



regoUniversity 2018

Jaspersoft Studio | Beginner

Your Guide: Sankhadeep Dhar

Part I Agenda

- Introduction
- Configuration Basics
 - Security Basics
- Installing Jaspersoft Studio
- Interface Introduction
 - Data Adapter Setup
- Server Connection Setup

*rego*University 2018

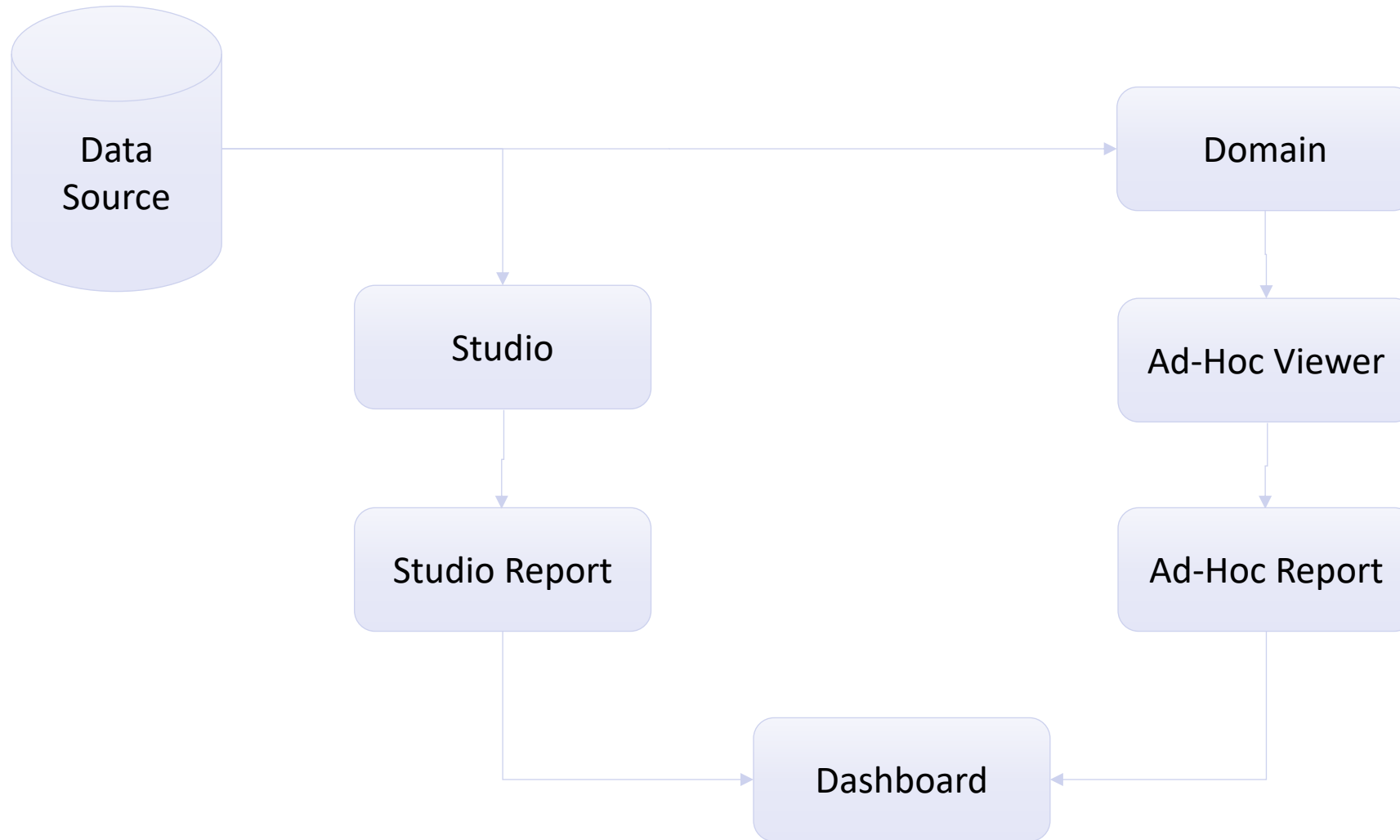
Part I: Introduction

- ✓ Jaspersoft – Overview of the Big Picture

What Is Jaspersoft?

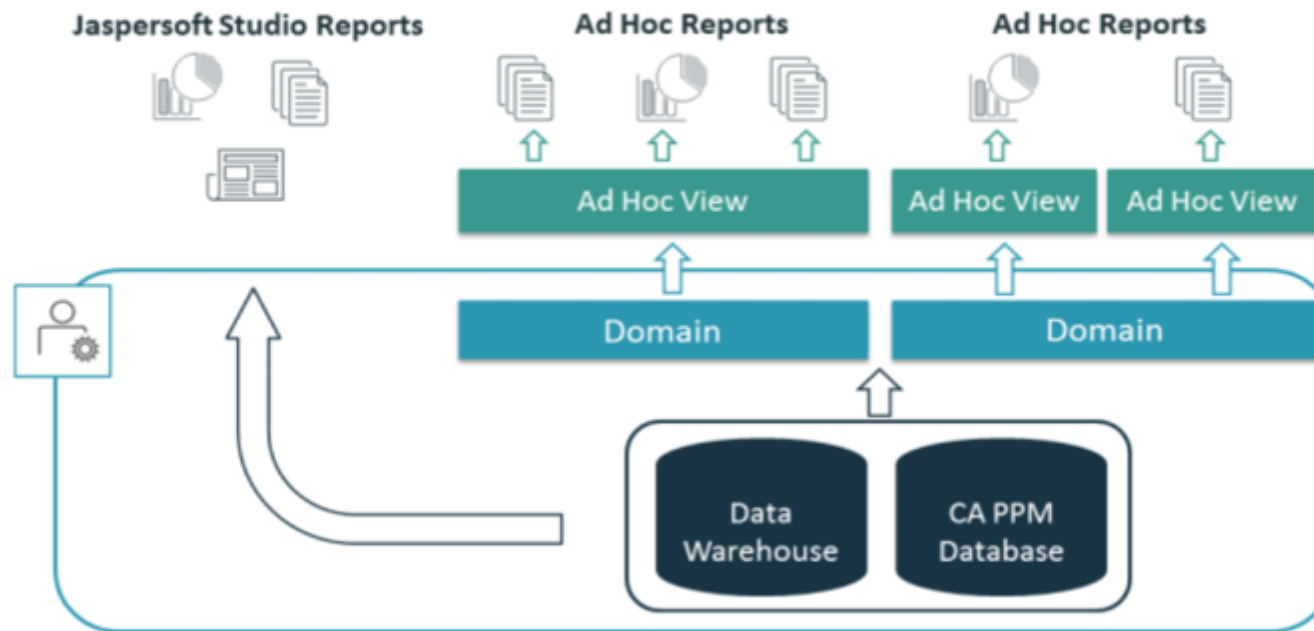
- JasperSoft is a reporting software that takes information from Clarity to provide easy to read, highly interactive reports for users.
- Jaspersoft reporting allows for sorting, filtering, formatting, moving/hiding columns, string searches, zooming in/out which can be saved for future use.
- Jaspersoft comes with pre-formatted Clarity reports or has the ability to do self-service ad-hoc reporting.
- Ad-hoc reports are easy to use with drag and drop capability, crosstab views, tables or charts.
- Reports can be published in PDF, XLS, XLSX, CSV, DOC X, RTF, ODT, ODS, or PPTX

Overview - The Big Picture



CA Jaspersoft Overview

Jaspersoft Overview

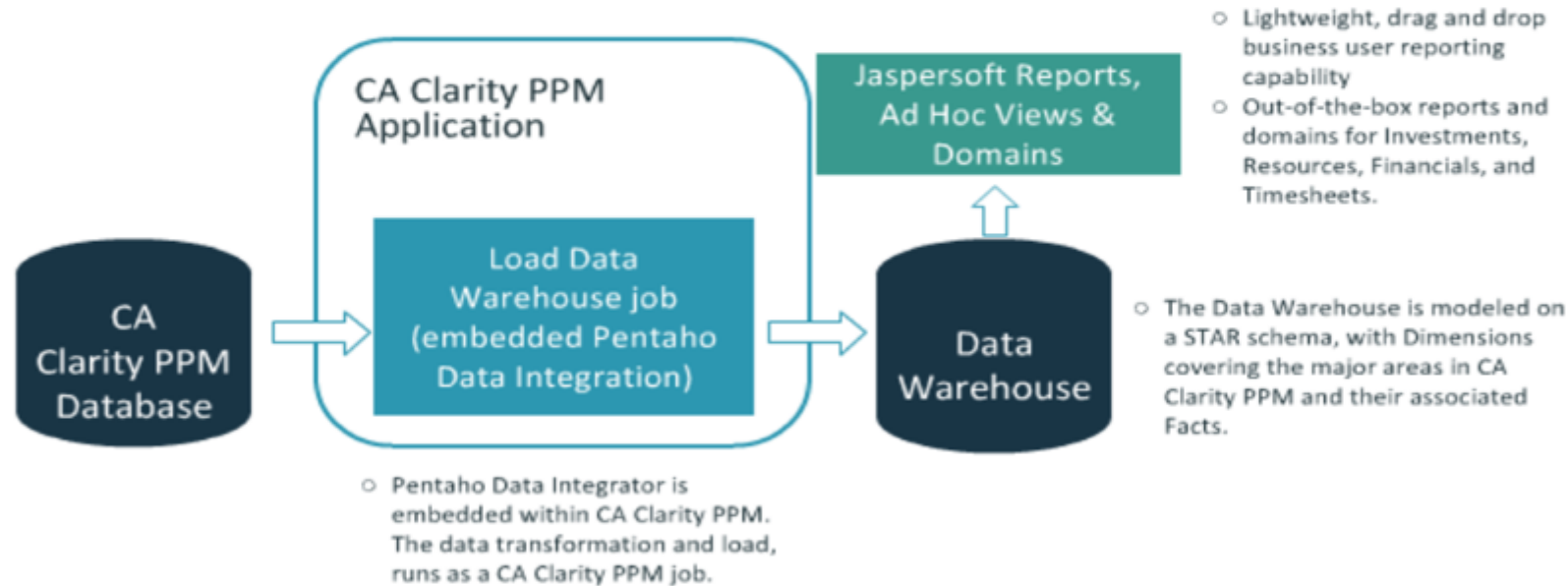


© 2015 CA. All rights reserved.



CA Data Warehouse Overview

Data Warehouse Overview



© 2015 CA. All rights reserved.



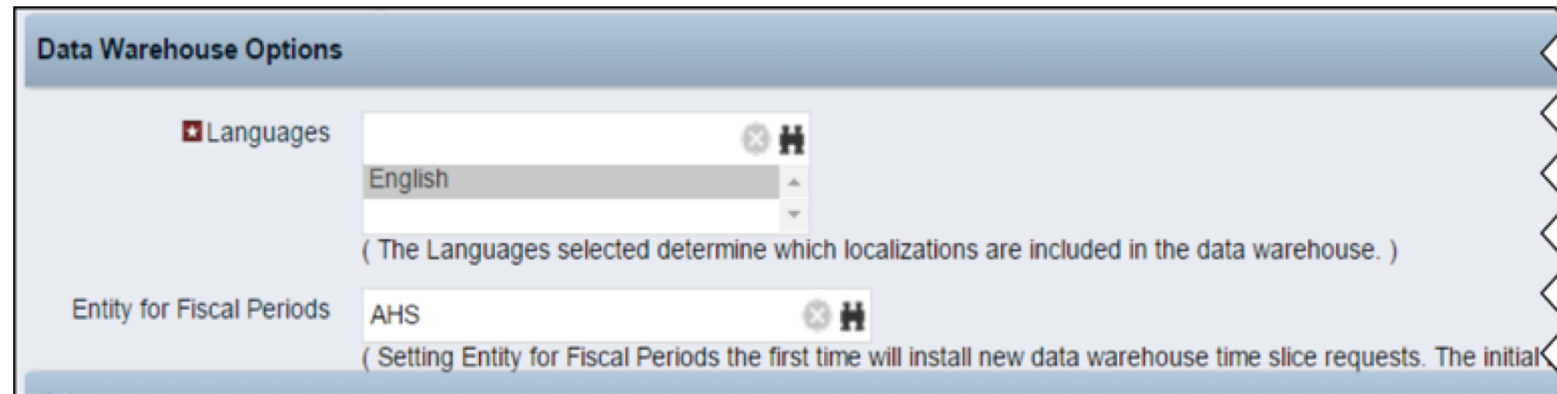
Part I: Configuring Basics

- ✓ Initial Setup Parameters
- ✓ Dynamic Query Lookups
 - ✓ Job Automation

Jaspersoft Configuration Considerations

Data Warehouse Options

- In order for the Jaspersoft Data Warehouse jobs to run, you must specify the Data Warehouse language and Fiscal Time period entity. To set these values, navigate to Administration > General Settings > System Options. Specify the language and financial period entity in the corresponding fields (this is typically English and the name of your entity). If you do not set these values, your Load Data Warehouse jobs will fail.



The screenshot shows the 'Data Warehouse Options' configuration window. It has a light blue header with the title 'Data Warehouse Options'. Below the header, there are two main sections. The first section is labeled 'Languages' with a small red square icon. It contains a dropdown menu with 'English' selected. Below the dropdown, there is a note in parentheses: '(The Languages selected determine which localizations are included in the data warehouse.)'. The second section is labeled 'Entity for Fiscal Periods' and contains a text input field with 'AHS' entered. To the right of the input field are two small icons: a circle with an 'x' and a house icon. Below this input field, there is another note in parentheses: '(Setting Entity for Fiscal Periods the first time will install new data warehouse time slice requests. The initial'.

Jaspersoft Configuration Considerations (Cont.)

Custom Attributes

- One of the key advantages of Jaspersoft over Business Objects is the ability to add custom attributes to the Domain without needing to modify the universe. However, in order to add the custom attribute, you need to check the new Include in the Data Warehouse box on the attributes property. If you do not check this box, the attribute is not added to the domain.

The screenshot shows the configuration window for an 'Object Attribute' named 'Escalate' under the 'Risk' object. The 'General' tab is active. The 'Attribute Name' field contains 'Escalate'. The 'Attribute ID' field contains 'ahs-escalate'. The 'Read-Only' checkbox is unchecked. The 'Include in the Data Warehouse' checkbox is checked. A callout box points to the 'Include in the Data Warehouse' checkbox with the text: 'Check to include the attribute in the warehouse.'

Jaspersoft Configuration Considerations (Cont.)

Dynamic Query Lookup

- If including a lookup which uses a Dynamic Query, you must ensure that your query includes the column LAST_UPDATED_DATE. If not included, the option to include in the data warehouse will be greyed out. While not required, you should also include a filter for LANGUAGE.

- Example:


```
SELECT @SELECT:a.name:country_name@,
       @SELECT:c.currency_code:currency_code@,
       @SELECT:c.currency_code || '(' || c.currency_symbol || ')':currency_name@,
       @SELECT:c.LAST_UPDATED_DATE:LAST_UPDATED_DATE@,
       @SELECT:LANG.LANGUAGE_CODE:LANGUAGE_CODE@,
       @SELECT:LANG.ID:LANGUAGE_ID@
FROM   cmn_countries b,
       cmn_captions_nls a,
       cmn_currencies c,
       CMN_LANGUAGES LANG
WHERE  a.table_name = 'CMN_COUNTRIES'
AND    @FILTER@
AND    a.pk_id = b.id
AND    a.language_code = @WHERE:PARAM:LANGUAGE@
AND    b.currency_id = c.id
AND    LANG.LANGUAGE_CODE=a.LANGUAGE_CODE
```

Jaspersoft Automation / Jobs

Automation

There are three new jobs Jaspersoft uses that you will need to schedule in order for Jaspersoft to work correctly. They are:

- Create and Update Jaspersoft Users Job
- Load Data Warehouse Job
- Load Data Warehouse Access Rights Job

Job Scheduling and Best Practices

Create and Update Jaspersoft Users Job

Create and Update Jaspersoft Users job does the following:

- Creates users in Jaspersoft for active CA PPM users
- Assigns Jaspersoft roles based on granted CA PPM Advanced Reporting rights
- Updates users in Jaspersoft that are inactivated or locked in CA PPM

Recommended Schedule

- Optimum scheduling may depend on frequency of user changes within your environment, but nightly run appears to be a good scheduling solution

Job Scheduling and Best Practices (Cont.)

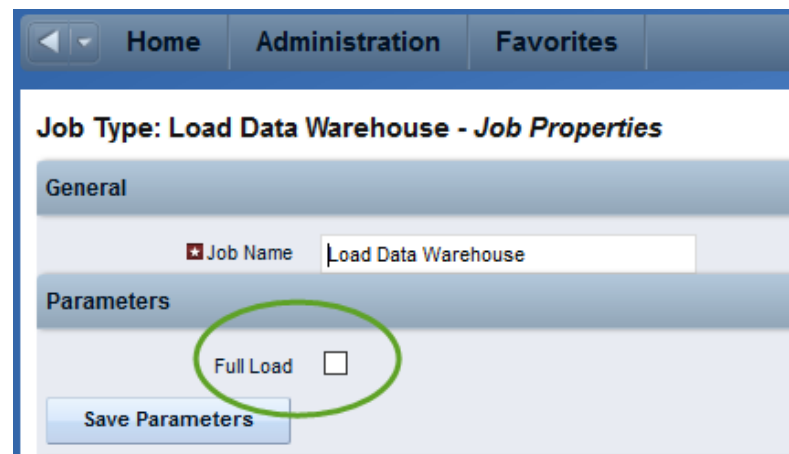
Load Data Warehouse Job

The Load Data Warehouse job does the following:

- Extracts data from the CA PPM database
- Transforms the extracted data into a de-normalized format
- Loads the de-normalized data into the CA PPM Data Warehouse

Recommended Schedule

- CA support recommends running the (full) load once daily after hours, and (lite) incrementally every hour



Job Scheduling and Best Practices (Cont.)

