Advanced Eclipse BIRT Report Customization

Virgil Dodson

Evangelist, BIRT Exchange

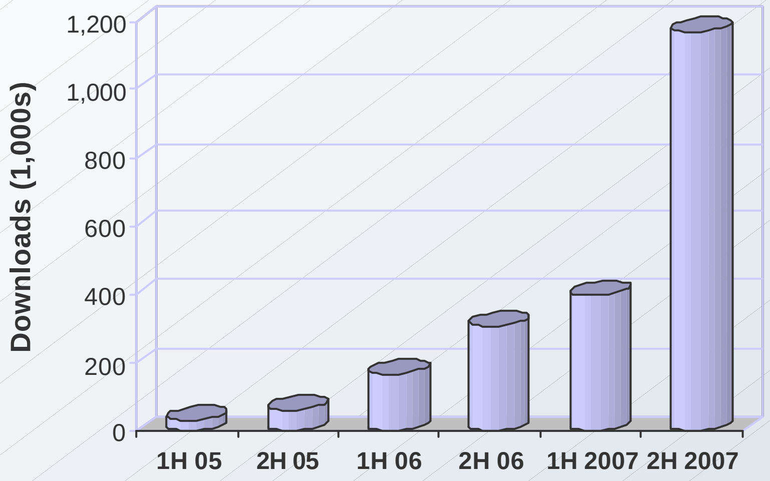
July 3, 2008

Java Forum Stuttgart

Open Source Reporting: Eclipse BIRT

- **B**usiness **I**ntelligence and **R**eporting **T**ools
- Open source initiative as part of the Eclipse Foundation
- Founded, organized and led by Actuate
- Project launched in October, 2004

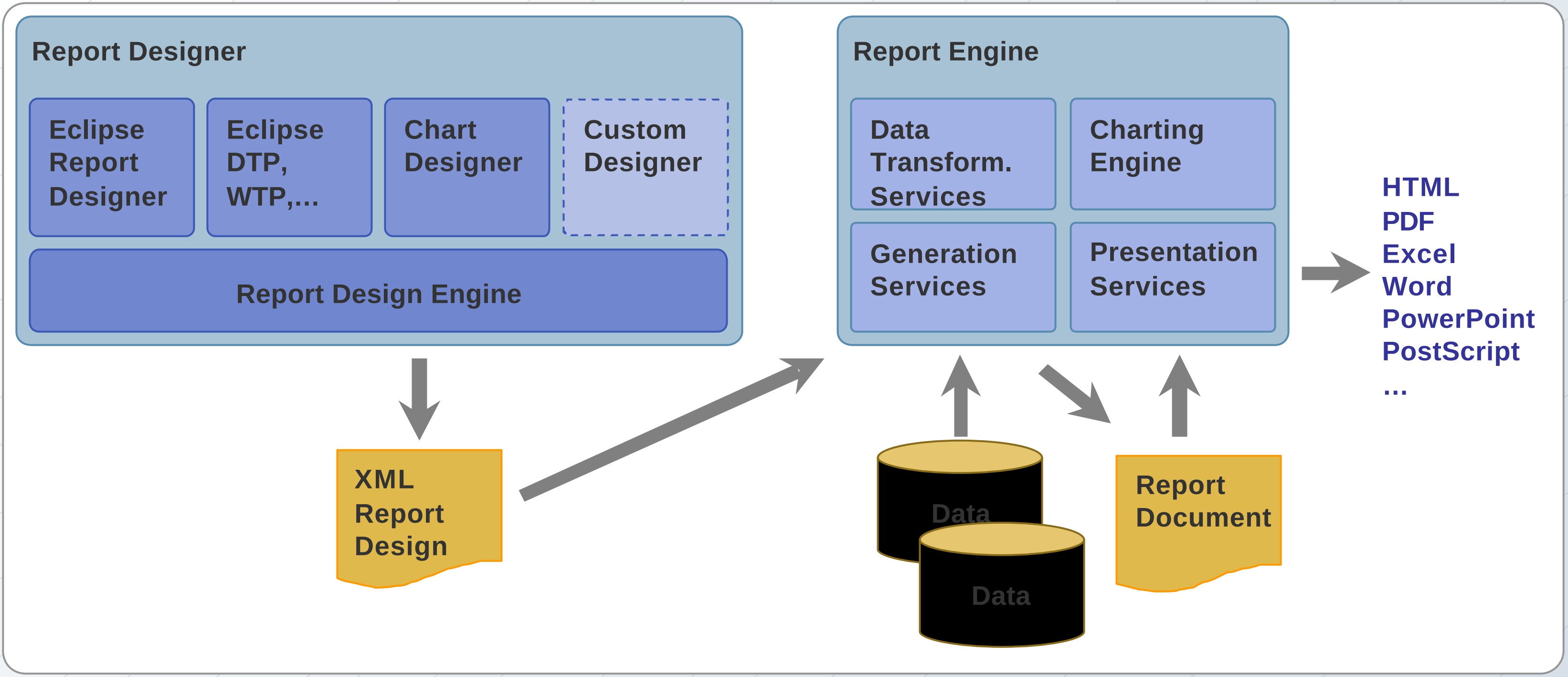
Over 3 Million Downloads as of April, 2007



