
SAP LUMIRA 2 - DESIGNER
USER MANUAL

TABLE OF CONTENTS

TABLE OF CONTENTS	2
EXERCISE 1 - INTRODUCTION	3
EXERCISE 2 - INTEROPERABILITY	8
EXERCISE 3 – CONTAINER COMPONENTS	10
EXERCISE 4 – FILTER COMPONENTS	45
EXERCISE 5 – WORKING WITH SCRIPTING	59
EXERCISE 6 – WORKING WITH MAP COMPONENT	85
EXERCISE 7 - SCORECARD.....	115
EXERCISE 8 - COMPOSITE.....	129
EXERCISE 9 – ADAPTIVE LAYOUT	141

Exercise 1 - INTRODUCTION

Objective of the Exercise

- Introduction to Lumira Designer
- Logging on and Layout
- Designer Modes

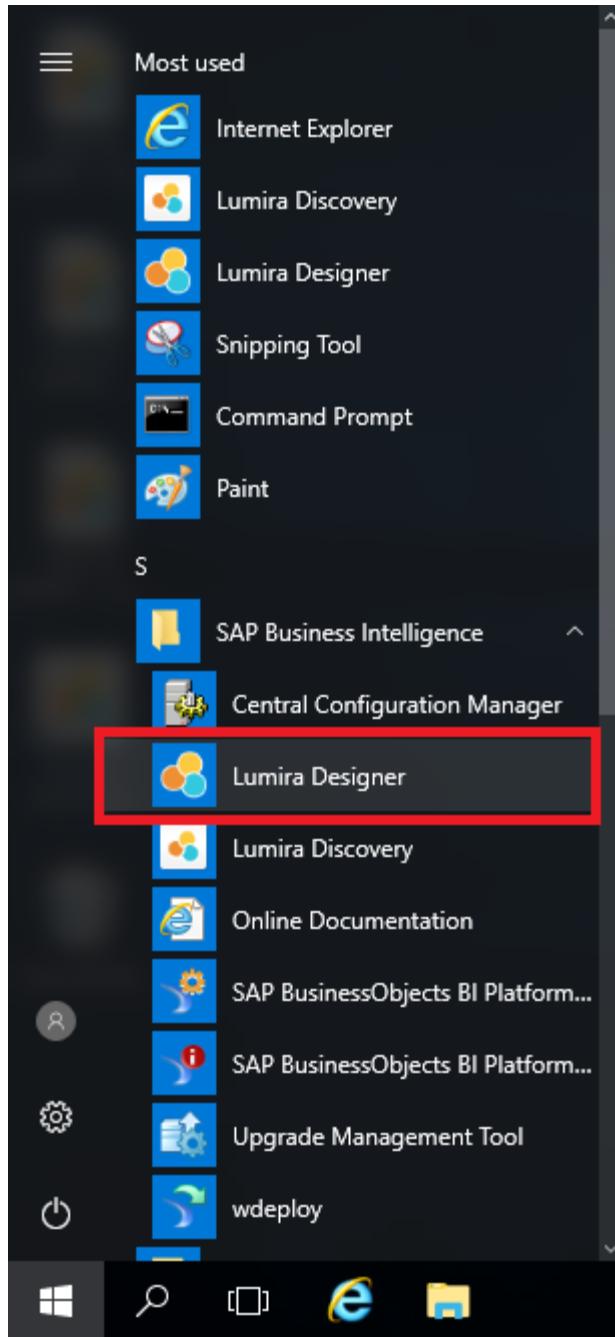
1 [Introduction to Lumira Designer](#)

SAP BusinessObjects Lumira enables customers to gain insights from trusted enterprise data sources and personal data, and to share those insights through interactive visualizations, stories, and tailored analysis applications with other users, on desktop browsers and mobile devices.

The Lumira Designer component is based on the Eclipse framework and is a developer-centric tool. Developers can create applications by dragging and dropping components onto the Layout and modify some of their properties using options built-in to Designer or using BI Action Language scripting.

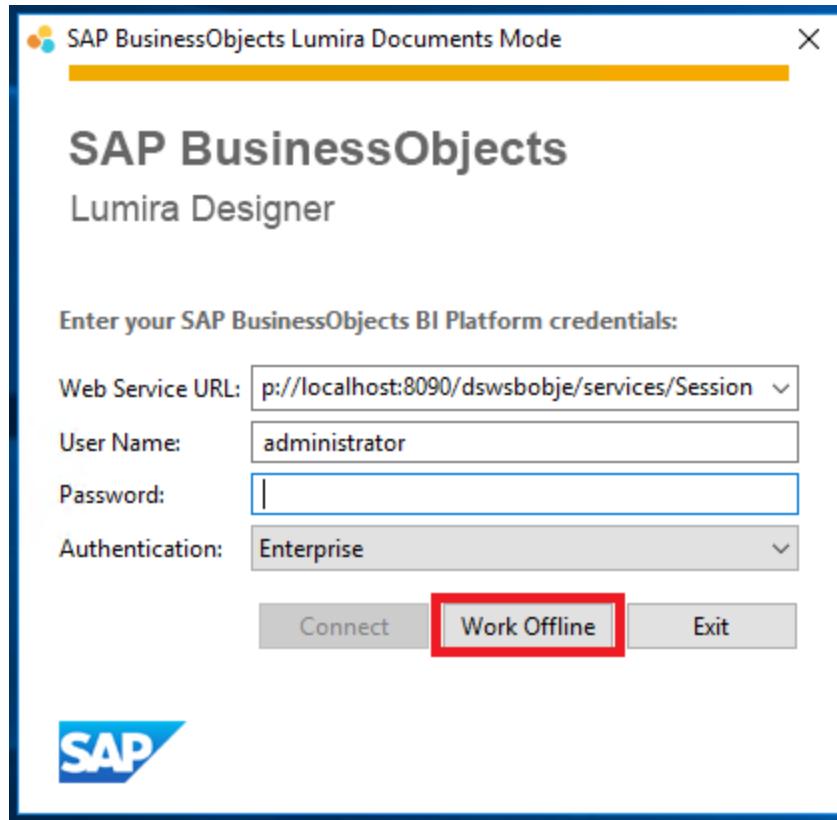
2 [Logging onto Lumira Designer](#)

Click on **Start button >> All programs >> SAP Business Intelligence >> Lumira Designer.**

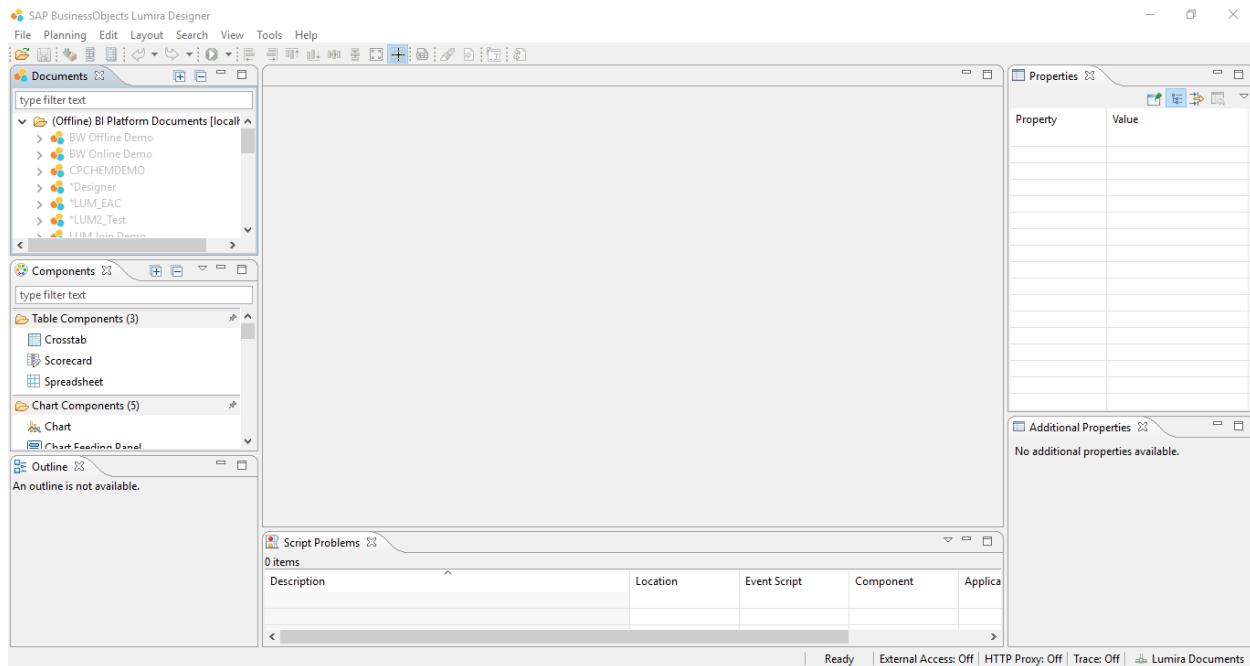


3 Layout of Lumira Designer

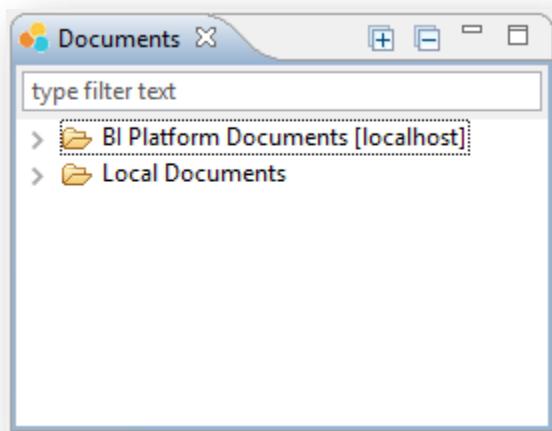
The SAP BusinessObjects Lumira Documents Mode popup opens up. We'll go over working on the BI Platform in a later section, for now select Work Offline. This will open up the Design application on Local Mode.



The **Home** page appears as follows. It is similar to the default layout of Design Studio. By default we see the Document, Components, Outline, Layout, Script, Properties and Additional Properties tabs.

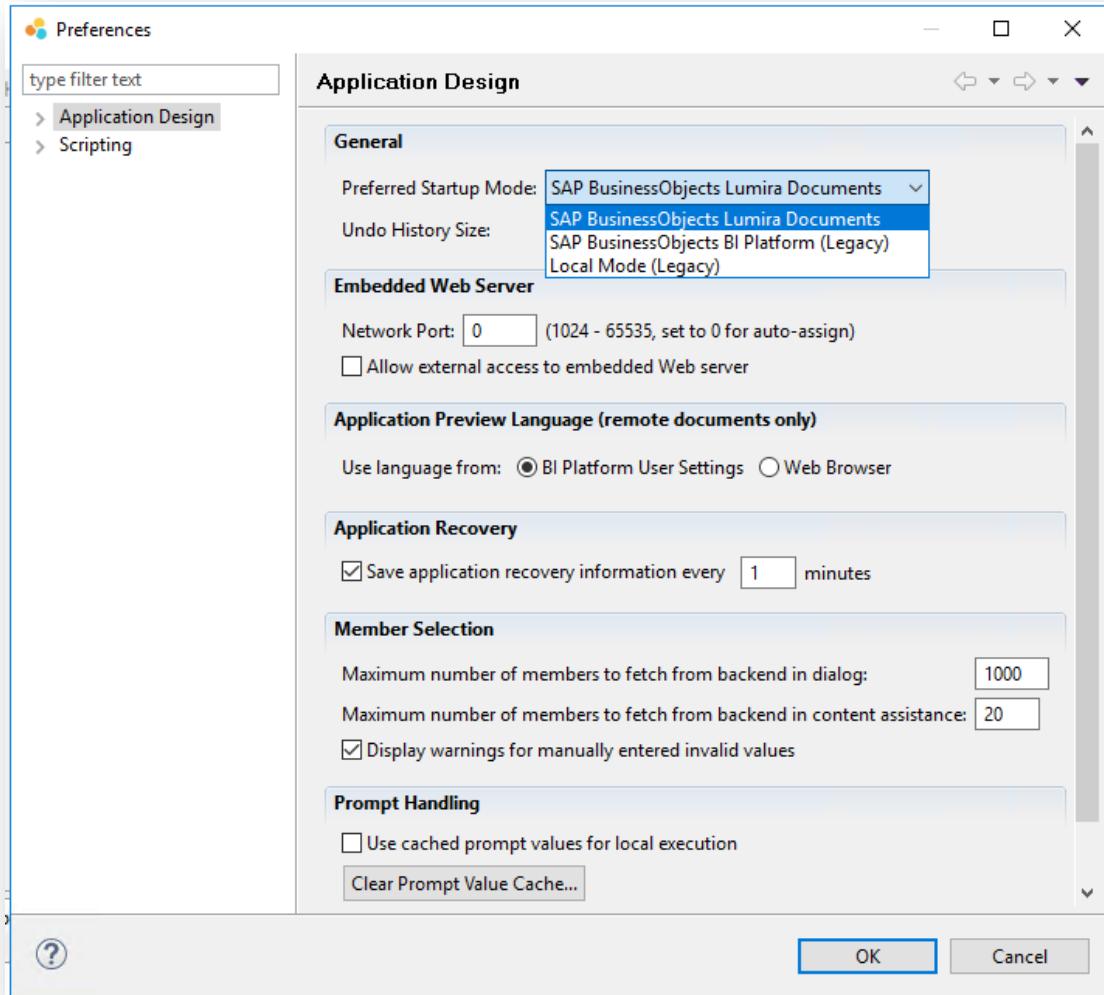


Note: New to Lumira 2.0 Designer is the Documents tab. This contains links to both documents created on the BI Platform and also Locally.



4 Designer Modes

Understanding Designer Modes is essential for understanding how SAP BusinessObjects Lumira 2.0 works and most other topics covered in this document. Designer Modes are present under **Tools>Preferences**. Under the Preferences dialog box select **Application Design>>General**. Select the dropdown next to **Preferred Startup Mode** to bring up all the different Designer Modes.



4.1 SAP BusinessObjects Lumira Documents Mode: This is the default mode for operating Lumira 2.0 Designer. In this mode, users create new applications in which they can make use of new 2.0 features, such as Composites and new Bookmarks. This new mode can handle both local documents and documents stored on the BI platform.

Note: *In 2.0, make sure that you create a document that is meant for productive use on the BI Platform from the very beginning. Bringing a document initially created on the local filesystem to the BI Platform at a later point in time has many limitations in 2.0.*

4.2 SAP BusinessObject BI Platform (Legacy)/ Local Mode (Legacy): These two were the standard modes in previous versions of Design Studio (DS 1.6, DS 1.5, etc.). Users work with those modes just like in 1.6, with the exception that SAP does not currently plan to enable creation of new 1.6 applications (through New or Save As options in the Application menu), but users will be able to open, edit and save existing 1.6 applications in these modes.

4.3 Set the Preferred Startup Mode to SAP BusinessObjects Lumira Documents.

Exercise 2 - INTEROPERABILITY

Objective of the Exercise

- Interoperability between Discovery and Designer
- Interoperability use-cases

1 Interoperability between Lumira Discovery and Lumira Designer

The output of Lumira 1.x versions – in the form of storyboards – could be published on the SAP BI Platform to be consumed by others. However, when these same storyboards require a professional touch, they were rebuilt using SAP Design Studio. For this professionalization, you might think of bringing multiple storyboards together in one dashboard, or in instances when cross-chart filtering, or complex filtering needed to be added to name a few. Design Studio was also needed in cases where a live connection to SAP BW was required.

Lumira 2.0 will dramatically simplify the above situation: end users who create storyboards and require these to be professionalized towards enterprise-wide dashboards, can now have this done very easily in the same Lumira 2.0 product. We call this **interoperability**: the seamless cooperation and interactivity of storyboards or mock ups, with enterprise wide authored dashboards. Interoperability thus means bi-directional communication between storyboards and dashboards.

We will go over some aspects of Interoperability in the introduction of Exercise 3 – Container Components.

Prerequisites: You have to install both the client tools in your local system.

2 Use cases for Interoperability

Let's make things a bit more concrete and discuss several use cases for this interoperability. It will help us understanding the value of interoperability:

1. Tuning a storyboard using Scripting

Lumira 2.0 Discovery allows for data exploration, which is blending and visualizations through storyboards. Its features are advanced yet focus on quick and easy creation of insights. Some more advanced features are missing on purpose; Discovery aims at end users and the easy way of insights gathering. If however an end user wants some advanced features applied to the storyboard, the Designer component of Lumira 2.0 offers the solution. For example, adding cross-chart filtering or high level drill down is not possible using Discovery but quite easily

applied in Designer. The solution is to open the storyboard in the Designer module, apply some simple guided script and save the storyboard.

2. Combining various storyboards

In the use case where the requirement is a dashboard that oversees a full LOB process. Spending, sales, finance or manufacturing metrics, they all need to be in and provide a 360° overview. The Discovery component might play a big role here, but probably will give you various storyboards that each cover the pieces (spend, sales etc) of the process. Designer now comes in by creating one dashboard application where you pick up the various storyboards.

3. Professionalize to an enterprise dashboard

The most obvious use case for interoperability is definitely the professionalization of your storyboards using SAP Lumira 2.0 Designer. The Designer module has basically no limitations when it comes transferring your storyboard(s) to enterprise wide dashboard applications. Designer allows users to add radio button groups, sliders, interactive commenting, responsive color-coded progress indicators, input fields, tab-strips, chart-pickers and so on. But also adding more data-sources, creating a scorecard or setting up an adaptive layout (when enlarging the browser, the graphs are automatically re-arranged).

4. The Governance hurdle

Designer can help set up roles and authorizations for users so that they only see the sections and data they are entitled to.

5. The data connection and blending challenge

One of the strong points SAP Lumira 2.0 Discovery is its data mashup, enrichment and blending capabilities. It is so easy to join data-sets from various source types (CSV, Xlsx, SQL, BW, HANA, Universes, Hadoop, SalesForce, JDBC, Cloudera, Vora, etc.,) and quickly discover the data. There is also blending where the same can be done on data sources with different grain. Well, all this enrichment and blending can be easily set-up in a Discovery storyboard and re-used in SAP Lumira 2.0 Designer.

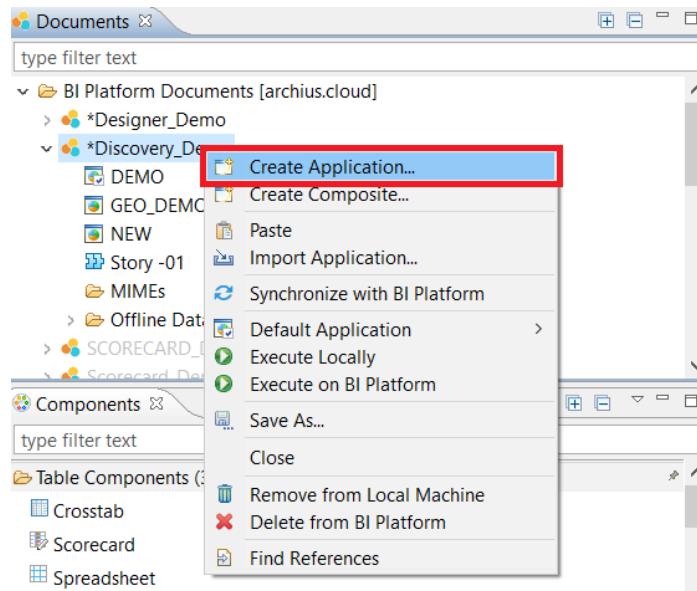
Exercise 3 – CONTAINER COMPONENTS

Objective of the Exercise

- Creating a new application in Lumira Designer
- Setting CSS Style
- Creating the Header
- Adding Grids, Panels, Pagebook and creating charts and crosstabs

1. Creating new application on the BI Platform

We will create a application we created using the Lumira Discovery tool. We will save this application under the Under the BI Platform Documents tab, right click on Discovery_Demo under the BI Platform Documents [localhost] and from the context menu select create new application.



Name the application as Designer_Demo and choose the Templates as Blank. Click on create.

2. Setting the CSS Style

Cascading Style Sheets (CSS) is a style sheet language used for describing the presentation of a document written in a markup language.

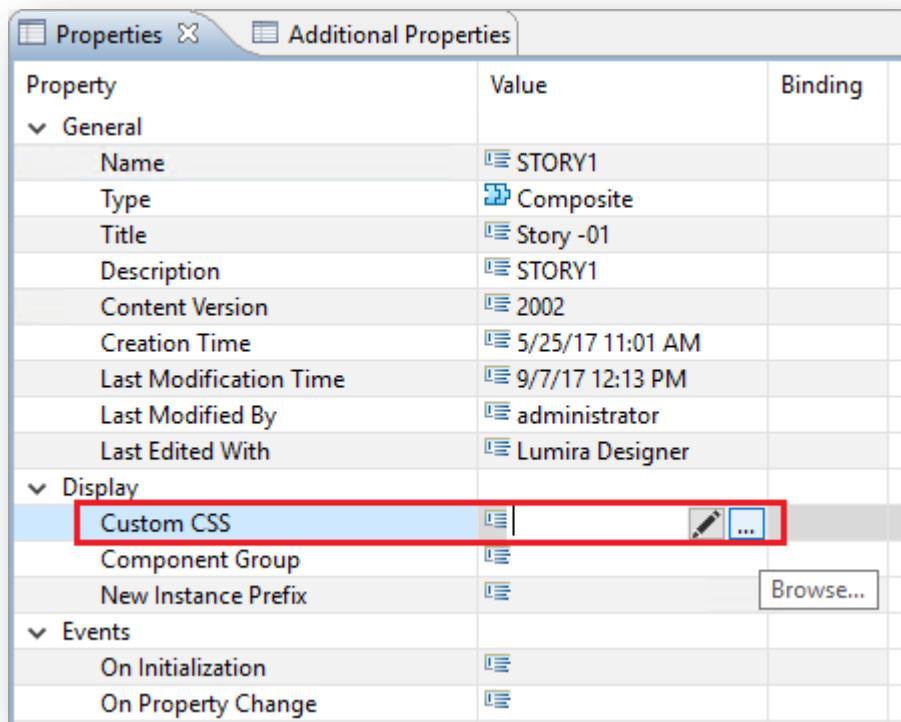
We use CSS files to enable us to format the application by changing look, font size, color, weight and style. You can browse and add a Custom Cascading style sheet file (one that was created and saved to the BI platform earlier) in the Custom CSS property of the Properties view of the Analysis Application. These styles will get applied to the components as per the styles defined in the CSS Style file.

