

Practical Guide to Getting Started with OpenTextTM Analytics

Use this guide to accelerate your design and deployment of dynamic data visualizations, dashboards, and analytics.

Analytics Designer provides you with core features, such as design layout, data access, and scripting, which allow you to build compelling and meaningful data visualizations that can be easily integrated into your rich-client or web applications.

This document gives you an overview of how to use Analytics Designer and OpenTextTM Information Hub (iHub) to build, design and deploy analytics content and focuses on the key capabilities of the design tool and the deployment server. It also includes tutorials to help you get your analytics application up and running quickly. If you are a report or application designer, developer, or architect, this practical guide is for you.

Table of Contents

Introduction.....	3
Take a Tour of the Analytics Designer	3
Navigator.....	4
Data Explorer	5
Data Sources and Data Objects.....	5
Resource Explorer.....	6
Layout.....	6
Palette.....	6
Chart Types	7
Property Editor	8
Outline	8
Tutorial 1: Creating a Simple Listing Design.....	9
Advanced Design Customization.....	10
Localization.....	10
Styles.....	10
Customization with Expressions, Scripting and Events.....	11
Extending Your Designs.....	11
Tutorial 2: Creating a Dashboard.....	12
Deploying Your Designs with Information Hub	17
Output Options	18

Introduction

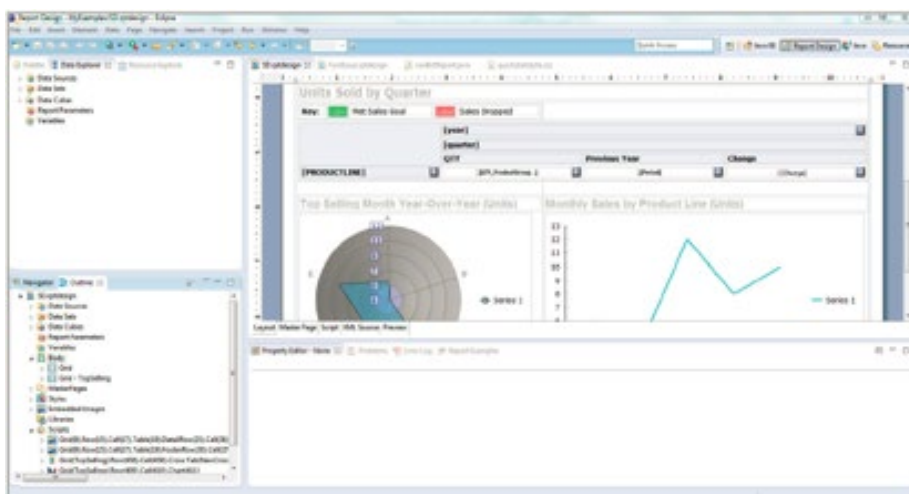
Analytics Designer is an easy-to-use, visual report development tool that meets a comprehensive range of requirements. The companion design tool to iHub includes task-specific editors, builders, and wizards that make it easy to create data visualizations that can be integrated into mobile and web applications.

With Analytics Designer (formerly known as BIRT Designer Professional), and iHub, OpenText extends the open source capabilities of BIRT and iHub. With these products, you gain additional data connectors, HTML5 charting, enhanced emitters, scheduling and delivery functionality, user-level security, interactive viewing, predictive analytics capabilities and more.

Take a Tour of the Analytics Designer

Analytics Designer provides you with:

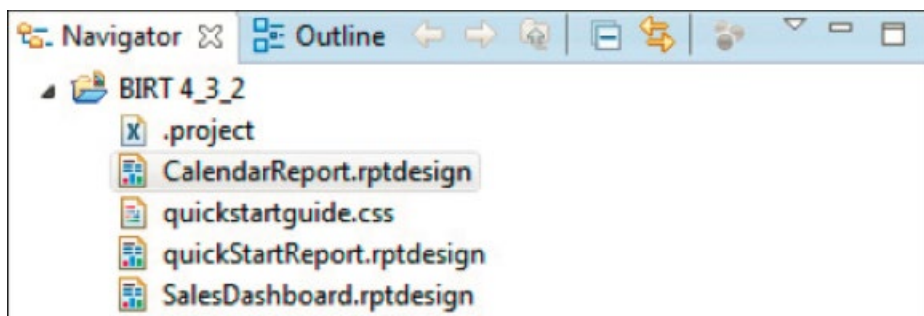
- Component-based model for reuse
- Ease of use features (drag-and-drop, WYSIWYG layout)
- Support for a wide range of reports, layouts, and formatting
- Programmatic control
- Data access across multiple data sources