Load Data Warehouse Access Rights Job

The Load Data Warehouse Access Rights job does the following:

- Extracts access rights for investments and resources from the CA PPM database
- Loads the extracted data into the CA PPM Data Warehouse

Recommended Schedule

- Nightly – After Hours

Part I: Security Basics

- ✓ Security Basics
- ✓ Groups / Roles
- ✓ Domain Security
- ✓ Folder Security

Security Basics

Security

There are several new security rights associated with Jaspersoft reporting.

Best Practice Recommendations

- New Clients – Remove security rights for access to the old Business Objects reports (*Reports – Access*). This eliminates the risk of users becoming confused by accessing the wrong reporting feature.
- Transitioning Clients – For clients transitioning from BO to Jaspersoft where access to old reports is still needed, train users on the two reporting tools and their various functions in relation to each other.

Security Basics

Security within the Jaspersoft Server can be separated into two sections:

- Jaspersoft Server Authentication/Authorization
- CA PPM Authentication/Authorization

Jaspersoft Server Authentication/Authorization:

- Any user within Jaspersoft Server is given a role of “ROLE_USER” by default, which gives them basic rights to view reports.
- To develop and deploy reports to the server, a user account with a role of “REPORT_DESIGNER” is required
- Other roles, which are self explanatory, include:
 - ROLE_DOMAIN_DESIGNER
 - ROLE_ADHOC_DESIGNER
 - ROLE_DASHBOARD_DESIGNER
 - ROLE_DATASOURCE_DESIGNER
 - ROLE_ADMINISTRATOR
 - ROLE_SUPERUSER

Security Basics (Cont.)

CA PPM 14.2+ Installation and Configuration will create several roles within Jaspersoft Server, which correlate to roles within CA PPM, which include:

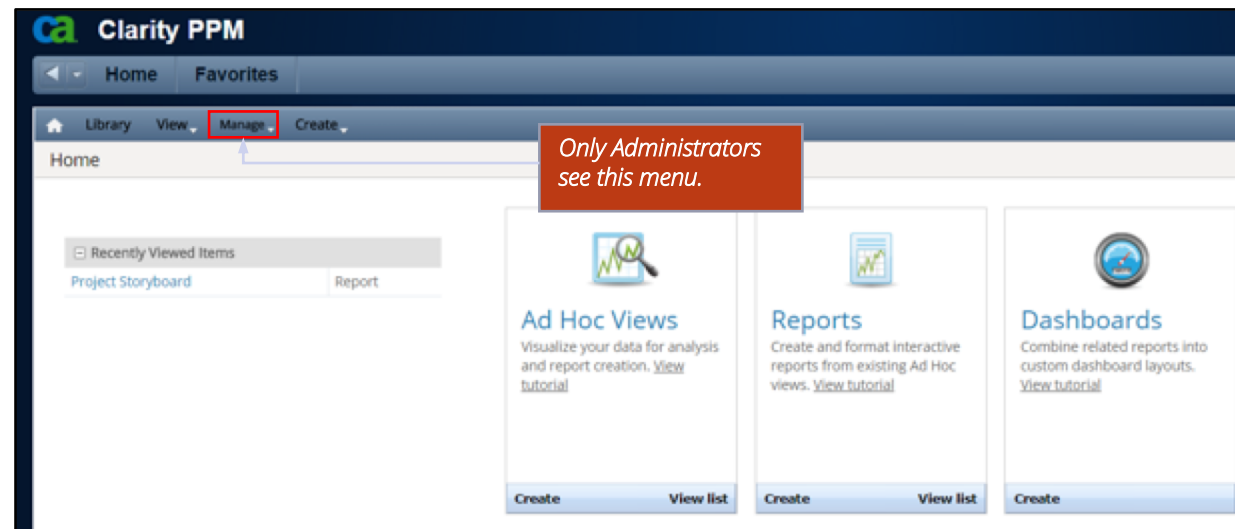
Clarity PPM Right	Description	Notes
Advanced Reporting - Ad Hoc Create	Allows resource to navigate to the Advanced Reporting page and create Ad Hoc Views. Resource is assigned the following Jaspersoft roles: ROLE_USER and ROLE_ADHOC_DESIGNER.	Needed to allow Ad Hoc reporting. Note that without the domain access user will not be able to create.
Advanced Reporting - Administer	Allows resource to navigate to the Advanced Reporting page and administer Advanced Reporting. Resource is assigned the following Jaspersoft roles: ROLE_USER and ROLE_ADMINISTRATOR.	By default this is the Clarity Administrator
Advanced Reporting - Dashboard Create	Allows resource to navigate to the Advanced Reporting page and create Dashboards. Resource is assigned the following Jaspersoft roles: ROLE_USER and ROLE_DASHBOARD_DESIGNER.	To create a dashboard, users need reports to add to the dashboard and to create those reports they will need domain access too.
Advanced Reporting - Data Source Create	Allows resource to navigate to the Advanced Reporting page and create Data Sources. Resource is assigned the following Jaspersoft roles: ROLE_USER and ROLE_DATASOURCE_DESIGNER.	By Default this is set to the admin account. This is for advanced users who will be creating new data sources.
Advanced Reporting - Domain Create	Allows resource to navigate to the Advanced Reporting page and create Domains. Resource is assigned the following Jaspersoft roles: ROLE_USER and ROLE_DOMAIN_DESIGNER.	By Default this is set to the admin account. This is for advanced users who will be creating new domains.
Advanced Reporting - Navigate	Allows resource to navigate to the Advanced Reporting page. Resource is assigned Jaspersoft role as ROLE_USER.	By default all users inherit this right but without Clarity rights to accompany, the user cannot navigate
Advanced Reporting - Report Create	Allows resource to navigate to the Advanced Reporting page and create Reports. Resource is assigned the following Jaspersoft roles: ROLE_USER and ROLE_REPORT_DESIGNER.	To create reports, users will need domain access too

Security Basics (Cont.)

As an example, granting project managers (via the PM group) the following rights within CA PPM:

- Advanced Reporting - Ad Hoc Create
- Advanced Reporting - Dashboard Create
- Advanced Reporting – Navigate

Would show them the following when accessing advanced reporting.



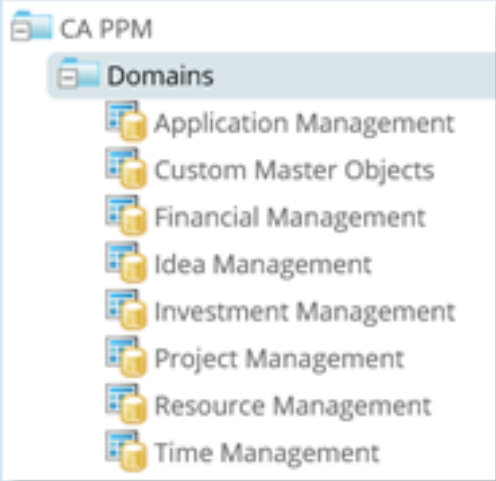
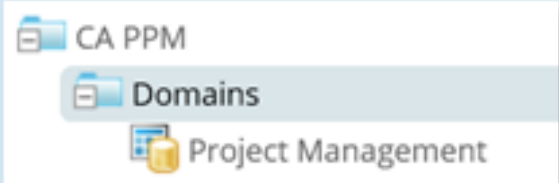
Security Basics – Domain Security

The following roles are in Advanced Reporting, but do not link back to CA PPM groups. The Advance Reporting administrator will have to grant these roles to every user manually in Jaspersoft.

Domain	Right
Investment Management	CSK_ROLE_INVESTMENT_MANAGEMENT
Project Management	CSK_ROLE_PROJECT_MANAGEMENT
Program Management	CSK_ROLE_PROGRAM_MANAGEMENT
Idea Management	CSK_ROLE_IDEA_MANAGEMENT
Application Management	CSK_ROLE_APPLICATION_MANAGEMENT
Custom Master Objects	CSK_ROLE_CUSTOM_MASTER_OBJECTS
Financial Management	CSK_ROLE_FINANCIAL_MANAGEMENT
Resource Management	CSK_ROLE_RESOURCE_MANAGEMENT
All Domains	CSK_ROLE_ALL_REPORTS_DOMAINS

Security Basics – Domain Security (Cont.)

To give project managers access to all domains, add them to *CSK_ROLEALL_REPORTS_DOMAINS*. The following table shows the difference between the two roles.

CSK_ROLE_ALL_REPORTS_DOMAINS	CSK_ROLE_PROJECT_MANAGEMENT
	

Security Basics – Folder Security

- Jaspersoft stores Views, Ad Hoc Reports, Dashboards, Data Sources, Domains, Input Controls, Reports, and Templates in the Jaspersoft Repository using a folder structure. This lets you organize reports, data sources, etc. and control access using permissions
- Out of the Box, the Jaspersoft Repository has two primary folders:
 - Public
 - Houses several report templates
 - Organization
 - CA PPM – Contains all components provided by CA, which includes: Data Sources, Ad Hoc Views, Dashboards, Domains, Input Controls, Reports, Resources and Templates
 - Shared – Content available to multiple users
 - Users – Separate folders for each user within the Jaspersoft Server
 - Temp – Temporary files/reports
- Additional folders, sub-folders can be created by administrators, based on the needs of the organization
- Additional security roles can be created by administrators, based on the needs of the organization

Part I:

Installing Jaspersoft Studio

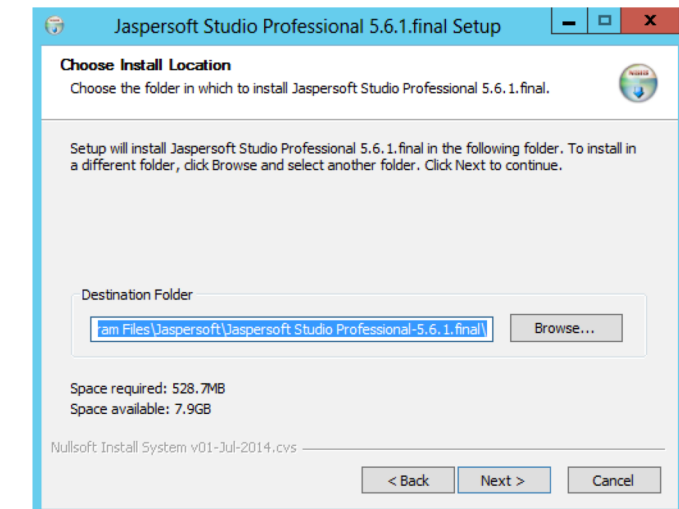
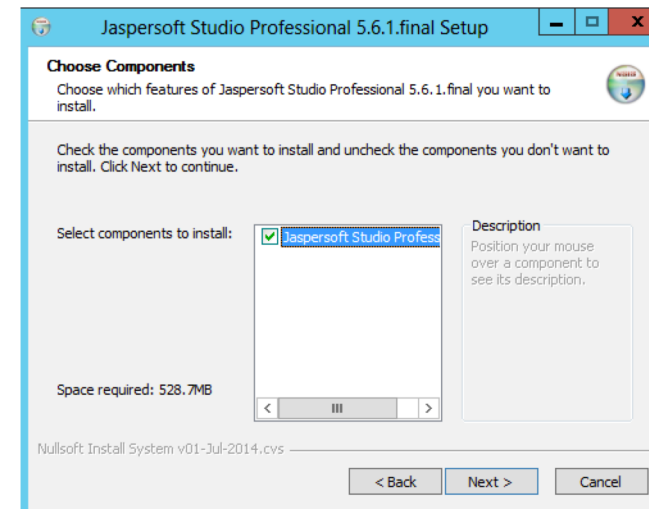
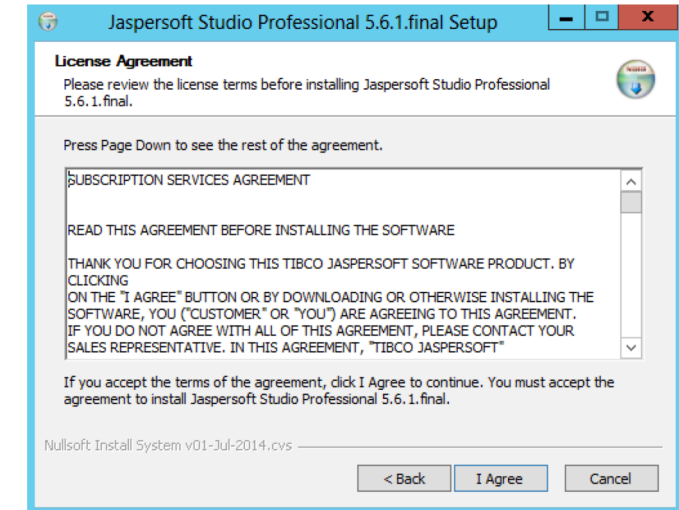
- ✓ Software Download
- ✓ Installation Overview

Jaspersoft Studio Installation

- Your official Jaspersoft Studio install and license file can be found within your CA Support Download Center
 - CA PPM 14.2: **Jaspersoft Studio 5.6.1**
 - CA PPM 14.3: **Jaspersoft Studio 6.1.0**
 - CA PPM 15.2: **Jaspersoft Studio 6.2.1**
 - CA PPM 15.3: **Jaspersoft Studio 6.4.2**
- For the purposes of this session, if needed, you can download the Commercial Trial version of the TIBCO Jaspersoft Studio here:
 - <http://reporting.regoconsulting.com/>
 - Download the appropriate version for your desktop platform
- The trial version should remain licensed for 30 days, after which, you'll need to install the CA provided version from your CA Support Download Center

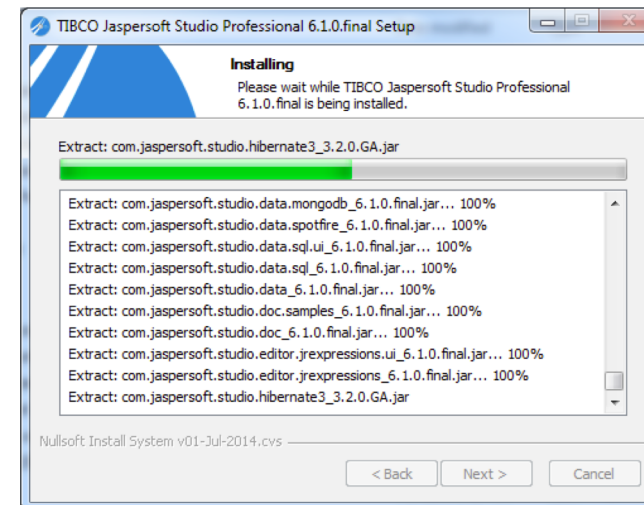
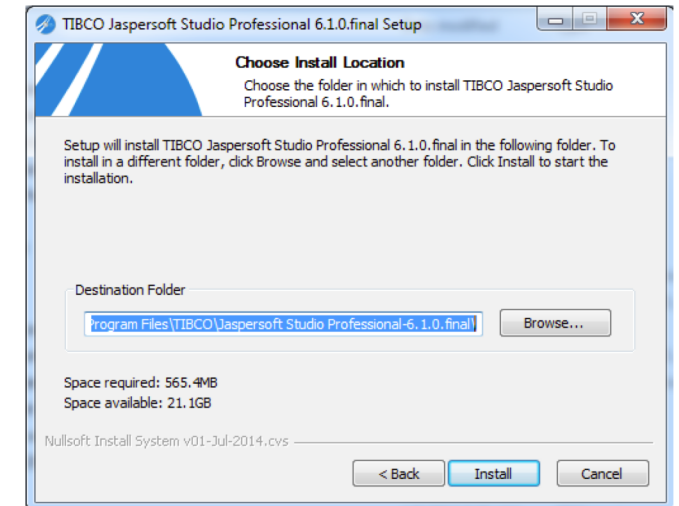
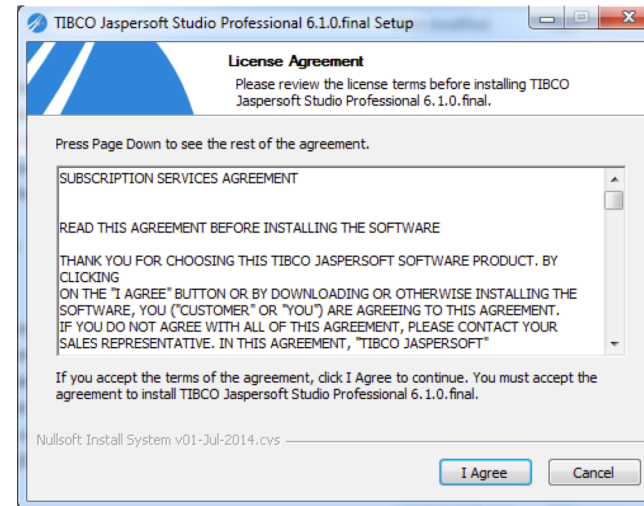
Jaspersoft Studio Installation – CA PPM 14.2

- The Studio installation should progress like any software install for your platform, and the defaults can be used for the base installation:



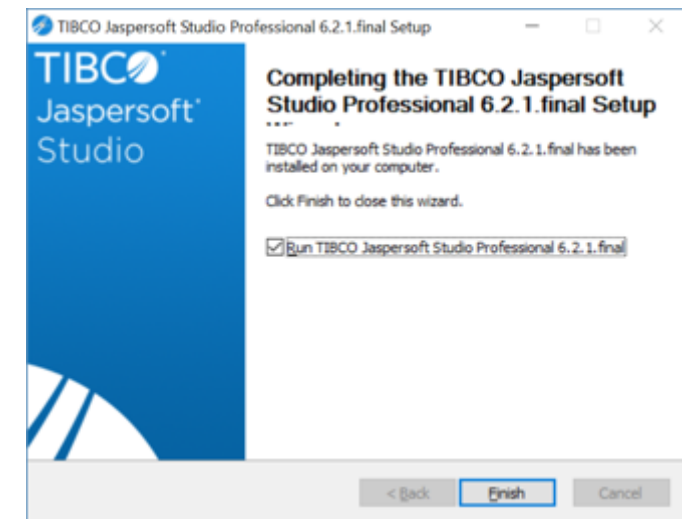
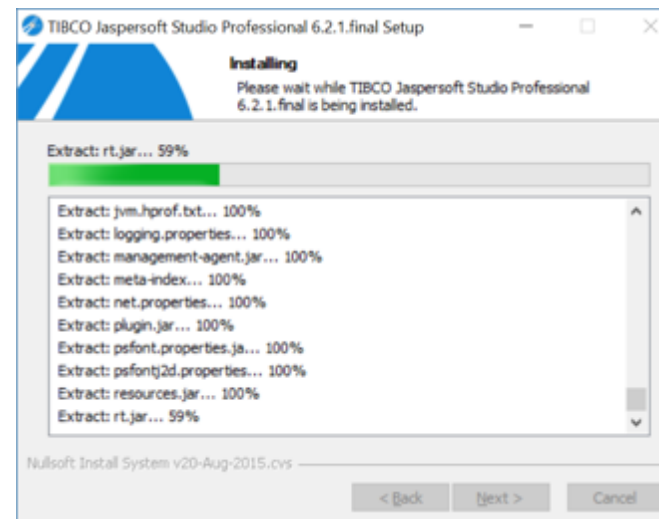
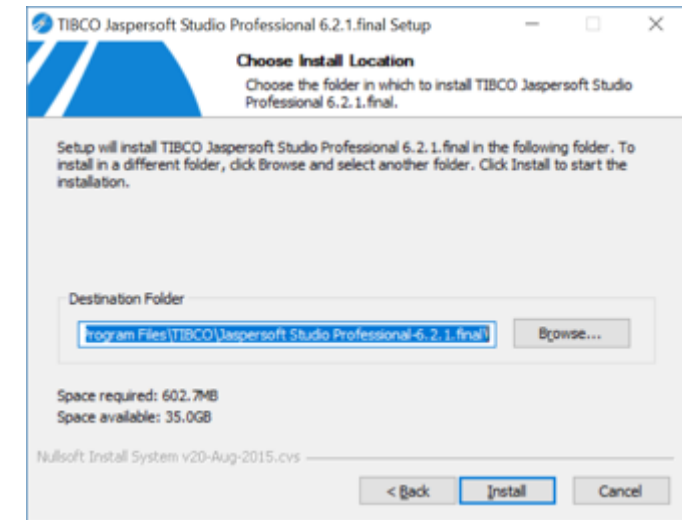
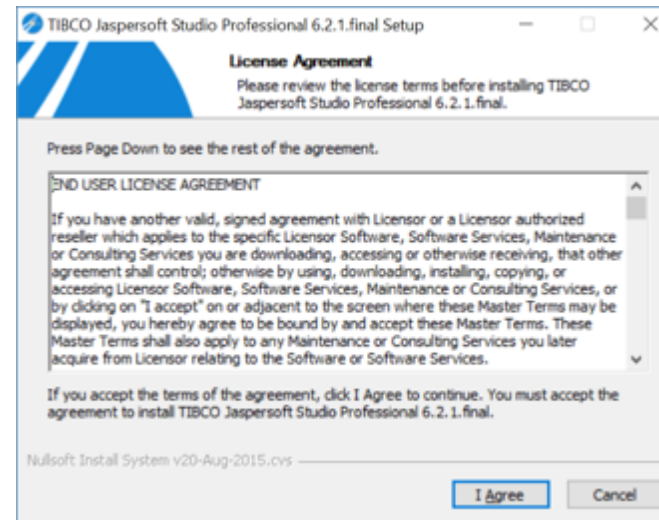
Jaspersoft Studio Installation – CA PPM 14.3

- The Studio installation should progress like any software install for your platform, and the defaults can be used for the base installation:



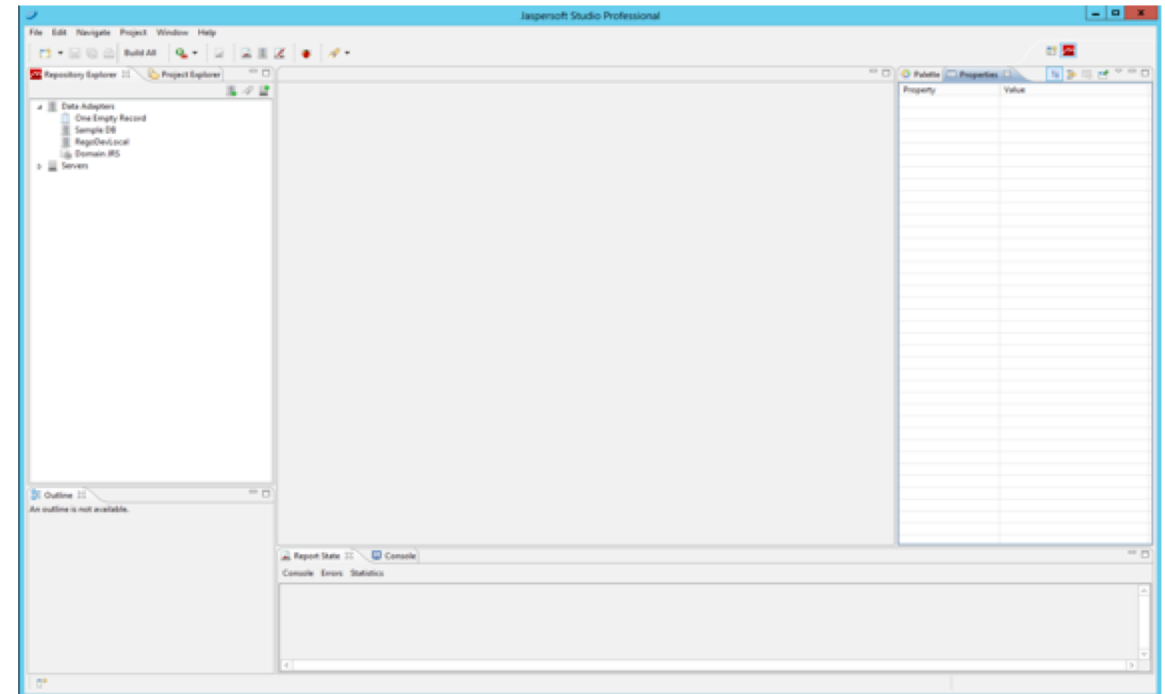
Jaspersoft Studio Installation – CA PPM 15.2

- The Studio installation should progress like any software install for your platform, and the defaults can be used for the base installation:



Jaspersoft Studio Interface

- After completion of the Studio software, launch the application
- The Studio software is built on top of the Eclipse platform, those with eclipse experience should feel right at home
- There are 7 panels in which you'll spend most of your report development time:
 - Repository Explorer
 - Project Explorer
 - Outline
 - Palette
 - Properties
 - Report Design
 - Report State



Jaspersoft Studio Interface (Cont.)

- **Repository Explorer**
Used to view and create Data Adapters, connect to and browse Jaspersoft Servers
- **Project Explorer**
This is your project folder structure, where you will create, and maintain your various reporting projects
- **Report Design**
The design view will be used for development, and initial testing of your reports
- **Outline**
This view will give you a hierarchical view of your report, providing a easy to read structural view of your report
- **Palette**
Drag and Drop components that can be included within your report
- **Properties**
Much time will be spent here tweaking individual settings of various reporting components
- **Report State**
During development, this view will provide error reports used for debugging

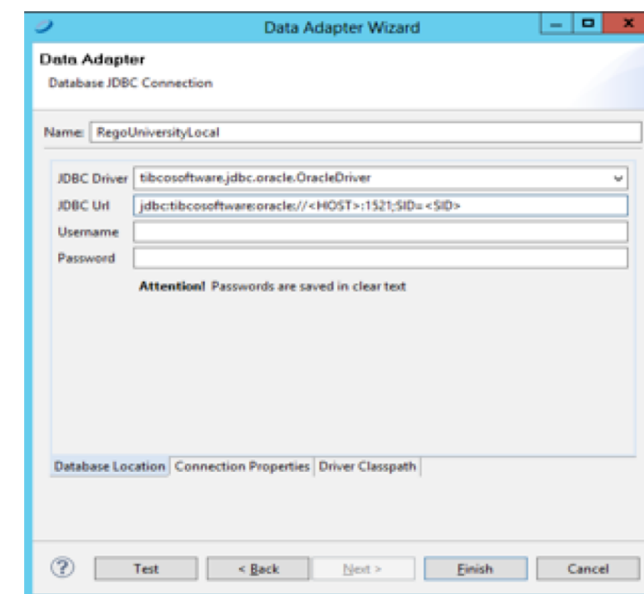
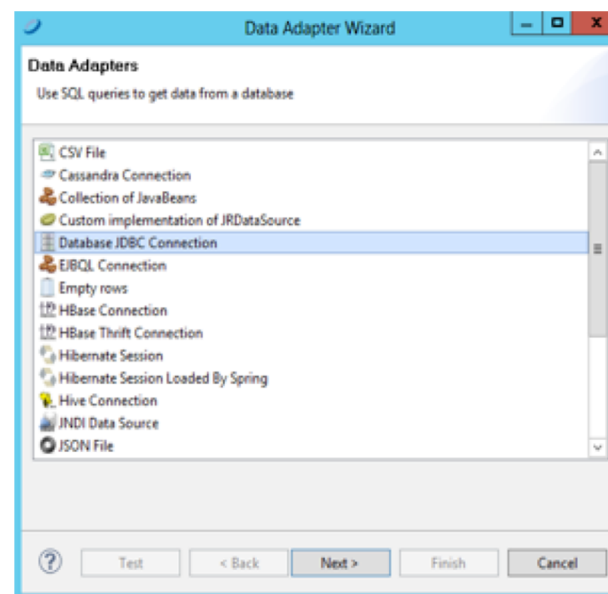
Part I:

Connecting to Data & Repository Sources

- ✓ Data Adapter Connection
- ✓ Repository Connection
- ✓ Interface Overview

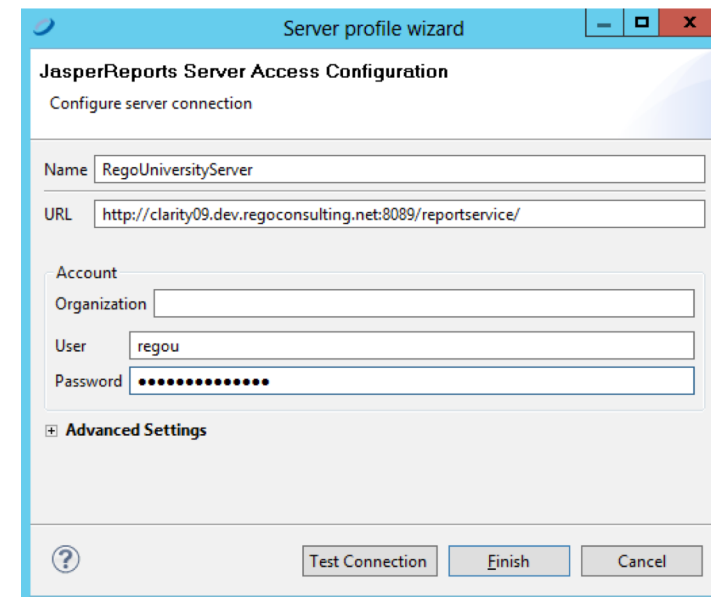
Jaspersoft Studio – Data Adapter

- In this exercise we'll create a new Data Adapter, which we will connect to an existing Clarity 14.2+ Oracle instance
- From the “Repository Explorer” right-click on Data Adapters, and select “Create Data Adapter”, then select “Database JDBC Connection”
- Give it a meaningful Name, then select “tibcosoftware.jdbc.oracle.OracleDriver” for the JDBC Driver
- See <http://reporting.regoconsulting.com/> for the login credentials
- Verify using “Test”
- Click “Finish”



Jaspersoft Studio – Repository Connection

- At this stage, having connected a Data Adapter to a Clarity Database, one can begin development locally
- To validate your locally developed report against the Jaspersoft Server, you must first connect your Studio environment to a Jaspersoft Server
- Within the “Repository Explorer” right-click on “Servers” and select “Create JasperReports Server Connection” and name the connection appropriately
- For this session, we’ll use the following information to connect to Rego development server:
 - URL: See <http://reporting.regoconsulting.com/> for details
 - Organization: rego
 - On Demand Users Excluded
 - Username and Password: See <http://reporting.regoconsulting.com/> for details



The screenshot shows the "Server profile wizard" dialog box, titled "JasperReports Server Access Configuration". The subtitle is "Configure server connection". The dialog contains the following fields and sections:

- Name:** RegoUniversityServer
- URL:** <http://clarity09.dev.regoconsulting.net:8089/reportservice/>
- Account section:**
 - Organization:** (empty field)
 - User:** regou
 - Password:** (masked with dots)
- Advanced Settings:** (collapsed section)
- Buttons:** Test Connection, Finish, Cancel

Jaspersoft Studio – Connect to On Demand

- You can connect Jasper Studio (Jaspersoft 6.2.1_5.2.1.x or higher) to client CA PPM transactional or data warehouse database without a VPN connection using CA JDBC Adapter.

Here is how you do it

Install the adapter:

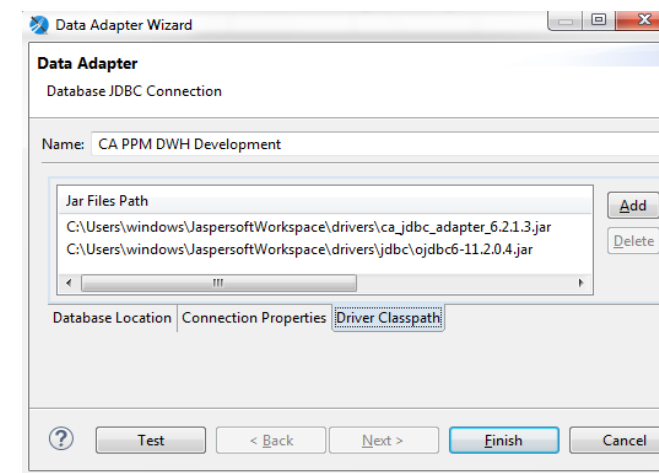
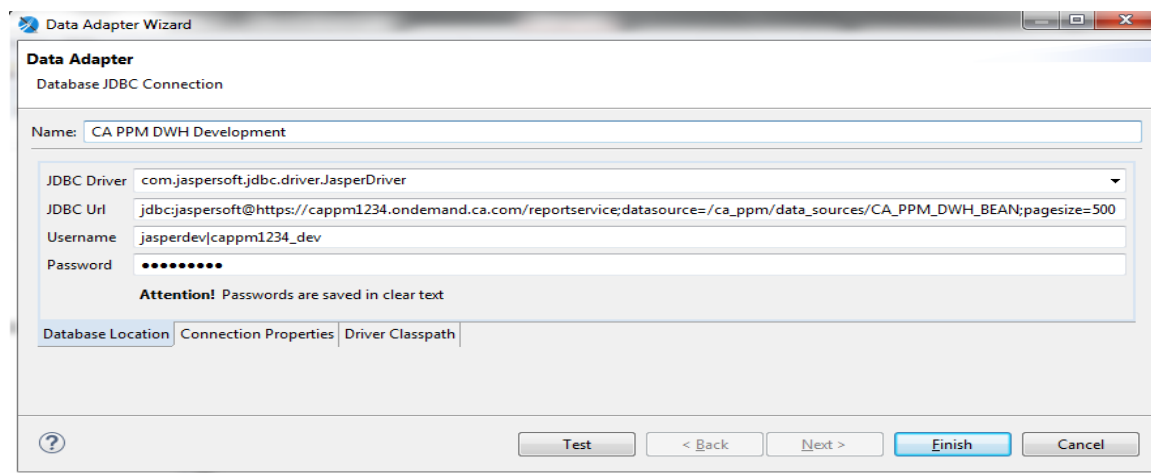
- Exit any instance of Jasper Studio that is running.
- Download the ca_jdbc_adapter_6.4.2.2.zip file from <http://reporting.regoconsulting.com/>
- Create folder “drivers” at your Jaspersoft workspace location. (Click File in Jasper Studio and select Switch Workspace to find workspace location)
- Copy the contents of ca_jdbc_adapter_6.4.2.2.zip to “drivers” folder
- Contents should include ca_jdbc_adapter_6.4.2.2.jar, jdbc\ojdbc8-12.2.0.1.0.jar
- Reopen the Jasper Studio

Create a CA JDBC Adapter Connection:

- From the “Repository Explorer” right-click on Data Adapters, and select “Create Data Adapter”, then select “Database JDBC Connection”

Jaspersoft Studio – Connect to On Demand contd..

- Give it a meaningful Name, then select “com.jaspersoft.jdbc.driver.JasperDriver” for the JDBC Driver
- Url : jdbc:jaspersoft@<report_service_url>/reportservice;datasource=/ca_ppm/data_sources/<bean_name>;pagesize=500
 <report_service_url> is available in System Options-Advanced Reporting.
 <bean_name> is CA_PPM_BEAN (Transactional DB) or CA_PPM_DWH_BEAN (DWH DB)
- Username: <USERNAME>|<ORGANIZATION_ID>
 Username: report developer user created in Advance Reporting. Organization Id is available in System Options-Advanced Reporting.
- Password : report developer user password
- Click the Driver Classpath tab, Add the CA JDBC jar files . Click Test to Verify and Click Finish



Part 1: Conclusion – Interface Live Demo



Questions?