For our application a CSS file was created and uploaded to **Public Folders/Demo** with the name *Lumiradesigner.css*. It has the following code:

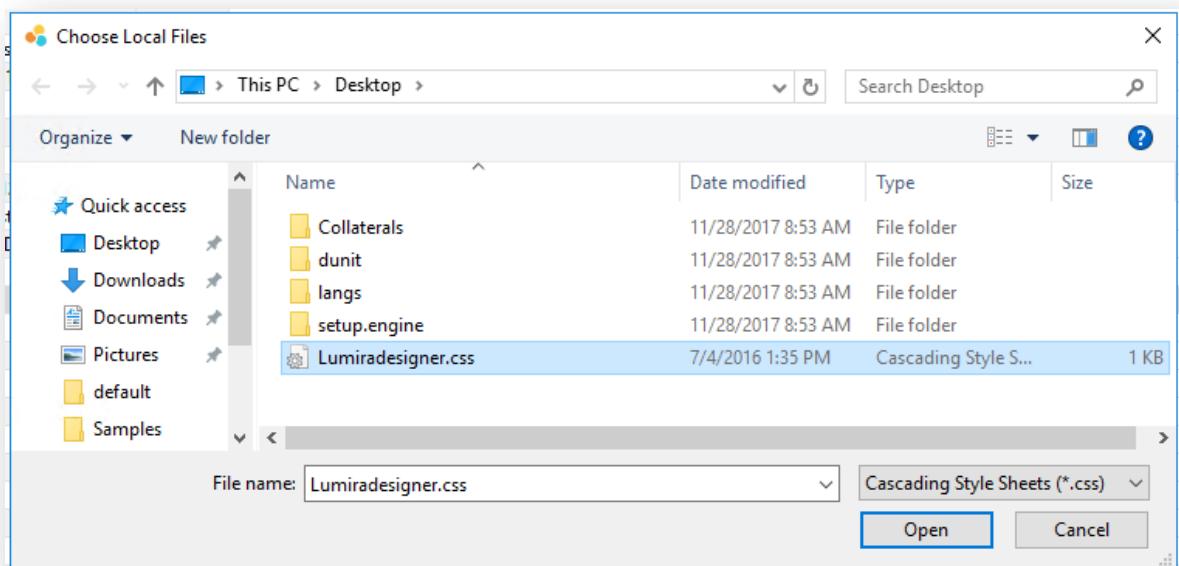
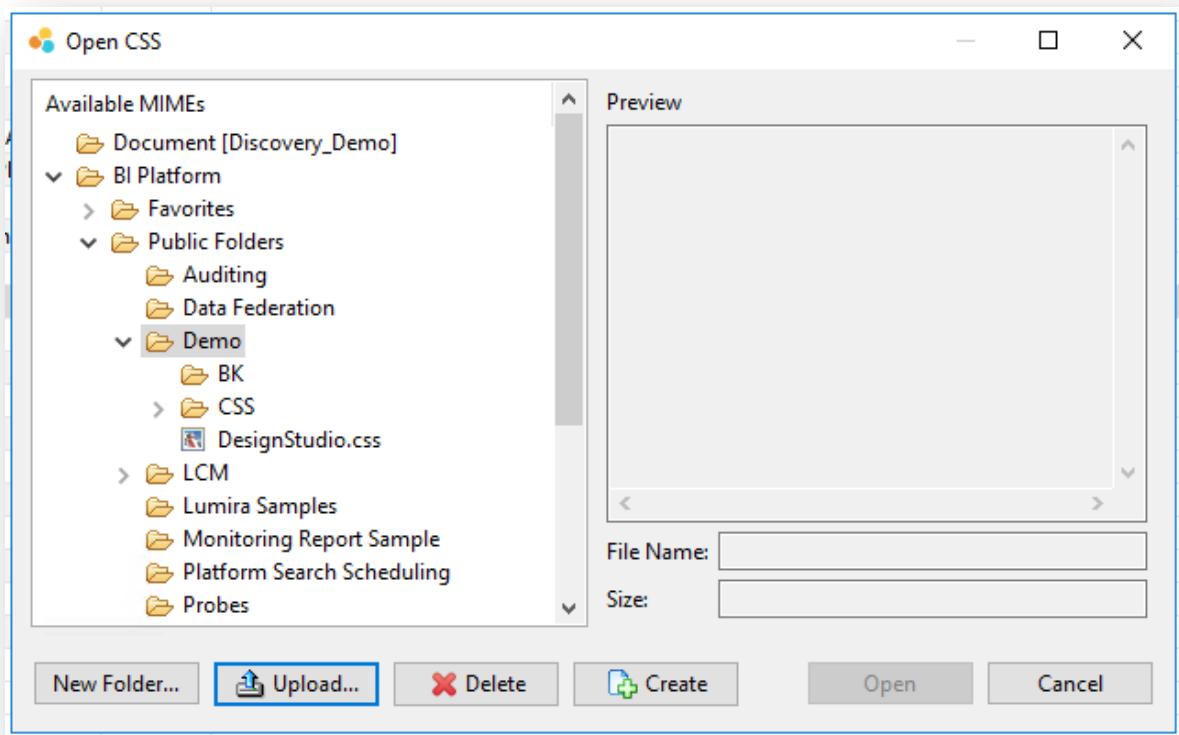
```
.panel-box {  
border: 1px solid #c9e1f4;  
text-align: center;  
border-radius: 1px;  
}  
  
@font-face {  
font-family: 'Open Sans';  
font-weight: normal;  
font-style: normal;  
}
```

(*NOTE: Apart from defining properties of the border, font, text alignment the file also sets the CSS Class Name as panel-box. This means we can adjust the above given settings to those specified in the Lumiradesigner.css in any component where we set the CSS Class Name to panel-box.*)

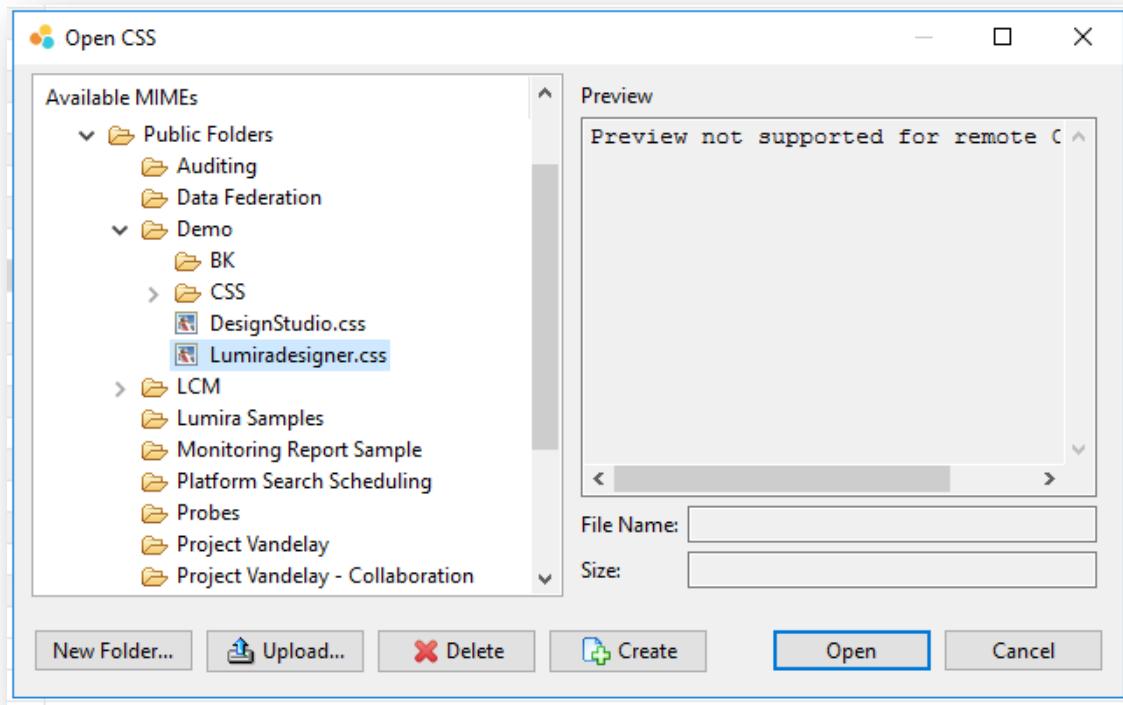
2.1 Under the Properties View>>Display tab, select the **Custom CSS** property click on the **Browse...** button.



2.2 Next, in the **Open CSS** dialog box that opens, navigate to **BI Platform>>Public Folders>>Demo** and select the **Upload...** button to upload the Local CSS file to the BI Platform. The **Choose Local Files** dialog box opens up. Select the *Lumiradesigner.css* file from where it is saved on the local system and click **Open**.



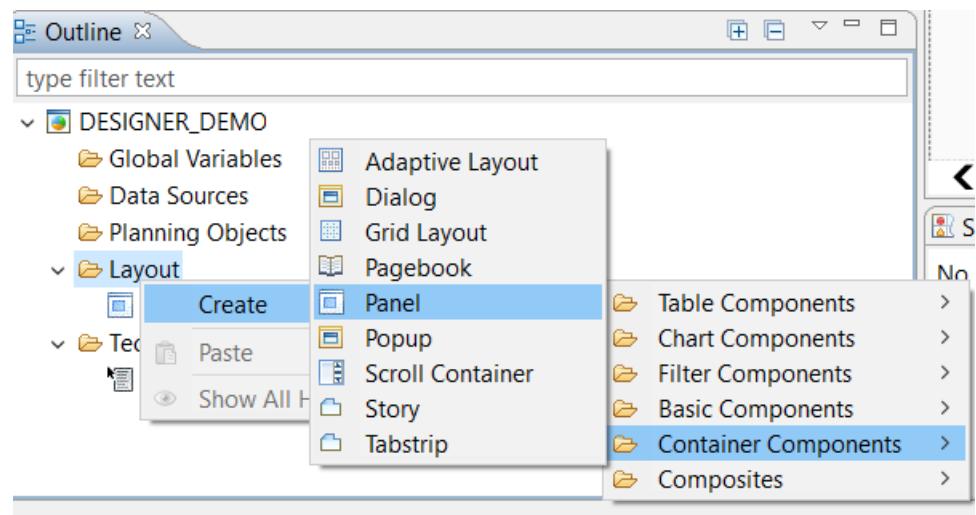
2.3 The file is successfully uploaded to the BI Platform, from under the **Open CSS** dialog box, select the *Lumiradesigner.css* file and click **Open** to add it to our application.



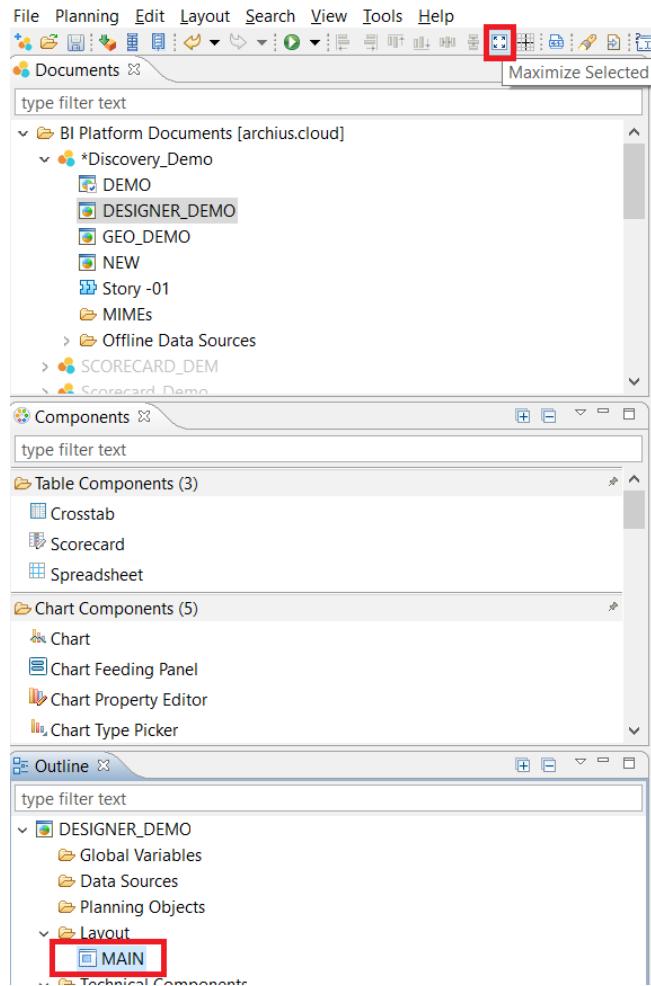
3 Creating a Header Panel

We add a Header to our Analysis Application using the Panel container component. Here is how:

3.1 From the Outline Panel, right click on the Layout folder and from the context menu select **Create>>Container Components>>Panel**.



3.2 Under the Properties View of the Panel component, set its **Name** as *MAIN*. Click on Maximize selected component as shown.



Following settings should appear.

Properties		Additional Properties
Property	Value	
General		
Name	MAIN	
Type	Panel	
Vendor	SAP SE	
Visible	true	
Display		
CSS Class		
CSS Style		
Events		
On Click		
Layout		
Top Margin	0	
Left Margin	0	
Bottom Margin	0	
Right Margin	0	
Width	auto	
Height	auto	

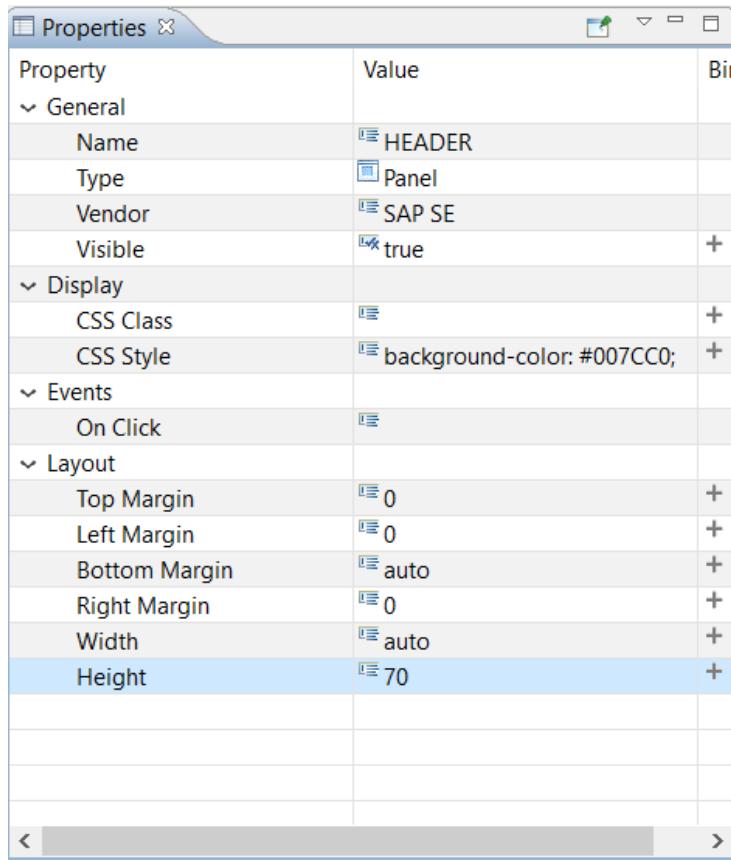
3.3 Similarly, under Display set its CSS Style to: **background-color: #d3d3d3;** to set the Panel to grey.

The screenshot shows two overlapping dialogs. The main dialog is the 'Properties' window for a component named 'MAIN'. It has sections for General, Display, Events, and Layout. Under Display, the 'CSS Style' field is selected and highlighted with a blue background. To the right, a smaller dialog titled 'CSS Style Editor' is open, containing the text 'Edit the CSS styles:' followed by the CSS rule 'background-color: #d3d3d3;'. There are 'OK' and 'Cancel' buttons at the bottom of the editor.

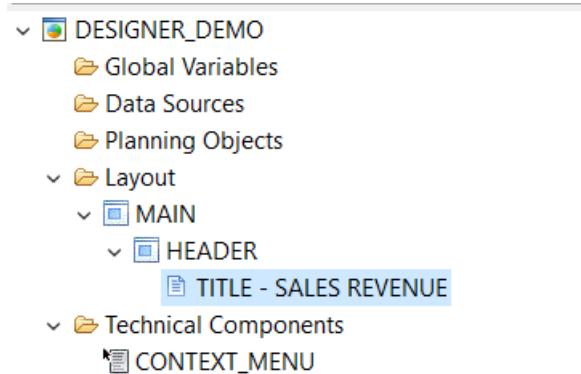
Property	Value
Vendor	SAP SE
Visible	true
Display	
CSS Class	
CSS Style	
Events	
On Click	
Layout	
Top Margin	0
Left Margin	0
Bottom Margin	0

3.4 Next, add another Panel component to the *MAIN* panel in a similar fashion. Set its **Name** field to **HEADER**. Set its **CSS Style** to **background-color: #007CC0;** Click on Maximize selected component  and set Height = 70. Following settings should appear.

 and set Height = 70. Following settings should appear.



3.5 Add a Text component to the HEADER panel by right clicking on it and selecting **Create>>Basic Components>>Text**. The Layout of STORYPAGE_1 should now look similar to:



3.6 In the Properties View of the Text component set its Name property to TITLE and Text property to SALES REVENUE. Set the following CSS properties under CSS Style:

```
font-family : Calibri;
font-size : 28px;
color:#FFFFFF;
```

font-weight: bold;

(**Note:** Here we change the font type, font size and color to give the heading a louder appeal.)

Properties		Additional Properties
Property	Value	
General		
Name	TITLE	
Type	Text	
Vendor	SAP SE	
Visible	true	
Display		
CSS Class		
Text	SALES REVENUE	
CSS Style	font-family : Calibri; f...	
Tooltip		

3.7 Click on Maximize selected component  and set Top Margin = 10 and Left Margin = 15. Following settings will appear.

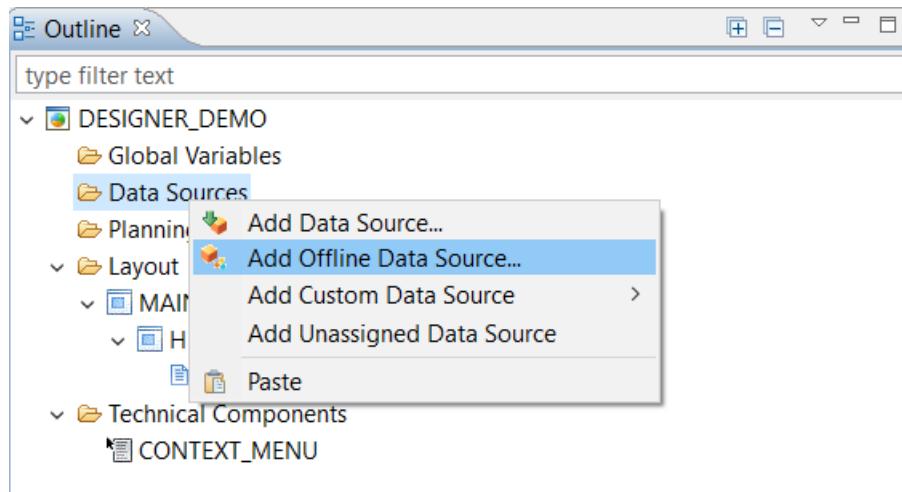
Layout	
Top Margin	10
Left Margin	15
Bottom Margin	0
Right Margin	0
Width	auto
Height	auto

3.8 The Header Panel should now look similar to:



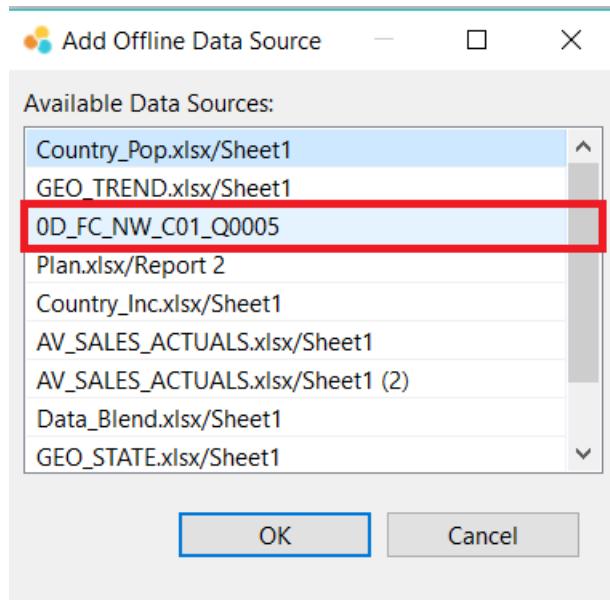
4. Adding Offline Data Sources

4.1 Right Click on the Data Sources folder. Click on Add Offline Data Source.



4.2

Click on the data source as shown

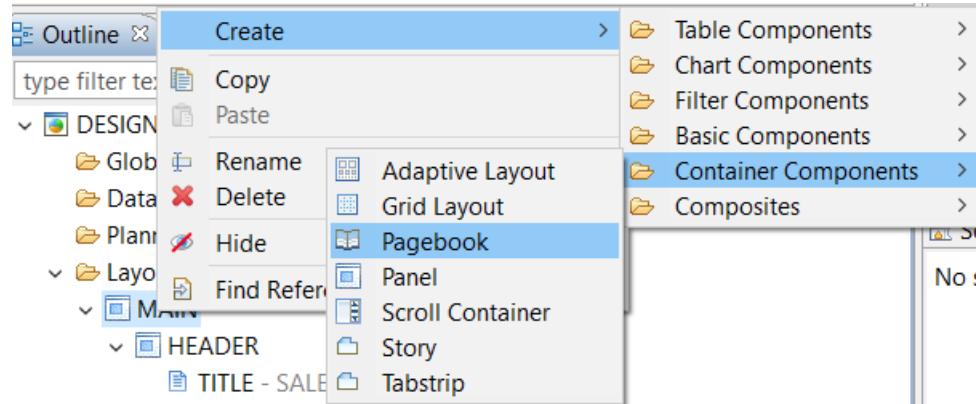


This Data Source will be available in the data source section.