Example of a Report Design in Analytics Designer

Navigator

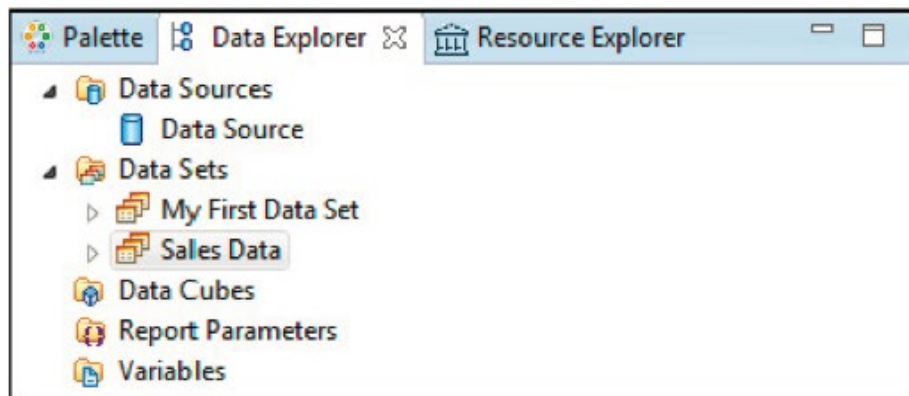
This is where you start when you open Analytics Designer for the first time. In this area, you'll see all of the projects in your workspace. When you want to create a new Report Design Project or a new file within a project, this is the place to go. Below you'll see a table of the different file types you can create.



Design File (*.rptdesign)	An XML file that contains the data connection information, design layout, and instructions. Created when making a design in Analytics Designer.
Template File (*.rpttemplate)	Ensures all reports you create start with some common elements, such as a company header or predefined styles. The starting point for a report.
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 executed design, including layout instructions and data. Can be transformed into final report output, such as HTML, PDF, and XLSX.
Dashboard File (*.dashboard)	A file created using Analytics Designer or iHub that contains one or more dashboards. In an Encyclopedia volume, a dashboard file supports individual and shared user access control to enable users adding content to their personal dashboards.
Data Object Design (*.datadesign)	An XML file created using Analytics Designer. This file type can contain a collection of data set definitions created from one or more data sources. The cubes in a data object support users creating dashboard gadgets, such as charts, cross tabs, dashboard KPIs, and tables.
Data Object Store (*.data)	A file that contains the output from running a data object design (.datadesign) file in an Encyclopedia volume. A .data file is a persistent data repository.

Data Explorer

Once you've created your project and design file in the Navigator View, this is where you'd go next. This is the area where you'll be able to create and view all of the Data Sources, Data Sets, Data Objects, Data Models, Data Cubes, and Report/Page Variables.



Data Sources

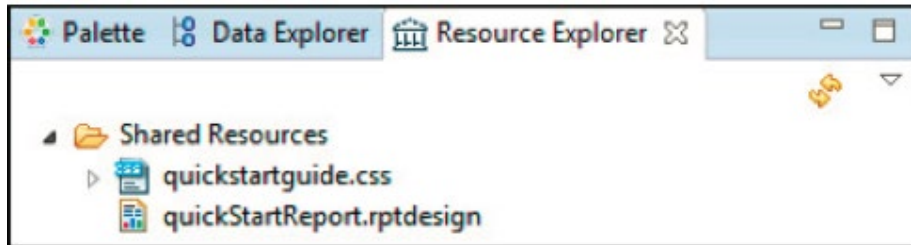
Analytics Designer makes it easy for you to access and combine multiple/disparate data sources.

The design tool natively supports multiple data sources, including big data sources (see table below) and can be extended using the ODA framework to support any data to which you have access. In addition to the list below, Analytics Designer also ships with a connection to the Classic Models sample database and includes a Joint Data Set, which allows you to join data across data sources, no matter the type.

Flat File	Supports tab, comma, semicolon, and pipe delimited data.
JDBC	Supports connections to relational databases.
Scripted	Allows you to implement custom logic, communicate with Java objects, or get access to data within your application. It can be used to access virtually any data source that is structures or contains an API.
Web Services	Supports connections to a web service. A wizard helps you point at a service through a WSDL and select the data.
XM	Supports data from XML.
Apache Hive™/Apache® Hadoop®	Allows access to Hadoop data through Hive using Hive Query Language (HQL). If you use Hive 2, please ensure that you copy the client library JAR files to the appropriate folders
POJO	Allows for easy connection to POJOs.
MongoDB®	Provides the ability for to access data contained in MongoDB.
Cassandra Scripted • Apache Cassandra™ • Datastax Cassandra Community • Datastax Cassandra Enterprise	Analytics Designer connects to and queries a Cassandra data source using the Hector API. You must copy the Hector client library JAR files to the appropriate folder. You can download the JAR files from here.
Microsoft® Excel®	Initially contributed by the community, this connector allows you to pull in data from Excel Workbooks.
Additional Data Sources	The number of data connectors has been extended by both the open source community and within commercial products allowing additional data connections to Amazon RDS, LDAP, Report Documents, Data Objects, LinkedIn, Facebook, GitHub, Spring Beans, and Big Data Analytics.