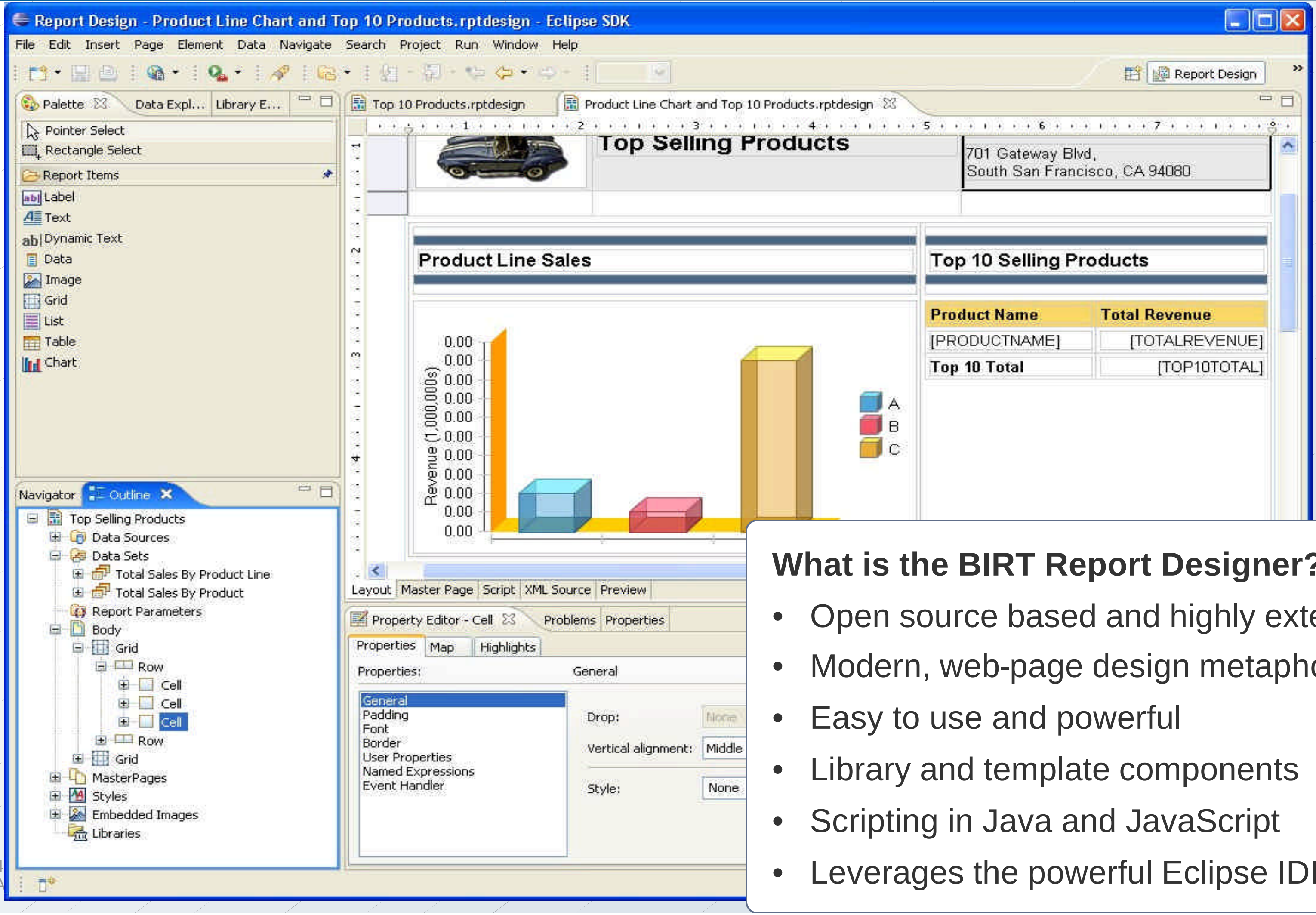
Project Goals

- Next generation reporting technology
- Eclipse-based Report Designer
- Web-centric design metaphor
- Open Source with rapid adoption
- Standards based & highly extensible
- Open XML design format
- Build community and ecosystem

High Level BIRT Architecture



BIRT Report Designer



Product Name	Total Revenue
[PRODUCTNAME]	[TOTALREVENUE]
Top 10 Total	[TOP10TOTAL]

Properties: General

Drop: None

Vertical alignment: Middle

Style: None

What is the BIRT Report Designer?

- Open source based and highly extensible
- Modern, web-page design metaphor
- Easy to use and powerful
- Library and template components
- Scripting in Java and JavaScript
- Leverages the powerful Eclipse IDE

Agenda

- Report Scripting
- Libraries and Templates
- I18N and Localization

Report Scripting

What is Scripting?

- Custom code to control various aspects of report creation
- Used as custom event handlers
- Java or JavaScript

Why use scripting?

- Maximum Flexibility
- Data is rarely is the perfect format
- Business rules are rarely an exact science
- Use for Exception handling

What can be done with Scripting?

- Create custom data sets
  - Data from Java Objects
  - Computed columns
  - Fix/combine data – parts of a field (reg exp, substr), remove MS Access image headers
- Modify report based on parameters or data
  - Show/hide/resize/drop controls (tables, columns, charts)
  - Conditionally change labels and images
- Highlight chart elements based on data
- Instantiate/Iterate global variables
  - Custom counts/sums

Choosing between Java and JavaScript

- Advantages of Scripting with **JavaScript**
  - Easier for single event
  - Simpler language construct
  - Looser typing
  - Less strict language rules
  - Available to RCP applications
- Advantages of Scripting with **Java**
  - Can use Java editor
  - Easier to find and view scripts
  - Access to the integrated debugger

Mix and Match! – You can use both Java and JavaScript Events… The JavaScript event overrides the Java event if the same event exist twice.

Events Overview

- Events are triggered throughout report execution
- Understanding the Event Order is important and depends on:
  - Engine Task Processes
  - BIRT Processing Phases
  - Event Types

Engine Task Processes

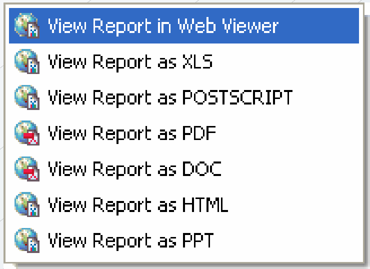
- Report Engine can be used in different ways
  - 3 tasks related to report execution and rendering

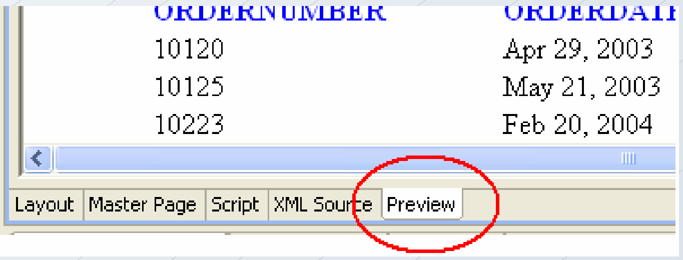
    RunTask
    RenderTask
    RunAndRenderTask
  - Using RunTask and then RenderTask means multiple processes to generate and view a report.
  - RunAndRenderTask happens in single process so event firing order is different

Engine Task Processes (cont.)

- Engine Tasks used with the Example Web Viewer
  - ‘**frameset**’ mapping uses RunTask and RenderTask… export from Viewer uses RenderTask
  - ‘**run**’, ‘**preview**’ mapping uses RunAndRender Task

- Engine Tasks used with the BIRT Designer
  - Web Viewer Preview uses RunTask and then RenderTask





ORDERNUMBER	ORDERDATI
10120	Apr 29, 2003
10125	May 21, 2003
10223	Feb 20, 2004

  - Preview tab, plus rest of Preview icons use RunAndRenderTask

BIRT Processing Phases

- BIRT report processing happens in three phases
  - Preparation Phase
  - Generation Phase
  - Presentation Phase

Preparation Phase

- Report items are prepared for execution

	Preparation Phase	Generation Phase	Presentation Phase
RunTask	X		
RenderTask			
RunAndRenderTask	X		

Generation Phase

- creates the individual instances of report items
- connects to data sources
- executes data sets
- processes data needed for the report

	Preparation Phase	Generation Phase	Presentation Phase
RunTask	X	**X**	
RenderTask			
RunAndRenderTask	X	**X**	

Presentation Phase

- Selects the proper emitter
- Produces the desired output

	Preparation Phase	Generation Phase	Presentation Phase
RunTask	X	X	
RenderTask			**X**
RunAndRenderTask	X	X	**X**

BIRT Event Types

- Parameter events
- ReportDesign events
- Data Source/Set events
- ReportItem events

- Each event type has one or more events that will fire during report processing.

Parameter Events

- validate()
  - Used for extra validation or modifying parameter value
  - First event triggered for reports with parameters
    - *After user enters parameter value*
    - *Before rest of report events run*
  - Only available in JavaScript
  - Expects true or false returned
    - *true – process as normal*
    - *false – throw parameter not set exception*

Report Design Events

- Fired for all reports
- initialize()
  - once for each RunTask, RenderTask, or RunAndRenderTask
- beforeFactory()
  - Once; after Preparation Phase and before Generation Phase
- afterFactory()
  - Once; after Generation Phase
- beforeRender()
  - Once for each render task, before Presentation Phase
- afterRender()
  - Once for each render task, after Presentation Phase

Data Source/Set Events

- All data sources and data sets have a common set of event handlers
- A scripted **Data Source** has two additional event handlers
- A scripted **Data Set** has three additional event handlers
- Data source/set events are fired prior to being used on a data bound item.
- If the data is not used on a report, these events will not fire
- Not advisable to write event handlers that rely on the data set event firing order.