*rego*University 2018

Let Rego be your guide.

Part II Agenda

Creating a Report

- Project Creation
- Report Creation
 - Query Setup
 - Field Creation
- Basic Report Objects
 - ✓ Static Text
 - ✓ Text Field
 - ✓ Frames
 - ✓ Shapes
 - ✓ Breaks
- Report Preview

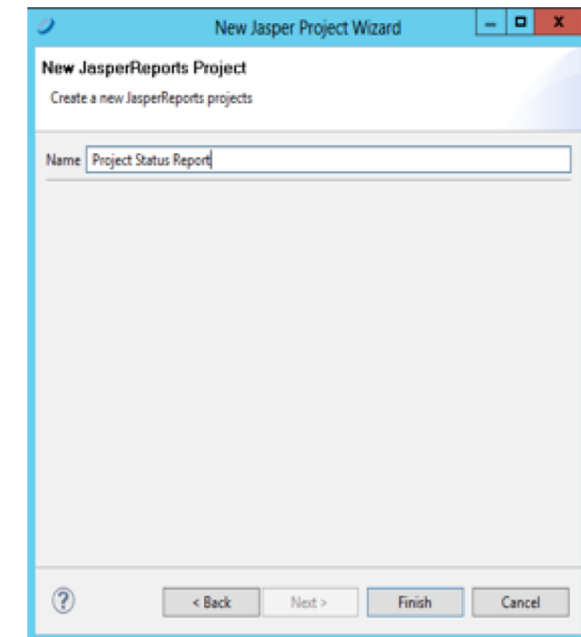
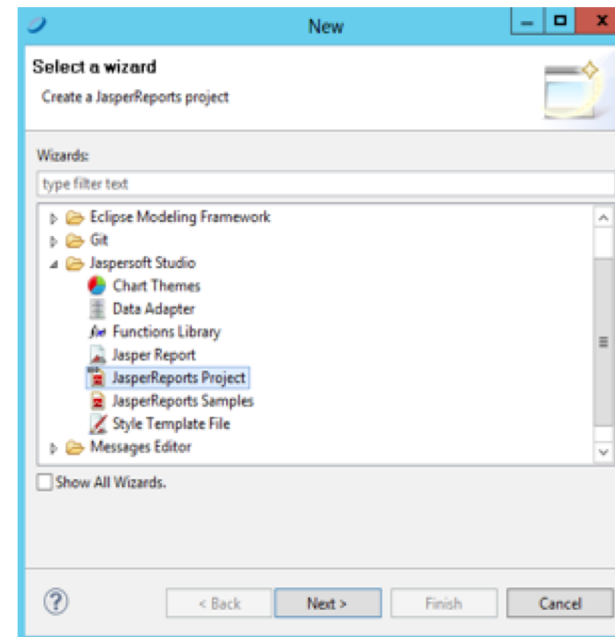
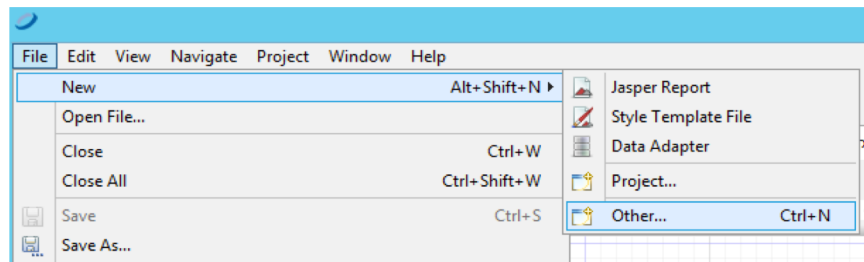
*rego*University 2018

Part II: Project Creation

- ✓ Creating your first project
- ✓ Add fields to your Report
- ✓ Preview your Report

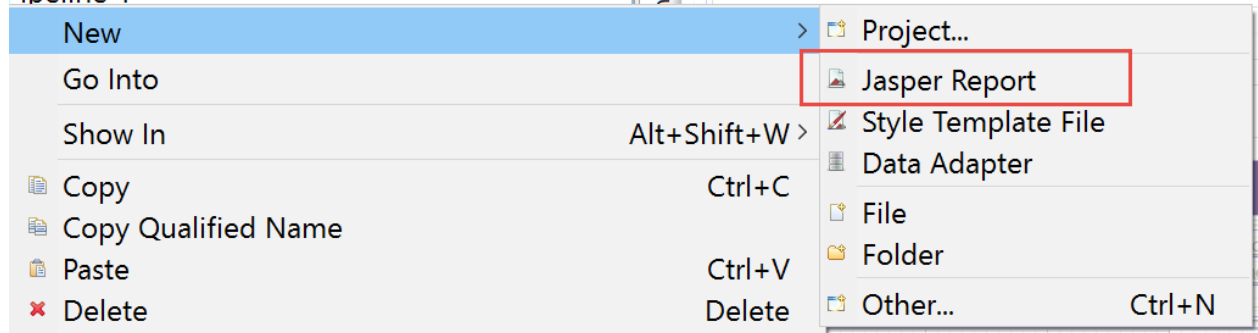
Creating The Project

- You should now have an installation of Jaspersoft Studio, a Data Adapter connected to a Clarity PPM database, and a Jaspersoft Server connection for deployment
- Start by creating a new Project, within the “Project Explorer” tab, by clicking on “File -> New -> Other”, then select “JasperReports Project” and click “Next”
 - **Be careful not to create a new generic Java project
- Name your project appropriately, such as “Training Project - <Initials>”, to avoid conflicts during deployment



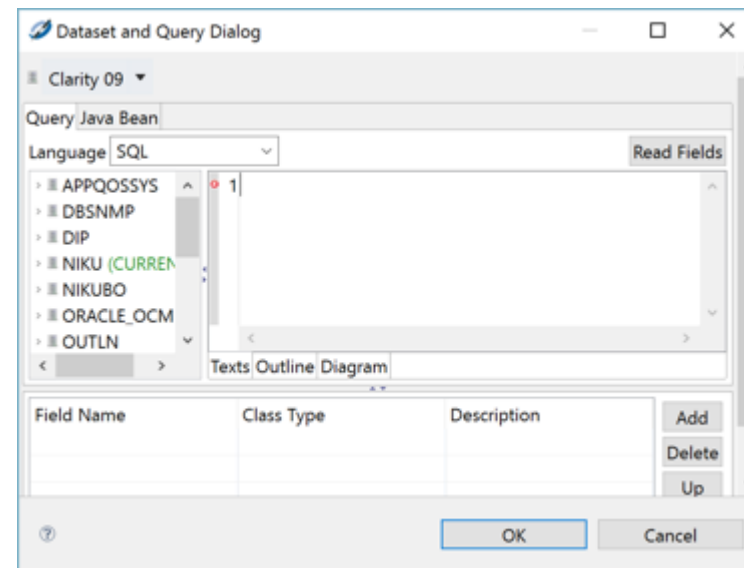
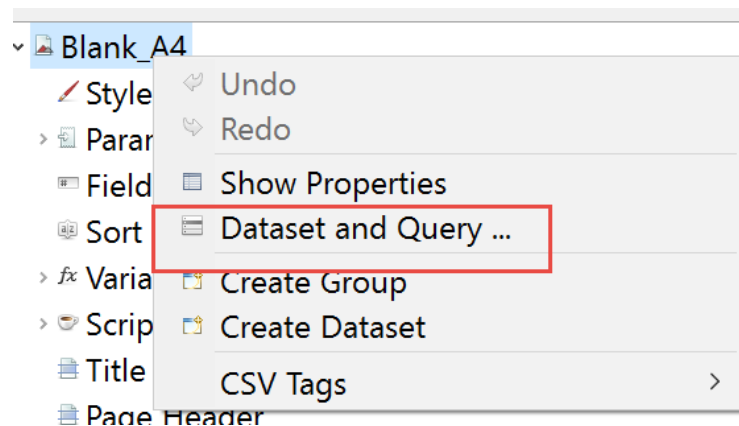
Creating The Report

- Since the Jaspersoft project has been created, you can now create the report (Jrxml file).
- Start by navigating to the Project Explorer. Right click the Project Folder you just created and select New -> Jasper Report
- Select the template you want the report to be based on. Templates determine Size and default items and formatting based on the one chosen. Click Next.
- Select the project folder you just created. Click Next.
- Next select the Data Adapter to be used to test this report with.
- Click Finish.



Data Setup

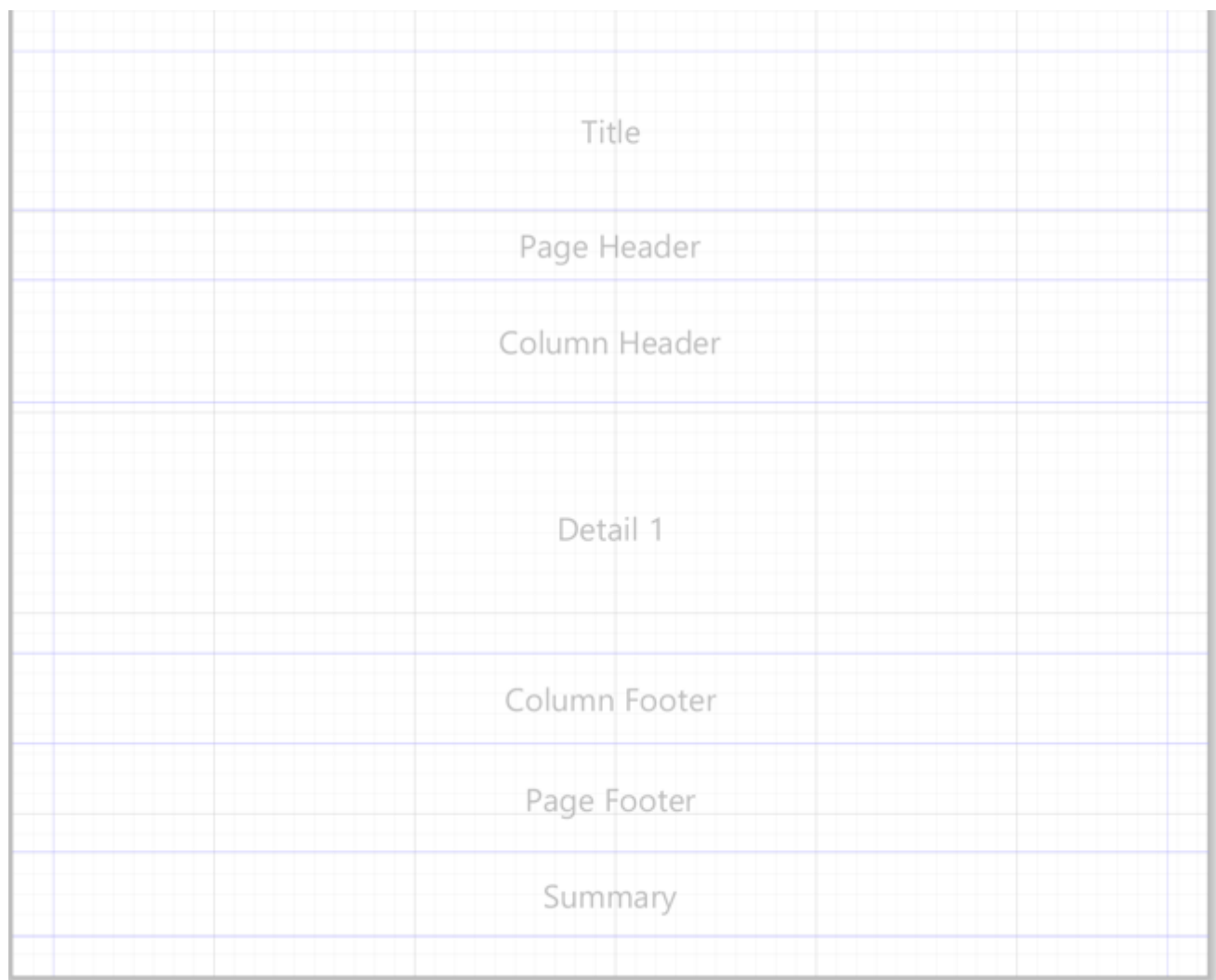
- The query can be setup in two ways:
 1. Create a query using SQL to get the data we need to put into the report.
 2. Create a domain to get the data we need to put into the report. (This is covered in the advanced)
- Navigate to the Outline tab.
- Right click on the Report (This will be the name of the jrxml).
- This will bring up the Dataset and Query interface.



Field Creation

- Now that we have the Dataset and Query interface up, we can input our query.
- Once we have put our query in the dialogue, click the Read Fields button.
 - Note: This will only read fields if you have access to the database.
 - Query for Exercise: http://reporting.regoconsulting.com/2018/Beginner_Queries.txt
- Steps to manually create fields.
 1. Click the Add button
 2. Name the field the exact alias name as in the query. (This includes caps)
 3. Change the Class Type to the appropriate data type.
- Data Types
 - Java.lang.String (All Strings)
 - Java.math.BigDecimal (Numbers with decimals) : Convert this to java.lang.Double
 - Java.lang.Boolean (Boolean type)
 - Java.util.collection (Use this for multi value field separated via commas)
 - Java.util.date (Date Type)
 - Java.lang.integer (Numbers without decimals)

Report Design



Title: First section of the report.

Page Header: Repeated on each page at the top.

Column Header: The header for the Detail below.

Detail: The data for each row in your query.

Column Footer: The footer for the Detail above.

Page Footer: Repeated on each page on the bottom.

Summary: The last section of the report.

Basic Report Objects

- To add objects to your report, navigate to the Outline tab and check the elements in the Palette tab



- The following objects are available.
 - Static Text: Object which allows you to put permanent text on the report.
 - Text Field: Object which let's you utilize, fields, parameters, and java functions with the expression editor.
 - Image: Object which let's you reference an image stored in the Repository.
 - Break: Object which creates a new page for the contents put after.
 - Rectangle: Object which lets you create a clip art of a rectangle.
 - Frame: Object which can size it's self to the contents inside. It also allows to keep grouping objects in a report to move together.

Basic Report Objects: Static Text

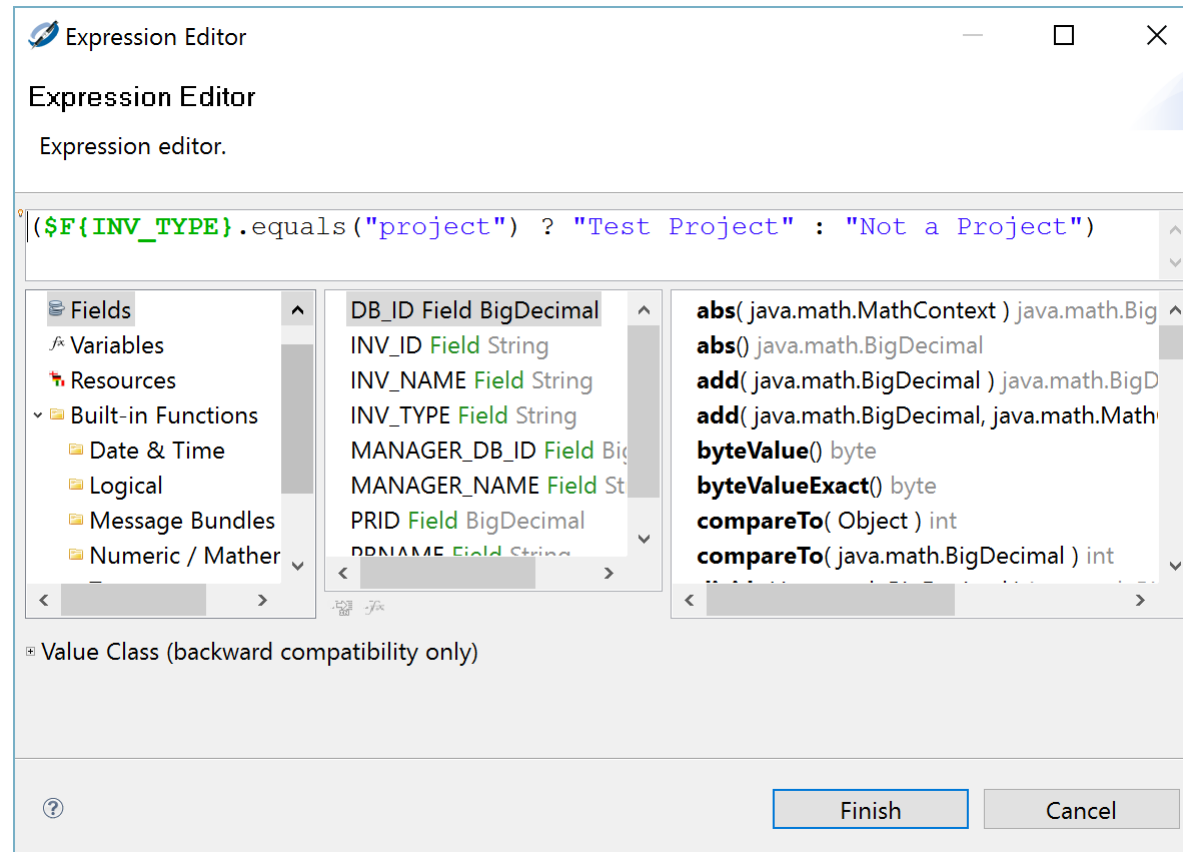
- The static Text Object allows you to put permanent unchanging text on a report. In our example lets add the Static Text object into the Title Band.



- Exercise: Add Static Text to your report.