5. Adding and aligning Page Book, Grid and Panel container components to the application and creating charts and crosstabs

We can make use of the custom .css defined in **Section 3** by adding Panels to the application and assigning our charts to different Panel components. We can also divide the Panels into Grids to further demarcate them. Here is how:

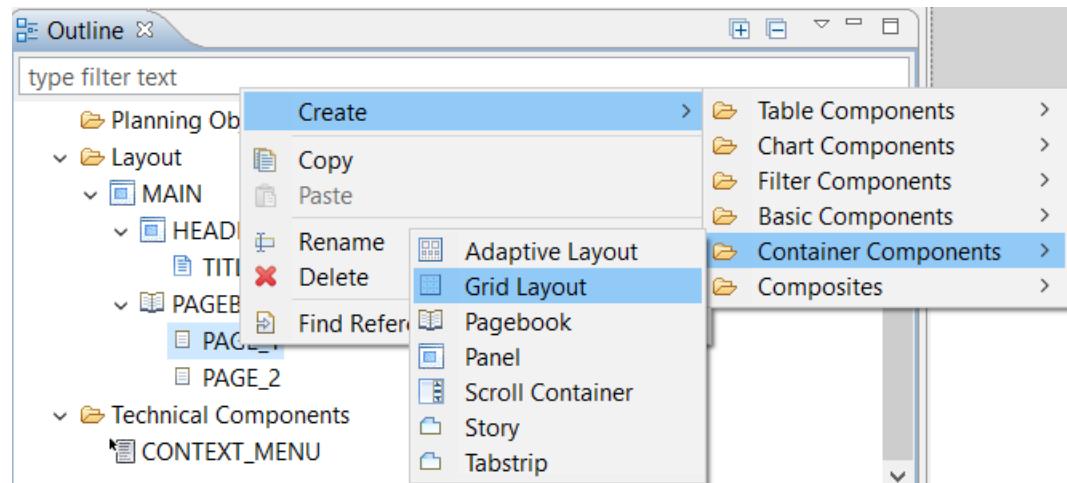
5.1 Right Click on the MAIN Panel and create a new Pagebook component under Container Components



5.2 Click on Maximize selected component and set the Top Margin = 70. Following settings should appear

Layout	
Top Margin	70
Left Margin	0
Bottom Margin	0
Right Margin	0
Width	auto
Height	auto

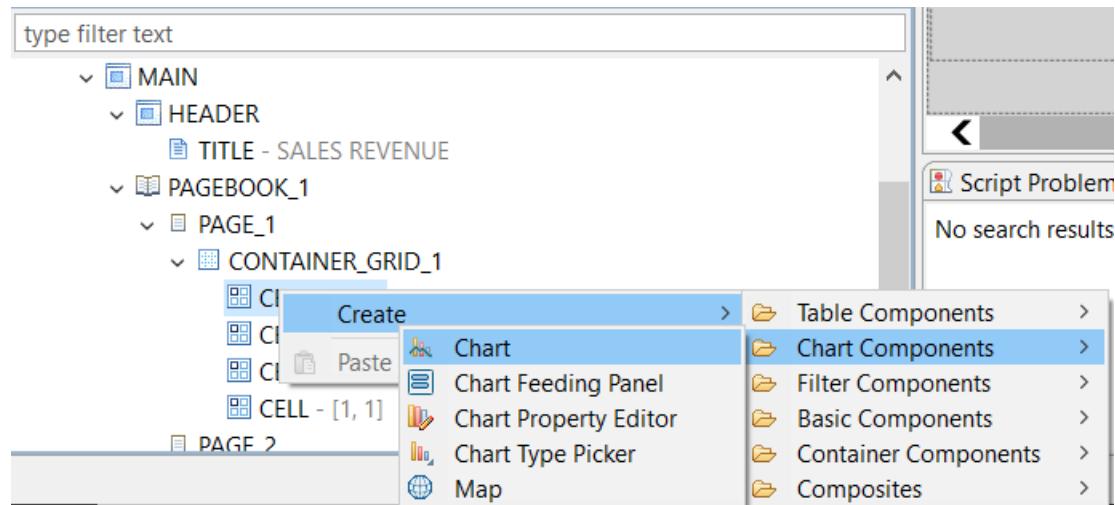
5.3 Under Page 1 component, add a new panel component by right click>> Container Component>> Grid Layout.



5.4 Rename the Grid Layout as CONTAINER_GRID_1. Click on Maximize Selected component  . Also set the number of rows = 2 and No of Columns = 2. Following settings should appear

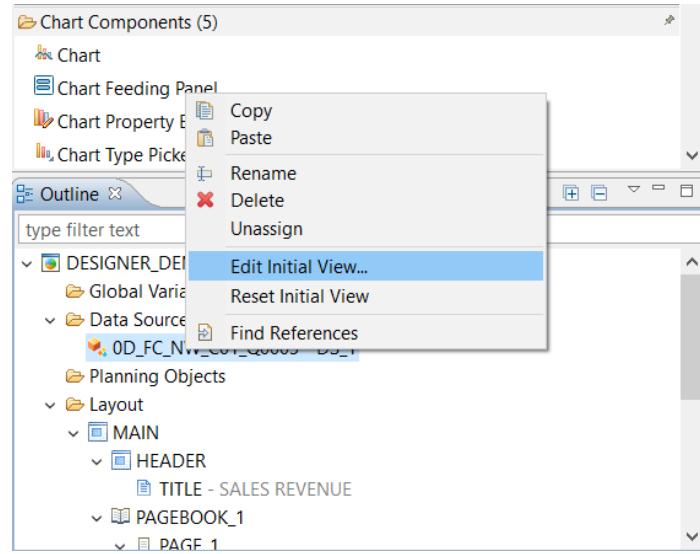
Properties	
Property	Value
General	
Name	CONTAINER_GRID_1
Type	Grid Layout
Vendor	SAP SE
Visible	true
Display	
CSS Class	
Layout	
Top Margin	0
Left Margin	0
Bottom Margin	0
Right Margin	0
Width	auto
Height	auto
Number of Rows	2
Number of Columns	2

5.5 Under Cell – [0,0] of Content_Grid_1 add a new chart component by Right click>> Create Components>> Chart >> Chart

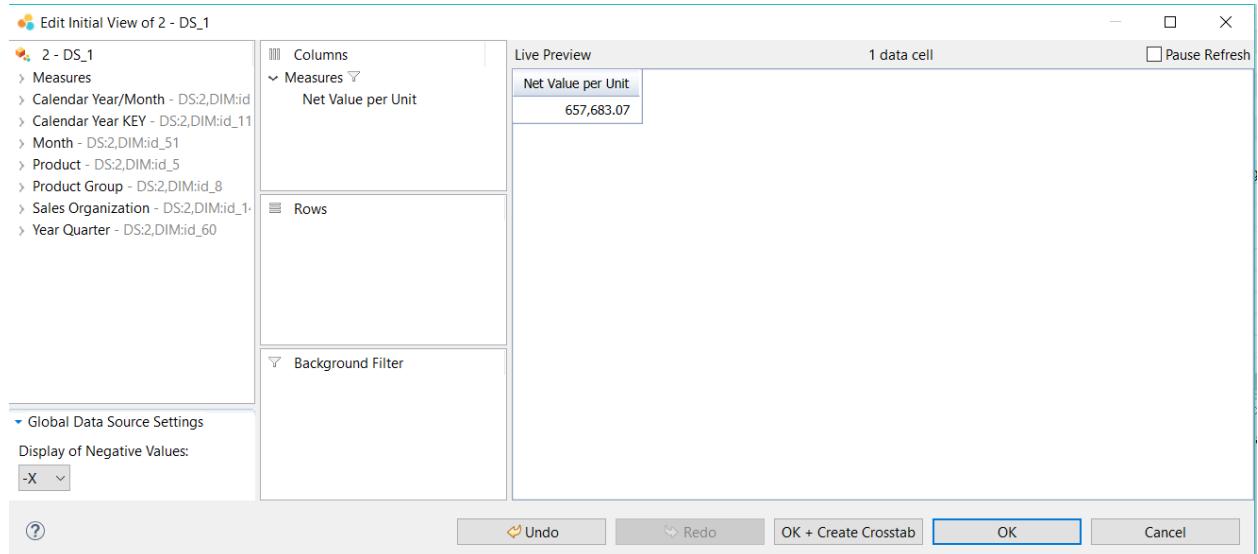


For chart click on Maximize selected component  . Set left margin = 20 and Right Margin = 20.

5.6 Right click on the Data Source DS_1 and Click on Edit Initial View



Set the Measure as **Net Value per unit**. Click on OK.



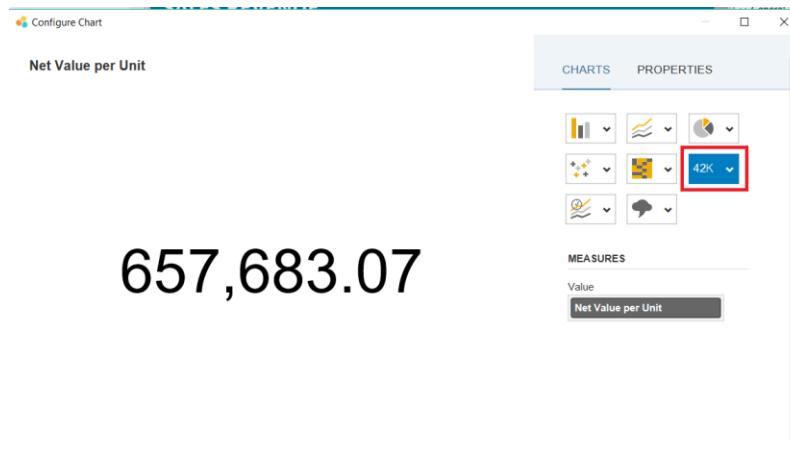
5.7 Under Chart 1, go to properties, Under Data Binding >> select the Data Source as DS_1.

Properties	
Property	Value
General	
Name	CHART_1
Type	Chart
Vendor	SAP SE
Visible	true
Data Binding	
Data Source	<none>
Data Selection	Add... DS_1 <none>
Display	
CSS Class	
Chart Configuration	Column Chart
Allow Data Source Modific	false
Conditional Formatting Vis	false
Conditional Formatting Set	
Show Totals	false
Show Scaling Factors	false
Dimension Label	Initial View Definitions
Events	
On Select	
Layout	
Top Margin	0

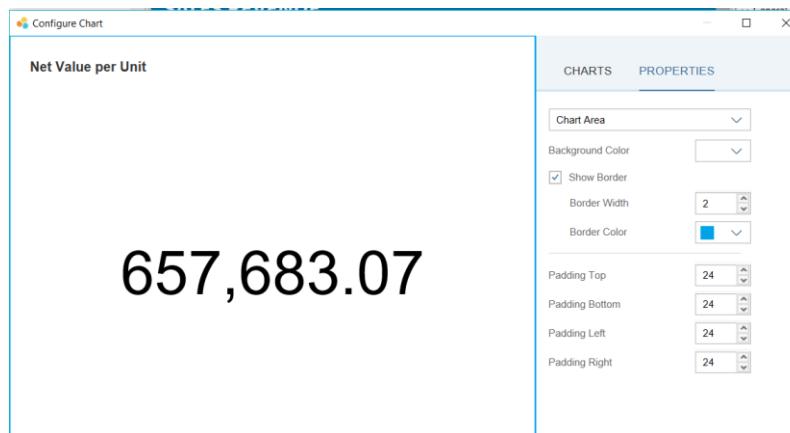
5.8 Next, Click on Chart Configuration

Properties	
Property	Value
General	
Name	CHART_1
Type	Chart
Vendor	SAP SE
Visible	true
Data Binding	
Data Source	<none>
Data Selection	
Display	
CSS Class	
Chart Configuration	Column Chart
Allow Data Source Modific	false
Conditional Formatting Vis	false
Conditional Formatting Set	
Show Totals	false
Show Scaling Factors	false
Dimension Label	Initial View Definitions
Events	
On Select	
Layout	
Top Margin	0

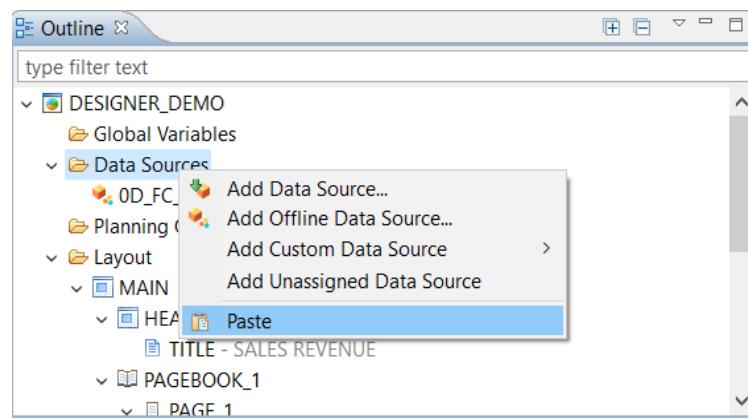
5.9 Select the chart as Numeric Component.



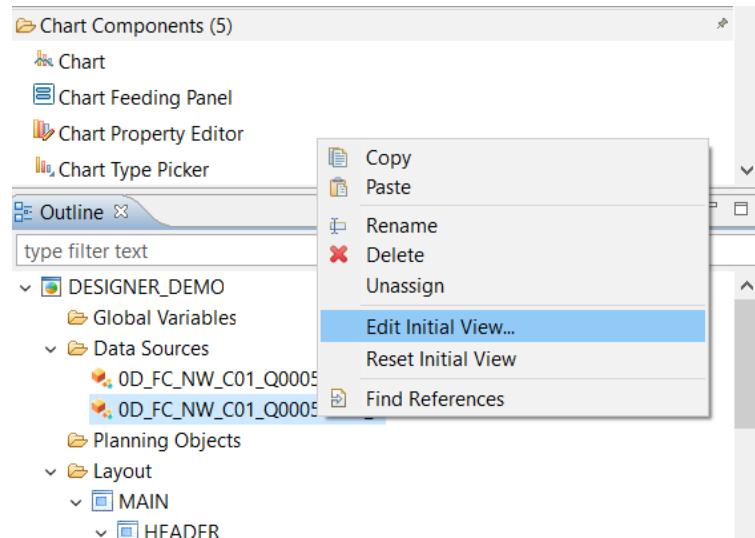
5.10 Under Properties tab, select Chart area. Click on Show border and configure as given. Also select Background color as White. Click on OK.



5.11 Copy the Data Source DS_1 and paste it under data source again



5.12 A new Data Source DS_2 will be created. Click on Edit Initial View.



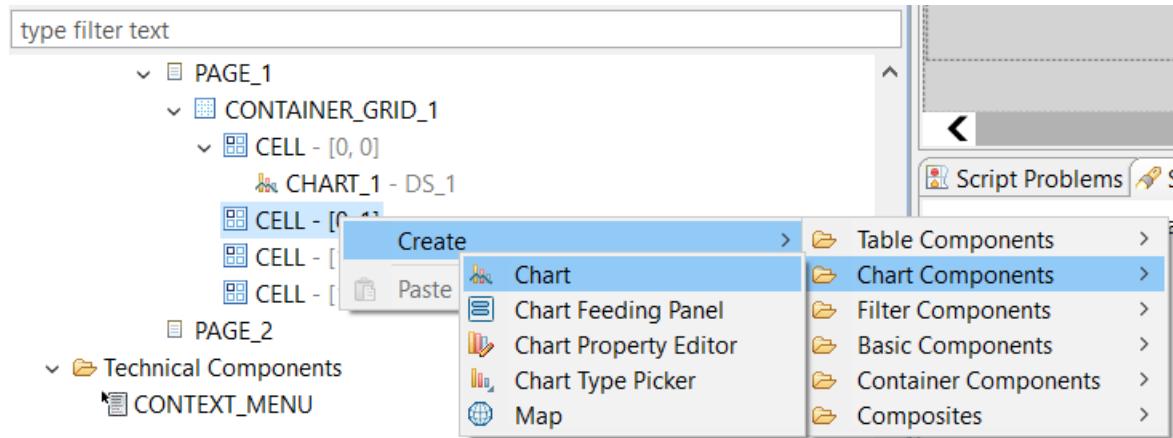
5.13 Set the view as follows

The screenshot shows the 'Edit Initial View of 2 - DS_2' dialog box. On the left, there is a sidebar with a tree view of measures and a 'Global Data Source Settings' section. The main area is divided into 'Columns' and 'Rows'. Under 'Columns', there is a 'Measures' section with 'Net Value Stat. Curr. [EUR]' and 'Plan' listed. Under 'Rows', there is a 'Year Quarter' section with four rows: 'Quarter 1', 'Quarter 2', 'Quarter 3', and 'Quarter 4', followed by an 'Overall Result' row. The 'Live Preview' section shows a table with the following data:

Year Quarter	Net Value Stat. Curr. [EUR]	Plan
Quarter 1	1,167,679,457.00	592,258,650,407.82
Quarter 2	1,156,317,096.00	599,788,452,938.40
Quarter 3	1,167,016,547.00	599,788,452,938.40
Quarter 4	1,183,798,195.00	599,788,452,938.40
Overall Result	4,674,811,295.00	2,391,624,009,223.04

At the bottom, there are buttons for '?', Undo, Redo, OK + Create Crosstab, OK, and Cancel.

5.14 Create a new chart component under Cell – [0,1] of Content_GRID_1.



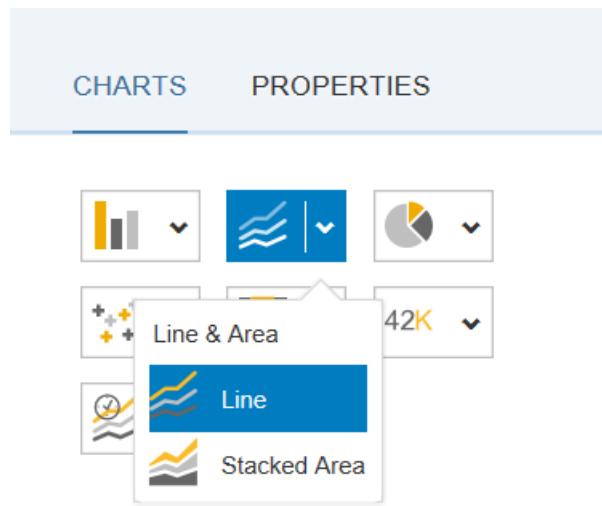
5.15 For Chart_2, Click on Maximize Selected component . Also set left Margin = 30 and Right Margin = 20. Following properties should appear

Properties	
Property	Value
Data Source	<none>
Data Selection	
Display	
CSS Class	
Chart Configuration	Column Chart
Allow Data Source Modific	false
Conditional Formatting Vis	false
Conditional Formatting Set	
Show Totals	false
Show Scaling Factors	false
Dimension Label	Initial View Definitions
Events	
On Select	
Layout	
Top Margin	0
Left Margin	30
Bottom Margin	0
Right Margin	20
Width	auto
Height	auto

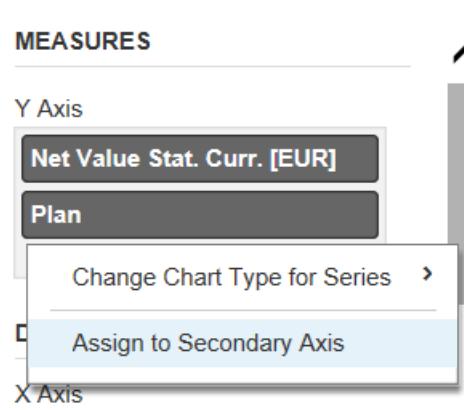
5.16. Set the data source as DS_2 and click on Chart Configuration.

Properties	
Property	Value
General	
Name	CHART_2
Type	Chart
Vendor	SAP SE
Visible	true
Data Binding	
Data Source	DS_2
Data Selection	
Display	
CSS Class	
Chart Configuration	Column Chart
Allow Data Source Modific.	false
Conditional Formatting Vis	false
Conditional Formatting Set	
Show Totals	false
Show Scaling Factors	false
Dimension Label	Initial View Definitions
Events	
On Select	
Layout	
Top Margin	0

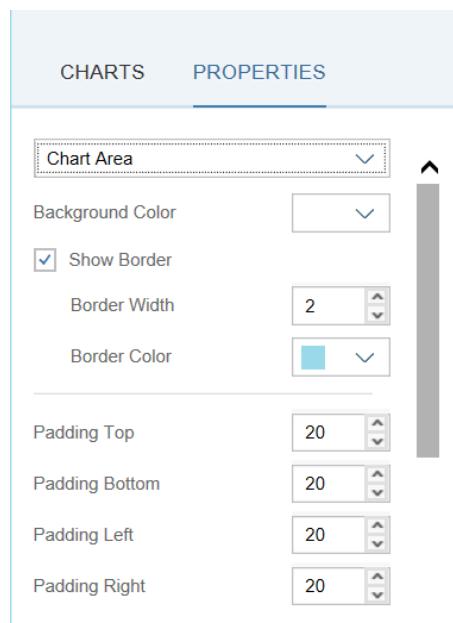
5.17 Click on Line under Line and Area.