Data Source Events

- beforeOpen()
- open() – only for Scripted Data Sources
- afterOpen()
- beforeClose()
- close() - only for Scripted Data Sources
- afterClose()

Data Set Events

- beforeOpen()
- open() – only for Scripted Data Sets
- afterOpen()
- fetch() – required for Scripted Data Sets
- onFetch() – as each row of data is retrieved
- beforeClose()
- close() - only for Scripted Data Sources
- afterClose()

ReportItem Events

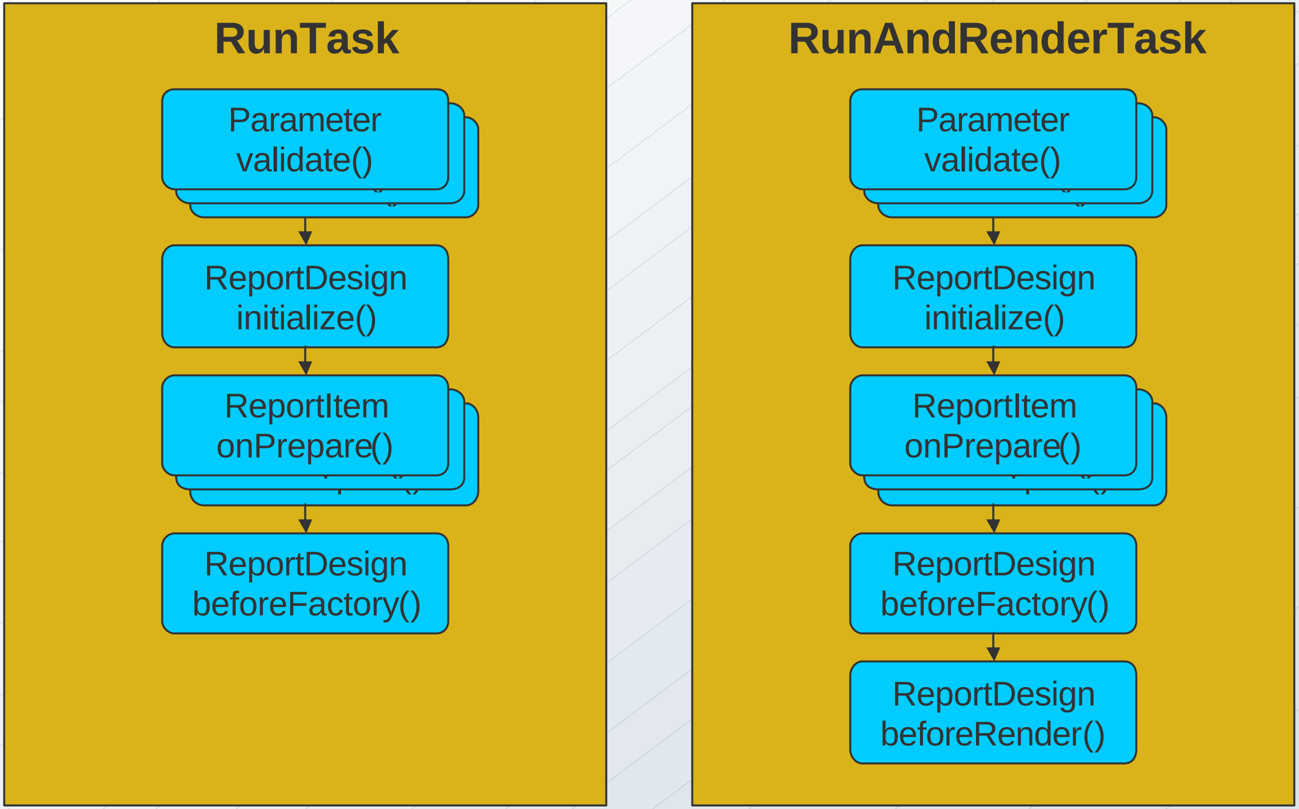
- Triggered for most report items
- onPrepare()
  - Fired at beginning of Preparation Phase before data binding.
  - Can be used to change the design of an item prior to creating instances of each item
- onCreate()
  - Fired during Generation Phase as item is being created
  - Can be used to change individual instance of item
- onRender()
  - Fired during Presentation Phase
  - Useful for operations regarding the output format
- onPageBreak()
  - Fires for all report items on a page when the break occurs

Preparation Phase

- Report items are prepared for execution
  - Parameter validation
  - Initialization
  - Report element preparation

	Preparation Phase	Generation Phase	Presentation Phase
RunTask	X		
RenderTask			
RunAndRenderTask	X		

Preparation Phase


RunTask
Parameter validate()
ReportDesign initialize()
ReportItem onPrepare()
ReportDesign beforeFactory()
RunAndRenderTask
Parameter validate()
ReportDesign initialize()
ReportItem onPrepare()
ReportDesign beforeFactory()
ReportDesign beforeRender()

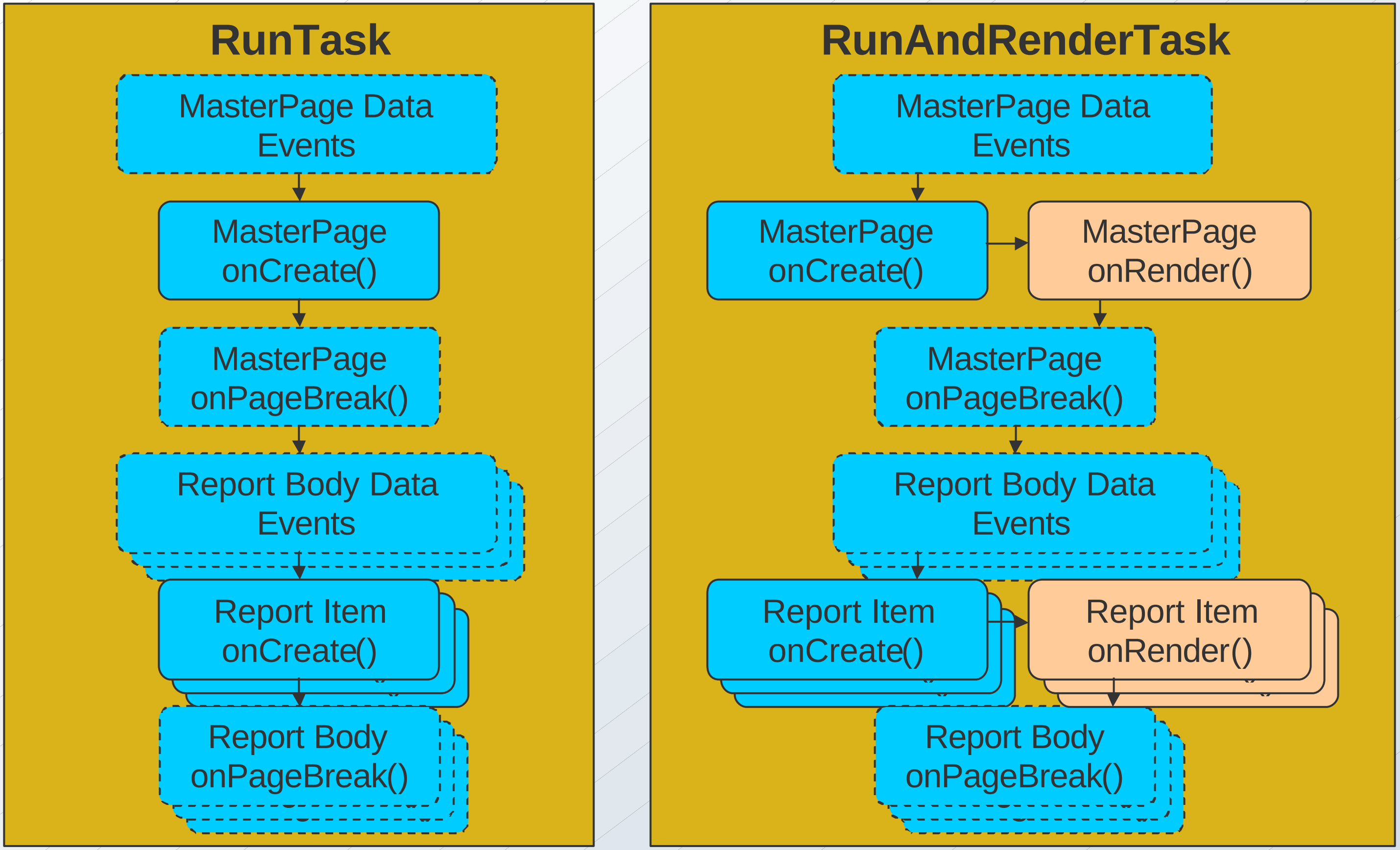
Generation Phase

- Connecting to Data Sources
- Executing Data Sets and Data Cubes
- Data Binding Evaluation
- Creation of Report Items