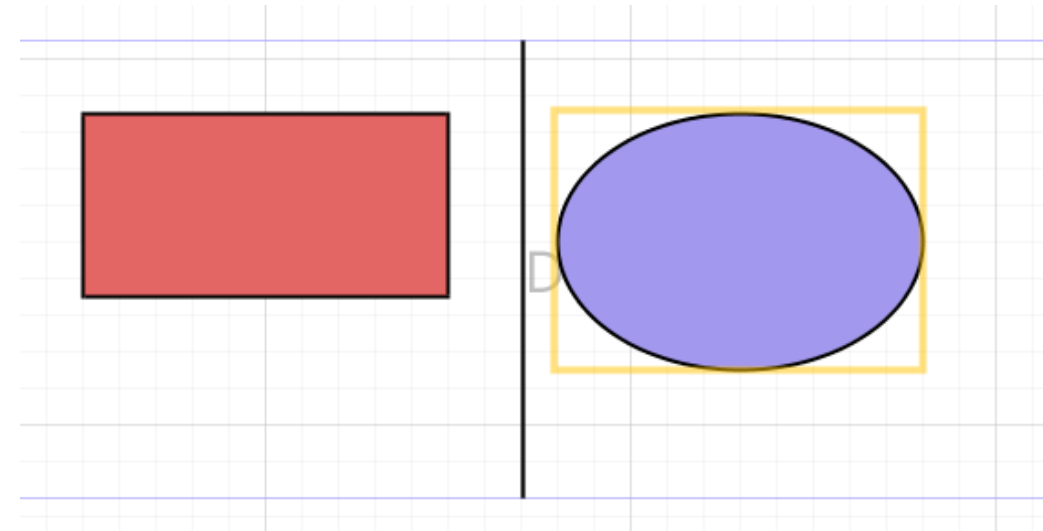
Basic Report Objects: Text Field

- The Text Field is a dynamic field you can place on a report. The text field is unique in that it lets the user pull in fields, parameters, and implement java functions to these values using an expression editor.
- Exercise: Add a text field that modifies the “Is Active” field to be a yes or no versus a 1 or 0.



Basic Report Objects: Shapes

- In Jaspersoft studio there are 3 shapes that can be added to a report.
 - Line
 - Rectangle
 - Ellipse
- Each shape can have it's size, direction, and color changed via the appearance tab.
- Exercise: Add a line to separate two area of a report.
- Exercise: Add a rectangle to the report and change the color.
- Exercise: Add an ellipse to the report and change the size.



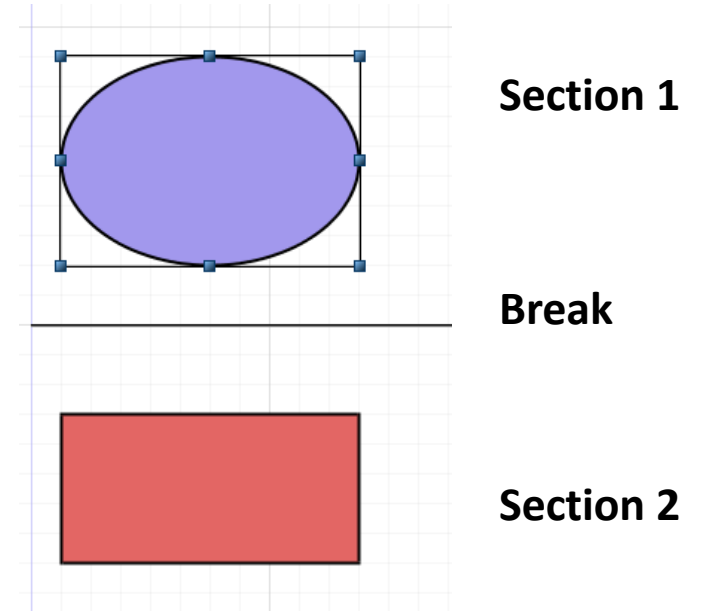
Basic Report Objects: Frames

- A frame allows the user to group objects in a single area. Utilizing frames in your report can keep your report organized and clean.
- A frame can be configured to auto size based on the size of its content. This allows the user to not worry about the text fields over stretching the section in which they are present.
- Example below. If the Objective section has more text than the Accomplishment section they will still remain the same size. This is due to both sections are in the same Frame.

<div data-bbox="435 748 1233 778" data-label="Text"> <div data-bbox="435 778 1233 808" data-label="Text"> <div data-bbox="435 808 1233 838" data-label="Text"> <div data-bbox="435 838 1233 858" data-label="Text"> <div data-bbox="435 858 1233 888" data-label="Text"> <div data-bbox="435 888 1233 918" data-label="Text"> <div data-bbox="435 918 1233 948" data-label="Text"> <div data-bbox="435 948 1233 978" data-label="Text"> <div data-bbox="435 978 1233 1008" data-label="Text"> <div data-bbox="435 1008 1233 1038" data-label="Text"> <div data-bbox="435 1038 1233 1068" data-label="Text"> <div data-bbox="435 1068 1233 1098" data-label="Text"> <div data-bbox="435 1098 1233 1128" data-label="Text"> <div data-bbox="435 1128 1233 1158" data-label="Text"> <div data-bbox="435 1158 1233 1188" data-label="Text"> <div data-bbox="435 1188 1233 1218" data-label="Text"> <div data-bbox="435 1218 1233 1248" data-label="Text"> <div data-bbox="435 1248 1233 1278" data-label="Text"> <div data-bbox="435 1278 1233 1308" data-label="Text"> <div data-bbox="435 1308 1233 1338" data-label="Text"> <div data-bbox="435 1338 1233 1368" data-label="Text"> <div data-bbox="435 1368 1233 1398" data-label="Text"> <div data-bbox="435 1398 1233 1428" data-label="Text"> <div data-bbox="435 1428 1233 1428" data-label="Text"> <div data-bbox="435 1457 1233 1428" data-label="Text"> <div data-bbox="435 1487 1233 1428" data-label="Text"> <div data-bbox="435 1517 1233 1428" data-label="Text"> <div data-bbox="435 1547 1233 1428" data-label="Text"> <div data-bbox="435 1577 1233 1428" data-label="Text"> <div data-bbox="435 1607 1233 1428" data-label="Text"> <div data-bbox="435 1637 1233 1428" data-label="Text"> <div data-bbox="435 1667 1233 1428" data-label="Text"> <div data-bbox="435 1697 1233 1428" data-label="Text"> <div data-bbox="435 1727 1233 1428" data-label="Text"> <div data-bbox="435 1757 1233 1428" data-label="Text"> <div data-bbox="435 1787 1233 1428" data-label="Text"> <div data-bbox="435 1817 1233 1428" data-label="Text"> <div data-bbox="435 1847 1233 1428" data-label="Text"> <div data-bbox="435 1877 1233 1428" data-label="Text"> <div data-bbox="435 1907 1233 1428" data-label="Text"> <div data-bbox="435 1937 1233 1428" data-label="Text"> <div data-bbox="435 1967 1233 1428" data-label="Text"> <div data-bbox="435 1997 1233 1428" data-label="Text"> <div data-bbox="435 2027 1233 1428" data-label="Text"> <div data-bbox="435 2057 1233 1428" data-label="Text"> <div data-bbox="435 2087 1233 1428" data-label="Text"> <div data-bbox="435 2117 1233 1428" data-label="Text"> <div data-bbox="435 2147 1233 1428" data-label="Text"> <div data-bbox="435 2177 1233 1428" data-label="Text"> <div data-bbox="435 2207 1233 1428" data-label="Text"> <div data-bbox="435 2237 1233 1428" data-label="Text"> <div data-bbox="435 2267 1233 1428" data-label="Text"> <div data-bbox="435 2297 1233 1428" data-label="Text"> <div data-bbox="435 2327 1233 1428" data-label="Text"> <div data-bbox="435 2357 1233 1428" data-label="Text"> <div data-bbox="435 2387 1233 1428" data-label="Text"> <div data-bbox="435 2417 1233 1428" data-label="Text"> <div data-bbox="435 2447 1233 1428" data-label="Text"> <div data-bbox="435 2477 1233 1428" data-label="Text"> <div data-bbox="435 2507 1233 1428" data-label="Text"> <div data-bbox="435 2537 1233 1428" data-label="Text"> <div data-bbox="435 2567 1233 1428" data-label="Text"> <div data-bbox="435 2597 1233 1428" data-label="Text"> <div data-bbox="435 2627 1233 1428" data-label="Text"> <div data-bbox="435 2657 1233 1428" data-label="Text"> <div data-bbox="435 2687 1233 1428" data-label="Text"> <div data-bbox="435 2717 1233 1428" data-label="Text"> <div data-bbox="435 2747 1233 1428" data-label="Text"> <div data-bbox="435 2777 1233 1428" data-label="Text"> <div data-bbox="435 2807 1233 1428" data-label="Text"> <div data-bbox="435 2837 1233 1428" data-label="Text"> <div data-bbox="435 2867 1233 1428" data-label="Text"> <div data-bbox="435 2897 1233 1428" data-label="Text"> <div data-bbox="435 2927 1233 1428" data-label="Text"> <div data-bbox="435 2957 1233 1428" data-label="Text"> <div data-bbox="435 2987 1233 1428" data-label="Text"> <div data-bbox="435 3017 1233 1428" data-label="Text"> <div data-bbox="435 3047 1233 1428" data-label="Text"> <div data-bbox="435 3077 1233 1428" data-label="Text"> <div data-bbox="435 3107 1233 1428" data-label="Text"> <div data-bbox="435 3137 1233 1428" data-label="Text"> <div data-bbox="435 3167 1233 1428" data-label="Text"> <div data-bbox="435 3197 1233 1428" data-label="Text"> <div data-bbox="435 3227 1233 1428" data-label="Text"> <div data-bbox="435 3257 1233 1428" data-label="Text"> <div data-bbox="435 3287 1233 1428" data-label="Text"> <div data-bbox="435 3317 1233 1428" data-label="Text"> <div data-bbox="435 3347 1233 1428" data-label="Text"> <div data-bbox="435 3377 1233 1428" data-label="Text"> <div data-bbox="435 3407 1233 1428" data-label="Text"> <div data-bbox="435 3437 1233 1428" data-label="Text"> <div data-bbox="435 3467 1233 1428" data-label="Text"> <div data-bbox="435 3497 1233 1428" data-label="Text"> <div data-bbox="435 3527 1233 1428" data-label="Text"> <div data-bbox="435 3557 1233 1428" data-label="Text"> <div data-bbox="435 3587 1233 1428" data-label="Text"> <div data-bbox="435 3617 1233 1428" data-label="Text"> <div data-bbox="435 3647 1233 1428" data-label="Text"> <div data-bbox="435 3677 1233 1428" data-label="Text"> <div data-bbox="435 3707 1233 1428" data-label="Text"> <div data-bbox="435 3737 1233 1428" data-label="Text"> <div data-bbox="435 3767 1233 1428" data-label="Text"> <div data-bbox="435 3797 1233 1428" data-label="Text"> <div data-bbox="435 3827 1233 1428" data-label="Text"> <div data-bbox="435 3857 1233 1428" data-label="Text"> <div data-bbox="435 3887 1233 1428" data-label="Text"> <div data-bbox="435 3917 1233 1428" data-label="Text"> <div data-bbox="435 3946 1233 1428" data-label="Text"> <div data-bbox="435 3976 1233 1428" data-label="Text"> <div data-bbox="435 4006 1233 1428" data-label="Text"> <div data-bbox="435 4036 1233 1428" data-label="Text"> <div data-bbox="435 4066 1233 1428" data-label="Text"> <div data-bbox="435 4096 1233 1428" data-label="Text"> <div data-bbox="435 4126 1233 1428" data-label="Text"> <div data-bbox="435 4156 1233 1428" data-label="Text"> <div data-bbox="435 4186 1233 1428" data-label="Text"> <div data-bbox="435 4216 1233 1428" data-label="Text"> <div data-bbox="435 4246 1233 1428" data-label="Text"> <div data-bbox="435 4276 1233 1428" data-label="Text"> <div data-bbox="435 4306 1233 1428" data-label="Text"> <div data-bbox="435 4336 1233 1428" data-label="Text"> <div data-bbox="435 4366 1233 1428" data-label="Text"> <div data-bbox="435 4396 1233 1428" data-label="Text"> <div data-bbox="435 4426 1233 1428" data-label="Text"> <div data-bbox="435 4456 1233 1428" data-label="Text"> <div data-bbox="435 4486 1233 1428" data-label="Text"> <div data-bbox="435 4516 1233 1428" data-label="Text"> <div data-bbox="435 4546 1233 1428" data-label="Text"> <div data-bbox="435 4576 1233 1428" data-label="Text"> <div data-bbox="435 4606 1233 1428" data-label="Text"> <div data-bbox="435 4636 1233 1428" data-label="Text"> <div data-bbox="435 4666 1233 1428" data-label="Text"> <div data-bbox="435 4696 1233 1428" data-label="Text"> <div data-bbox="435 4726 1233 1428" data-label="Text"> <div data-bbox="435 4756 1233 1428" data-label="Text"> <div data-bbox="435 4786 1233 1428" data-label="Text"> <div data-bbox="435 4816 1233 1428" data-label="Text"> <div data-bbox="435 4846 1233 1428" data-label="Text"> <div data-bbox="435 4876 1233 1428" data-label="Text"> <div data-bbox="435 4906 1233 1428" data-label="Text"> <div data-bbox="435 4936 1233 1428" data-label="Text"> <div data-bbox="435 4966 1233 1428" data-label="Text"> <div data-bbox="435 4996 1233 1428" data-label="Text"> <div data-bbox="435 5026 1233 1428" data-label="Text"> <div data-bbox="435 5056 1233 1428" data-label="Text"> <div data-bbox="435 5086 1233 1428" data-label="Text"> <div data-bbox="435 5116 1233 1428" data-label="Text"> <div data-bbox="435 5146 1233 1428" data-label="Text"> <div data-bbox="435 5176 1233 1428" data-label="Text"> <div data-bbox="435 5206 1233 1428" data-label="Text"> <div data-bbox="435 5236 1233 1428" data-label="Text"> <div data-bbox="435 5266 1233 1428" data-label="Text"> <div data-bbox="435 5296 1233 1428" data-label="Text"> <div data-bbox="435 5326 1233 1428" data-label="Text"> <div data-bbox="435 5356 1233 1428" data-label="Text"> <div data-bbox="435 5386 1233 1428" data-label="Text"> <div data-bbox="435 5416 1233 1428" data-label="Text"> <div data-bbox="435 5446 1233 1428" data-label="Text"> <div data-bbox="435 5476 1233 1428" data-label="Text"> <div data-bbox="435 5506 1233 1428" data-label="Text"> <div data-bbox="435 5536 1233 1428" data-label="Text"> <div data-bbox="435 5566 1233 1428" data-label="Text"> <div data-bbox="435 5596 1233 1428" data-label="Text"> <div data-bbox="435 5626 1233 1428" data-label="Text"> <div data-bbox="435 5656 1233 1428" data-label="Text"> <div data-bbox="435 5686 1233 1428" data-label="Text"> <div data-bbox="435 5716 1233 1428" data-label="Text"> <div data-bbox="435 5746 1233 1428" data-label="Text"> <div data-bbox="435 5776 1233 1428" data-label="Text"> <div data-bbox="435 5806 1233 1428" data-label="Text"> <div data-bbox="435 5836 1233 1428" data-label="Text"> <div data-bbox="435 5866 1233 1428" data-label="Text"> <div data-bbox="435 5896 1233 1428" data-label="Text"> <div data-bbox="435 5926 1233 1428" data-label="Text"> <div data-bbox="435 5956 1233 1428" data-label="Text"> <div data-bbox="435 5986 1233 1428" data-label="Text"> <div data-bbox="435 6016 1233 1428" data-label="Text"> <div data-bbox="435 6046 1233 1428" data-label="Text"> <div data-bbox="435 6076 1233 1428" data-label="Text"> <div data-bbox="435 6106 1233 1428" data-label="Text"> <div data-bbox="435 6136 1233 1428" data-label="Text"> <div data-bbox="435 6166 1233 1428" data-label="Text"> <div data-bbox="435 6196 1233 1428" data-label="Text"> <div data-bbox="435 6226 1233 1428" data-label="Text"> <div data-bbox="435 6256 1233 1428" data-label="Text"> <div data-bbox="435 6286 1233 1428" data-label="Text"> <div data-bbox="435 6316 1233 1428" data-label="Text"> <div data-bbox="435 6346 1233 1428" data-label="Text"> <div data-bbox="435 6376 1233 1428" data-label="Text"> <div data-bbox="435 6406 1233 1428" data-label="Text"> <div data-bbox="435 6435 1233 1428" data-label="Text"> <div data-bbox="435 6465 1233 1428" data-label="Text"> <div data-bbox="435 6495 1233 1428" data-label="Text"> <div data-bbox="435 6525 1233 1428" data-label="Text"> <div data-bbox="435 6555 1233 1428" data-label="Text"> <div data-bbox="435 6585 1233 1428" data-label="Text"> <div data-bbox="435 6615 1233 1428" data-label="Text"> <div data-bbox="435 6645 1233 1428" data-label="Text"> <div data-bbox="435 6675 1233 1428" data-label="Text"> <div data-bbox="435 6705 1233 1428" data-label="Text"> <div data-bbox="435 6735 1233 1428" data-label="Text"> <div data-bbox="435 6765 1233 1428" data-label="Text"> <div data-bbox="435 6795 1233 1428" data-label="Text"> <div data-bbox="435 6825 1233 1428" data-label="Text"> <div data-bbox="435 6855 1233 1428" data-label="Text"> <div data-bbox="435 6885 1233 1428" data-label="Text"> <div data-bbox="435 6915 1233 1428" data-label="Text"> <div data-bbox="435 6945 1233 1428" data-label="Text"> <div data-bbox="435 6975 1233 1428" data-label="Text"> <div data-bbox="435 7005 1233 1428" data-label="Text"> <div data-bbox="435 7035 1233 1428" data-label="Text"> <div data-bbox="435 7065 1233 1428" data-label="Text"> <div data-bbox="435 7095 1233 1428" data-label="Text"> <div data-bbox="435 7125 1233 1428" data-label="Text"> <div data-bbox="435 7155 1233 1428" data-label="Text"> <div data-bbox="435 7185 1233 1428" data-label="Text"> <div data-bbox="435 7215 1233 1428" data-label="Text"> <div data-bbox="435 7245 1233 1428" data-label="Text"> <div data-bbox="435 7275 1233 1428" data-label="Text"> <div data-bbox="435 7305 1233 1428" data-label="Text"> <div data-bbox="435 7335 1233 1428" data-label="Text"> <div data-bbox="435 7365 1233 1428" data-label="Text"> <div data-bbox="435 7395 1233 1428" data-label="Text"> <div data-bbox="435 7425 1233 1428" data-label="Text"> <div data-bbox="435 7455 1233 1428" data-label="Text"> <div data-bbox="435 7485 1233 1428" data-label="Text"> <div data-bbox="435 7515 1233 1428" data-label="Text"> <div data-bbox="435 7545 1233 1428" data-label="Text"> <div data-bbox="435 7575 1233 1428" data-label="Text"> <div data-bbox="435 7605 1233 1428" data-label="Text"> <div data-bbox="435 7635 1233 1428" data-label="Text"> <div data-bbox="435 7665 1233 1428" data-label="Text"> <div data-bbox="435 7695 1233 1428" data-label="Text"> <div data-bbox="435 7725 1233 1428" data-label="Text"> <div data-bbox="435 7755 1233 1428" data-label="Text"> <div data-bbox="435 7785 1233 1428" data-label="Text"> <div data-bbox="435 7815 1233 1428" data-label="Text"> <div data-bbox="435 7845 1233 1428" data-label="Text"> <div data-bbox="435 7875 1233 1428" data-label="Text"> <div data-bbox="435 7905 1233 1428" data-label="Text"> <div data-bbox="435 7935 1233 1428" data-label="Text"> <div data-bbox="435 7965 1233 1428" data-label="Text"> <div data-bbox="435 7995 1233 1428" data-label="Text"> <div data-bbox="435 8025 1233 1428" data-label="Text"> <div data-bbox="435 8055 1233 1428" data-label="Text"> <div data-bbox="435 8085 1233 1428" data-label="Text"> <div data-bbox="435 8115 1233 1428" data-label="Text"> <div data-bbox="435 8145 1233 1428" data-label="Text"> <div data-bbox="435 8175 1233 1428" data-label="Text"> <div data-bbox="435 8205 1233 1428" data-label="Text"> <div data-bbox="435 8235 1233 1428" data-label="Text"> <div data-bbox="435 8265 1233 1428" data-label="Text"> <div data-bbox="435 8295 1233 1428" data-label="Text"> <div data-bbox="435 8325 1233 1428" data-label="Text"> <div data-bbox="435 8355 1233 1428" data-label="Text"> <div data-bbox="435 8385 1233 1428" data-label="Text"> <div data-bbox="435 8415 1233 1428" data-label="Text"> <div data-bbox="435 8445 1233 1428" data-label="Text"> <div data-bbox="435 8475 1233 1428" data-label="Text"> <div data-bbox="435 8505 1233 1428" data-label="Text"> <div data-bbox="435 8535 1233 1428" data-label="Text"> <div data-bbox="435 8565 1233 1428" data-label="Text"> <div data-bbox="435 8595 1233 1428" data-label="Text"> <div data-bbox="435 8625 1233 1428" data-label="Text"> <div data-bbox="435 8655 1233 1428" data-label="Text"> <div data-bbox="435 8685 1233 1428" data-label="Text"> <div data-bbox="435 8715 1233 1428" data-label="Text"> <div data-bbox="435 8745 1233 1428" data-label="Text"> <div data-bbox="435 8775 1233 1428" data-label="Text"> <div data-bbox="435 8805 1233 1428" data-label="Text"> <div data-bbox="435 8835 1233 1428" data-label="Text"> <div data-bbox="435 8865 1233 1428" data-label="Text"> <div data-bbox="435 8895 1233 1428" data-label="Text"> <div data-bbox="435 8925 1233 1428" data-label="Text"> <div data-bbox="435 8954 1233 1428" data-label="Text"> <div data-bbox="435 8984 1233 1428" data-label="Text"> <div data-bbox="435 9014 1233 1428" data-label="Text"> <div data-bbox="435 9044 1233 1428" data-label="Text"> <div data-bbox="435 9074 1233 1428" data-label="Text"> <div data-bbox="435 9104 1233 1428" data-label="Text"> <div data-bbox="435 9134 1233 1428" data-label="Text"> <div data-bbox="435 9164 1233 1428" data-label="Text"> <div data-bbox="435 9194 1233 1428" data-label="Text"> <div data-bbox="435 9224 1233 1428" data-label="Text"> <div data-bbox="435 9254 1233 1428" data-label="Text"> <div data-bbox="435 9284 1233 1428" data-label="Text"> <div data-bbox="435 9314 1233 1428" data-label="Text"> <div data-bbox="435 9344 1233 1428" data-label="Text"> <div data-bbox="435 9374 1233 1428" data-label="Text"> <div data-bbox="435 9404 1233 1428" data-label="Text"> <div data-bbox="435 9434 1233 1428" data-label="Text"> <div data-bbox="435 9464 1233 1428" data-label="Text"> <div data-bbox="435 9494 1233 1428" data-label="Text"> <div data-bbox="435 9524 1233 1428" data-label="Text"> <div data-bbox="435 9554 1233 1428" data-label="Text"> <div data-bbox="435 9584 1233 1428" data-label="Text"> <div data-bbox="435 9614 1233 1428" data-label="Text"> <div data-bbox="435 9644 1233 1428" data-label="Text"> <div data-bbox="435 9674 1233 1428" data-label="Text"> <div data-bbox="435 9704 1233 1428" data-label="Text"> <div data-bbox="435 9734 1233 1428" data-label="Text"> <div data-bbox="435 9764 1233 1428" data-label="Text"> <div data-bbox="435 9794 1233 1428" data-label="Text"> <div data-bbox="435 9824 1233 1428" data-label="Text"> <div data-bbox="435 9854 1233 1428" data-label="Text"> <div data-bbox="435 9884 1233 1428" data-label="Text"> <div data-bbox="435 9914 1233 1428" data-label="Text"> <div data-bbox="435 9944 1233 1428" data-label="Text"> <div data-bbox="435 9974 1233 1428" data-label="Text"> <div data-bbox="435 10004 1233 1428" data-label="Text"> <div data-bbox="435 10034 1233 1428" data-label="Text"> <div data-bbox="435 10064 1233 1428" data-label="Text"> <div data-bbox="435 10094 1233 1428" data-label="Text"> <div data-bbox="435 10124 1233 1428" data-label="Text"> <div data-bbox="435 10154 1233 1428" data-label="Text"> <div data-bbox="435 10184 1233 1428" data-label="Text"> <div data-bbox="435 10214 1233 1428" data-label="Text"> <div data-bbox="435 10244 1233 1428" data-label="Text"> <div data-bbox="435 10274 1233 1428" data-label="Text"> <div data-bbox="435 10304 1233 1428" data-label="Text"> <div data-bbox="435 10334 1233 1428" data-label="Text"> <div data-bbox="435 10364 1233 1428" data-label="Text"> <div data-bbox="435 10394 1233 1428" data-label="Text"> <div data-bbox="435 10424 1233 1428" data-label="Text"> <div data-bbox="435 10454 1233 1428" data-label="Text"> <div data-bbox="435 10484 1233 1428" data-label="Text"> <div data-bbox="435 10514 1233 1428" data-label="Text"> <div data-bbox="435 10544 1233 1428" data-label="Text"> <div data-bbox="435 10574 1233 1428" data-label="Text"> <div data-bbox="435 10604 1233 1428" data-label="Text"> <div data-bbox="435 10634 1233 1428" data-label="Text"> <div data-bbox="435 10664 1233 1428" data-label="Text"> <div data-bbox="435 10694 1233 1428" data-label="Text"> <div data-bbox="435 10724 1233 1428" data-label="Text"> <div data-bbox="435 10754 1233 1428" data-label="Text"> <div data-bbox="435 10784 1233 1428" data-label="Text"> <div data-bbox="435 10814 1233 1428" data-label="Text"> <div data-bbox="435 10844 1233 1428" data-label="Text"> <div data-bbox="435 10874 1233 1428" data-label="Text"> <div data-bbox="435 10904 1233 1428" data-label="Text"> <div data-bbox="435 10934 1233 1428" data-label="Text"> <div data-bbox="435 10964 1233 1428" data-label="Text"> <div data-bbox="435 10994 1233 1428" data-label="Text"> <div data-bbox="435 11024 1233 1428" data-label="Text"> <div data-bbox="435 11054 1233 1428" data-label="Text"> <div data-bbox="435 11084 1233 1428" data-label="Text"> <div data-bbox="435 11114 1233 1428" data-label="Text"> <div data-bbox="435 11144 1233 1428" data-label="Text"> <div data-bbox="435 11174 1233 1428" data-label="Text"> <div data-bbox="435 11204 1233 1428" data-label="Text"> <div data-bbox="435 11234 1233 1428" data-label="Text"> <div data-bbox="435 11264 1233 1428" data-label="Text"> <div data-bbox="435 11294 1233 1428" data-label="Text"> <div data-bbox="435 11324 1233 1428" data-label="Text"> <div data-bbox="435 11354 1233 1428" data-label="Text"> <div data-bbox="435 11384 1233 1428" data-label="Text"> <div data-bbox="435 11414 1233 1428" data-label="Text"> <div data-bbox="435 11443 1233 1428" data-label="Text"> <div data-bbox="435 11473 1233 1428" data-label="Text"> <div data-bbox="435 11503 1233 1428" data-label="Text"> <div data-bbox="435 11533 1233 1428" data-label="Text"> <div data-bbox="435 11563 1233 1428" data-label="Text"> <div data-bbox="435 11593 1233 1428" data-label="Text"> <div data-bbox="435 11623 1233 1428" data-label="Text"> <div data-bbox="435 11653 1233 1428" data-label="Text"> <div data-bbox="435 11683 1233 1428" data-label="Text"> <div data-bbox="435 11713 1233 1428" data-label="Text"> <div data-bbox="435 11743 1233 1428" data-label="Text"> <div data-bbox="435 11773 1233 1428" data-label="Text"> <div data-bbox="435 11803 1233 1428" data-label="Text"> <div data-bbox="435 11833 1233 1428" data-label="Text"> <div data-bbox="435 11863 1233 1428" data-label="Text"> <div data-bbox="435 11893 1233 1428" data-label="Text"> <div data-bbox="435 11923 1233 1428" data-label="Text"> <div data-bbox="435 11953 1233 1428" data-label="Text"> <div data-bbox="435 11983 1233 1428" data-label="Text"> <div data-bbox="435 12013 1233 1428" data-label="Text"> <div data-bbox="435 12043 1233 1428" data-label="Text"> <div data-bbox="435 12073 1233 1428" data-label="Text"> <div data-bbox="435 12103 1233 1428" data-label="Text"> <div data-bbox="435 12133 1233 1428" data-label="Text"> <div data-bbox="435 12163 1233 1428" data-label="Text"> <div data-bbox="435 12193 1233 1428" data-label="Text"> <div data-bbox="435 12223 1233 1428" data-label="Text"> <div data-bbox="435 12253 1233 1428" data-label="Text"> <div data-bbox="435 12283 1233 1428" data-label="Text"> <div data-bbox="435 12313 1233 1428" data-label="Text"> <div data-bbox="435 12343 1233 1428" data-label="Text"> <div data-bbox="435 12373 1233 1428" data-label="Text"> <div data-bbox="435 12403 1233 1428" data-label="Text"> <div data-bbox="435 12433 1233 1428" data-label="Text"> <div data-bbox="435 12463 1233 1428" data-label="Text"> <div data-bbox="435 12493 1233 1428" data-label="Text"> <div data-bbox="435 12523 1233 1428" data-label="Text"> <div data-bbox="435 12553 1233 1428" data-label="Text"> <div data-bbox="435 12583 1233 1428" data-label="Text"> <div data-bbox="435 12613 1233 1428" data-label="Text"> <div data-bbox="435 12643 1233 1428" data-label="Text"> <div data-bbox="435 12673 1233 1428" data-label="Text"> <div data-bbox="435 12703 1233 1428" data-label="Text"> <div data-bbox="435 12733 1233 1428" data-label="Text"> <div data-bbox="435 12763 1233 1428" data-label="Text"> <div data-bbox="435 12793 1233 1428" data-label="Text"> <div data-bbox="435 12823 1233 1428" data-label="Text"> <div data-bbox="435 12853 1233 1428" data-label="Text"> <div data-bbox="435 12883 1233 1428" data-label="Text"> <div data-bbox="435 12913 1233 1428" data-label="Text"> <div data-bbox="435 12943 1233 1428" data-label="Text"> <div data-bbox="435 12973 1233 1428" data-label="Text"> <div data-bbox="435 13003 1233 1428" data-label="Text"> <div data-bbox="435 13033 1233 1428" data-label="Text"> <div data-bbox="435 13063 1233 1428" data-label="Text"> <div data-bbox="435 13093 1233 1428" data-label="Text"> <div data-bbox="435 13123 1233 1428" data-label="Text"> <div data-bbox="435 13153 1233 1428" data-label="Text"> <div data-bbox="435 13183 1233 1428" data-label="Text"> <div data-bbox="435 13213 1233 1428" data-label="Text"> <div data-bbox="435 13243 1233 1428" data-label="Text"> </div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div>
--