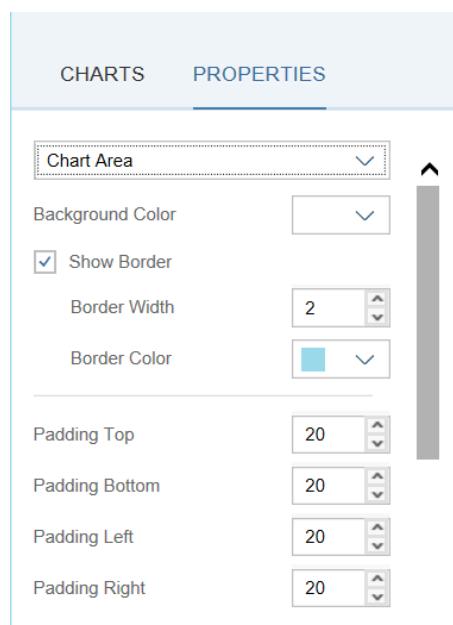
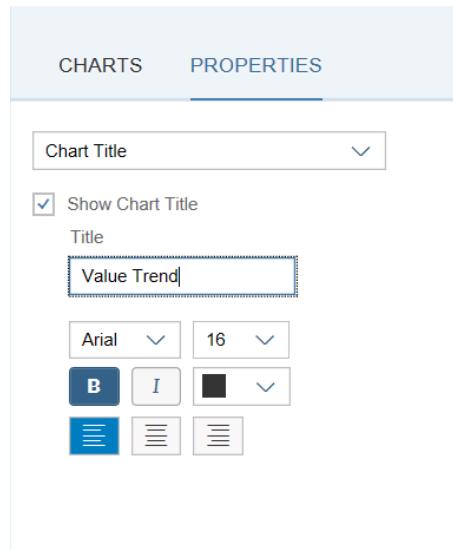
5.18 Click on Plan under Measures in Y-axis and Click on Assign to Secondary axis



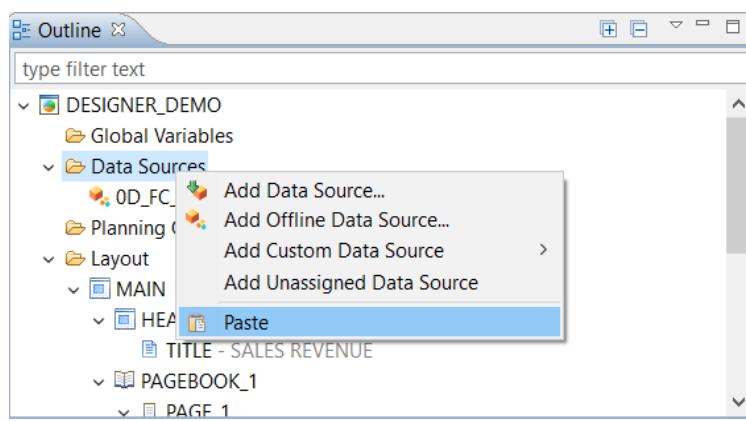
5.19 Under Properties tab, Select Chart Area and set Background color as white and check on show border. Set the color and width as shown



5.20 Under Chart Title set the title as Value Trend



5.21 Copy the Data Source DS_1 and paste it under data source again



5.22 A new Data Source DS_3 will be created. Click on Edit Initial View and set the view as follows

Sales Organization	Net Value per Unit	Open order quantity [PC]
Berlin	659,338.09	3,168.00
London	671,069.14	1,584.00
New York	649,595.79	3,168.00
Paris	653,729.63	3,168.00
San Francisco	661,397.93	3,168.00
Overall Result	657,683.07	14,256.00

5.23 Create a new chart component under Cell – [1,0] of Content_GRID_1. For Chart_3, Click on

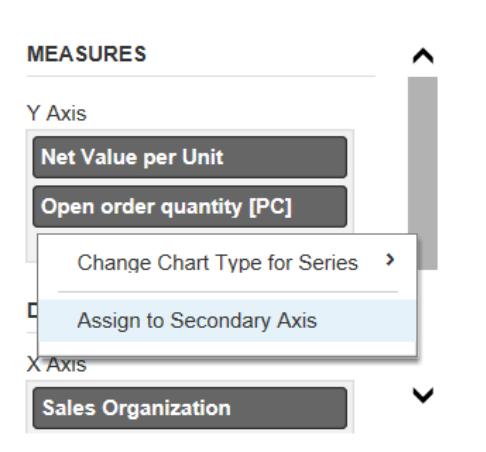
Maximize Selected component . Also set left Margin = 30 and Right Margin = 20. Following properties should appear

Properties	
Property	Value
Data Source	<none>
Data Selection	
Display	
CSS Class	
Chart Configuration	Column Chart
Allow Data Source Modific	false
Conditional Formatting Vis	false
Conditional Formatting Set	
Show Totals	false
Show Scaling Factors	false
Dimension Label	Initial View Definitions
Events	
On Select	
Layout	
Top Margin	0
Left Margin	30
Bottom Margin	0
Right Margin	20
Width	auto
Height	auto

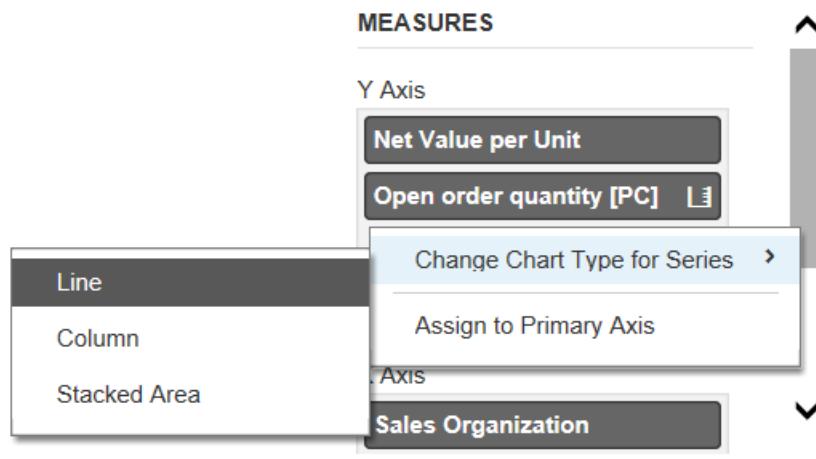
5.24 Set the data source as DS_3 and click on Chart Configuration.

Properties	
Property	Value
General	
Name	CHART_3
Type	Chart
Vendor	SAP SE
Visible	true
Data Binding	
Data Source	DS_3
Data Selection	
Display	
CSS Class	
Chart Configuration	Column Chart
Allow Data Source Modific	false
Conditional Formatting Vis	false
Conditional Formatting Set	
Show Totals	false
Show Scaling Factors	false
Dimension Label	Initial View Definitions
Events	
On Select	
Layout	
Top Margin	0

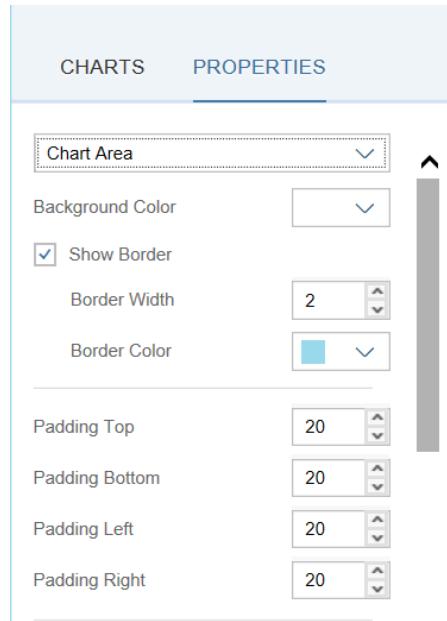
5.25 Click on Open order quantity under Measures in Y-axis and Click on Assign to Secondary axis



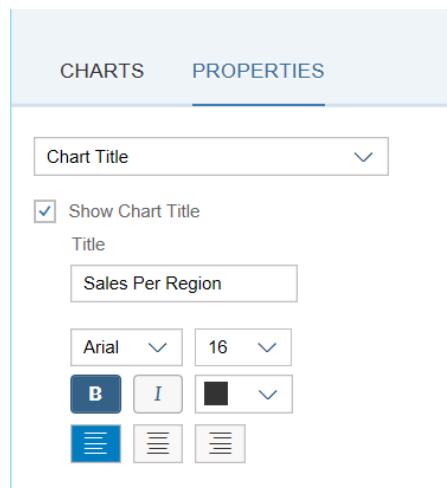
5.26 Click on Open order quantity under Measures in Y-axis and Click on Change chart type for Series. Choose Line



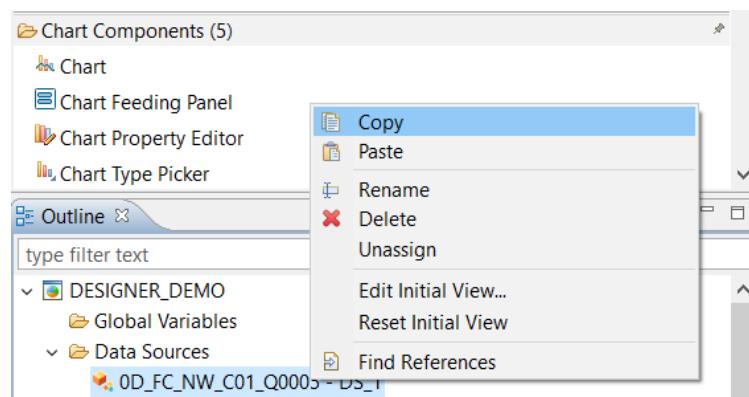
5.27 Under Properties tab, Select Chart Area and set Background color as white and check on show border. Set the color and width as shown

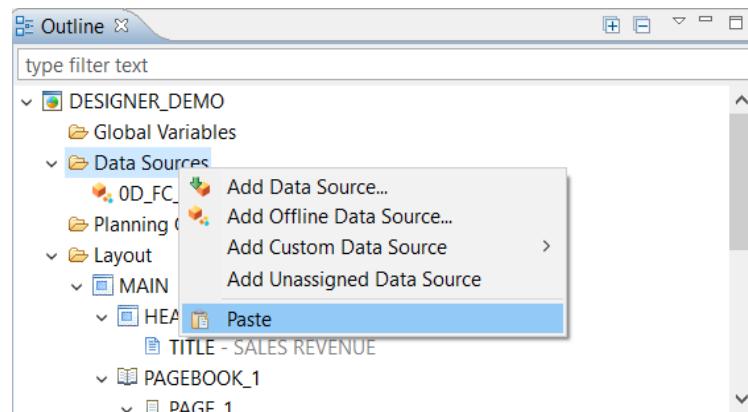


5.28 Under Chart Title set the title as Value Trend



5.29 Copy the Data Source DS_1 and paste it under data source again





5.30 A new Data Source DS_4 will be created. Click on Edit Initial View and set the view as follows

Product Group	Net Value stat curr
Mobile Devices 1	379,485,801,816
Mobile Devices 2	372,621,885,372
Monitors 1	378,299,201,808
Notebooks 1	11,318,041,812
Notebooks 2	11,994,144,360
Ultrabooks 1	376,128,947,568
Ultrabooks 2	333,868,602,771
Wide LED Backlight 1	388,074,368,888
Overall Result	2,251,790,993,595

5.31 Create a new chart component under Cell – [1,1] of Content_GRID_1. For Chart_4, Click on

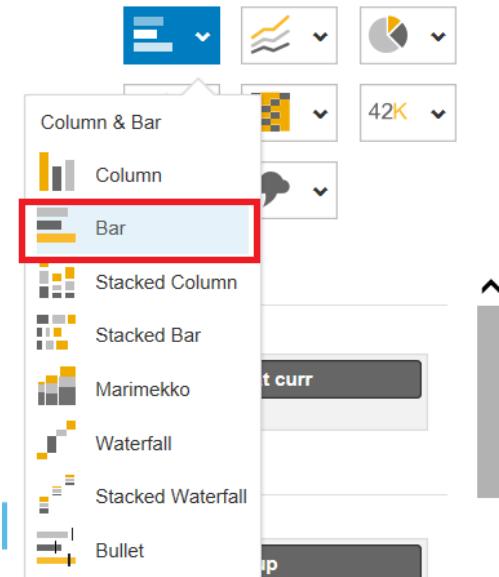
Maximize Selected component . Also set Right Margin = 20. Following properties should appear

Properties	
Property	Value
Data Source	<none>
Data Selection	...
Display	
CSS Class	...
Chart Configuration	Column Chart
Allow Data Source Modific	false
Conditional Formatting Vis	false
Conditional Formatting Set	...
Show Totals	false
Show Scaling Factors	false
Dimension Label	Initial View Definitions
Events	
On Select	...
Layout	
Top Margin	0
Left Margin	30
Bottom Margin	0
Right Margin	20
Width	auto
Height	auto

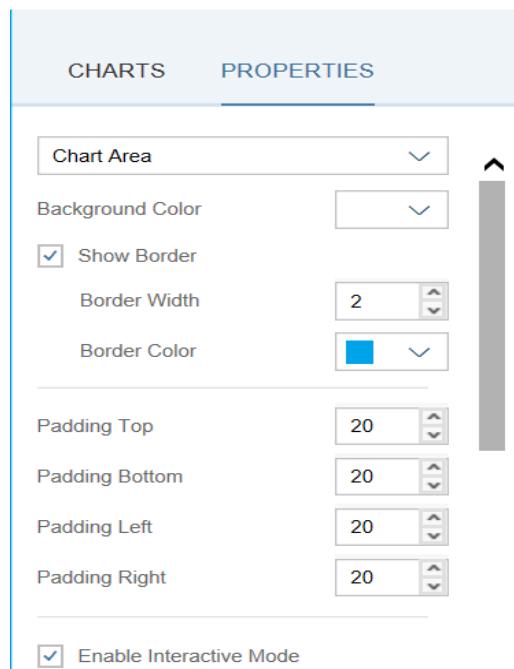
5.32 Set the data source as DS_4 and click on Chart Configuration.

Properties	
Property	Value
General	
Name	CHART_4
Type	Chart
Vendor	SAP SE
Visible	true
Data Binding	
Data Source	DS_4
Data Selection	...
Display	
CSS Class	...
Chart Configuration	Column Chart
Allow Data Source Modific	false
Conditional Formatting Vis	false
Conditional Formatting Set	...
Show Totals	false
Show Scaling Factors	false
Dimension Label	Initial View Definitions
Events	
On Select	...
Layout	
Top Margin	0

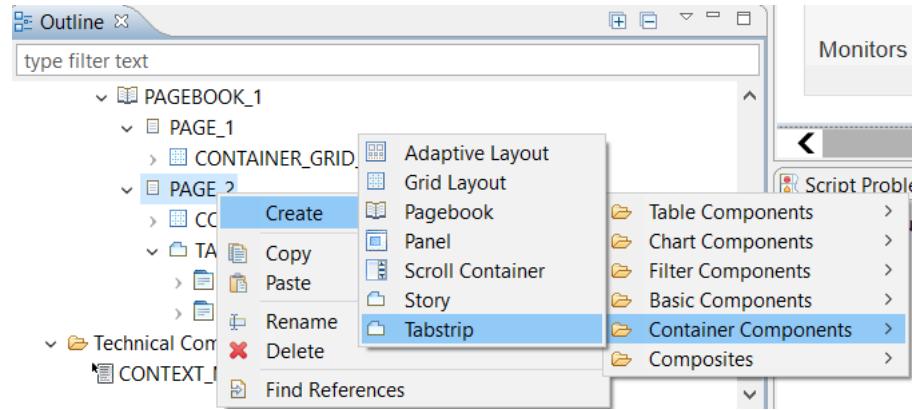
5.33 Under Charts, click on Column and Bar and select Bar



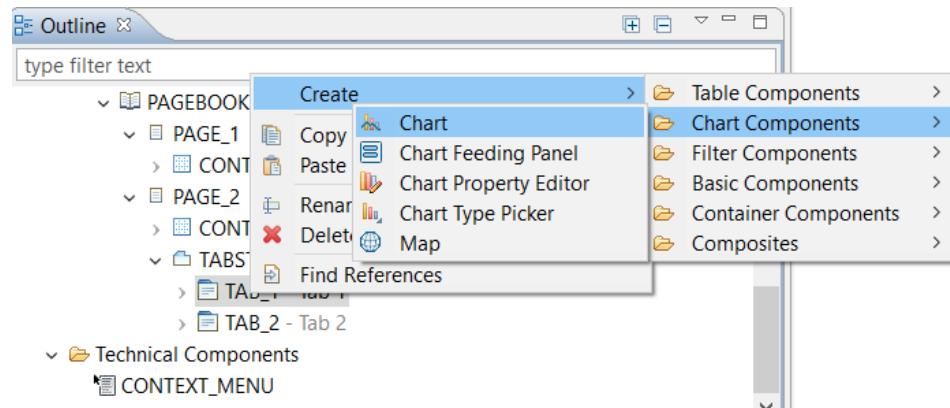
5.34 Under Properties tab, Select Chart Area and set Background color as white and check on show border. Set the color and width as shown



5.35 Under Page 2 component, add a new Tabstrip Component by right click>> Container Component>> Tabstrip.

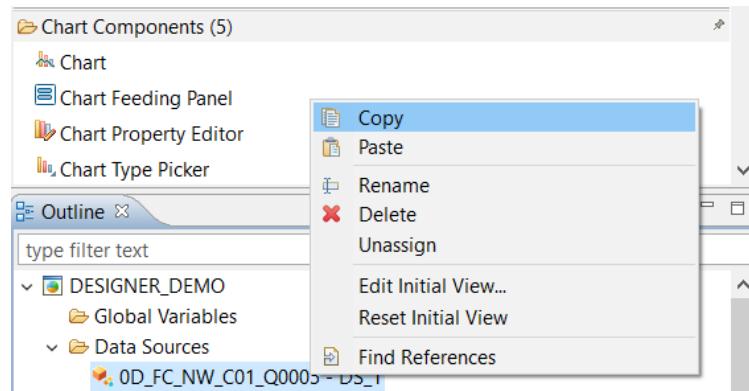


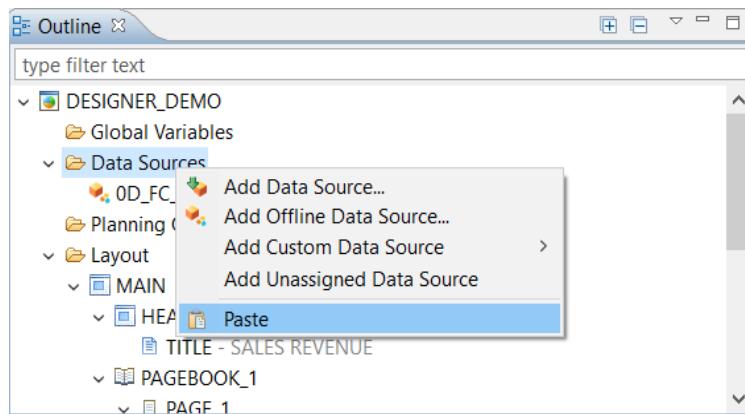
5.36 Click on Maximize Selected component . Under Tab_1 inside Tabstrip_1, add a new chart component by Right click>> Create >> Chart Components >> Chart



5.37 For chart click on Maximize selected component .

5.38 Copy the Data Source DS_1 and paste it under data source again





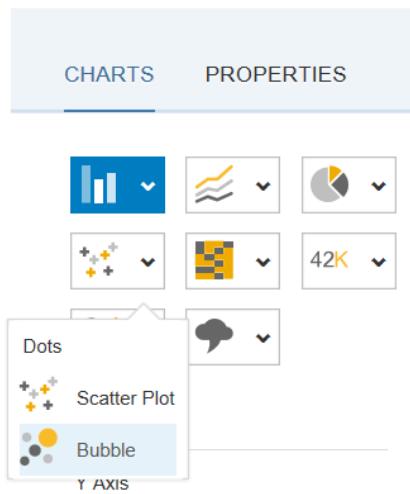
5.39A new Data Source DS_5 will be created. Click on Edit Initial View. Set the Initial view as shown Below.

Product	Calendar Year KEY	Net Value stat curr	Net Value per Unit	Open order quantity [PC]
	2010	33,288,914,832	640,652.40	192.00
	2011	33,288,914,832	611,127.75	192.00
	2012	33,288,914,832	660,297.99	192.00
	2013	33,288,914,832	652,177.32	192.00
	2014	33,288,914,832	678,981.04	192.00
	2015	33,288,914,832	617,474.27	192.00
	2016	33,288,914,832	760,870.29	192.00
	2017	33,288,914,832	639,751.66	192.00
	2018	33,288,914,832	617,743.71	192.00
	2019	33,288,914,832	656,608.97	192.00
	2020	33,288,914,832	703,727.69	192.00
Result		366,178,063,152	657,987.72	2,112.00
	2010	930,313,032	604,739.67	24.00
	2011	930,313,032	515,569.17	24.00
	2012	930,313,032	729,818.42	24.00
	2013	930,313,032	634,235.58	24.00
	2014	930,313,032	634,926.83	24.00

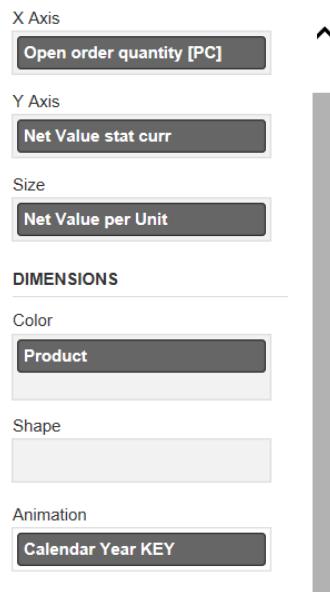
5.40 Under Chart 1, go to properties, Under Data Binding >> select the Data Source as DS_5. Next, Click on Chart Configuration

Property	Value
General	
Name	CHART_5
Type	Chart
Vendor	SAP SE
Visible	true
Data Binding	
Data Source	DS_5
Data Selection	
Display	
Chart configuration	
Chart Configuration	Column Chart
Allow Data Source Modific	false
Conditional Formatting Vis	false
Conditional Formatting Set	
Show Totals	false
Show Scaling Factors	false
Dimension Label	Initial View Definitions
Events	
On Select	

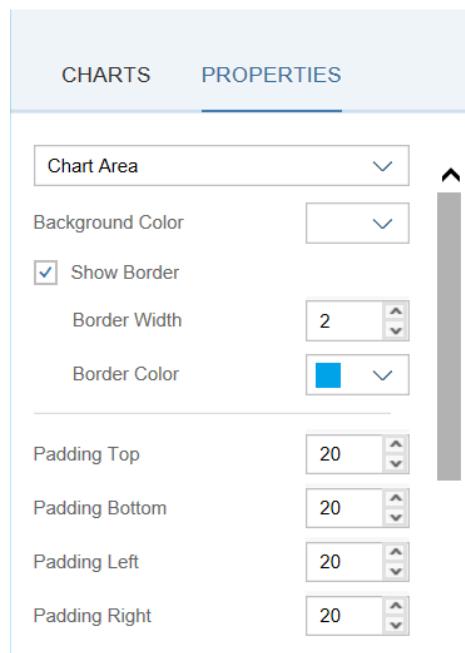
5.41 Select the chart under Dots as Bubble.



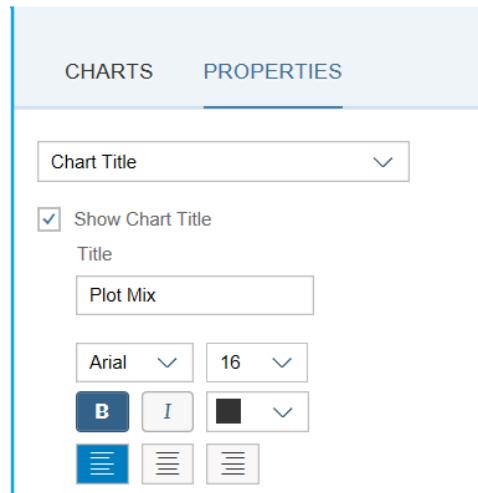
5.42 Set the properties as follows:



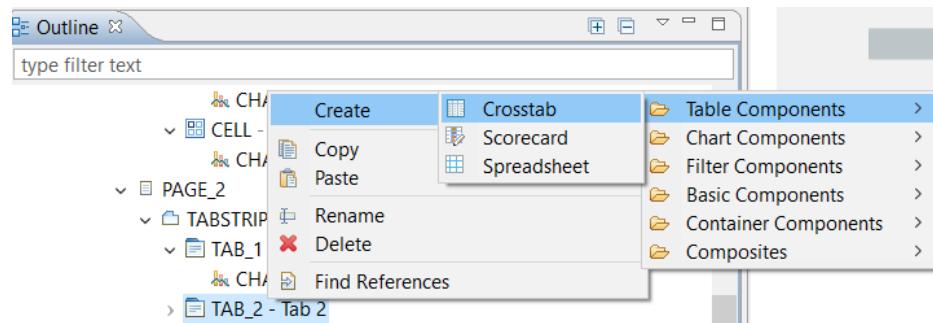
5.43 Under Properties tab, select Chart area. Click on Show border and configure as given. Also select Background color as White. Click on OK.