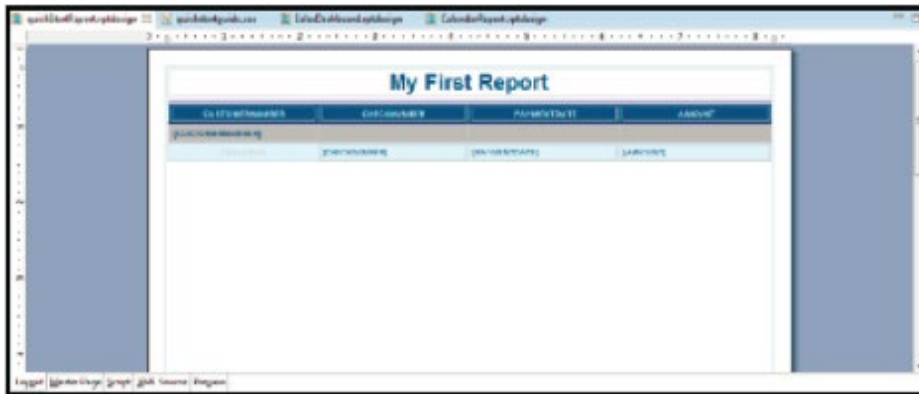
Resource Explorer

Next to the Data Explorer, you'll see the Resource Explorer. This is where you'll find all of your resources that can be used in your design, such as, libraries, images, js files, etc.



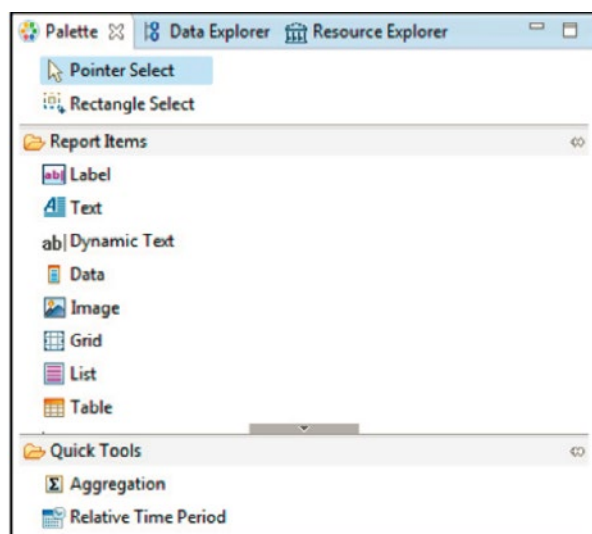
Layout

The main portion of the designer is the Layout view. This is where you create your WYSIWYG design. Below this window, you'll also see tabs that will allow you to see your Master Page, the Script on a selected element, the design's XML Source, and a Preview of your design.



Palette

Now that you've created your project, your design, and connected to your data, you're ready to start building the design layout. To insert an element into your layout, you can simply drag and drop elements from the Palette. The table on page 10 shows a listing of the available items, including a breakdown of the available chart types.



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 and interactive charting.
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 more than 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, Maps, Flash gadgets/objects, and interactive HTML buttons.

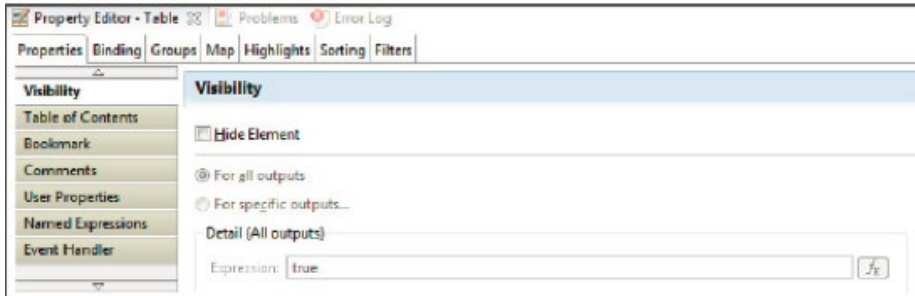
Chart Types

Below is a listing of the chart types that come with Analytics Designer. There are many combinations for each chart type, making it very likely that the designer has a chart that will work for you. Several of the charts are offered in 2D, 2D w/depth, and 3D dimensions. Most charts also have more than one sub-type, whether it's a bar chart with options of side-by-side, stacked, and percent stacked, or a stock chart with options of candlestick and bar-stick. If you don't find what you're looking for, you can create a custom chart plug-in.