- MasterPage content first…
- …then Top-to-Bottom, Left-to-Right

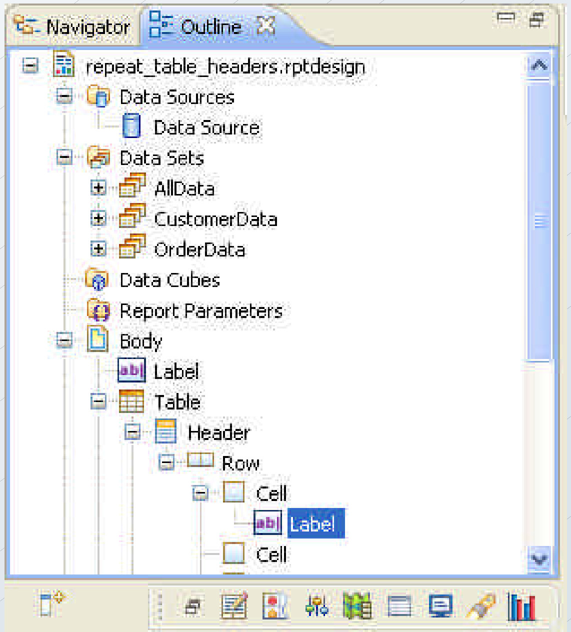
	Preparation Phase	Generation Phase	Presentation Phase
RunTask	X	**X**	
RenderTask			
RunAndRenderTask	X	**X**	

Generation Phase



Report Element Processing

- Processing is Iterative
- Nested Elements are processed before moving to the next element

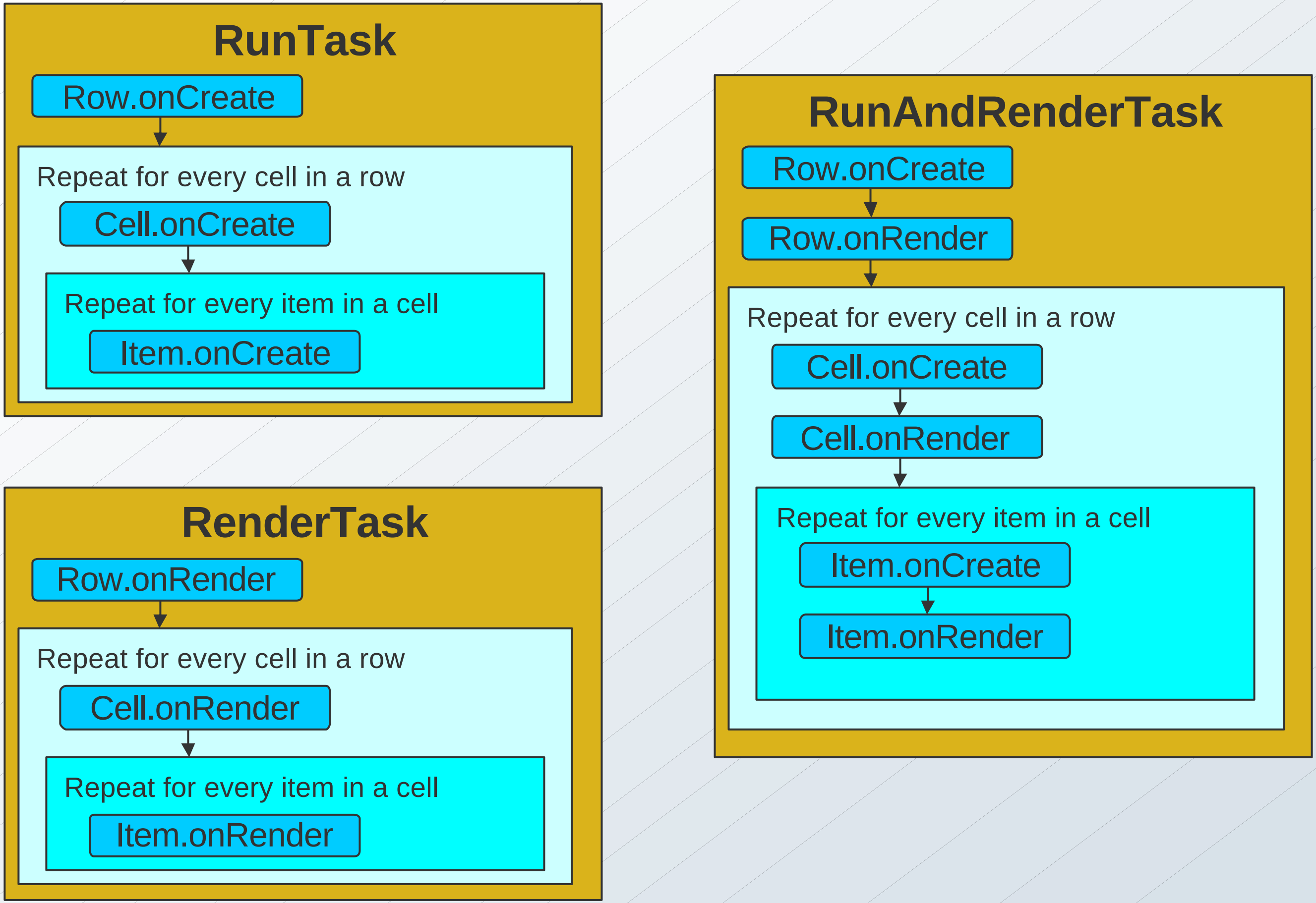


```
repeat_table_headers.rptdesign
  Data Sources
    Data Source
  Data Sets
    AllData
    CustomerData
    OrderData
  Data Cubes
  Report Parameters
  Body
    Label
    Table
      Header
        Row
          Cell
            Label
          Cell
```

onPageBreak Event

- Can be set on most elements
- Triggered in Generation Phase for RunAndRenderTask
- Triggered in Presentation Phase for RenderTask
- Only fired for output that supports pagination
- onPageBreak event fires just prior to the onCreate event for the first master page element on the next page