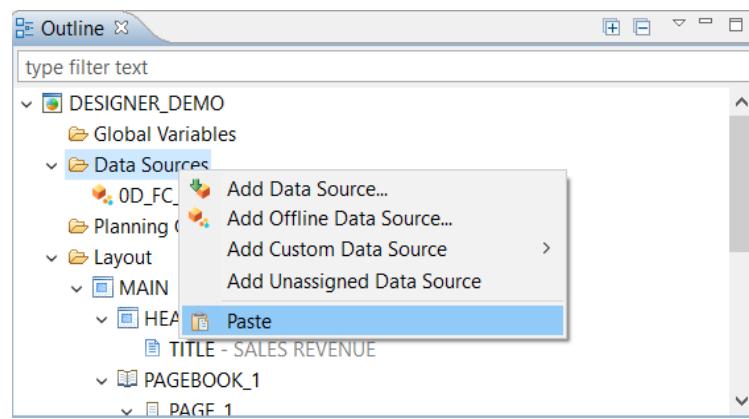
5.44 Select Chart Title from Dropdown and set Title as Plot Mix



5.45 Create a new Cross Tab component under Tab_2 by Right Click>> Create>> Table components>> Crosstab



5.46 Click on Maximize selected component . Copy the Data Source DS_1 and paste it under data source again



5.47 A new Data Source DS_6 will be created. Click on Edit Initial View and set the view as follows

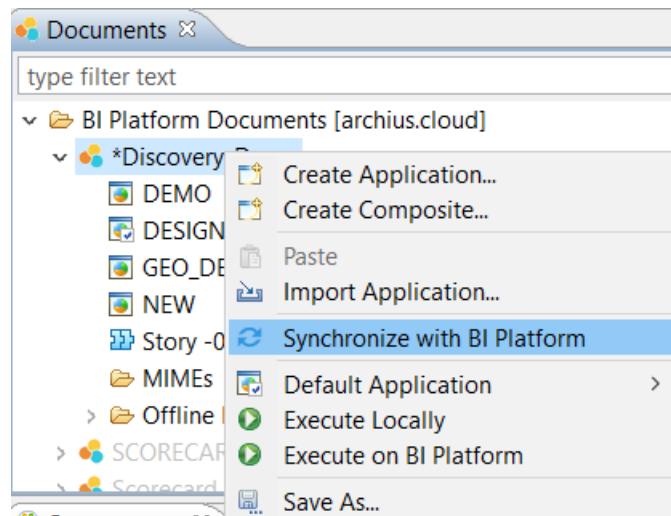
Product Group	Sales Organization	Net Value stat curr	Net Value per Unit	Qu. in base units [PC]	Open order stat curr [EUR]
Mobile Devices 1	Berlin	92,313,089,616	676,455.30	262.00	
	London	10,233,443,352	587,318.83	132.00	
	New York	92,313,089,616	679,241.05	264.00	
	Paris	92,313,089,616	644,763.19	264.00	
	San Francisco	92,313,089,616	653,686.30	264.00	
Result	379,485,801,816	655,031.77	1,186.00		
Mobile Devices 2	Berlin	90,171,178,020	631,066.39	264.00	
	London	11,937,173,292	685,099.48	132.00	
	New York	90,171,178,020	660,967.33	262.00	
	Paris	90,171,178,020	650,007.19	263.00	
	San Francisco	90,171,178,020	652,985.33	264.00	
Result	372,621,885,372	652,783.22	1,185.00		
Monitors 1	Berlin	91,544,515,788	681,029.21	263.00	
	London	12,121,138,656	706,360.06	130.00	
	New York	91,544,515,788	637,163.48	264.00	
	Paris	91,544,515,788	667,789.76	263.00	

5.48 Assign Data source DS_6 to crosstab_1

Property	Value
Name	CROSSTAB_2
Type	Crosstab
Vendor	SAP SE
Visible	true
Data Binding	DS_6

6 Running the Application

Right Click on the Application and Synchronize it with BI Platform.



Right Click on Application again and click on Execute on BI Platform. The output should be similar to the following.



Click on Right Arrow to navigate to Page 2. Page should be something similar to

SALES REVENUE

Tab 1 Tab 2

Open order quantity [PC], Net Value stat curr and Net Value per Unit by Calendar Year KEY and Product



Click in Tab 2. Following screen will be shown

SALES REVENUE

Tab 1 Tab 2

Product Group	Sales Organization	Net Value stat curr EUR^{a}	Net Value per Unit EUR^{a}	Qu. in base units [PC] EUR^{a}	Open order stat curr [EUR] EUR^{a}
Mobile Device...	Berlin	92,313,089,616	676,455.30	262.00	23,040,072.00
	London	10,233,443,352	587,318.83	132.00	10,078,403.00
	New York	92,313,089,616	679,241.05	264.00	23,311,557.00
	Paris	92,313,089,616	644,763.19	264.00	22,128,289.00
	San Francisco	92,313,089,616	653,686.30	264.00	22,434,523.00
	Result	379,485,801,816	655,031.77	1,186.00	100,992,844.00
Mobile Devices 2	Berlin	90,171,178,020	631,066.39	264.00	21,658,218.00
	London	11,937,173,292	685,099.48	132.00	11,756,314.00
	New York	90,171,178,020	660,967.33	262.00	22,512,561.00
	Paris	90,171,178,020	650,007.19	263.00	22,223,766.00
	San Francisco	90,171,178,020	652,985.33	264.00	22,410,475.00
	Result	372,621,885,372	652,783.22	1,185.00	100,561,334.00
Monitors 1	Berlin	91,544,515,788	681,029.21	263.00	23,284,403.00
	London	12,121,138,656	706,360.06	130.00	11,937,490.00
	New York	91,544,515,788	637,163.48	264.00	21,867,477.00
	Paris	91,544,515,788	667,789.76	263.00	22,831,740.00
	San Francisco	91,544,515,788	646,092.84	264.00	22,173,924.00
	Result	378,299,201,808	663,298.87	1,184.00	102,095,034.00

Exercise 4 – Filter Components

Objective of the Exercise

- Creating a Filter Panel
- Adding Dropdowns and List Box
- Input/Output binding
- Icon to Show/Hide Filter Panel

1 [Adding and configuring Panel and Text Box components to create a Filter Panel](#)

In this exercise we will create a Filter Panel to filter values in the Discovery_Demo application based on dimensions that are not displayed in all of the analytic components. The following steps will demonstrate how we are able to create new views on our Analysis Components based on the values we choose from the Filter Panel.

1.1 Select the MAIN panel under Layout and right click on it. From the context menu add a Panel component to it by selecting **Create>>Container Components>>Panel**.

1.2 Set the Name property of the newly added Panel as FILTER and set the CSS class to panel-box.
From under the Properties View of FILTER, set the CSS Style:

background-color: #007cc0; Click on Maximize selected component  and set Top Margin =70 and Width = 190.

▼ Display	
CSS Class	panel-box
CSS Style	background-color: #...
▼ Events	
On Click	
▼ Layout	
Top Margin	70
Left Margin	0
Bottom Margin	0
Right Margin	auto
Width	190
Height	auto

1.3 Under General properties set the Visible property to false.

Property	Value
General	
Name	FILTER
Type	Panel
Vendor	SAP SE
Visible	false
Display	false
CSS Class	true

(Note: This hides the filter panel at start-up of the application)

1.4 Add a Text box to the FILTER Panel and set the Name and Text property in the Properties View to FILTERS, set the margins and CSS Style as shown below:

```
font-family : Calibri;
font-size : 24px;
color:#FFFFFF;
font-weight: bold;
```

Display	
CSS Class	
Text	FILTERS
CSS Style	font-family : Calibri;f...
Tooltip	
Events	
On Click	
Layout	
Top Margin	50
Left Margin	45
Bottom Margin	auto
Right Margin	auto
Width	80
Height	30

1.5 Similarly, add another Text box to the FILTER panel and set the Name and Text property as QUARTER. Set the CSS Class as panel-box. Click on the Edit button of the CSS Style property, CSS Style editor dialog box opens, in this dialog box add the below script to set the Font size, Font style, and Font color. Also set its margins as shown:

```
font-family : Calibri;  
font-size : 16px;  
color:#FFFFFF;
```

Display	
CSS Class	panel-box
Text	QUARTER
CSS Style	font-family : Calibri;f...
Tooltip	A
Events	
On Click	
Layout	
Top Margin	125
Left Margin	45
Bottom Margin	auto
Right Margin	auto
Width	80
Height	20

1.6 Copy and paste the QUARTER text box twice more onto the FILTER panel. Set the Name and Text properties of the two new text boxes as PRODUCT GROUP and REGION respectively. Set the margins of the two text boxes as shown below:

PRODUCT GROUP:

Display	
CSS Class	panel-box
Text	PRODUCT GROUP
CSS Style	font-family : Calibri;f...
Tooltip	A
Events	
On Click	
Layout	
Top Margin	225
Left Margin	30
Bottom Margin	auto
Right Margin	auto
Width	120
Height	20

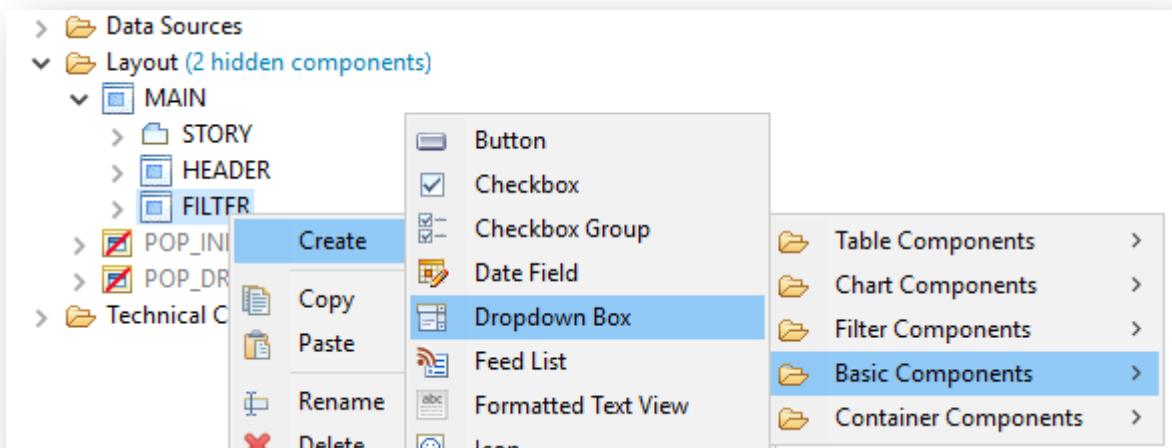
REGION:

Display	
CSS Class	panel-box
Text	REGION
CSS Style	font-family : Calibri; f...
Tooltip	A
Events	
On Click	
Layout	
Top Margin	325
Left Margin	45
Bottom Margin	auto
Right Margin	auto
Width	80
Height	20

1.7 With this we have configured the backbone of our filter panel. In the next section we will add the dropdown boxes and list boxes to configure the filters.

2 [Adding and aligning the Dropdowns and List Boxes](#)

2.1 Right click on the FILTER panel under the MAIN panel and select **Create>>Basic Components>>Dropdown Box**.



2.2 Set its name as YR_QUARTER. Set its CSS Class as *panel-box* and set its margins as shown:

Layout	
Top Margin	150
Left Margin	5
Bottom Margin	auto
Right Margin	auto
Width	150
Height	auto

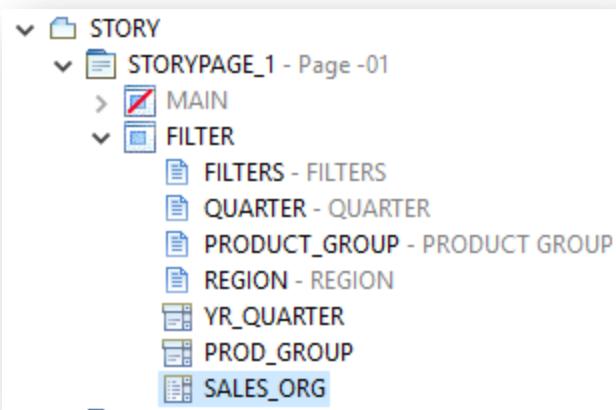
2.3 Similarly, add another Dropdown Box to the FILTER panel and rename it to PROD_GROUP. Set its CSS Class as *panel-box* and margins as shown below:

Layout	
Top Margin	250
Left Margin	5
Bottom Margin	auto
Right Margin	auto
Width	150
Height	auto

2.4 Finally, add a List Box to the FILTER panel from under **Create>>Basic Components>>List Box**. Rename it to SALES_ORG. Set its CSS Class as *panel-box* and margins as shown below. (**Note:** A list box comes in handy when users want to choose more than one value to filter)

Layout	
Top Margin	350
Left Margin	5
Bottom Margin	auto
Right Margin	auto
Width	170
Height	150

2.5 The Layout of the FILTER panel should appear as shown below:



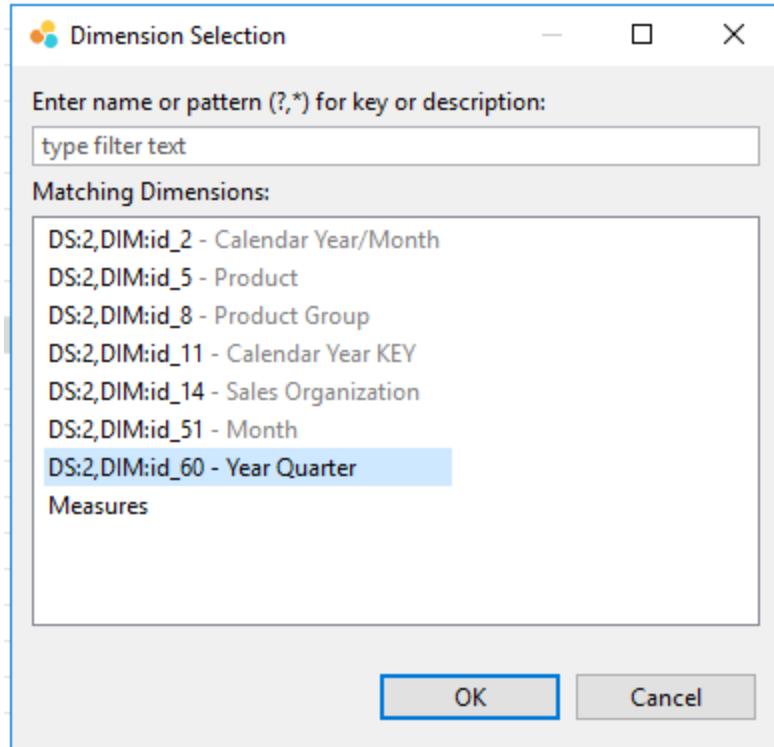
3 Populating and Filtering Data in the Dropdown Boxes and the List Box using Input and Output Binding

Data Binding is a new feature introduced with the release of Design Studio version 1.5. Data Binding is divided into Input Binding and Output binding. These in turn make it easier to bind Properties to Data Sources without having to use scripts, making the whole process of Filtering on Data Sources much easier. This is further explained in the following steps:

- 3.1 Click on dropdown YR_QUARTER present in the FILTER panel and under Properties View, go to **Display>Items**. Under the Binding column click on the '+' symbol to Bind values to the dropdown. This sets the value of Items as <bound> and opens up two new options called Source and Target.

Display		
CSS Class		Bind
Items		+
Tooltip		+

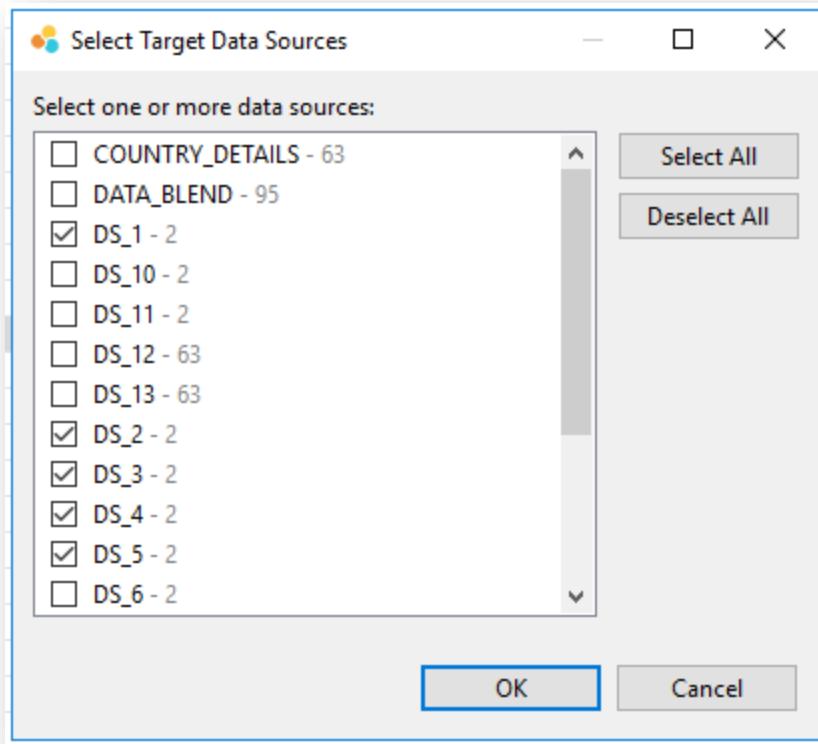
- 3.2 Expanding the Source options opens up a new set of Data Binding options. Here set the Type as Dimension Members Binding. Set the Data Source as DS_1. Under Dimensions click on the Browse... button and select Year Quarter and click OK.



3.3 Set the Member Display to Text and set the All Members Text to All Quarters. (**Note:** Here the Data Source does not matter as long as we select the Data Source containing the Year Quarter dimension we require. All Members Text will simply display All Quarters as the default selection, meaning every value is chosen in the output.)

Items		<bound>	
Source			
Type	Dimension Members Bi...		
Data Source	DS_1		
Dimension	DS:2,DIM:id_60		
Maximum Number of Mem...	100		
Member Display	Text		
All Members Text	All Quarters		
Follow Filter State	true		
Target			

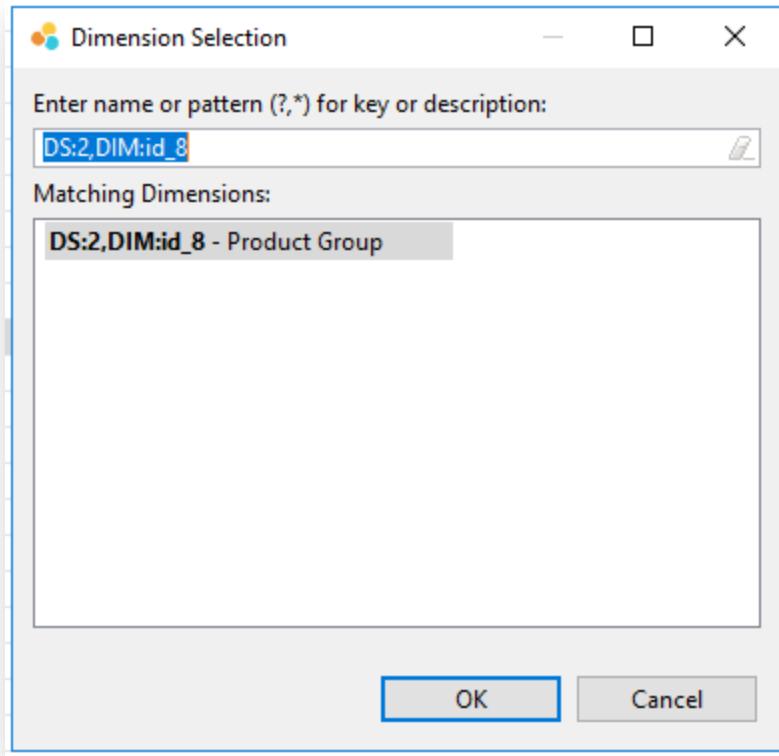
3.4 Next, expand the Target option to configure the Output binding options. Set the Type to Filter Binding. Under Target Data Sources click on the Edit Target Data Sources button, click on **DS_1**, **DS_2**, **DS_3**, **DS_4** and **DS_5**. Click OK. (**Note:** These five data sources are the ones used by our Charts and Crosstab on STORYPAGE_1)



3.5 Select the Year Quarter dimension under the Dimension field. Our target binding options are as follows:

Items	<code><bound></code>
> Source	
▼ Target	
Type	Filter Binding
Target Data Sources	<code>DS_1;DS_2;DS_3;DS_4;...</code>
Dimension	<code>DS:2,DIM:id_60</code>
Validate Members	<input checked="" type="checkbox"/> true

3.6 Next, select the PROD_GROUP dropdown. Similar to our YR_QUARTER dropdown, set its Input and Output Binding properties. One notable exception being that in both Source and Target tabs, select the Product Group dimension.