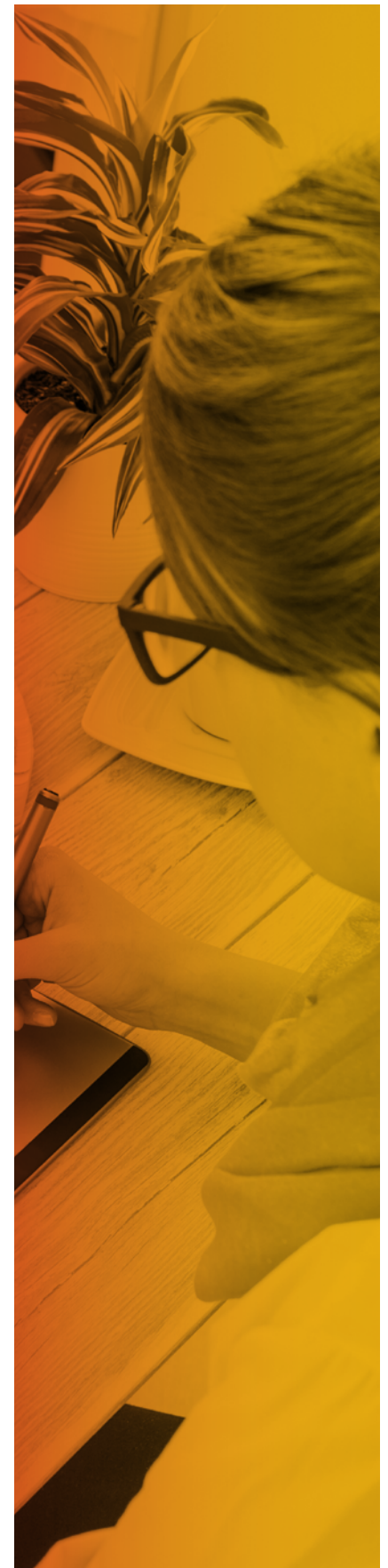
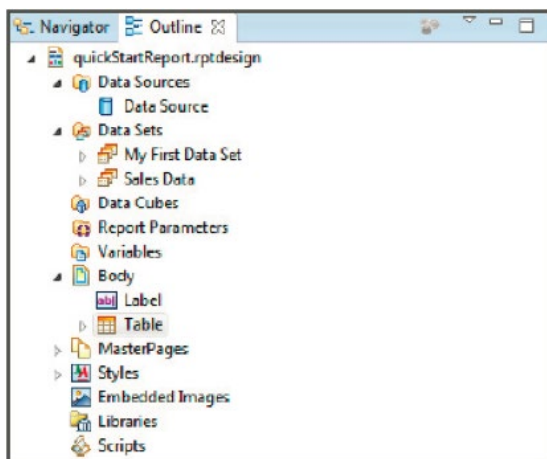
Property Editor

Just below the layout window, you'll find the property editor. This is a very powerful area, allowing you to customize your design or individual elements with just a few clicks, including filtering, grouping, sorting, mapping, highlighting, etc.



Outline

The outline shows a tree-structure view of your design. This is also where you can go to create/edit master pages, add styles, import CSS stylesheets, embed images, and locate scripts within your design.



Tutorial 1: Creating a Simple Listing Design

Now that you've taken a quick tour of the designer, let's use that knowledge to build a VERY simple design. **This tutorial assumes you've already installed Analytics Designer and are in the Report Design Perspective.**

1. Right click in the Navigator and create a new Project by selecting Report Project under Business Intelligence and Reporting Tools. In Analytics Designer, it's OpenText Analytics Designer -> BIRT Project. Enter a name for your project and hit Finish.
2. Now, right click on your new project in the Navigator and create a new report. Be sure your project is selected, give the design a name, and hit Next. Here you'll have several template options for getting started. As stated above in the BIRT Files list, you can create your own templates to add to this list. Choose the blank template and hit Finish.
3. Go to the Data Explorer, right-click on Data Sources and create a new data source. Choose the Classic Models Database, give your Data Source a name, and hit Finish. Next, right-click on Data Sets and create a new data set. Make sure your data source is selected, give your data set a name, and hit Next. Enter the following query, then hit Finish: `select * from payments` Click Preview Results, then click OK.
4. Drag a Label from the Palette into the layout window and enter My First Report. Drag your data set from the Data Explorer into your layout, below the label. Select all of the columns in the pop-up and click OK.
5. Select the table in the layout by hovering over the lower-left corner and selecting the "Table" tab. You'll notice the Property Editor now shows that it's for the table. Go to the Groups tab in the property editor and select Add. Name your group, then select CUSTOMERNUMBER from the Group On drop down. Hit OK and delete the CUSTOMERNUMBER data element from the detail row (the bottom one). Go to Run -> View Report -> In Web Viewer.

In five short steps, you've created a very simple listing table. Take some time to explore other options in the property editor, e.g., add a chart, or add a CSS style to see how quickly you can take this simple design to the next level.



Advanced Design Customization

Now that we've gone through the basics and you've created your first design, it's time to learn about how you can further customize and personalize your design to fit the needs of your application with things like localization, internationalization, styles, and scripting.

Localization

Analytics Designer supports internationalization of report data, including support for bidirectional text. It 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. The designer 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 report:

Styles

Reports can be richly formatted to adapt the look and feel of your existing web application or company color scheme.

Below are some examples of CSS styles, including a custom style title label that you can apply manually to your title label, like in the report we created above.

Assign resource files to entire report	Report -> Properties -> Resources -> Resource Files
Assign individual keys to a label	Label -> Properties -> Localization -> Text Key
Built-in styles	Built-in styles can be shared in a report library for managing style across multiple reports.
CSS style sheet	You can import CSS files at design time or reference existing CSS files at run time.