Row Event Order Processing



RunTask

Row.onCreate

Repeat for every cell in a row

Cell.onCreate

Repeat for every item in a cell

Item.onCreate

RenderTask

Row.onRender

Repeat for every cell in a row

Cell.onRender

Repeat for every item in a cell

Item.onRender

RunAndRenderTask

Row.onCreate

Row.onRender

Repeat for every cell in a row

Cell.onCreate

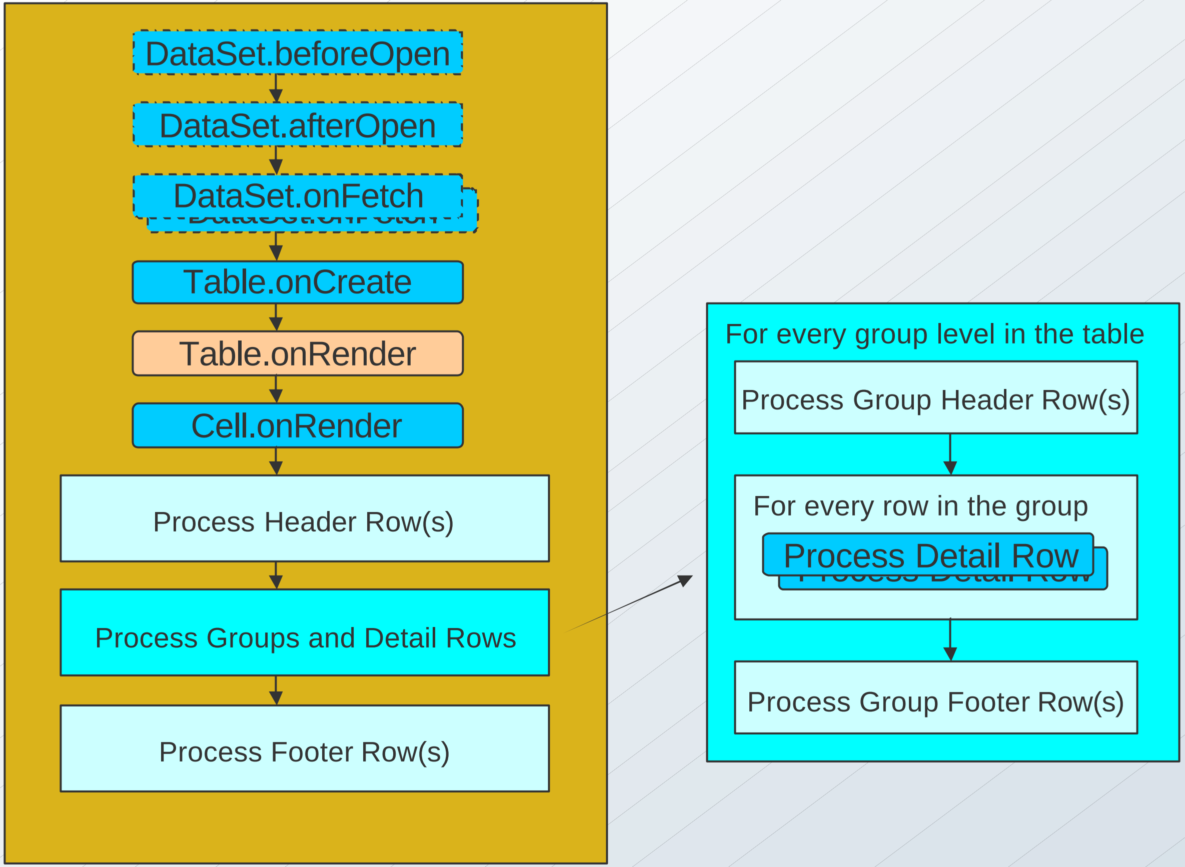
Cell.onRender

Repeat for every item in a cell

Item.onCreate

Item.onRender

Table or List Order Processing



DataSet.beforeOpen

DataSet.afterOpen

DataSet.onFetch

Table.onCreate

Table.onRender

Cell.onRender

Process Header Row(s)

Process Groups and Detail Rows

Process Footer Row(s)

For every group level in the table

Process Group Header Row(s)

For every row in the group

Process Detail Row

Process Group Footer Row(s)

Completion of Generation Phase

- Data Sources beforeClose() and afterClose() events are triggered
- afterFactory() event is triggered (RunTask Only)
- afterRender() event is triggered (RunAndRenderTask only)

Presentation Phase

- Selects the proper emitter
- Produces the desired output
- onRender events triggered for all report items
- Initialize() is triggered RenderTask only

	Preparation Phase	Generation Phase	Presentation Phase
RunTask	X	X	
RenderTask			**X**
RunAndRenderTask	X	X	**X**

BIRT Scripting in Action!

Libraries and Templates

What are Libraries?

- “Container” for reusable report components
- Library can have:
  - Data Sources
  - Data Sets
  - Visual Report Items
  - Styles
  - Master Pages
- Library is “Dynamic”
- Can use multiple libraries in a report

What are Templates?

- Structure for a “standard” report layout
- Starting point for report designs
- Simple to Complex layouts
- Template can have:
  - Data Sources
  - Data Sets
  - Visual Report Items
  - Master Pages
- Template is a Report Design in any stage of completion
- Use Templates in BIRT Designer and BusinessReports
- Standard set of templates included in Designer

Difference between Libraries and Templates

- Report Designs KEEP their link to Libraries
- Report Designs LOSE their link to Templates

- Changes in Libraries automatically promote changes to Designs
- Changes in Templates only effect new Designs

- Templates ARE Report Designs
- Libraries ARE NOT Report Designs

- Templates CAN use Library elements
- Libraries CANNOT use a Template

When to use Templates and Libraries?

- More than a few reports
- Logos and headers might change
- Separate report development tasks
- Complex data structures
- Desire consistent layout or look and feel
- Use same components repeatedly

Benefits of using Libraries and Templates

- Save time
- Reduce errors
- Reduce complexity
- Enforce standards
- More responsive to reporting requests

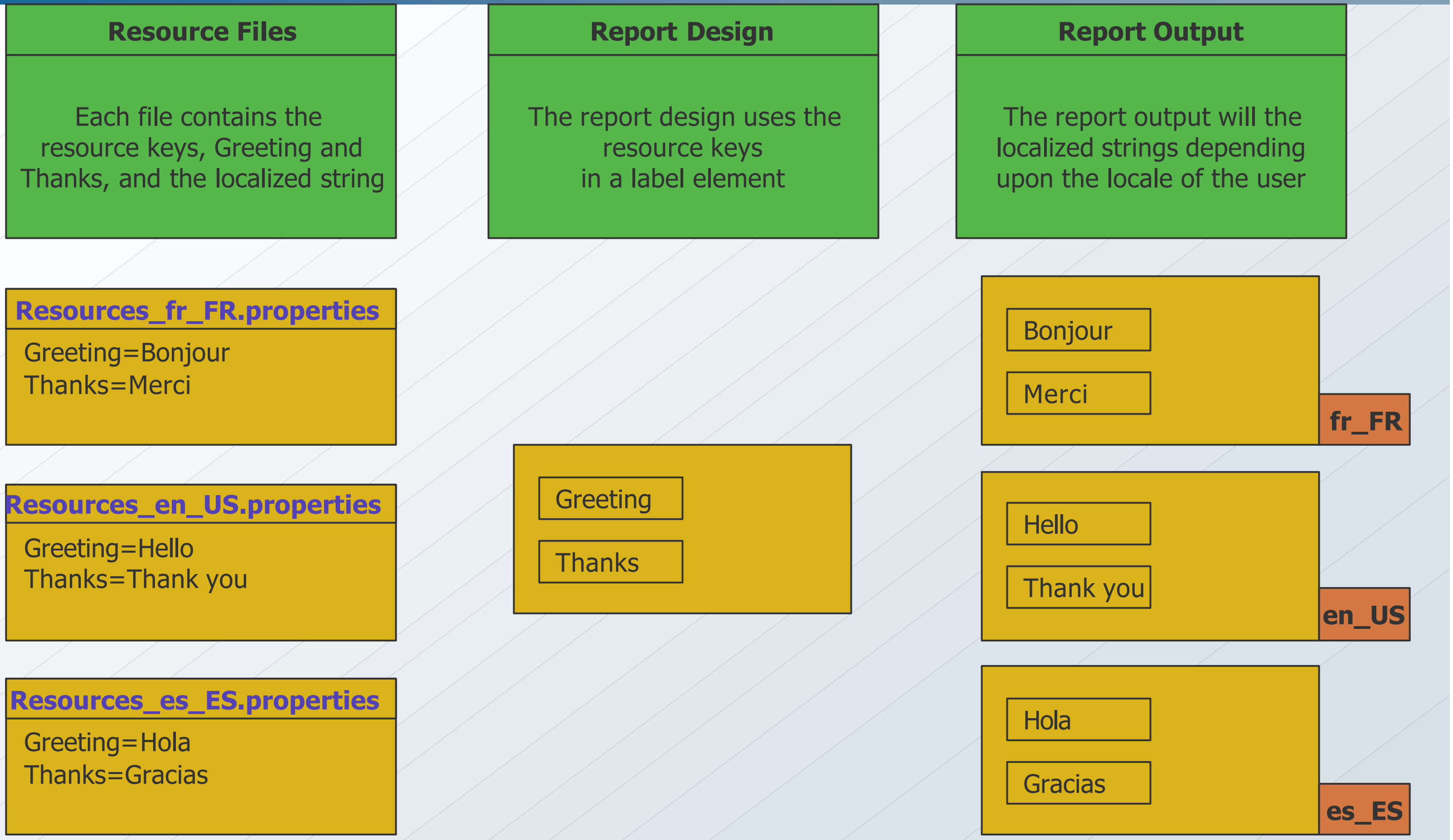
BIRT Libraries in Action!