Basic Report Objects: Breaks

- A break allows the user to separate two sections to always be on the next page or column.
- Use a break to separate sections in a report manually instead of requiring the data to do so.
- Exercise: Separate a section on your report to always be on the next page regardless of the amount of data.

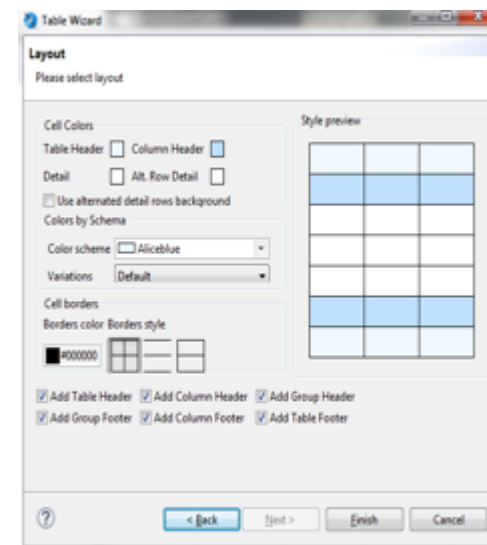
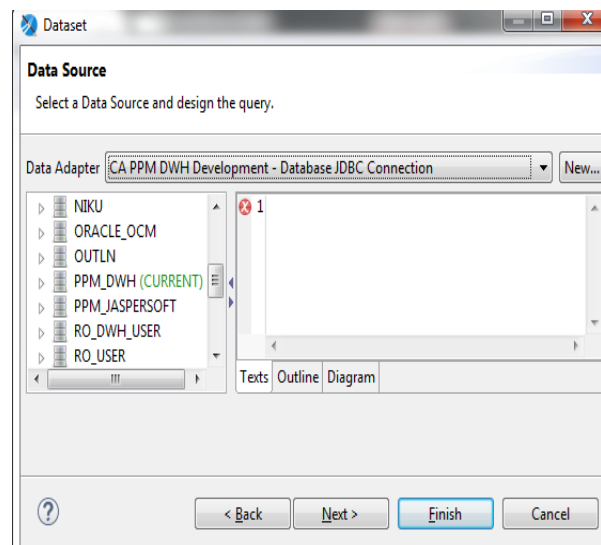
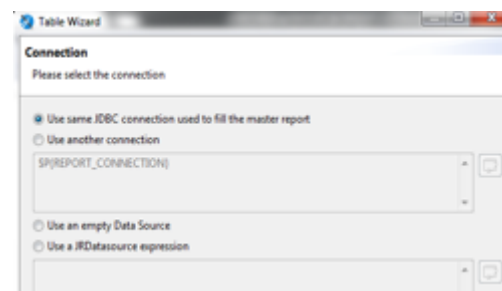
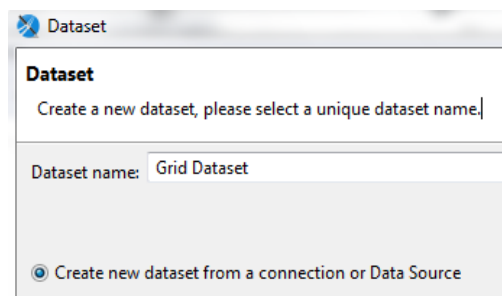
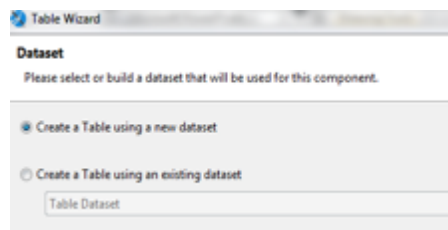


Basic Report Objects: Table

A Table element allows you to organize information in rows and columns. It is extremely powerful component that can replace the usage of sub reports in many situations.

Here is how you add and configure the Table:

- Drag the Table element from Basic element Palette
- You can either create a table using a new dataset or an existing dataset
- Create a table using new dataset provides you the option of selecting Dataset name, Data source to use, Put in the required query, Connection type, Table columns and Table layout.



Basic Report Objects: Table (Cont.)