.table-header { background : #34598d; border-bottom : solid; border-top : solid; border-top-width : thin; border-color : #483D8B; font-family : sans-serif; font-size : x-small; font-weight : bold; color : #e8f5fd;}	.table-detail { background : #e8f5fd; font-family : sans-serif; font-size : x-small; color : #34598d;}	.crosstab-header { background : #34598d; font-family : sans-serif; font-size : small; font-weight : bold; color : #e8f5fd;}
.table-group-header-1 { background : silver; border-bottom : solid; border-bottom-width : thin; border-top : solid; border-top-width : thin; border-color : #483D8B; font-family : sans-serif; font-size : x-small; font-weight : bold; color : #34598d;}	.table-footer { background : #34598d; 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 : #e8f5fd;}	.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;}
	.crosstab-detail { background : #ffffff; font-family : sans-serif; font-size : x-small; color : #34598d;}	.header-label { font-family : sans-serif; font-size : 20pt; font-weight : bold; text-align : right; color : #34598d;}

If you apply this style to the report that was created in the earlier tutorial, the output will be:

Customer Name	Customer Number	Order Number	Order Date	Order Amount
John Doe	12345	67890	2023-01-01	100.00
Jane Smith	23456	78901	2023-01-02	200.00
Bob Johnson	34567	89012	2023-01-03	150.00
Alice Brown	45678	90123	2023-01-04	300.00
Charlie Davis	56789	01234	2023-01-05	250.00
Diana Evans	67890	12345	2023-01-06	180.00
Frank Green	78901	23456	2023-01-07	220.00
Grace Hill	89012	34567	2023-01-08	120.00
Henry King	90123	45678	2023-01-09	280.00
Ivy Lee	01234	56789	2023-01-10	160.00
Jack Miller	12345	67890	2023-01-11	210.00
Karen Wilson	23456	78901	2023-01-12	190.00
Leo Young	34567	89012	2023-01-13	240.00
Mia Zane	45678	90123	2023-01-14	170.00
Noah Bell	56789	01234	2023-01-15	260.00
Olivia Clark	67890	12345	2023-01-16	140.00
Peter Hall	78901	23456	2023-01-17	290.00
Quinn Scott	89012	34567	2023-01-18	110.00
Rachel Turner	90123	45678	2023-01-19	230.00
Samuel White	01234	56789	2023-01-20	130.00
Tina Black	12345	67890	2023-01-21	270.00
Uma Gray	23456	78901	2023-01-22	150.00
Victor Blue	34567	89012	2023-01-23	200.00
Wendy Red	45678	90123	2023-01-24	180.00
Xavier Yellow	56789	01234	2023-01-25	250.00
Yara Green	67890	12345	2023-01-26	160.00
Zoe Purple	78901	23456	2023-01-27	220.00

To see other style examples, visit <http://developer.actuate.com/community/devshare> and enter keyword: "style."

Customization with Expressions, Scripting and Events

Analytics Designer includes functionality that is available through drag and drop or by setting some properties, but also supports more advanced customizations through expressions, scripting, and events. The expression builder allows you to do conditional report processing just about anywhere you need to, instead of hard coding values. For example, the expression below will include the ADDRESSLINE2 field in the address if it's not null.

```
if (dataSetRow["ADDRESSLINE2"] == null) {dataSetRow["ADDRESSLINE1"] + ", " +
dataSetRow["CITY"] + ", " + dataSetRow["STATE"] + " " + dataSetRow["POSTALCODE"];}
else {dataSetRow["ADDRESSLINE1"] + " " + dataSetRow["ADDRESSLINE2"] + ", " +
dataSetRow["CITY"] + ", " + dataSetRow["STATE"] + " " + dataSetRow["POSTALCODE"]};
```

Scripting of content 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 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.

```
dataClass = new Packages.SimpleClass();myData = dataClass.readData();...var
dataRow = myData.get(currentrow);

var customerName = dataRow[0];var customerNumber = dataRow[1];var orderNumber
= dataRow[2];var orderDate=dataRow[3];var orderAmount=dataRow[4];

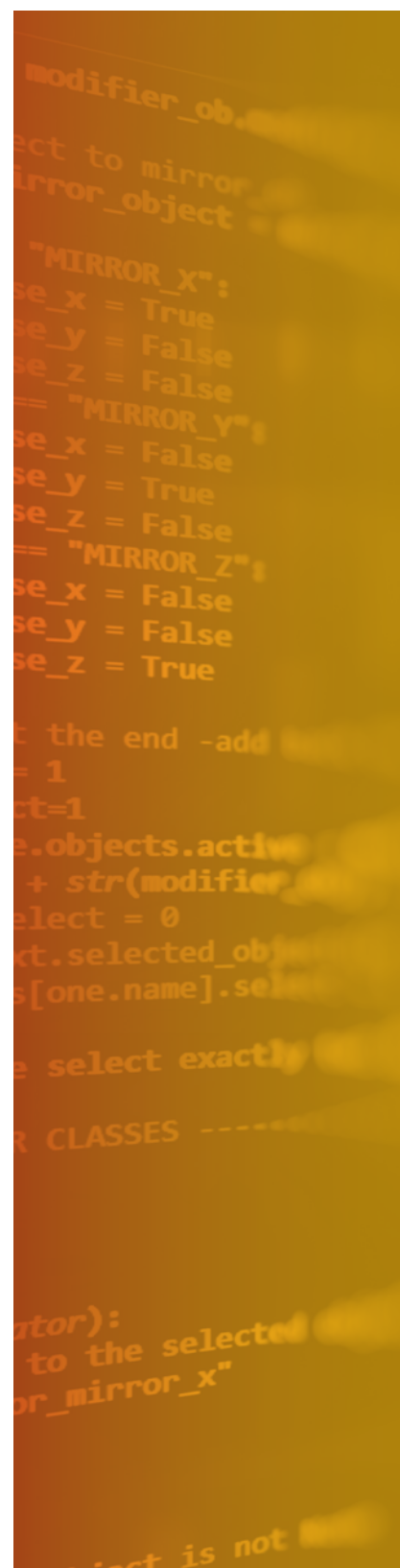
row["customerName"]=customerName;row["customerNumber"]=customerNumber;ro
w["orderNumber"]=orderNumber;row["orderDate"]=orderDate;row["orderAmount"]=o
rderAmount;
```