I18N and Localization

Introduction

- BIRT report designs and templates can be localized
- You can localize static text in:
  - Labels
  - Text element
  - Chart elements
  - Parameters
- Localization also applies to date/time, currency and numeric fomat.

Localization Process



Resource Files

Each file contains the resource keys, Greeting and Thanks, and the localized string

Resources_fr_FR.properties

Greeting=Bonjour
Thanks=Merci

Resources_en_US.properties

Greeting=Hello
Thanks=Thank you

Resources_es_ES.properties

Greeting=Hola
Thanks=Gracias

Report Design

The report design uses the resource keys in a label element

Report Output

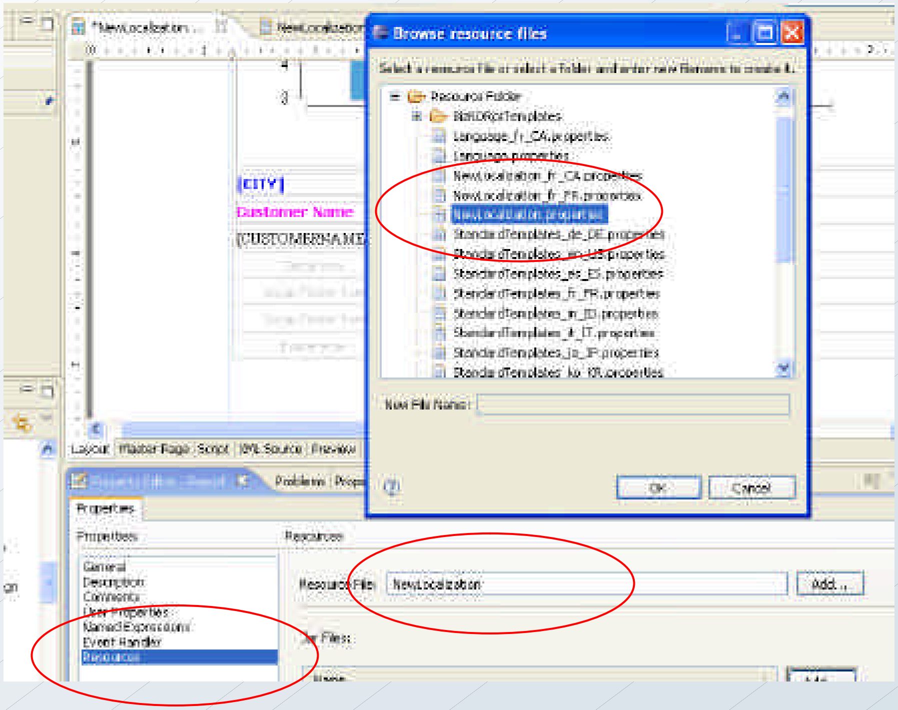
The report output will the localized strings depending upon the locale of the user

Resource Files

- Text file with **.properties** extension
  - e.g. MyResources.properties
- Default resource file contains keys and values
  - (key=value)
- Create one resource file for each supported language.
  - <filename>_<ISO 639 language code>_<ISO 3166 region code>.properties
  - e.g. MyResources_fr_FR.properties
- For each language resource file, provide the translation for each key
- You can create the files externally or in BIRT Designer.
- The files must be located in the Resources folder

Using a Resource File

- To use a resource file, you need to add its reference to the report design.



Browse resource files

Select a resource file or select a folder and enter new filename to create it.

- Resource Folder
  - BIRTDRptTemplates
  - Language_fr_CA.properties
  - Language.properties
  - NewLocalization_fr_CA.properties
  - NewLocalization_fr_FR.properties
  - NewLocalization.properties
  - StandardTemplates_de_DE.properties
  - StandardTemplates_en_US.properties
  - StandardTemplates_es_ES.properties
  - StandardTemplates_fr_FR.properties
  - StandardTemplates_in_ID.properties
  - StandardTemplates_it_IT.properties
  - StandardTemplates_ja_JP.properties
  - StandardTemplates_ko_KR.properties

New File Name:

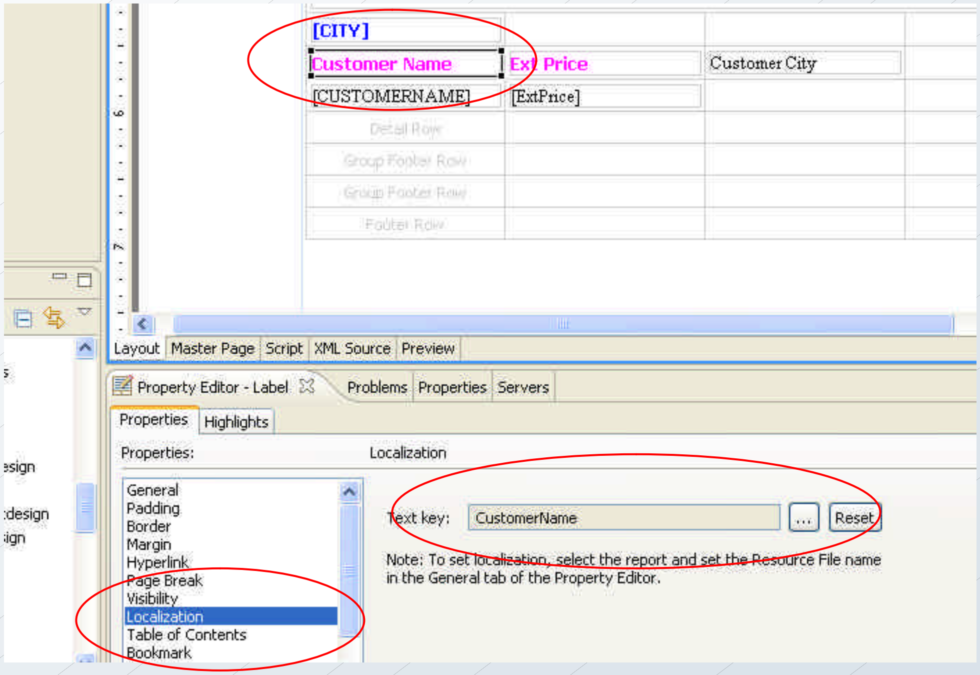
Properties: General, Description, Comments, User Properties, Named Expressions, Event Handler, Resources