3.7 Set the All Members Text option to *All Product Groups*. The PROD_GROUP dropdown properties should look similar to below:

Items	
Source	<bound>
Type	Dimension Members Bi...
Data Source	DS_1
Dimension	DS:2,DIM:id_8
Maximum Number of Members	100
Member Display	Text and key
All Members Text	All Product Groups
Follow Filter State	true
Target	
Type	Filter Binding
Target Data Sources	DS_1;DS_2;DS_3;DS_4;...
Dimension	DS:2,DIM:id_8
Validate Members	true

3.8 The SALES_ORG List Box can be bound in the same way as the two dropdowns. Set the Dimension value in Source and Target to Sales Organization. Set the All Members Text to All Regions and set the Selection Mode to Multiple Selection

Items	<bound>
Source	
Type	Dimension Members Bi...
Data Source	DS_1
Dimension	DS:2,DIM:id_14
Maximum Number of Membr	100
Member Display	Text and key
All Members Text	All Regions
Follow Filter State	true
Target	
Type	Filter Binding
Target Data Sources	DS_1;DS_2;DS_3;DS_4;...
Dimension	DS:2,DIM:id_14
Validate Members	true
Selection Mode	Multiple Selection

3.9 We have now used Input Binding to filter data on our Charts and Crosstab without having to use any scripting.

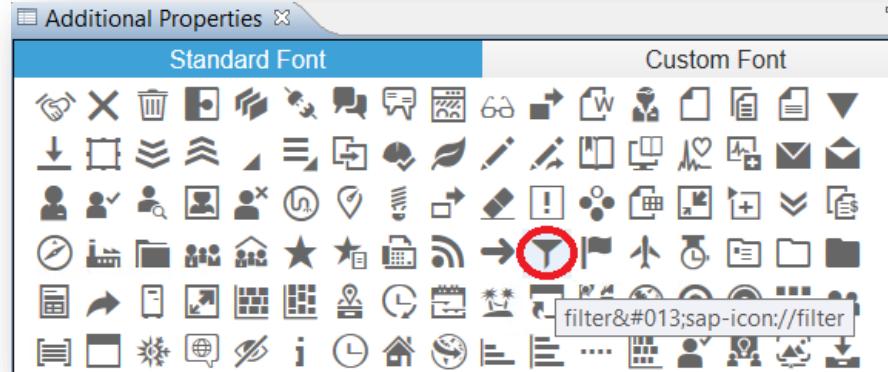
4 [Adding an Icon to Show/Hide the Filter Panel](#)

Instead of creating and uploading an Image file and adding it to the Image Component in Design Studio, we have the option to select from a wide array of preset Icons to work with and configure. Here we use an Icon Component to Show or Hide the Filter Panel as needed. Here is how:

4.1 Select the HEADER panel under the *MAIN* Panel. From the Basic Components tab add an Icon component to it. Rename it to *FILTER_ICON* and set the Margins as shown:

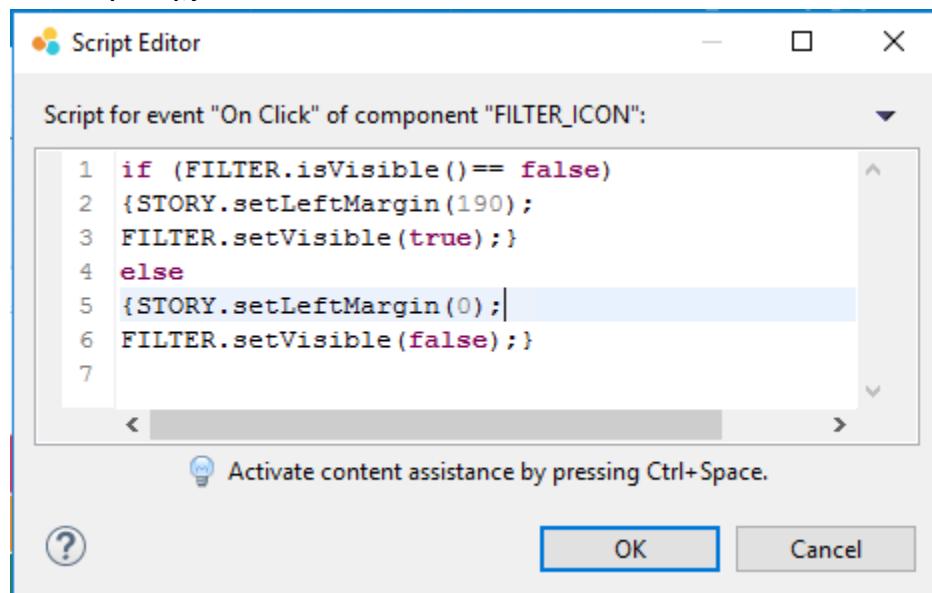
Layout	
Top Margin	25
Left Margin	5
Bottom Margin	auto
Right Margin	auto
Width	42
Height	42

4.2 Under the Additional Properties of the Icon select the Filter Icon as shown below:



4.3 Go back to the Properties View of FILTER_ICON and under Events>>On Click and click on the Edit the Script... button. Put in the following script in the Script Editor dialog box.

```
if (FILTER.isVisible() == false)
{STORY.setLeftMargin(190);
FILTER.setVisible(true);}
else
{STORY.setLeftMargin(0);
FILTER.setVisible(false);}
```

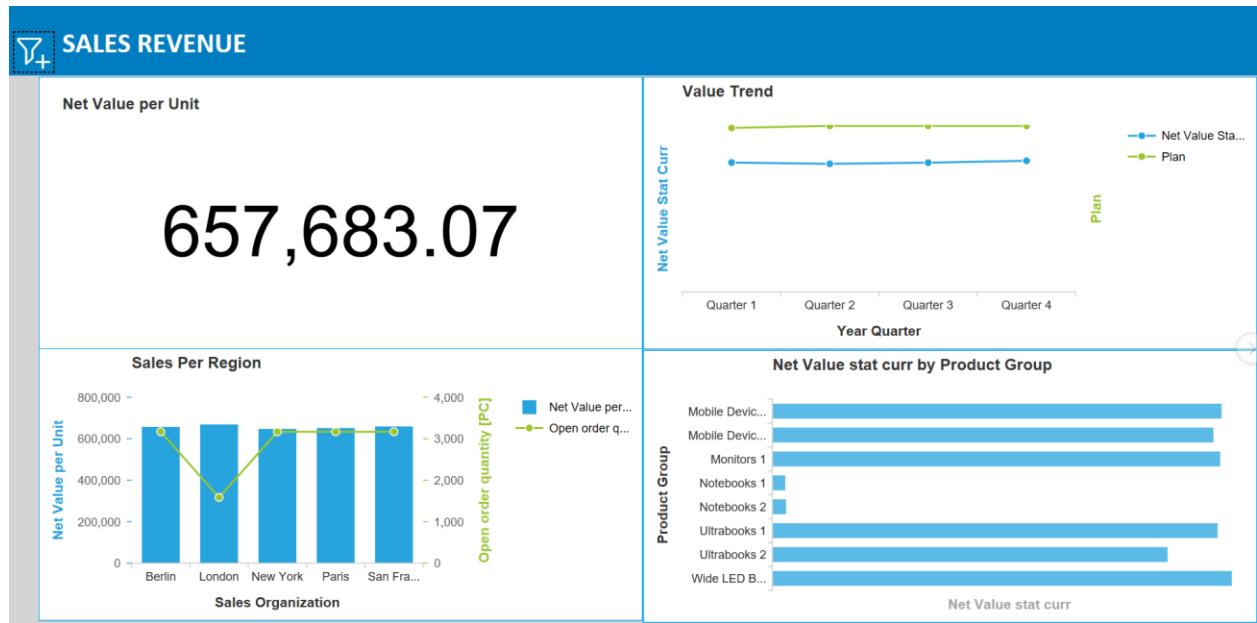


(**Note:** Here we are setting up an If/Then clause where our Story is moved to the right to accommodate the Filter Panel when the Icon is clicked once. If the Icon is clicked on a second time the Filter Panel is hidden again and the Story is reset to its original position.)

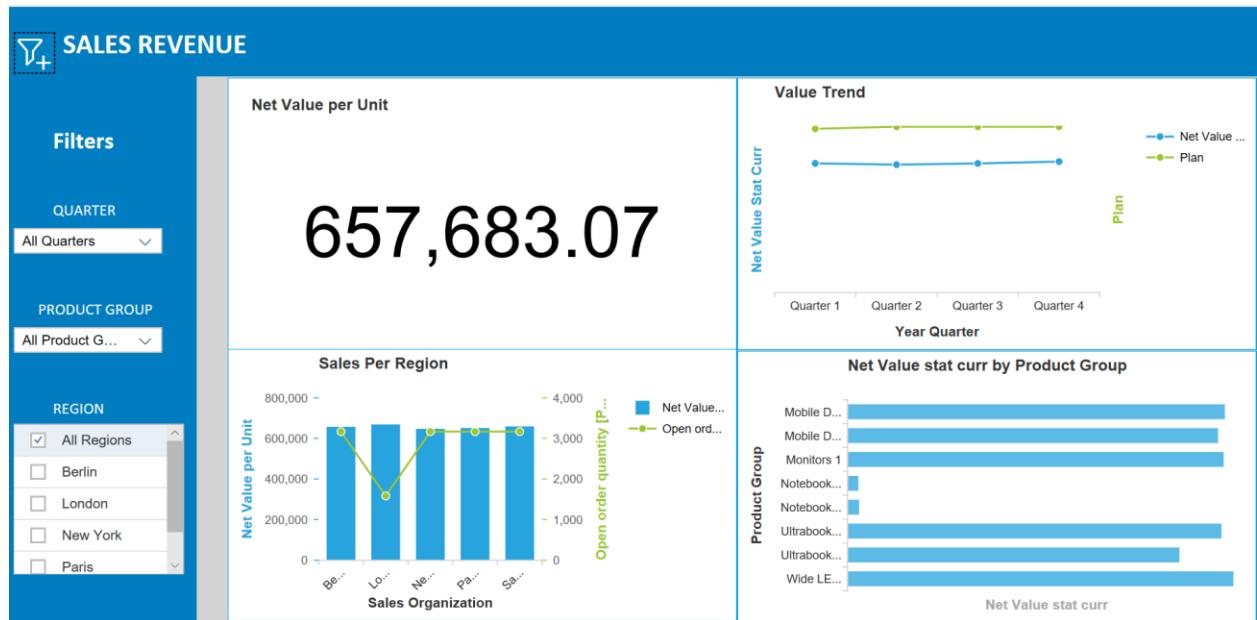
4.4 Lastly, let us now Hide the Filter panel and Show the MAIN panel.

5 Testing the Filter Panel

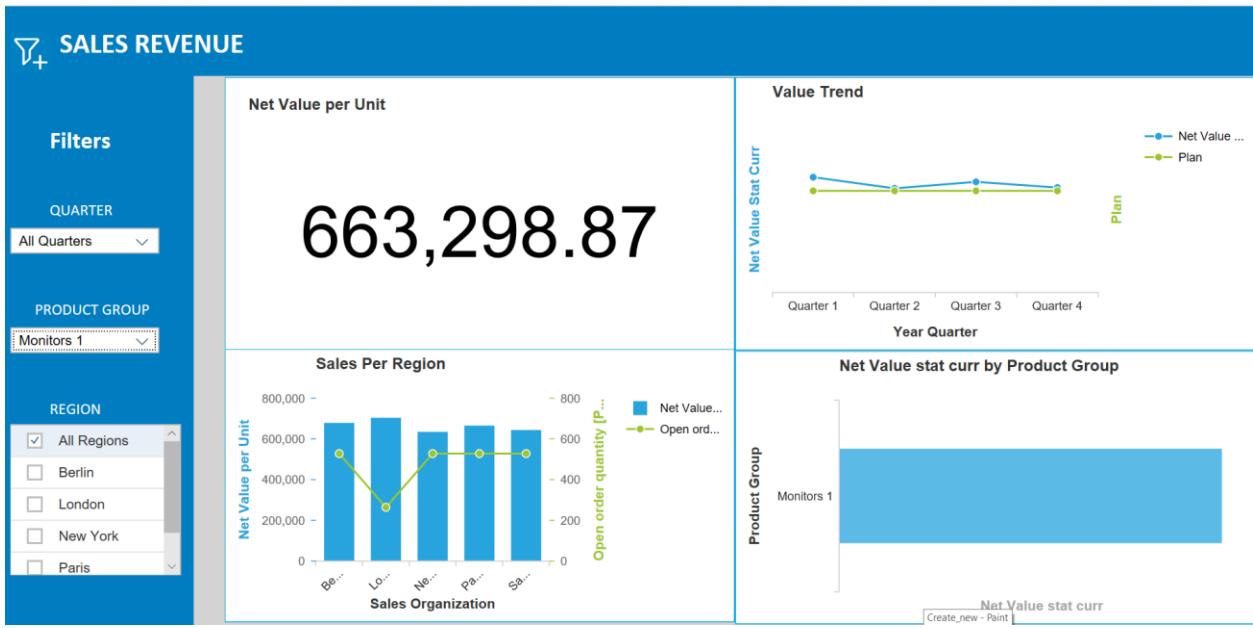
5.1 Save the application, synchronize it with the BI Platform and Execute the application on the BI Platform. The application initially appears as shown below:



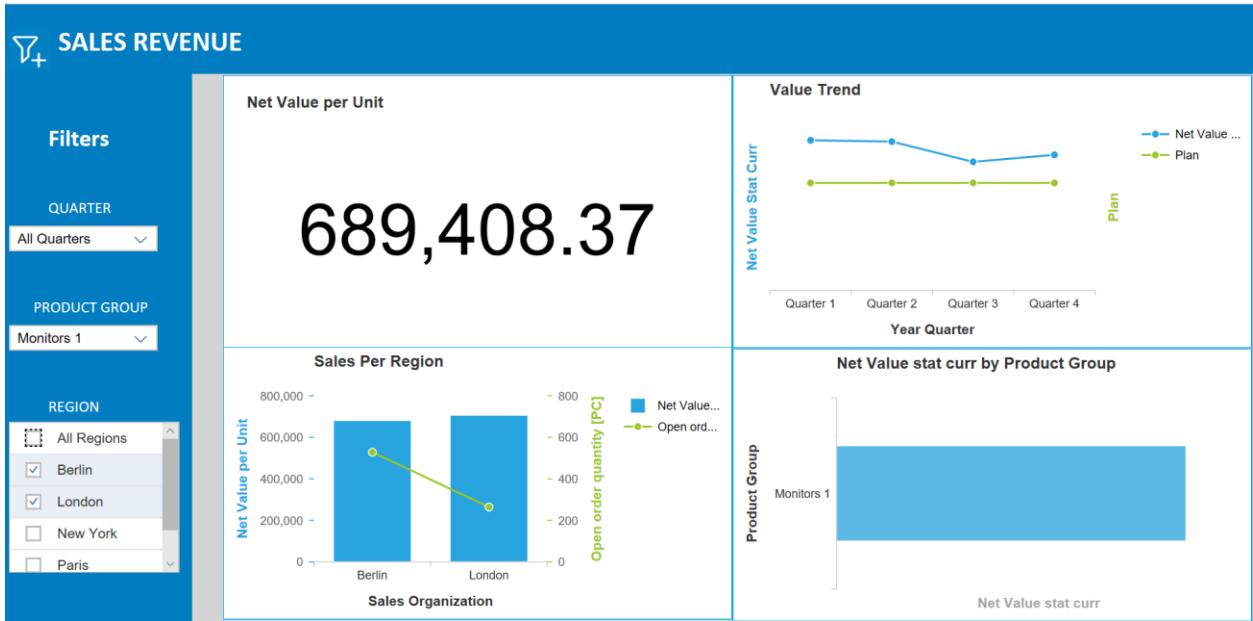
5.2 Clicking on the Filter Icon makes the Filter Panel visible. It also adjusts our Grid Layout so as to ensure the Filter Panel does not obstruct any of the Charts.



5.3 Click on the PROD_GROUP dropdown and select Monitors 1. Notice the data in our charts get filtered according to our selection:



5.4 Keeping the PROD_GROUP Filter we can click on the SALES_ORG List Box to test out the Multiple Selection feature. Click on Berlin and London after de-selecting All Regions:



5.5 Click on the Filter Icon once more to hide the Filter Panel.

SALES REVENUE

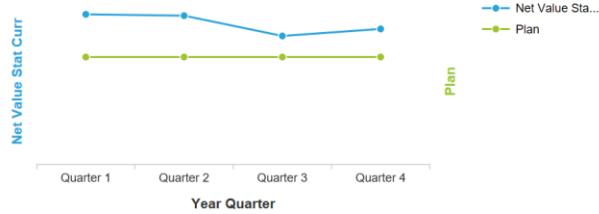
Net Value per Unit

689,408.37

Sales Per Region



Value Trend



Net Value stat curr by Product Group

Product Group

Monitors 1

Net Value stat curr

Create_new - Paint

Exercise 5 – WORKING WITH SCRIPTING

Objective of the Exercise

- Introduction
- Enabling Drilldown on Chart
- Swap Dimensions
- Page Navigation with Icon
- Getting Application Info
- Drilldown using Popup component
- Background Processing

1 [Introduction to Scripting](#)

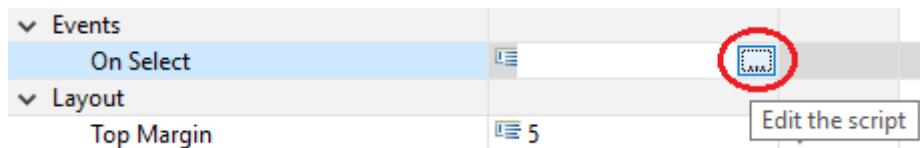
In this Exercise we take a more in-depth look at Scripting in Lumira Designer with Navigating pages in the Story, Enabling Drilldown on the Chart components, Getting Info on the Application, Swapping Dimensions in a Chart and Background Processing.

2 [Enabling Drill down on Chart](#)

Users can filter the Crosstab component by drilling down on selected columns in the Chart by using scripting. In our Designer_Demo application let us write a script such as to drill down on the Net Value stat curr by Product Group and filter values in the Sales Per Region based on the Sales Organization selected in the Chart, here is how:

2.1 From the Layout view, select the CHART_3 component present under
MAIN>>PAGEBOOK_1>>PAGE_1>>CONTAINER_GRID_1>>Cell [1, 0] and go to the Properties View.

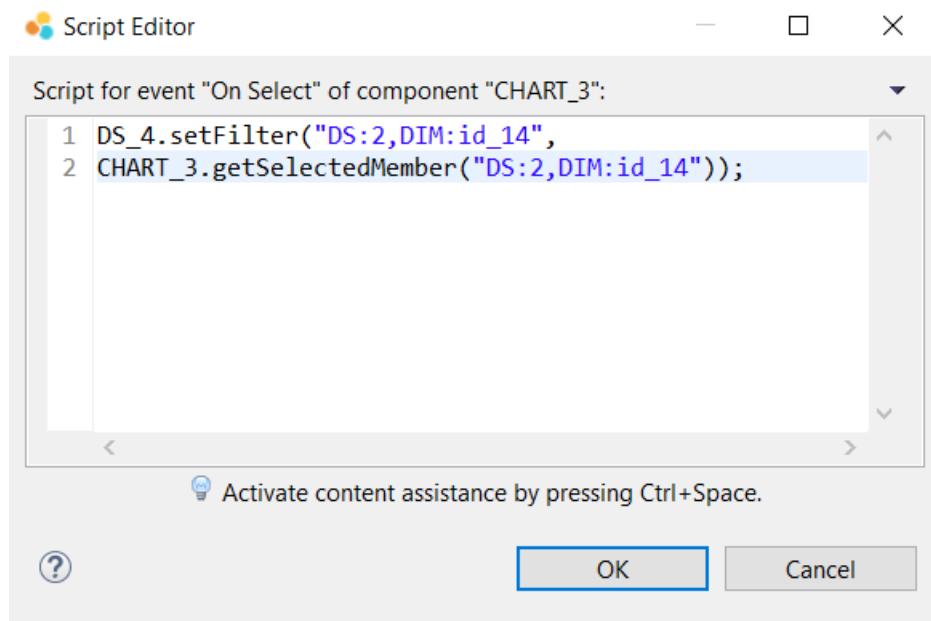
2.2 Under Events>>On Select click on the Edit the Script button.



2.3 The Script Editor dialog box opens up. Put in the following script. Click OK.

```
DS_4.setFilter("DS:2,DIM:id_14",
CHART_3.getSelectedMember("DS:2,DIM:id_14"));
```

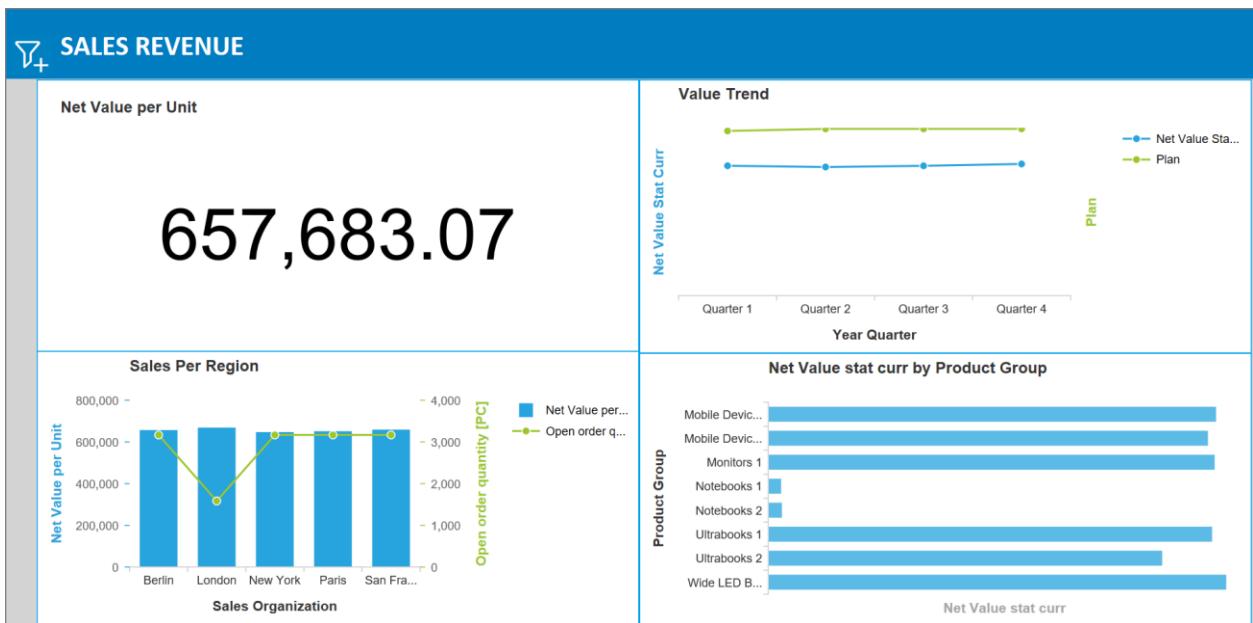
2.4 The script editor appears as so:



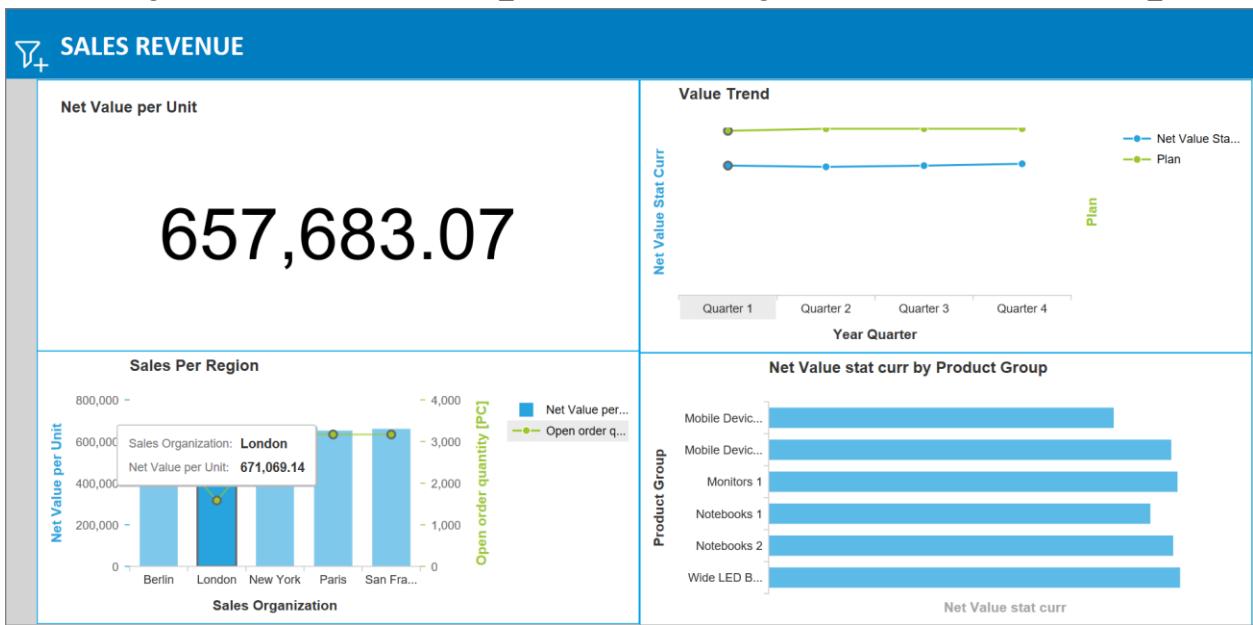
(Note: Here we set a filter on the dataset associated with the Chart_4 (DS_4) based on the Sales Organization category item (DS:2, DIM:id_14) selected in the CHART_3 component)

2.5 Right click the Designer_Demo application from within the Documents tab and select Execute on the BI Platform.

2.6 The application initially appears as shown (**Note:** The Chart_3 initially shows multiple Sales Organization values):



2.7 Selecting the London column in Chart_3 results in the Change in values in chart values Chart_4



3 Adding an Icon to navigate between Story pages

Users can switch between Story pages in a quick and easy way using scripting. We do this by configuring an Icon Component. Here is how:

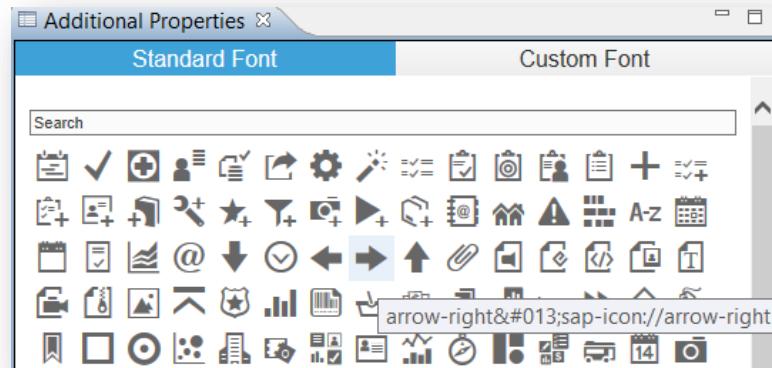
3.1 Add an Icon Component to the HEADER Panel under the MAIN Panel and set its their margins as shown:

Layout	
Top Margin	25
Left Margin	auto
Bottom Margin	auto
Right Margin	230
Width	42
Height	42

3.2 Rename the Icon Component as NAVIGATOR. Under Display set the Tooltip option to Navigator.

Under Additional Properties select the following Icon:

sap-icon://arrow-right



3.3 Under Events>>On Click... of the NAVIGATOR icon select the Edit the Script button and under the Script Editor dialog box that opens up, type in the following:

```
var i = PAGEBOOK_1.getSelectedPageIndex();
if (i == 0)
{
    PAGEBOOK_1.setSelectedPageByName("PAGE_2");
    NAVIGATOR.setIconUri("sap-icon://arrow-left");
}
else {
    PAGEBOOK_1.setSelectedPageByName("PAGE_1");
    NAVIGATOR.setIconUri("sap-icon://arrow-right");
}
```

Script Editor

Script for event "On Click" of component "NAVIGATOR":

```

1 var i = PAGEBOOK_1.getSelectedPageIndex();
2 if (i == 0)
3 {
4     PAGEBOOK_1.setSelectedPageByName("PAGE_2");
5     NAVIGATOR.setIconUri("sap-icon://arrow-left");
6 }
7 else {
8     PAGEBOOK_1.setSelectedPageByName("PAGE_1");
9     NAVIGATOR.setIconUri("sap-icon://arrow-right");
10}
11

```

Activate content assistance by pressing Ctrl+Space.

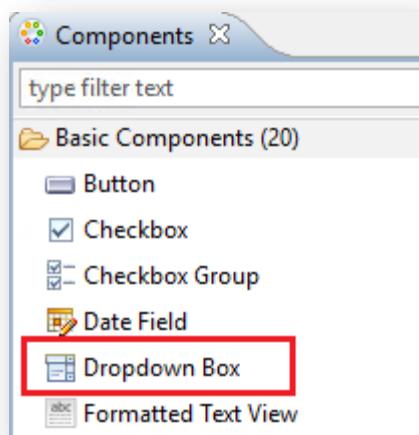
OK Cancel

(Note: Upon clicking on the icon the script above checks the Story page index. If the Index is 0(i.e. we are on the first page) then it sets the Navigator Icon to a right arrow and goes to the next page. Else, it goes back to the first page and sets the icon to a left arrow.)

4 [Adding a Dropdown Box to Swap Dimensions](#)

Users can swap dimensions in a given Chart using scripting. In this section we will swap dimensions for our CHART_6 component, here is how:

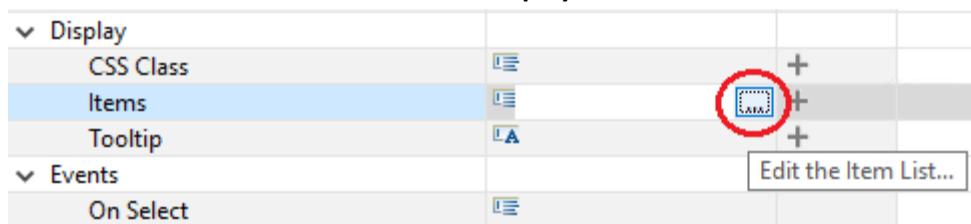
- 4.1 From under the Components tab, go to Basic Components, select a Dropdown Box and add it to Cell [0, 0] of CONTENT_GRID_7 under STORYPAGE_2.



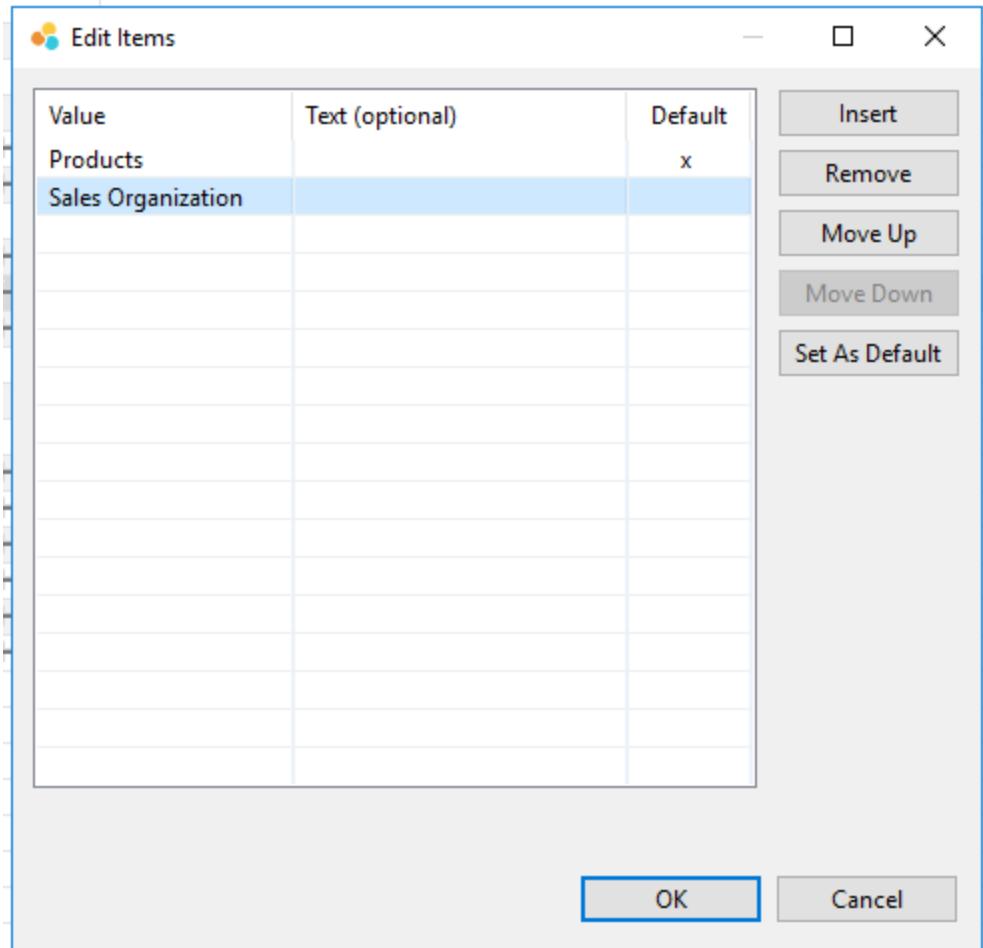
4.2 Rename the component to SWAP_DIMENSIONS and from under the Properties view, set the **CSS Class** as *panel-box* and margins of the component as shown here:

Layout	
Top Margin	0
Left Margin	auto
Bottom Margin	auto
Right Margin	15
Width	150
Height	auto

4.3 Next, select the Edit the Item List button under **Display>>Items**.

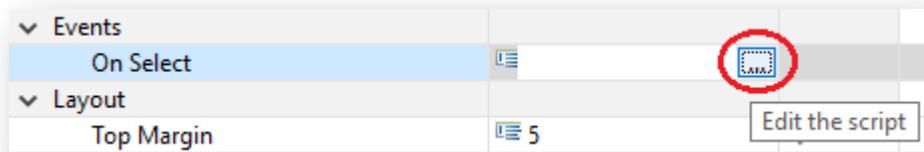


4.4 The Edit Items dialog box opens up. Select the Insert button and insert two values into the dropdown box list, *Products* and *Sales Organization*. Click **OK**.



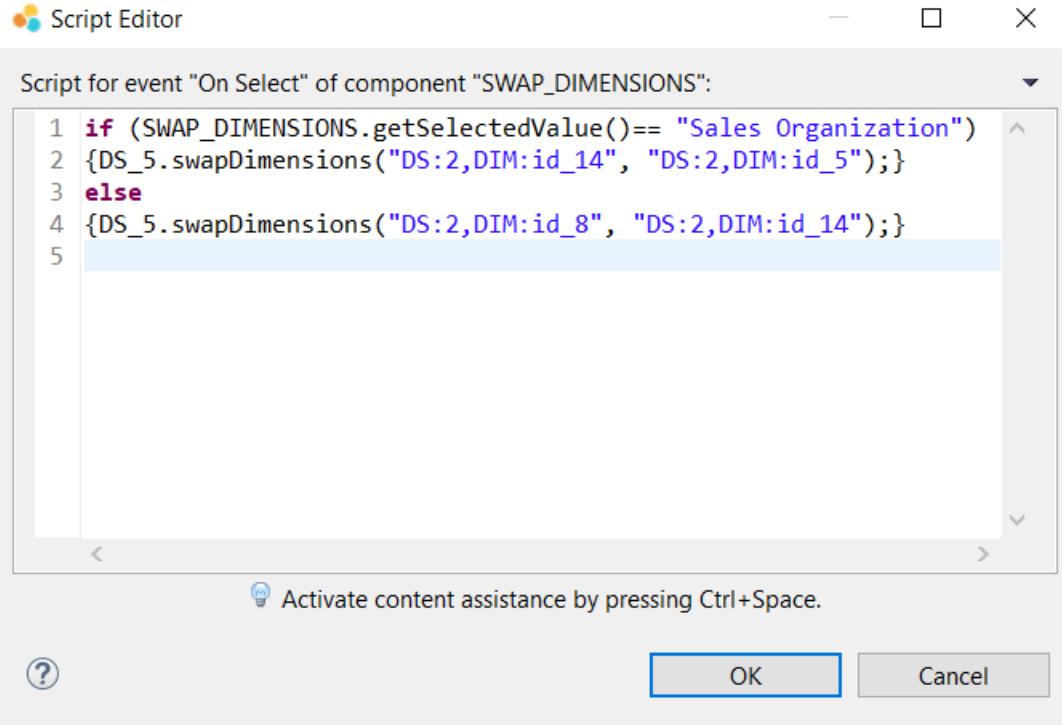
(Note: Keep the Products value as the default, ensuring that the dropdown box displays the value Products during runtime initially.)

4.5 Next, under Events>>On Select click the Edit the Script button.



4.6 Under the Script Editor dialog box, input the following script.

```
if (SWAP_DIMENSIONS.getSelectedValue() == "Sales Organization")
{DS_5.swapDimensions("DS:2,DIM:id_14", "DS:2,DIM:id_5");}
else
{DS_5.swapDimensions("DS:2,DIM:id_8", "DS:2,DIM:id_14");}
```



(Note: This script filters the data source of the Bubble Chart (DS_5) to swap dimensions between Products and Sales Organization based on the value selected in the SWAP_DIMENSIONS dropdown box.)

5 [Enabling Drilldown using a Popup Component](#)

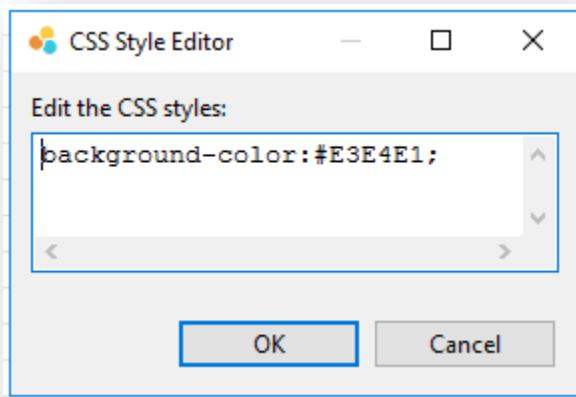
In this section we use scripting to drill down on our Crosstab component. Selecting a region in the crosstab brings up a Chart comparing Net Value and Plan values for that particular region.

- 5.1 Right click on the layout folder and select **Create>>Container Components>>Popup**. Rename it to **POP_DRILLDOWN** and set its margins as shown:

Property	Value
General	
Name	POP_DRILLDOWN
Type	Popup
Vendor	SAP SE
Display	
CSS Class	
Modal	false
Autoclose	false
Background Design	Translucent
Layout	
Top Margin	200
Left Margin	200
Bottom Margin	auto
Right Margin	auto
Width	900
Height	400

5.2 Add a Panel component to POP_DRILLDOWN. Set its CSS Class to *panel-box*. Set its CSS Style to the following:

background-color:#E3E4E1;



5.3 Set its margins as shown:

Layout	
Top Margin	3
Left Margin	3
Bottom Margin	3
Right Margin	3
Width	auto
Height	auto

5.4 Next, from under Chart Components, add a Chart to the panel. Set its name to CHART_DRILLDOWN and set its margins as shown:

Layout	
Top Margin	22
Left Margin	0
Bottom Margin	0
Right Margin	0
Width	auto
Height	auto

5.5 From under the Data Sources folder copy/paste the DS_1 data source. Rename the new data source to DS_DRILLDOWN. Edit the initial view and set *Net Value Stat Curr* and *Plan* under Measures. Set *Product Group* as the dimension. Set *Sales Organization* under Background Filter.

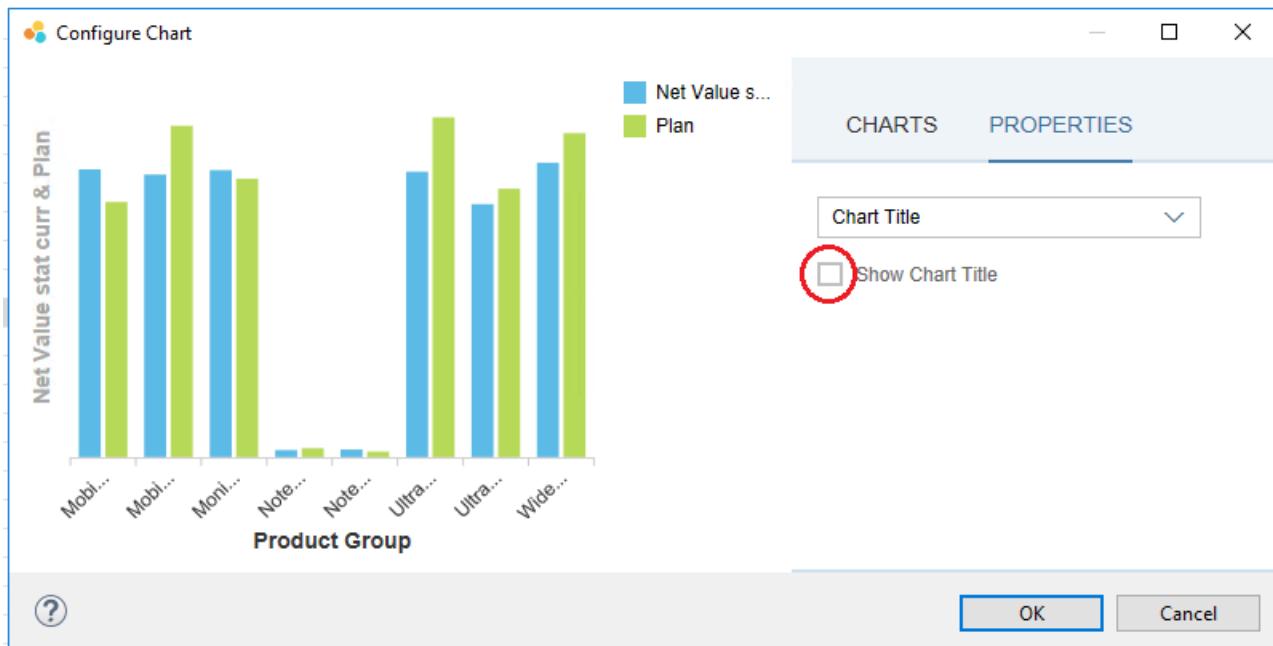
The screenshot shows the 'Edit Initial View' dialog for the data source 'DS_DRILLDOWN'. The left pane displays a hierarchical tree structure with nodes such as 'Measures', 'Rows', and 'Background Filter'. The 'Background Filter' node has 'Sales Organization - DS:2,DIM' selected. The right pane shows a 'Live Preview' table with 16 data cells, displaying sales data for different product groups. The table has columns for 'Product Group', 'Net Value stat curr', and 'Plan'. The data includes rows for Mobile Devices 1, Mobile Devices 2, Monitors 1, Notebooks 1, Notebooks 2, Ultrabooks 1, Ultrabooks 2, and Wide LED Backlight 1, with corresponding values for Net Value and Plan. At the bottom of the dialog, there are buttons for '?', 'Undo', 'Redo', 'OK + Create Crosstab', 'OK' (which is highlighted in blue), and 'Cancel'.

Product Group	Net Value stat curr	Plan
Mobile Devices 1	379,485,801,816	336,616,684,847.52
Mobile Devices 2	372,621,885,372	436,564,464,549.84
Monitors 1	378,299,201,808	367,045,998,105.60
Notebooks 1	11,318,041,812	13,808,011,010.64
Notebooks 2	11,994,144,360	9,235,491,157.20
Ultrabooks 1	376,128,947,568	447,593,447,605.92
Ultrabooks 2	333,868,602,771	353,900,718,937.26
Wide LED Backlight 1	388,074,368,088	426,859,193,009.04

5.6 Go to the Properties View of our CHART_DRILLDOWN chart. Set the data source as DS_DRILLDOWN. Under **Display>>Chart Configuration** click on the **Configure Chart...** button.

Property	Value	Binding
General		
Name	CHART_DRILLDOWN	
Type	Chart	
Vendor	SAP SE	
Visible	true	+
Data Binding		
Data Source	DS_DRILLDOWN	
Data Selection		
Display		
CSS Class		
Chart Configuration	Column Chart	
Allow Data Source Modification	false	
Conditional Formatting Visible	false	
Conditional Formatting Settings		

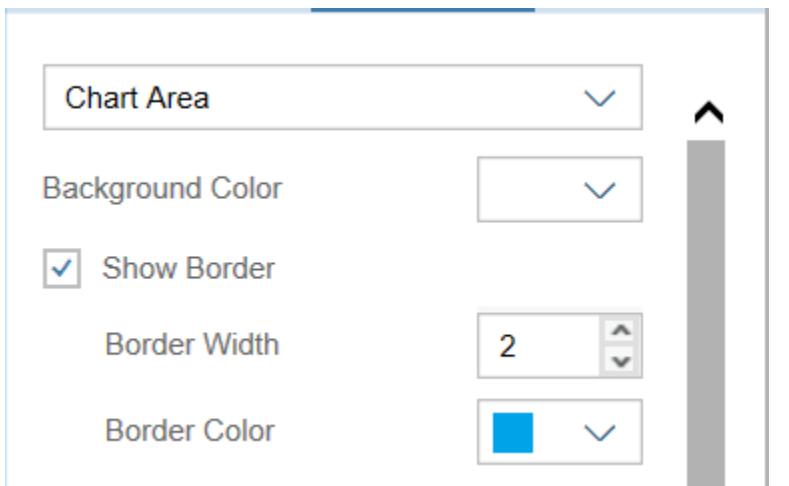
5.7 Under the Configure Chart dialog box go to the **Properties** Tab and select **Chart Title**. Uncheck the **Show Chart Title** checkbox.