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

```
if (dph.getOrthogonalValue() < 3000) {fill.set(255,0,0); //red} else if (dph.getOrthogonal-
Value() < 7000) {fill.set(255,255,0); //yellow} else {fill.set(0,255,0); //green}
```

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

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



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

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

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

```
<value-of>if (row["difference"] > 0) {"gain"} else {"loss"}</value-of>
```

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

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.

```
<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>
```

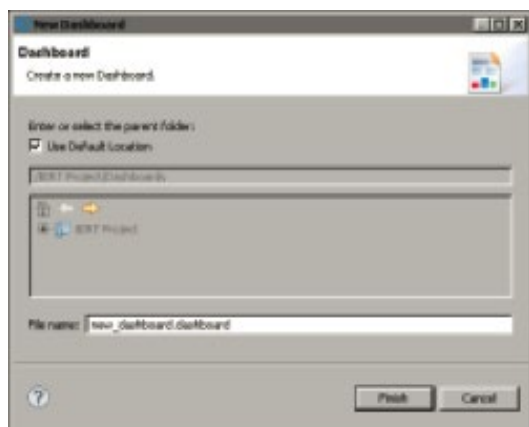
Or use html <script> tags to create client-side script, like creating a function to hide a certain table that will be called by a html button.

You can also use head.js() in the report's clientScripts method to load external JavaScript libraries for use on the client side. For an example on this, see this blog post: <http://developer.actuate.com/community/forum/index.php?blog/14/entry-496-using-external-javascript-libraries-with-birt-jquery-jvectormaps/>

For more scripting examples, visit <http://developer.actuate.com/community/devshare> and enter keyword: "scripting."

Creating a Dashboard

You create a new dashboard file using the dashboard editor in Analytics Designer. These files are stored in the project folder. If you plan to display charts verify that your BIRT data object file exists in the project.



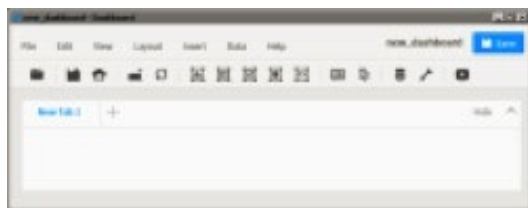
To create a new dashboard:

1. In Analytics Designer choose File New Dashboard.
2. The current project and the default folder for dashboards is selected.

If you want to use a different project or folder to store the dashboard file, remove the selection Use Default Location to access the tree view of the available folders.

If you want to change the dashboard file name, type the new name in File name.

3. Choose Finish.



4. Choose Save to save the new dashboard file to the project.

Adding a dashboard tab

A dashboard is divided into one or more pages called tabs. These tab pages enable you to organize the gadgets. For example, one tab page contains gadgets necessary to make a new customer order and another tab page can contain gadgets displaying a customer's order history. You can name each tab page to identify its contents and change the order that the tab appears in relation to other tabs.

To add a dashboard tab:

1. In the dashboard editor, choose Edit New Tab to create an empty, new tab page.
2. Choose Edit Rename Tab to change the tab name.

Choosing a dashboard layout

Dashboard layout defines how gadgets appear on a dashboard. Each gadget uses either a column or freeform layout. Gadgets in column layouts do not overlap and appear either above or below another gadget in the same column. You can place gadgets in freeform layout anywhere on the dashboard. If a freeform gadget overlaps another gadget, the user can move the gadget to the front or back of the other gadgets.

Dashboards support a one-, two-, or three-column layout in addition to a freeform layout. You can use the columns to organize gadgets on the dashboard. Dashboard columns are a percentage of the user's web browser size. If the web browser changes size, the dashboard columns are resized.

Gadgets in a resized column also resize to match the new width of the column.

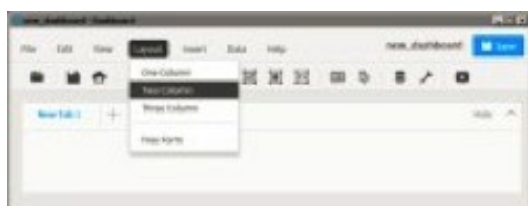
Choose the freeform layout if you need to move or resize gadgets anywhere on the dashboard. Freeform layout supports overlapping gadgets and changing the width of individual gadgets.