- Creating a table using the existing dataset uses the already available dataset in the report

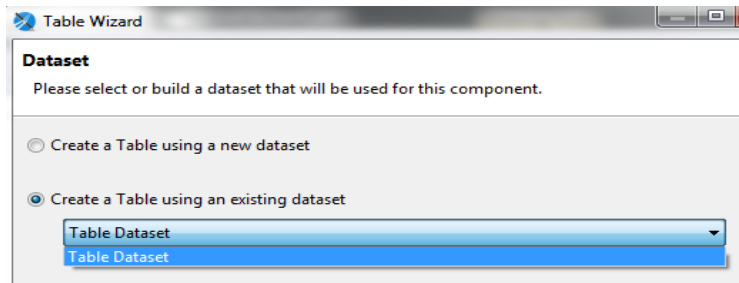
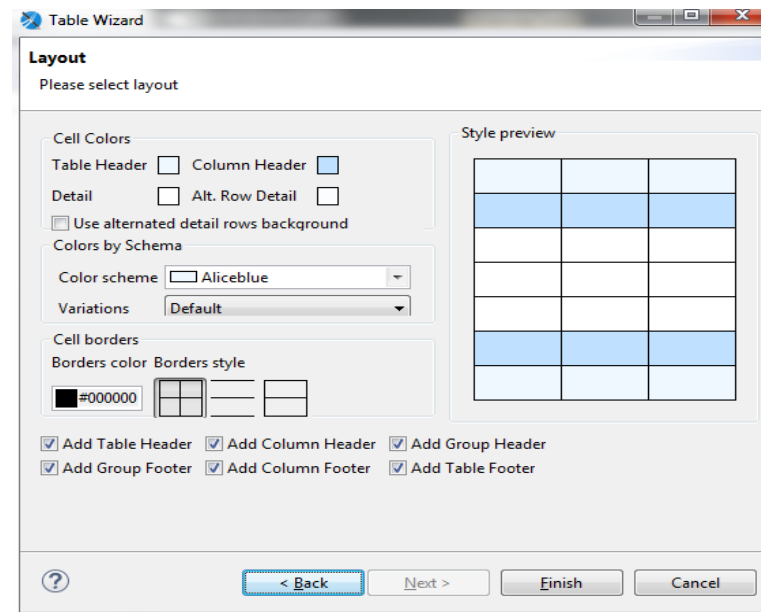


Table Layout:

- Table layout screen provides you the additional features that you can opt for the table.
- User can select the Colors for the Header (Table/Column) and Detail sections. You can also select different colors for the alternating rows
- User can choose Header/Footer for Table, Column or Group

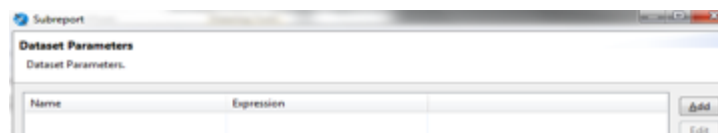
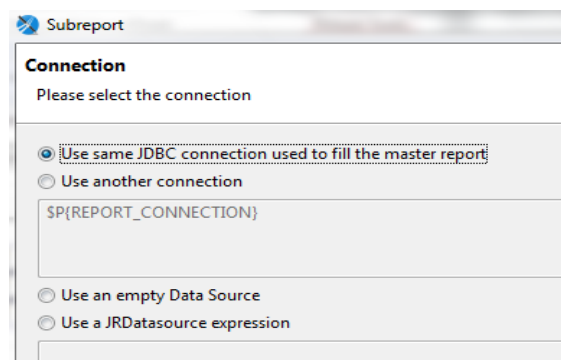
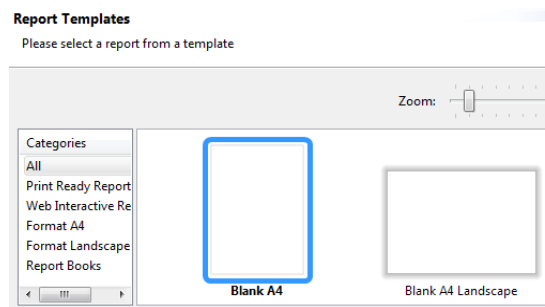
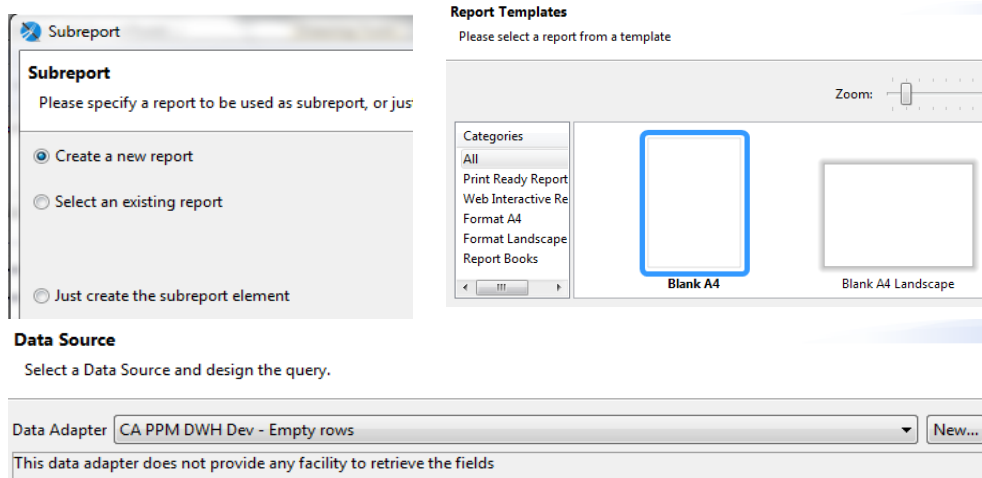


Basic Report Objects: Sub Report

- A Sub report is a report included inside another report. This allows the creation of very complex layouts with different portions of a single document filled using different data sources and reports

Here is how you add and configure the sub report:

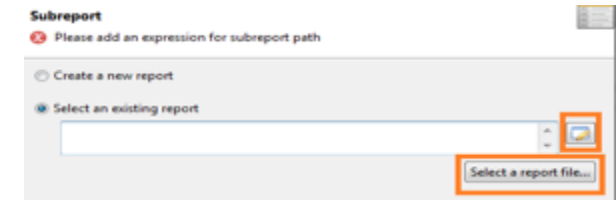
- Drag the sub report element from Basic element Palette
- You can either create a new report or select an existing report
- Create a new report provides you the option of selecting Report template, appropriate File name, Data source to use, Connection type and Parameters used for the report



- New report will be created upon completion of above steps.

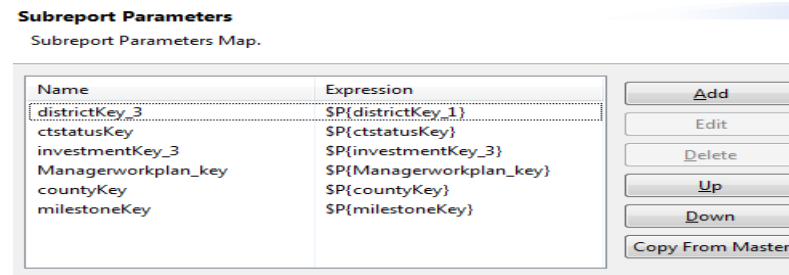
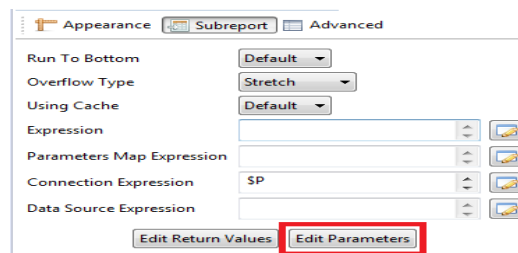
Basic Report Objects: Sub Report (Cont.)

- Select an existing report provides you the option of selecting already created jasper report
 - Use expression editor to type in the full path of already created report
 - Use “Select a report file” to select the sub report
- Upon selecting the sub report, choose the Connection Type and Parameters for the sub report and click on Finish to embed the sub report



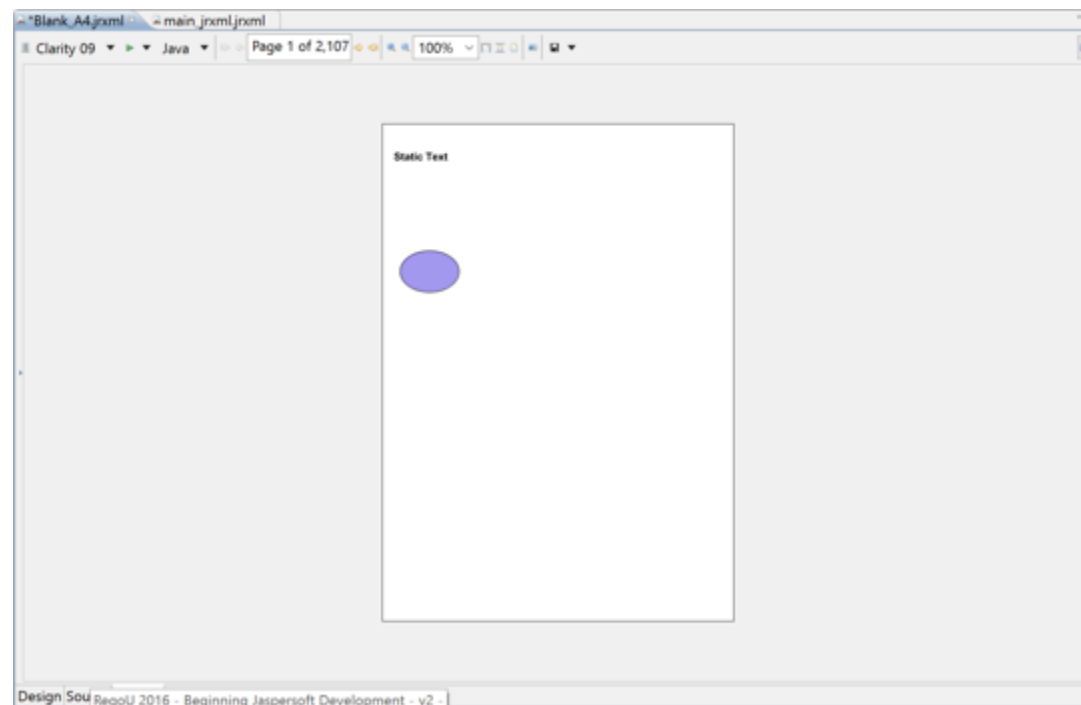
Subreport Parameters:

- Common use of subreport parameter is to pass the key of a record printed in the parent report in order to execute a query in the subreport
- To pass parameters from the master report to a subreport, you create a set of parameter name/object pairs that feed the parameters map of the subreport.
- To do this, click the **Edit Parameters** button on the Subreport tab of the Properties view to open the Subreport Parameters dialog. Add in the new Parameter or simply do “Copy from Master” to copy the parameters from the Main report



Previewing A Report

- To preview your report inside of Studio you must have a data adapter set. To run the report navigate to the report Design view.
- Select the Preview tab on the bottom .
- The report will try to run based on the data adapter set on the report.



Part 2: Conclusion –Live Demo



Questions?

*rego*University 2018

Let Rego be your guide.

Part III Agenda

Advanced Concepts

- Group data on a report
- Modifying the order of data in the report
 - Appearance properties overview
 - Adding an image to your report
- Adding a dynamic image to your report
 - Deploy the report to the server

*rego*University 2018

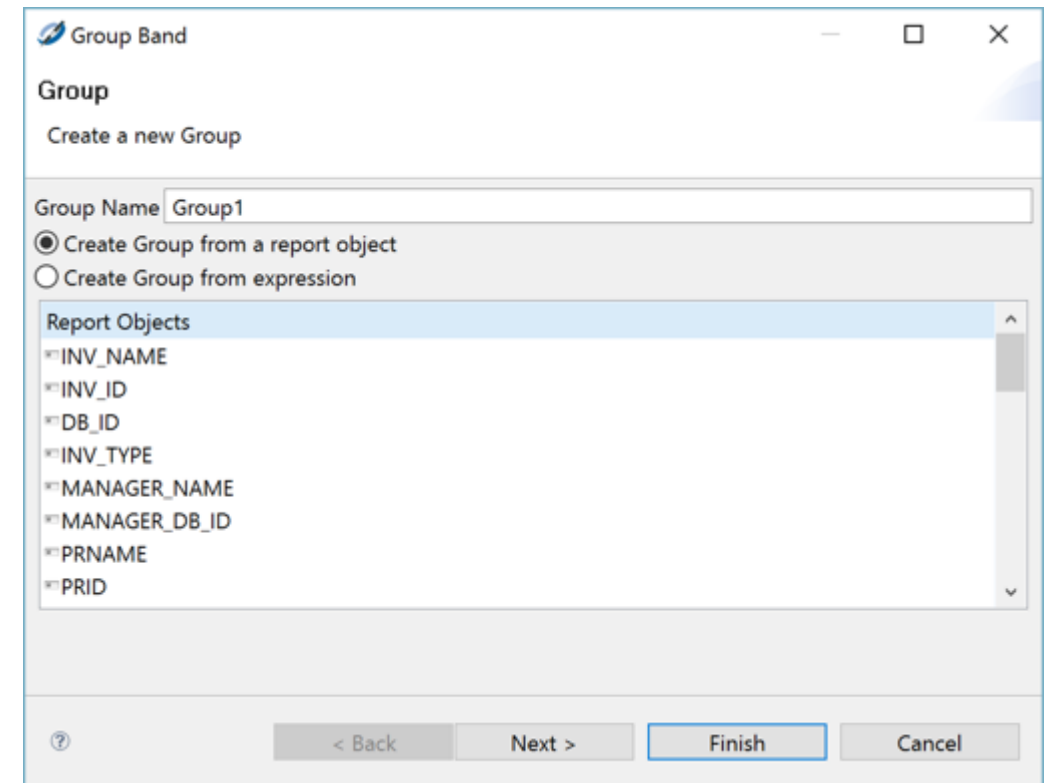
Part III: Advanced Concepts

- ✓ Group details by a field
 - ✓ Modifying the order
- ✓ Overview of the appearance properties

Grouping Data

- Grouping data on a report allows the user to have sections on a report based on a certain field or parameter.
- Example: A report needs to see all projects under a certain Project Manager. Instead of a line item for each Investment with the Project Manager on each, it can be grouped.

1. Navigate to the Outline Tab.
2. Right click on the Report (This will be the name of the jrxml).
3. Click the Create Group link
4. Select a field to group the data by or create an expression which will group the data.



Modifying the Order

- Modifying the order the data comes in can be done in two ways.
 1. Order By in the Query
 2. Create an Order By field
- To create an order by field, navigate to the Outline tab and right click on the Sort Fields element. Select the field you want to sort on.
- In the properties of the sort field you can select if it goes by ascending or descending.
- Exercise: Modify the order of your report.

New Sort Field

Create a SortField from Field or Variable

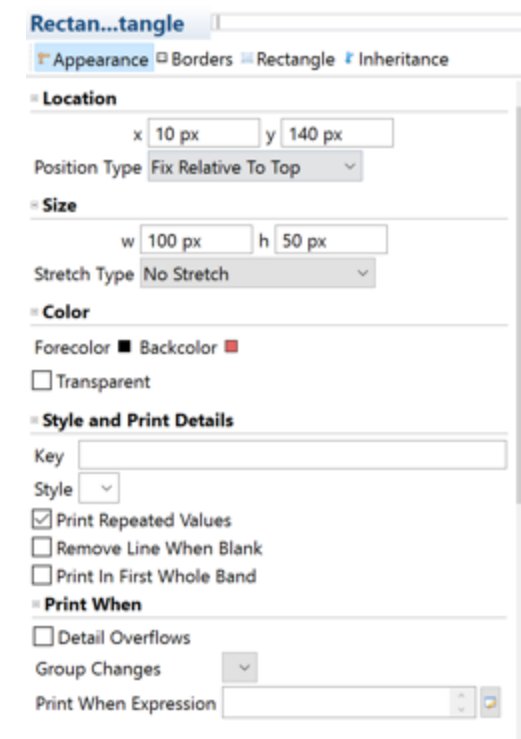
Dataset Fields and Variables

- INV_NAME
- INV_ID
- DB_ID
- INV_TYPE
- MANAGER_NAME
- MANAGER_DB_ID
- PRNAME
- PRID
- PAGE_NUMBER
- COLUMN_NUMBER
- REPORT_COUNT
- PAGE_COUNT
- COLUMN_COUNT

Finish Cancel

Appearance Properties Overview

- The appearance properties can be seen on all objects / elements in the system. It has a generic amount of properties that can be changed to modify the appearance of the object.
- To navigate to the appearance properties, right click on the object and select Show Properties.
 - Location: Object position on the report layout.
 - Size: The size of the object.
 - Stretch Type: Determines how the object should act if a section expands.
 - Color: The font color and background color can be set here.
 - Style and Print Details: The style and logic if something should be suppressed is stored here.
 - Print When: The detail overflow which controls if a field goes over the allotted space it can expand itself. The Print When Expression is a boolean expression that determines if the object should be visible or not.