Resources

Resource File: NewLocalization

Using a Resource File – cont'd

- Assign a resource key to a label or text element

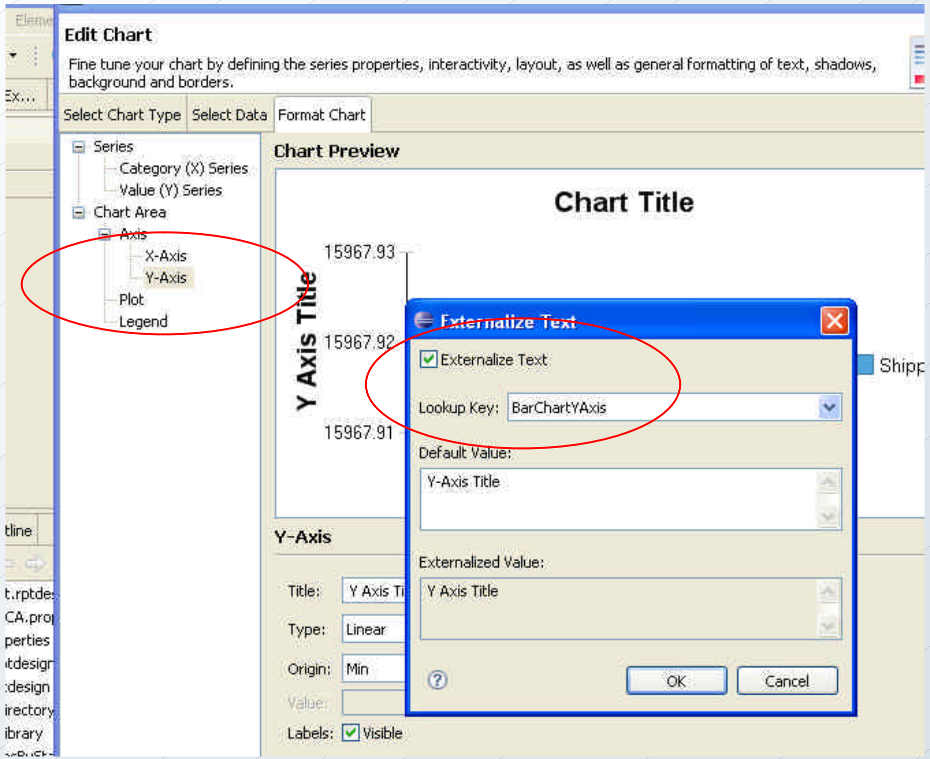


Text key: CustomerName

Note: To set localization, select the report and set the Resource File name in the General tab of the Property Editor.

Using a Resource File – cont'd

- Assign a resource key to a chart element



Edit Chart

Fine tune your chart by defining the series properties, interactivity, layout, as well as general formatting of text, shadows, background and borders.

Y-Axis

Title: Y Axis Ti

Type: Linear

Origin: Min

Value:

Labels: ☑ Visible

☑ Externalize Text

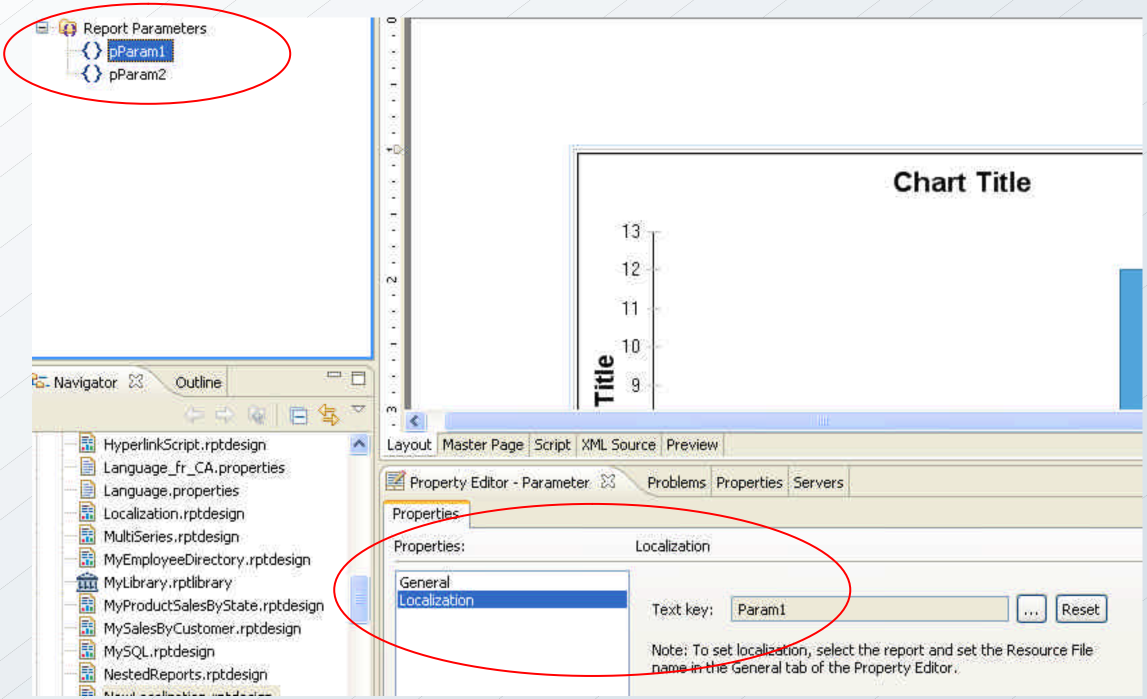
Lookup Key: BarChartYAxis

Default Value: Y-Axis Title

Externalized Value: Y Axis Title

Using a Resource File – cont'd

- Adding a resource key to a report parameter

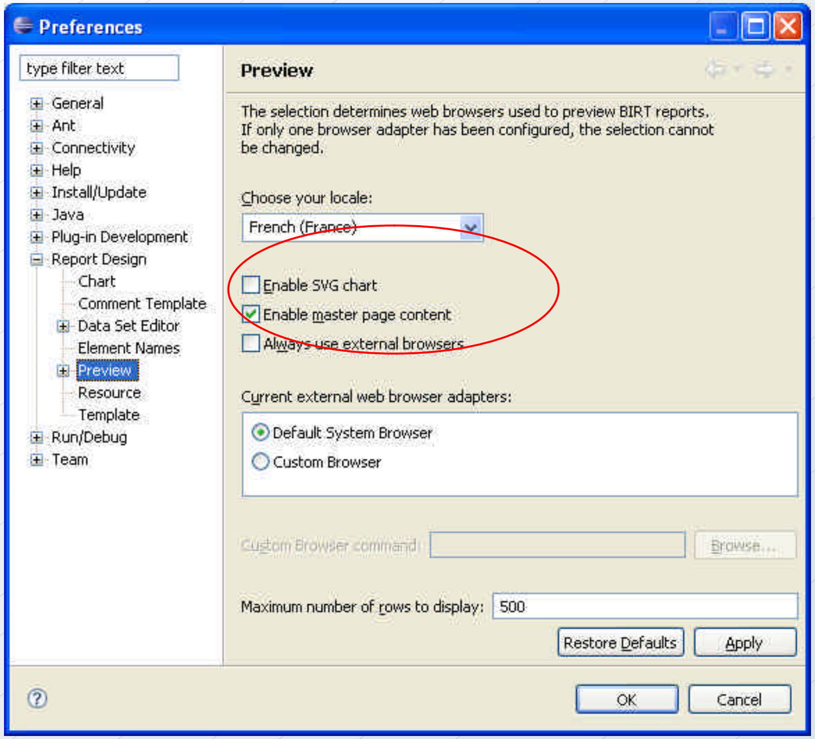


Text key: Param1

Note: To set localization, select the report and set the Resource File name in the General tab of the Property Editor.

