default location to store temporary images generated by the report engine. If this setting is left blank, the default location of webapp/report/images is used.
BIRT_VIEWER_LOG_DIR	Specifies the default location to store report engine log files. If this setting is left blank, the default location of webapp/logs is used.

Simple Viewer JSP Tag Example

```
<%@ taglib uri="/birt.tld" prefix="birt" %>
…
<birt:viewer
id="birtViewer" pattern="preview"
reportDesign="TopNPercent.rptdesign"
height="600" width="800"
format="html"
title="My Viewer Tag"
isHostPage="false"
showTitle="true" showToolBar="true"
showNavigationBar="true"
showParameterPage="true">
</birt:viewer>
```

Interactive Viewer JavaScript Example

(for BIRT Designs deployed on BIRT iServer)

```
<html> <head>
<script type="text/javascript" language="JavaScript" src="http://
localhost:8900/iPortal/jsapi"></script>
</head>
<body onload="init();">
<div id="myDivContainer" style="border-width: 1px; border-style: solid;"></
div>
<script type="text/javascript" language="JavaScript">
var myViewer;
function init(){
actuate.load("viewer");
actuate.initialize( "http://localhost:8080/iPortal", null, null, null,
createViewer );
}
function createViewer(){
myViewer = new actuate.Viewer( "myDivContainer" );
myViewer.setReportName("/Dashboard/QuickReport.rptdesign" );
myViewer.submit();
}
</script>
</body> </html>
```

BIRT Report Output Formats

In addition to delivering paginated report content to a web browser, BIRT also supports several other output formats. These formats listed below are support by both the Report Engine API as well as the BIRT Web Viewer.

Paginated HTML output	An example web viewer is included with BIRT allowing for on-demand paginated web output.
Paginated HTML output	An example web viewer is included with BIRT allowing for on-demand paginated web output.
Microsoft Office	Word, Excel, and Powerpoint.
HTML	Suitable for creating HTML pages of report data deployable to any server.
PDF	Adobe PDF output suitable for emailing or printing.
Postscript	Output can be directed to a printer that supports postscript.
Open Document	Text, Spreadsheet, and Presentation.

BIRT Extension Points

The APIs in BIRT define extension points that let the developer add custom functionality to the BIRT framework. These extensions can be in the form of custom data sources, report items, chart types, output formats, and functions. Once implemented, these custom extensions will show along with the built-in types. For example, you can create a custom report item, like a rotated text label, that will show up in the BIRT Palette along with the existing items. Below are some of the "more common" extension points.

Data Sources	BIRT supports the Open Data Access (ODA) architecture, which means it can be extended to support custom data sources.
Functions	BIRT allows you to create custom functions that extend those available in BIRT Expressions.
Report Items	Report Items can be extended, allowing you to create your own custom report item.
Chart Types	Additional chart types can be added to BIRT as plug-ins.
Output Emitters	BIRT can be extended to include your own custom output type. For example, a simple CSV emitter exists and can be added to BIRT.

Additional BIRT Resources

Eclipse BIRT Project Site	http://www.eclipse.org/birt
BIRT Exchange Community Site	http://www.birt-exchange.org
Submitting/Searching BIRT Bugs	https://bugs.eclipse.org/bugs/enter_bug.cgi?product=BIRT
Online BIRT Documentation	http://www.birt-exchange.com/modules/documentation/

ABOUT THE AUTHOR

Michael Williams graduated from The University of Kansas with a degree in computer engineering. Currently, he works as a BIRT Evangelist at Actuate, where he has been working with BIRT for the past 4 years. One of his roles is to provide technical content for the BIRT Exchange website, in the form of, DevShare articles, monitoring the forums, and maintaining a blog. Other roles include putting together the BIRT Report newsletter, attending software conferences as a technical presence at the BIRT-Exchange booth, and the occasional speaking session.
http://www.birt-exchange.org

RECOMMENDED BOOK

BIRT: A Field Guide

More than ten million people have downloaded BIRT (Business Intelligence and Reporting Tools) from the Eclipse web site, and more than one million developers are estimated to be using BIRT. Built on the open source Eclipse platform, BIRT is a powerful report development system that provides an end-to-end solution–from creating and deploying reports to integrating report capabilities in enterprise applications.

ISBN-13: 978-1-936502-40-0
ISBN-10: 1-936502-40-2
50795
9 781936 502400
$7.95

Version 1.0