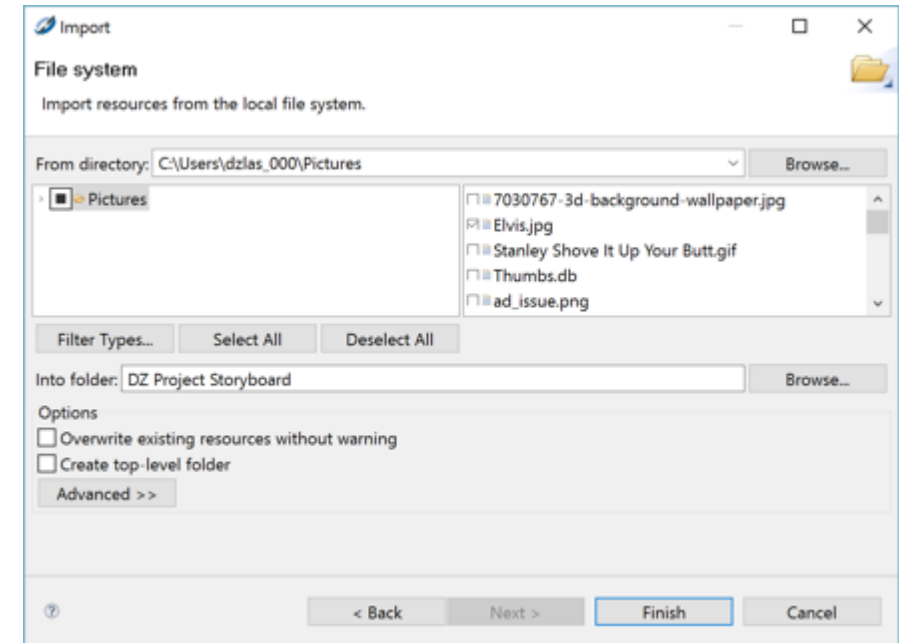


Adding an Image

- The user can add images into their reports using the Image object. The image needs to be a local resource stored in the project to be pulled over.
- Recommended approach is exporting the image into the project explorer and dragging it onto the report. The reason this is recommended is because the expression will auto fill itself with the location of the image file.

To add an image to your project follow the steps below.

1. Right Click on the Project of the report you want the image to display on and select Import.
2. In the Import Wizard click the File System option to import from.
3. Find the Image in your file system
4. Select the image with a checkbox.
5. Confirm the Into Folder is your project report folder.
6. Click Finish.
7. In the Project Explorer drag over the new image to add to the report.



Part III: Advanced Concepts

- ✓ Add a dynamic image
- ✓ Adding an Image to the Repository

Adding a Dynamic Image

- By adding an image from project explorer the expression is automatically filled out with the path of the image being used. This path however can be changed based on values in the report.
- Two ways to create dynamic images.
 - Using an expression based on a field to determine the path of the image.
 - Using the print when expression on each image to determine if the image should appear.
- Exercise: Add a checkbox image to the report. Using display only based on a Boolean equation on “Is Active”.
- Exercise: Based on the days late, show either the green, yellow, or red status indicator from the PPM resource images.

Adding an Image to the Repository

- If an image is going to be used on multiple reports, such as a logo, or a common status indicator, it is best practice to put this into a public folder on the repository.
- Using the repository folder to store the image is also the easiest way to determine the expression to find the image when running the report in Clarity.
- To add the image to the repository and add it to your report, follow the steps below.
 1. Navigate to the Repository and Click Add Folder. Name it Custom Images.
 2. On the newly created folder, click Add Resource -> File -> Image.
 3. In the Image, upload the image you want the report to reference.

Note: First look at the CA PPM images to make sure you can use one of their images first.
 4. Once the Image is in the repository, navigate back to Studio.
 5. In the Repository Explorer, navigate to the Image.
 6. Drag the image onto the report.
 7. Check the Expression. Notice it points towards a “repo” folder. If you tried running this in Studio, it would fail, but on the report server it will run correctly.

Adding an Image as Base64 text

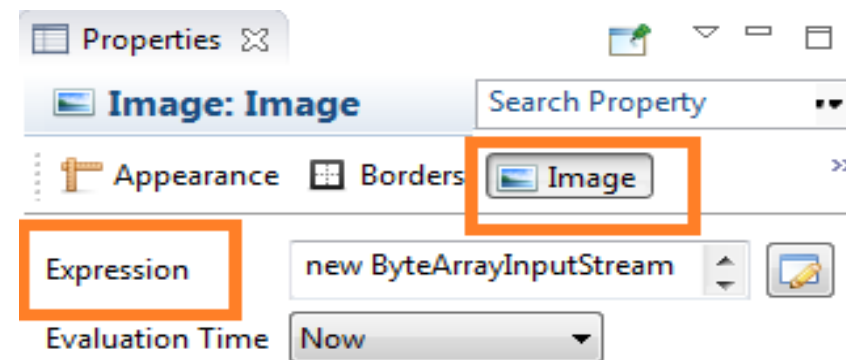
You can also embed image as Base64 text in jasper report. Following are the advantages of it:

- Need not save any image in the repository
- Report loads faster and quicker
- To embed image as Base64 text to your report, follow the steps below.
 1. Download the image to your desktop. Go to [Base 64 encoder](#)
 2. Upload the image and click on Encode. Copy the generated Base64 text to notepad
 3. Open the Studio. Import the Base64 package into the report (Report →Source tab→ Add the tag: `<import value="org.apache.commons.codec.binary.Base64"/>`)
 4. Drag the image element onto the report. Change the image expression as *new*
`ByteArrayInputStream(newBase64().decodeBase64("<base64text>".getBytes("UTF-8")))`
`<base64text>` is the generated Base64 text

```

1 <?xml version="1.0" encoding="UTF-8"?>
2 <!-- Created with JasperSoft Studio version 6.2.1.final using JasperReports Libra
3 <!-- 2018-02-20T16:36:00 -->
4 <jasperReport xmlns="http://jasperreports.sourceforge.net/jasperreports" xmlns:xs
5   <property name="com.jaspersoft.studio.data.sql.tables" value="" />
6   <property name="com.jaspersoft.studio.data.defaultdataadapter" value="Caltran
7   <property name="com.jaspersoft.studio.unit." value="pixel"/>
8   <property name="com.jaspersoft.studio.unit.pageWidth" value="inch"/>
9   <property name="com.jaspersoft.studio.unit.pageHeight" value="inch"/>
10  <property name="net.sf.jasperreports.export.xls.remove.empty.space.between.co
11  <property name="net.sf.jasperreports.export.xls.white.page.background" value=
12  <property name="net.sf.jasperreports.export.xls.detect.cell.type" value="true
13  <property name="net.sf.jasperreports.page.break.no.pagination" value="apply"/
14  <import value="org.apache.commons.codec.binary.Base64"/>
15  <style name="crossTab_ch" mode="Opaque" backgroundColor="#000000">
16    <box>
17      <pen lineWidth="0.5" lineColor="#000000"/>
18      <stroke lineWidth="0.5" lineColor="#000000"/>

```

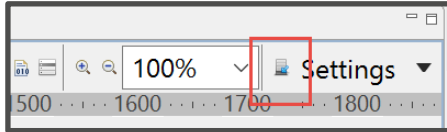


Part III: Advanced Concepts

✓ Deploying a Report

Deploy the Report to a Jaspersoft Server

- Deploying a report to a Jaspersoft server is a very seamless and easy process.
 1. Navigate to the Repository of the jaspersoft server you want to deploy the report to.
 2. Next click the Publish Report Icon in the top right of the report next to Settings.



3. Navigate to the folder you want to create the report.
4. Make sure you have the checkbox “Create Report Unit” checked.
5. Name the report Unit what you would name the report.

Note: The id is what will be the name of the file when exported out. Please name appropriately.

6. In the Data Source selection, select the data source from repository.
7. Click Finish.

- Exercise: Deploy a report to the environment.



Part 3: Conclusion –Live Demo



Questions?

*rego*University 2018

Let Rego be your guide.

Part IV Agenda

Advanced Concepts

- Out of the Box Input Control
- Creating your own Input Control
 - Creating a graph

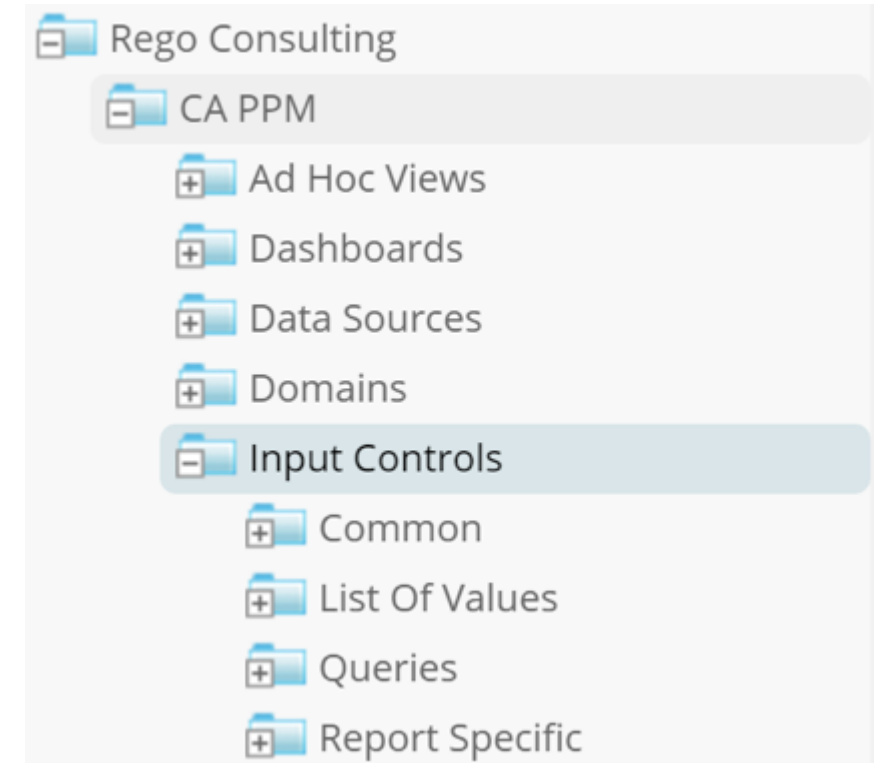
*rego*University 2018

Part IV: Advanced Concepts

- ✓ Add a parameter and input control
 - ✓ Create a Graph

Parameters and Input Controls

- Input Controls are the way jaspersoft sends parameters between the system and the report. They are external from the report.
- Parameters are variables on a report that allow the user to filter the reports data. These are setup on the report itself.
- There are a lot of out of the box input controls created already. Use these as much as possible.
 - Common: Holds the actual Input Control objects
 - List of Values: Static Values
 - Queries: The queries being used in the Input Controls to populate the options
 - Report Specific: Input Controls used for specific actions in reports.



How to Add an Input Control

- The following steps should be followed to add an input control to a report.

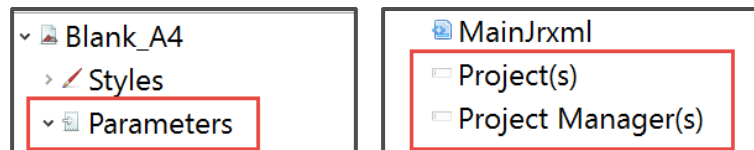
1. Create a parameter in the report. Write down the id that you give the parameter.

Note: If the input control is from the out of the box selections, make the parameter id and name the same as the input control's id.

2. Modify the query to accept the parameter either using the filter expression tab or directly in the sql.

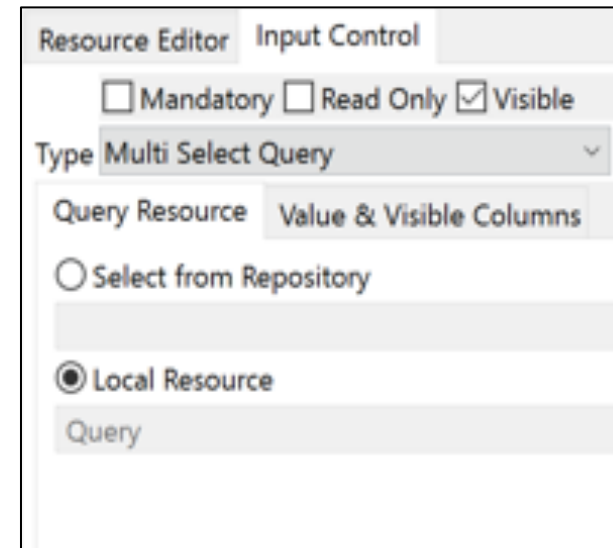
```
WHERE ($X{IN, null, c_inv_mgr} OR $X{IN, main_q.manager_id_key, c_inv_mgr})
AND ($X{IN, null, param_project_phase} OR $X{IN, main_q.lkp_project_phase, param_project_phase})
AND ($X{IN, null, c_work_status} OR $X{IN, main_q.obj_work_status, c_work_status})
```

3. Add / create the input control in the Report Unit.



Exercise: Add an out of the box input control for Project Manager.

Exercise: Adding a custom input control to filter for Is Active and Investment.



Charts Part 1

- HTML 5 Graphs in jaspersoft are an element similar to a text field but more complex.
- A user can use the master report's query as a data source to populate a graph or utilize creating a new dataset as well.
- The data from the graph can be set to change based on section, page, or even a full report.

To create an HTML 5 Graph, please follow the steps below.

1. Navigate to the Palette
2. Drag an HTML 5 graph element from the section Components Pro on to the report.
3. Select the type of graph when a dialogue pops up.

Note: This type can be changed at will at anytime.

Charts Part 2

Part 2 of charts will be going to the Chart Properties and navigating to the data set of the chart.

1. Right click on the Chart and select Edit Chart Properties. Chart Properties tab will appear.
2. The Chart properties allows the user to edit various visual properties of the chart.
3. Click on the tab Chart Data.
4. Chart Data allows the user to configure the data and parameters being sent into the chart.

Reset Type: Determines when the chart should refresh it's data.

Increment Type: Determines what level the chart data is being used at. Most commonly use none.

Sub Dataset: Determines the data set the chart is using. This can use the main report data set or a sub data set.

Note: We will create sub data set's in the advanced jaspersoft class.

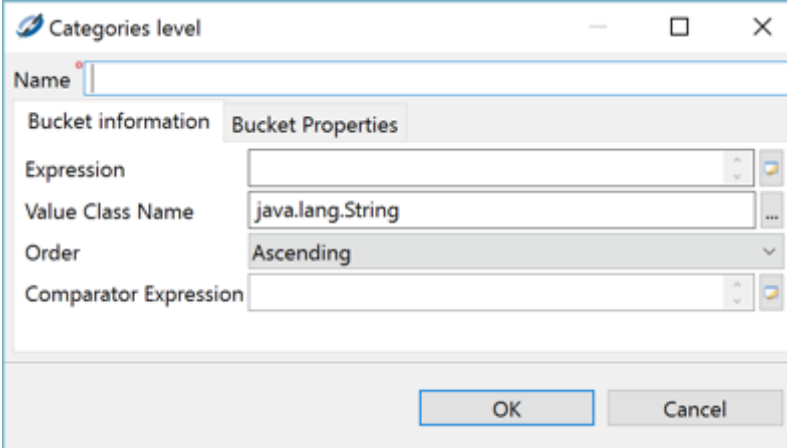
The screenshot shows the 'Chart Data' configuration window with three tabs: 'Chart Properties', 'Chart Data', and 'Hyperlink'. The 'Chart Data' tab is active, showing a 'Configuration' sub-tab. It contains the following fields:

- Reset Type**: A dropdown menu with 'Report' selected.
- Reset Group**: A dropdown menu.
- Increment Type**: A dropdown menu with 'None' selected.
- Increment Group**: A dropdown menu.
- Filter expression**: A text input field with a search icon on the right.
- Dataset Run**: A section containing a **Sub Dataset** dropdown menu with '[Report main dataset]' selected.

Charts Part 3 Category Levels

Part 3 of charts will be going over how to populate a Category

1. On the Chart Data tab, click the sub tab named Configuration.
2. The Categories Levels are the X Axis. You can modify Level 1 or create your own.
3. Click the Add button if you want to create a category, or click on the category you want to edit and click the Edit button.
4. In the Name field apply a value for the category.
Note: Name is just the representation in studio, no affect on the report.
5. In the Expression field, apply the field that will represent the category name.
6. Depending on the Expression Field's data type, edit the Value Class to represent the field type.
7. The Order field orders the categories. This can be set to None if you are already ordering them within the query.



The screenshot shows a 'Categories level' dialog box. It has a 'Name' field at the top. Below it are two tabs: 'Bucket information' and 'Bucket Properties'. The 'Bucket Properties' tab is active, showing fields for 'Expression', 'Value Class Name' (set to 'java.lang.String'), 'Order' (set to 'Ascending'), and 'Comparator Expression'. At the bottom right are 'OK' and 'Cancel' buttons.

Charts Part 4 Measures

Part 4 of charts will be going over how to populate a measure in the chart.

1. On the Chart Data tab, click the sub tab named Configuration.
2. The Measures are the Y Axis. You can modify Measure1 or create your own.
3. Click the Add button to make your own, or click on the measure you want to edit and click the Edit Button.
4. In the Name field apply a value for the category.
Note: Name is just the representation in studio, no affect on the report.
5. In the Label Expression field, apply either a field that will represent the label of the value, or input a String value of the label.
Ex: "Average Posted Timesheet".
6. In the Calculation field, select the calculation to be applied on the Valued field.
Note: If the data set has already done the calculation, select None.
7. In the Value Expression, select the field from the data set to be used as the value for the measure.
8. In the Value Class name, select the data type that the field selected in the Value Expression uses.

Name

☐ Hidden

Value definition Advanced properties

Label Expression

Calculation

Value Expression

Value Class Name

Increment Factory Class Name

Questions?



*rego*University 2018

Let Rego be your guide.

Thank You For Attending regoUniversity

Instructions for PMI credits

- Access your account at pmi.org
- Click on **Certifications**
- Click on **Maintain My Certification**
- Click on **Visit CCR's** button under the **Report PDU's**
- Click on **Report PDU's**
- Click on **Course or Training**
- Class Name = **regoUniversity**
- Course Number = **Session Number**
- Date Started = **Today's Date**
- Date Completed = **Today's Date**
- Hours Completed = **1 PDU per hour of class time**
- Training classes = **Technical**
- Click on **I agree** and **Submit**



Let us know how we can improve!
Don't forget to fill out the class survey.



Phone

888.813.0444



Email

info@regouniversity.com



Website

www.regouniversity.com