Previewing a Report in Different Locales

- In BIRT, you can preview the report in a different locale by changing the viewing preference in Windows → Preferences → Report Design → Preview



Preview

The selection determines web browsers used to preview BIRT reports. If only one browser adapter has been configured, the selection cannot be changed.

Choose your locale: French (France)

□ Enable SVG chart

☑ Enable master page content

□ Always use external browsers

Current external web browser adapters:

● Default System Browser

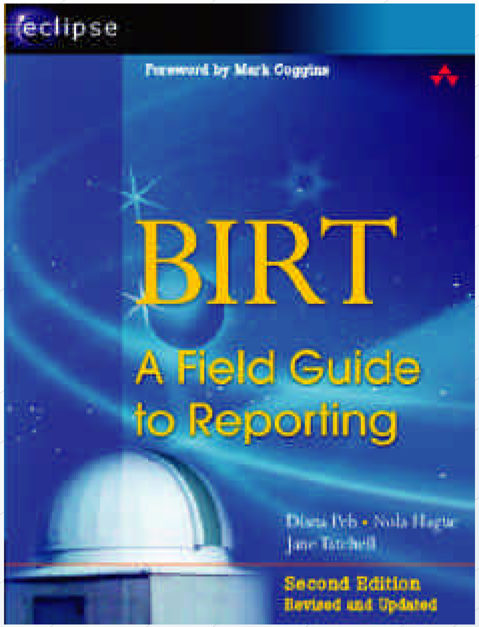
○ Custom Browser

Custom Browser command:

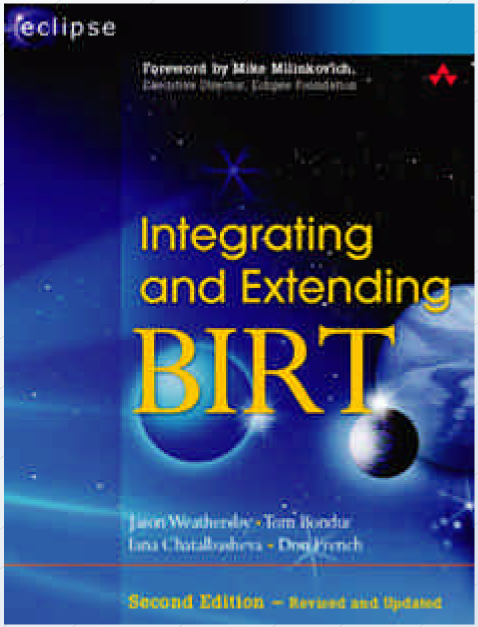
Maximum number of rows to display: 500

BIRT I18N in Action

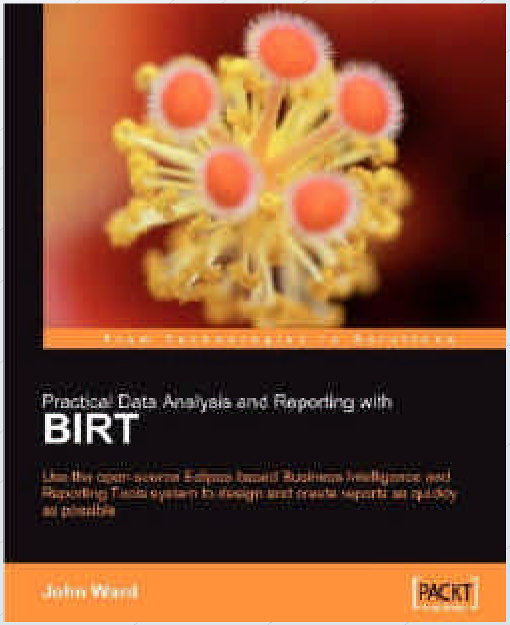
New and Updated Publications



Second Edition
Revised and Updated


eclipse
Foreword by Mike Milinkovich,
Executive Director, Eclipse Foundation
Integrating and Extending
BIRT
Jason Weathersby • Tom Bondur
Iana Chatalbasheva • Don French
Second Edition — Revised and Updated

Second Edition
Revised and Updated

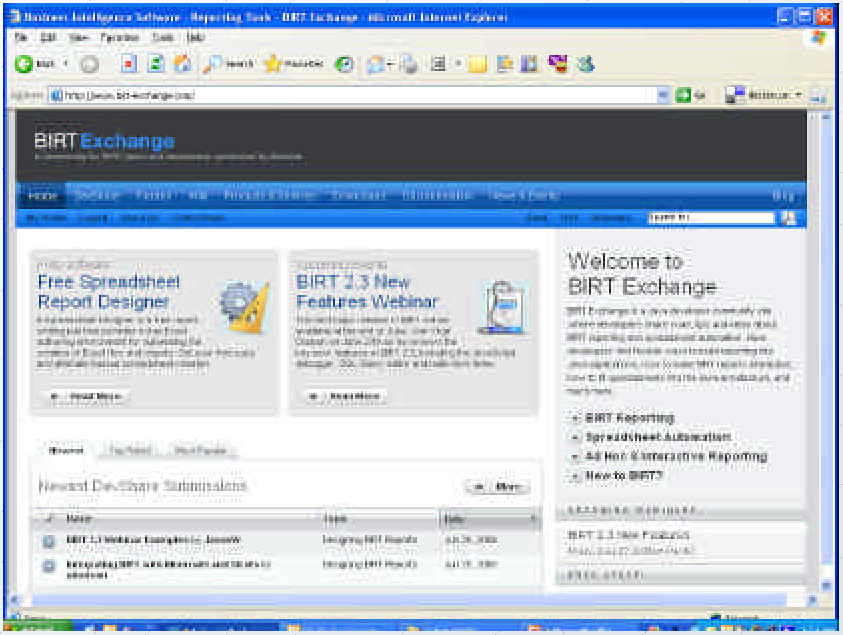

From Technologies to Solutions
Practical Data Analysis and Reporting with
BIRT
Use the open-source Eclipse-based Business Intelligence and Reporting Tools system to design and create reports as quickly as possible
John Ward
PACKT

New Book

BIRT Exchange Community Site

Centralized Knowledge Hub for BIRT Developers

- Access Downloads, Demos, Tutorials, Tips & Techniques, Webinars
- Easy for users to contribute content, share knowledge
- Enables developers to be more productive and build applications faster


Business Intelligence Software - Reporting Tools - BIRT Exchange - Microsoft Internet Explorer
BIRT Exchange
Free Spreadsheet Report Designer
BIRT 2.3 New Features Webinar
Welcome to BIRT Exchange
BIRT Reporting
Spreadsheet Automation
Ad Hoc & Interactive Reporting
New to BIRT?
Newest DevShare Submissions
BIRT 2.3 New Features

www.birt-exchange.com

Download

- Documentation
- Software

Share Knowledge

- Reports, Code, Tips
- Forums

Find

- Search, Sort
- Rate, Comment