For example, a single-column dashboard expands to fill the width of the web browser, and the gadgets in the column are resized accordingly. Floating gadgets, such as gadgets in a freeform layout, do not change their width or location on the dashboard when the browser size changes.

For complex visualization layout, use BIRT design features available in Analytics Designer. For example, use Analytics Designer to put multiple charts into a grid element or cross tab container. You can display the finished BIRT design file on a dashboard using a report gadget.

To change a dashboard layout:

In the dashboard editor, choose Layout Two Columns



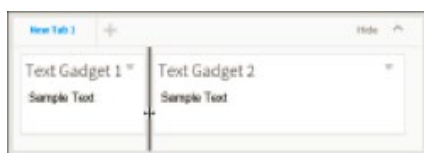
To resize a column in a dashboard:

This example begins with a dashboard that uses a two-column layout.

1. In the dashboard editor, position the pointer over the vertical space between two gadgets – the column bar appears.



2. Drag the bar to the left, to a new location



Existing gadgets are resized to fit within the new column widths.

Formatting a dashboard tab

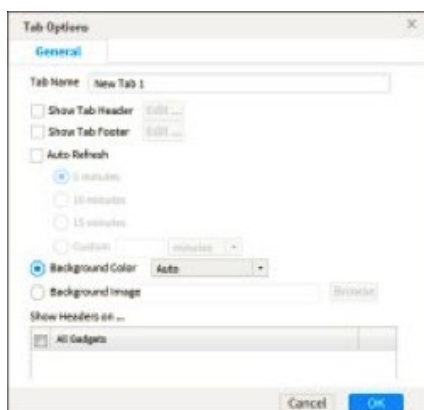
You can personalize the tab page of a dashboard with the following formatting options:

- Auto refresh, to refresh the dashboard at a selected interval
- Background color, to set the background color of the dashboard page
- Background image, to display an image as the background of the dashboard page
- Show Headers On, to show headers on selected gadgets
- Show Tab Footer, to include HTML text at the bottom of the dashboard page
- Show Tab Header, to include HTML text at the top of the dashboard page Tab
- Name, to customize the name of the dashboard tab

For example, you can add a row of HTML hyperlinks to the top of your dashboard or other HTML content with customized CSS styles inside the tab header.

Activating auto refresh sets the dashboard to refresh at the selected interval. Data and reports update at the selected interval. Set refresh settings to a speed that your BIRT iHub supports. Each refresh requests an update for all content on the dashboard tab page. Check with your BIRT iHub administrator for the supported refresh frequency.

These formatting options are available from the dashboard editor when you choose Edit Options.



Adding data objects to a dashboard

When you add a BIRT data object to a dashboard, all new gadgets added to that dashboard can use the data object. This can save time when adding multiple gadgets that use the same data object. If you add multiple data objects to a dashboard, each time you add a new gadget, the data objects you have added appear in the category Current Data Selection.

To add a data object to a dashboard:

1. In the dashboard editor, Choose Data Manage Data to select data objects to assign to the dashboard



2. In Available Data, select a data object inside the project and choose the right arrow. The selected data object appears in Selected Data



If multiple versions of a BIRT data object are available, you can select which version to use in the dashboard. You can browse the contents of a data object to verify it contains the data you require.

Importing an existing dashboard

You can add a shared dashboard file to a dashboard that you are editing. Choose Insert Dashboard From Gallery to import a dashboard file into your new dashboard. This enables you to quickly add existing dashboards as new pages in your own dashboard file.

Imported dashboards appear on your dashboard as tab pages with a share icon in the tab title. You cannot edit the content of a shared dashboard. Dashboard tabs with the shared icon link to the original dashboard file and changes to the original dashboard file appear in the imported dashboard when it is refreshed. The figure below shows two imported tabs called Customers and Orders.

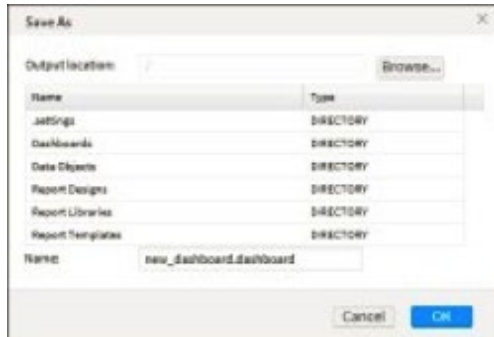


To change the content of imported tab pages, either duplicate the tab pages or edit the original dashboard file. Choose Edit Duplicate Tab to copy the selected tab into a new tab page. You can edit a copied tab page because it is no longer linked to changes in the original dashboard file.

For example, you want to include an existing dashboard file in your new dashboard but you want to also change the layout and replace a chart gadget with a table gadget. After importing the existing dashboard, you then duplicate it. A new tab page appears in your dashboard with the same content as the imported dashboard. You can edit the duplicated tab page. Finally, you do not need the imported tab page with the share icon and can delete it.

Saving a dashboard

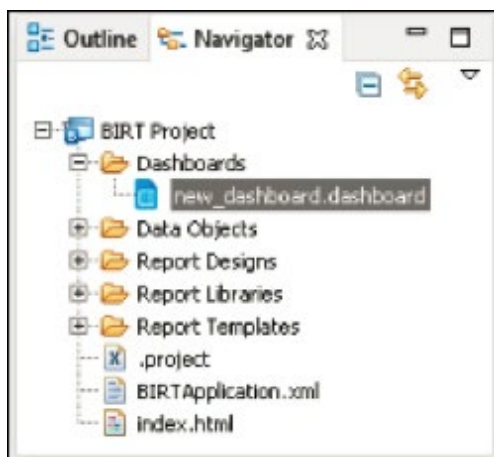
Save changes to a dashboard by choosing File Save. To save the dashboard with a new name, choose File Save as, navigate to a new location and give a new name for the dashboard file



When you save a dashboard file using Analytics Designer, you save the file to a folder in a BIRT project or a BIRT application. You can then deploy the project or application to a BIRT iHub or cloud server, such as BIRT onDemand or export the dashboard file.

Opening a dashboard file

You can open a dashboard file in a BIRT project or BIRT application using the Navigator. Navigate to the folder containing the file and double-click the dashboard file



Dashboard files require additional files such as BIRT data objects and BIRT documents. These files must exist where the dashboard file expects to find them.

If you need to send dashboard files for editing or review to someone that does not have access to your iHub server, put the dashboard and its resources into a BIRT application file. The recipient of the BIRT application file can then open the dashboard in Analytics Designer.

Deploying Your Designs with Information Hub

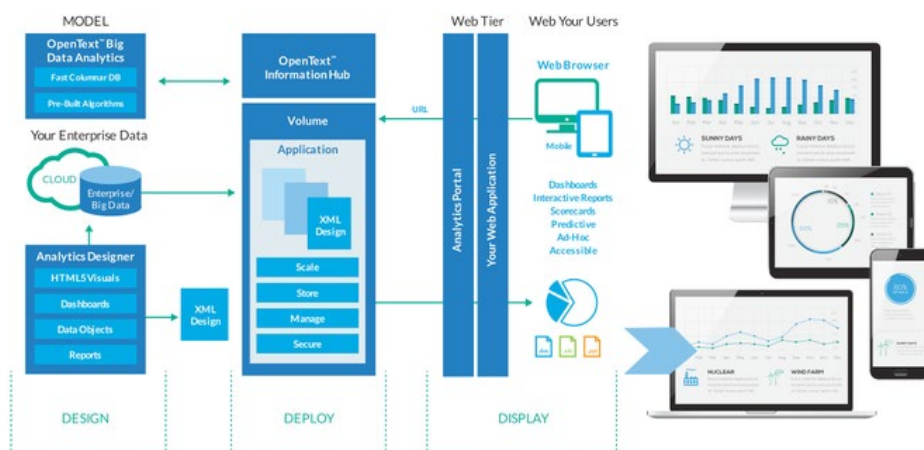
Now that you've created your designs, you're ready to deploy. There are several different ways to generate the output. You can run these designs directly from Analytics Designer, but you can also run them from the command line, generate them from your Java application using the APIs, integrate or deploy to an iHub system.

iHub

iHub is a server platform that supports large numbers of end users and a high volume of content that needs to be generated automatically and on demand. This server environment is dedicated to deploying, monitoring, managing, and administering content built with Analytics Designer while enabling end user functionality, such as interactive viewing, analytics, self-service dashboards, and web-based ad hoc content creation. iHub also ships with a customizable portal that serves as an entry point for end users to access, generate and view content.

iHub supports both Microsoft® Windows® and Linux® environments. All of its services are made available via Web Services and JavaScript APIs for seamless integration. Users can modify and interact with iHub-generated content without any help from IT. With its own repository, version control, and scheduling capabilities, iHub lets developers and end users run reports on a time or event basis at a larger scale than the other options. When reports are ready to be viewed, iHub can notify individuals with a web page link where they can view and/or download the reports in multiple formats. The addition of security capabilities for defining users or integrating with existing security servers ensures privacy of reports and end user protection.

iHub scales out to handle a large volume of reports and users. The clustering and fail-over capabilities make it an ideal solution for your most important applications. You can choose to use some or all of the functionality built into iHub, including interactive viewing and dashboards; and instead of building components yourself, you can use iHub's automated features. It can also connect to data sources across multiple servers and deliver output. It is easier to deploy and makes it simpler for developers to work in a shared environment.



JavaScript API Example (for Designs deployed to iHub)

```
<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("myReport.rptdesign" ); myViewer.submit(); }</script> </body> </html>
```

Output Options

In addition to delivering paginated report content to a web browser, the iHub platform (Analytics designer and iHub) also supports several other output formats. These formats listed below are supported by both the Report Engine API as well as the BIRT Web Viewer.

Paginated HTML output	A standard web viewer ships with iHub allowing for on-demand paginated web output.
Microsoft® Office	Microsoft® Word, Excel, and Microsoft® 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.

www.opentext.com/contact