5.8 Also Select Chart Area from Drop Down and set the following properties:

Background-color: White

Border-width:2, color: blue



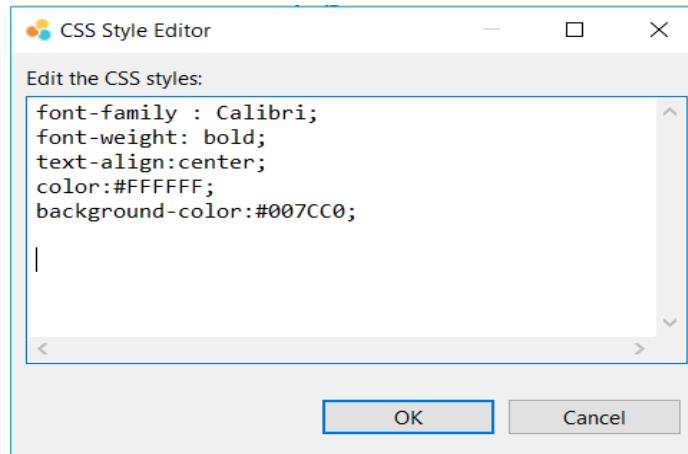
5.9 Next, add a Button component to POP_DRILLDOWN. Rename it to *CLOSE_DRILL*. Set its **Text** property to **Close**. Set its margins as shown:

General	
Name	<code>CLOSE_DRILL</code>
Type	Button
Vendor	SAP SE
Visible	✓ true
Enabled	✓ true
Display	
CSS Class	
Text	Close
Icon	
Tooltip	
Button Type	Default
Events	
On Click	
Layout	
Top Margin	3
Left Margin	auto
Bottom Margin	auto
Right Margin	3
Width	75
Height	20

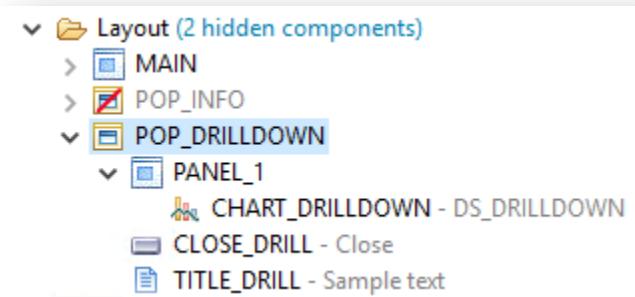
5.10 Finally, add a Text component to POP_DRILLDOWN. Rename it to *TITLE_DRILL*. Set its margins as shown:

Layout	
Top Margin	5
Left Margin	5
Bottom Margin	auto
Right Margin	auto
Width	400
Height	20

5.11 Set the CSS Style of TITLE_DRILL as follows:



5.12 The layout of our POP_DRILLDOWN popup should now appear as shown:



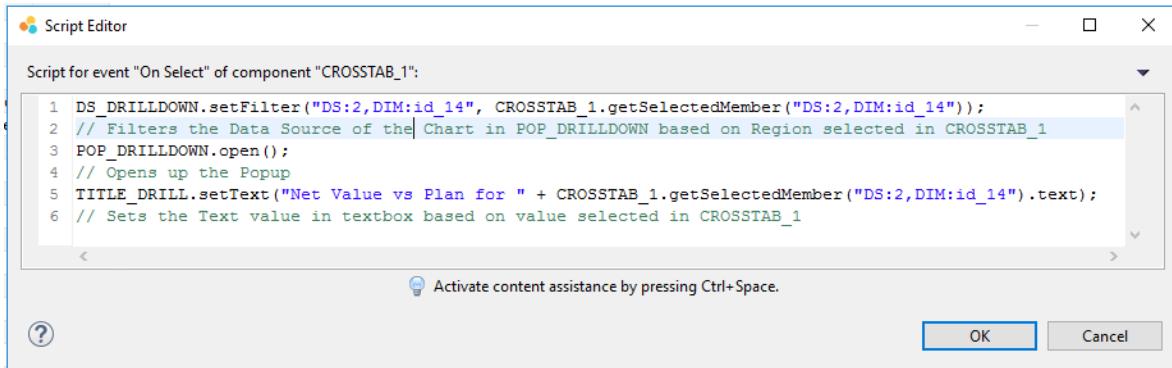
5.13 Select CROSSTAB_1 under Cell [1, 0] of CONTENT_GRID_1 under STORYPAGE_1. Go to Properties View>>Events>>On Select... and put in the following script. Click OK.

```
DS_DRILLDOWN.setFilter("DS:2,DIM:id_14",
CROSSTAB_1.getSelectedMember("DS:2,DIM:id_14"));
// Filters the Data Source of the Chart in POP_DRILLDOWN based on Region selected in
CROSSTAB_1
```

```

POP_DRILLDOWN.open();
// Opens up the Popup
TITLE_DRILL.setText("Net Value vs Plan for " +
CROSSTAB_1.getSelectedMember("DS:2,DIM:id_14").text);
// Sets the Text value in textbox based on value selected in CROSSTAB_1

```

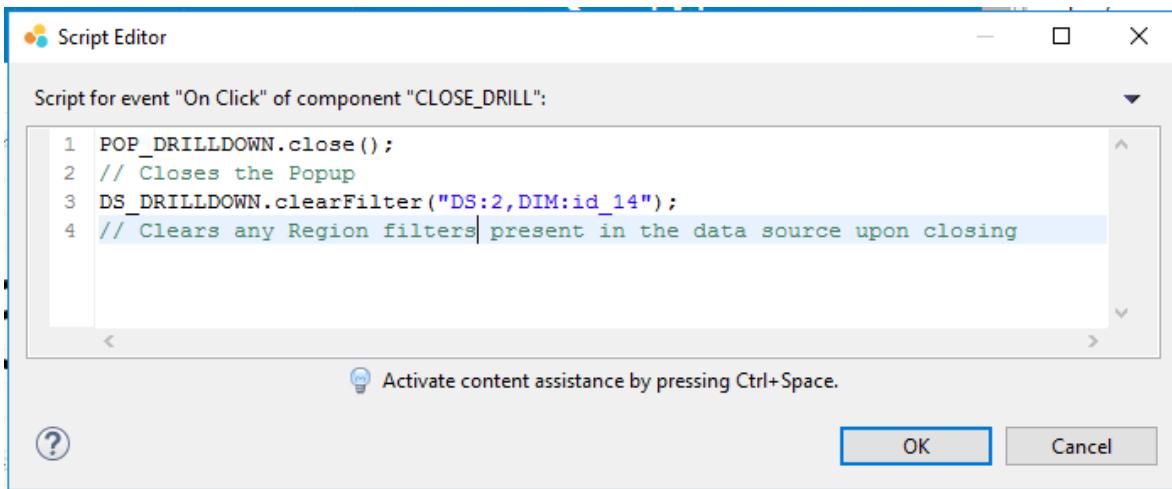


5.14 Next, go to CLOSE_DRILL button present under POP_DRILLDOWN. Under **Properties** View>>Events>>On Select... put in the following script. Click **OK**.

```

POP_DRILLDOWN.close();
// Closes the Popup
DS_DRILLDOWN.clearFilter("DS:2,DIM:id_14");
// Clears any Region filters present in the data source upon closing

```



5.15 The drilldown popup has been successfully configured. Right click on POP_DRILLDOWN in the Outline View and from the content menu select **Hide**.

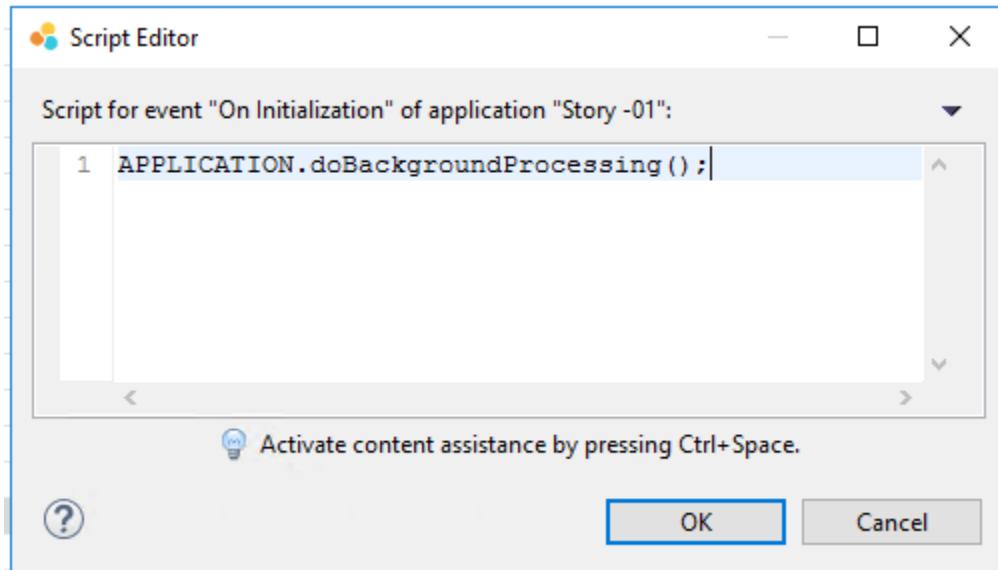
6 Background Processing

Background processing enables us to load some of the Data Sources after initial startup of the application. By this we enable the application to load faster.

6.1 In the present demo application the DS_DRILLDOWN Data Source is assigned to CHART_DRILLDOWN (Present under POP_DRILLDOWN). As the popup is not shown when the application is initially loaded, there is no requirement to make the data source load at the start-up of the Application. We can make this Data Source load after loading the remaining Data Sources. This makes the application load faster and decreases end user wait time.

6.2 To enable this click on **Story -01** in the Discovery_Demo Application. In the Properties View under **Events>>On Initialization**, click on Edit the Script button and add the below given script.

```
APPLICATION.doBackgroundProcessing();
```

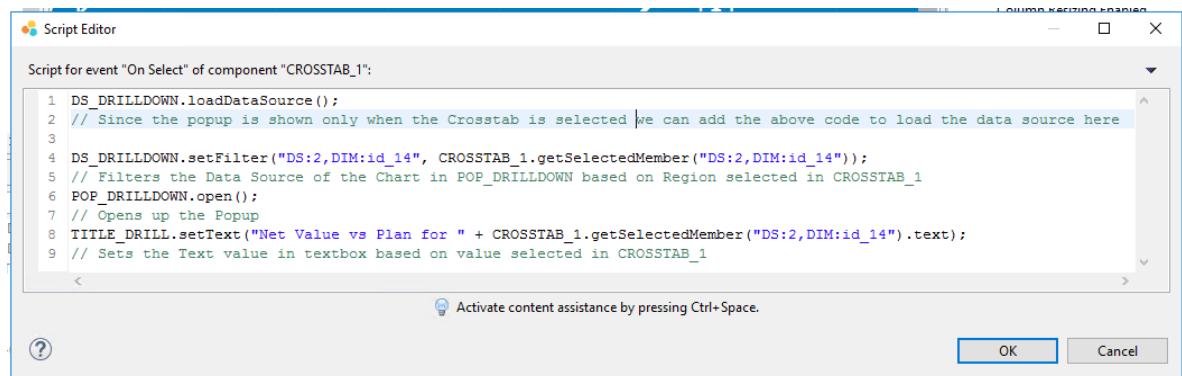


(**Note:** This script enables us to run the script present under the Events>>On Background Processing property)

6.3 Next, go to CROSSTAB_1 under Cell [1, 0] of CONTENT_GRID_1 under STORYPAGE_1. Go to Properties View>>Events>>On Select... and put in the following script. Click OK.

```
DS_DRILLDOWN.loadDataSource();
```

```
// Since the popup is shown only when the Crosstab is selected we can add the above code to load the data // source here
```



6.4 In the Properties View of DS_DRILLDOWN data source under Data Binding set the **Load in Script** property to **True**.

Property	Value
General	
Name	DS_DRILLDOWN
Type	Lumira DataSet
Vendor	SAP SE
Data Binding	
Load in Script	✓ true
Data Source	
Processing Group	<default group>

7 Getting and Displaying Application Info

We want to configure a Popup Component such that it displays pertinent information on the Analysis Application when accessed through an Icon Component.

7.1 Add a Popup to the application directly under Layout and rename it as **POP_INFO**. In the Properties View under Display set the Modal property as True and set the Margins as shown below:

Layout	
Top Margin	300
Left Margin	650
Bottom Margin	auto
Right Margin	auto
Width	600
Height	300

7.2 Add a Panel component to the POP_INFO popup. Set its name to INFO_PANEL set its CSS Style and margins as shown:
background-color:#E3E4E1;

Layout	
Top Margin	3
Left Margin	3
Bottom Margin	3
Right Margin	3
Width	auto
Height	auto

7.3 Add a Grid Layout to the INFO_PANEL. Rename it as INFO_GRID and in the properties view set the margins and number of rows as shown below:

Layout	
Top Margin	0
Left Margin	0
Bottom Margin	0
Right Margin	0
Width	auto
Height	auto
Number of Rows	
Row Height	1
> Number of Columns	
	1

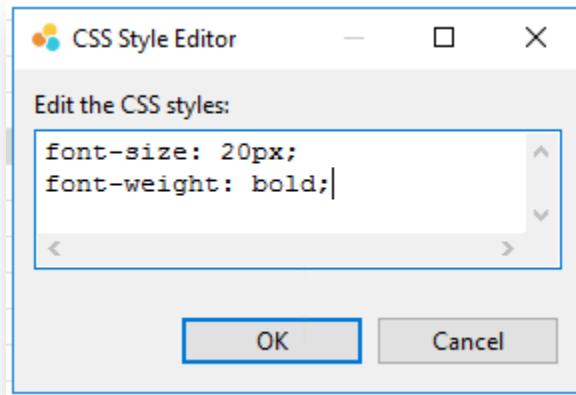
7.4 Now, add 5 Text Components in each of the Rows of the Grid Layout, set the margins of all the text boxes as shown below:

Layout	
Top Margin	0
Left Margin	0
Bottom Margin	0
Right Margin	0
Width	auto
Height	auto

7.5 Rename the 5 Text Components respectively to *APPLICATION_NAME*, *DATA_SOURCES*, *DESCRIPTION*, *LAST_DATAUPDATED* and *APPLIED_FILTERS*.

7.6 In the Properties View of each of the Text boxes, under Display, set the CSS Style as the following. This enables us to increase the size and weight of the font.

```
font-size: 20px;  
font-weight: bold;
```

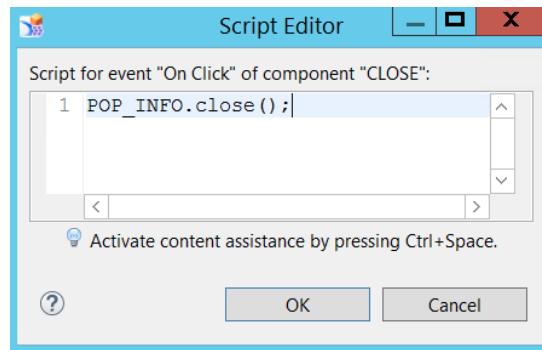


7.7 Add a Button component to Cell [0, 0] in the INFO_GRID. Set its Name as *INFO_CLOSE* its Text property as Close and set the margins as shown below:

Layout	
Top Margin	auto
Left Margin	540
Bottom Margin	35
Right Margin	auto
Width	50
Height	20

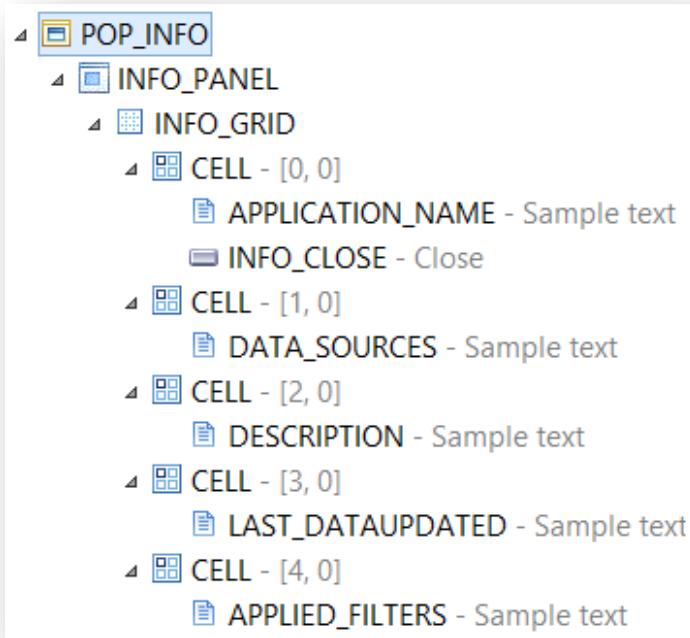
7.8 Add the following script in the Events>>On Click Script Editor Dialog box:

```
POP_INFO.close();
```



(Note: This script simply enables us to close the Popup on click)

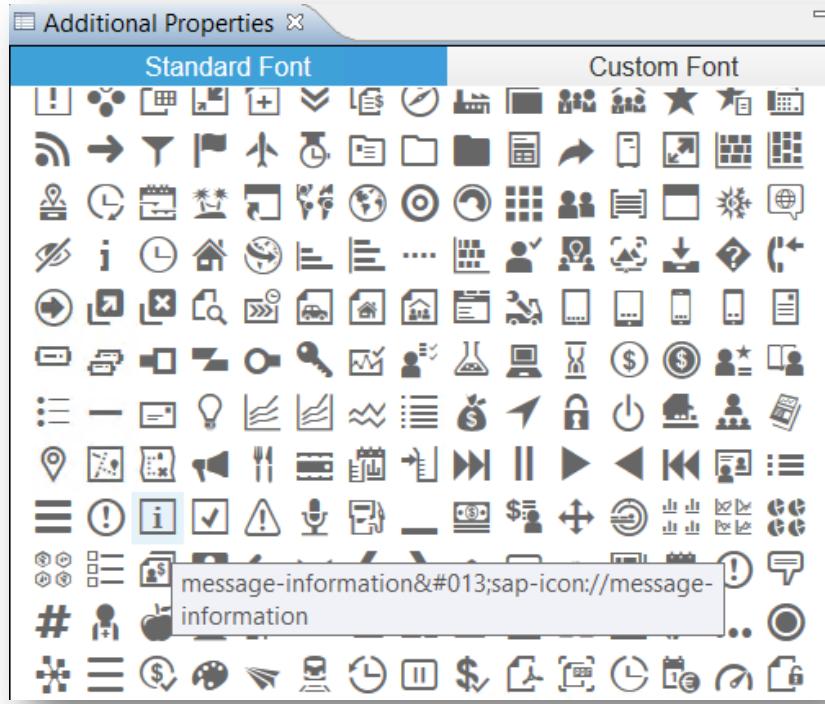
7.9 The outline view for the Popup should appear similar to what's shown below:



7.10 Add an Icon Component to the HEADER Panel. Set its name as INFO_ICON and under Properties View>>Display set the Tooltip to Get Info. Set its margins as shown below:

Layout	
Top Margin	25
Left Margin	auto
Bottom Margin	auto
Right Margin	160
Width	42
Height	42

7.11 Under Additional Properties choose the following Icon:



7.12 Under Events>>On Click click on the Edit the Script button. In the Script Editor put in the following script:

```

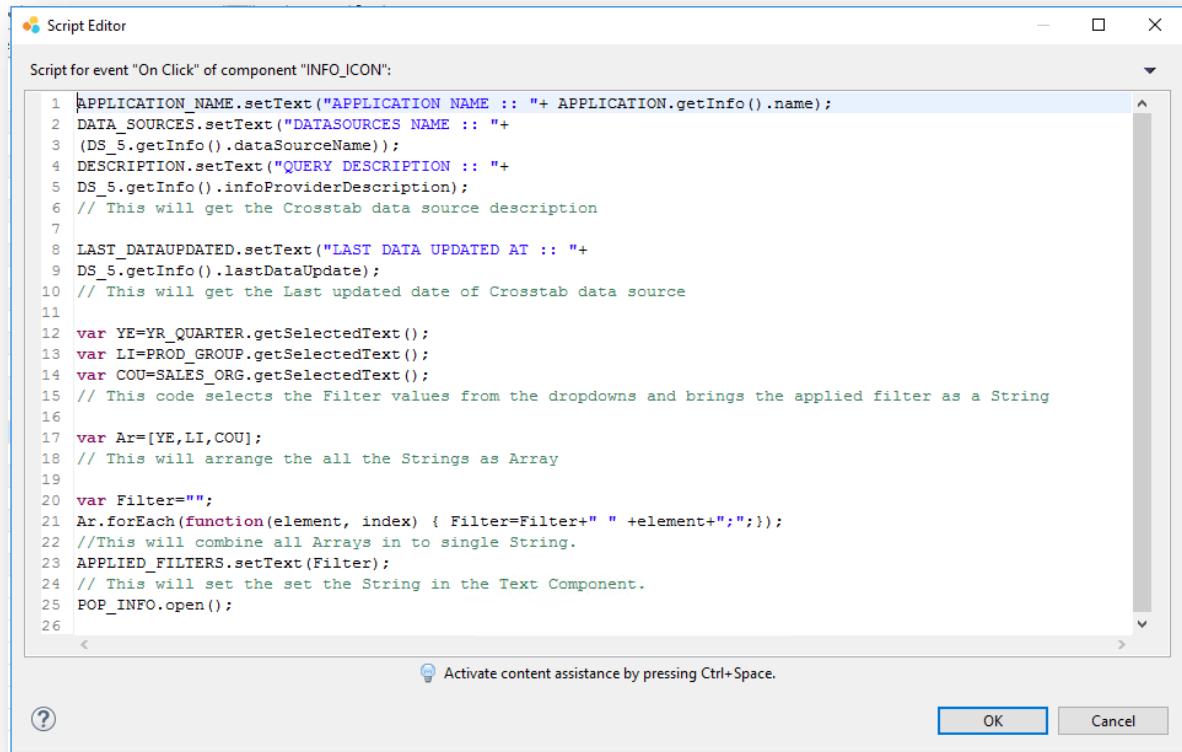
APPLICATION_NAME.setText("APPLICATION NAME :: "+ APPLICATION.getInfo().name);
DATA_SOURCES.setText("DATASOURCES NAME :: "+
(DS_5.getInfo().dataSourceName));
DESCRIPTION.setText("QUERY DESCRIPTION :: "+
DS_5.getInfo().infoProviderDescription);
// This will get the Crosstab data source description
LAST_DATAUPDATED.setText("LAST DATA UPDATED AT :: "+
DS_5.getInfo().lastDataUpdate);
// This will get the Last updated date of Crosstab data source
var YE=YR_QUARTER.getSelectedText();
var LI=PROD_GROUP.getSelectedText();
var COU=SALES_ORG.getSelectedText();
// This code selects the Filter values from the dropdowns and brings the applied filter as a
String
var Ar=[YE,LI,COU];
// This will arrange the all the Strings as Array
var Filter="";
Ar.forEach(function(element, index) { Filter=Filter+" "+element+";"});
//This will combine all Arrays in to single String.

```

```

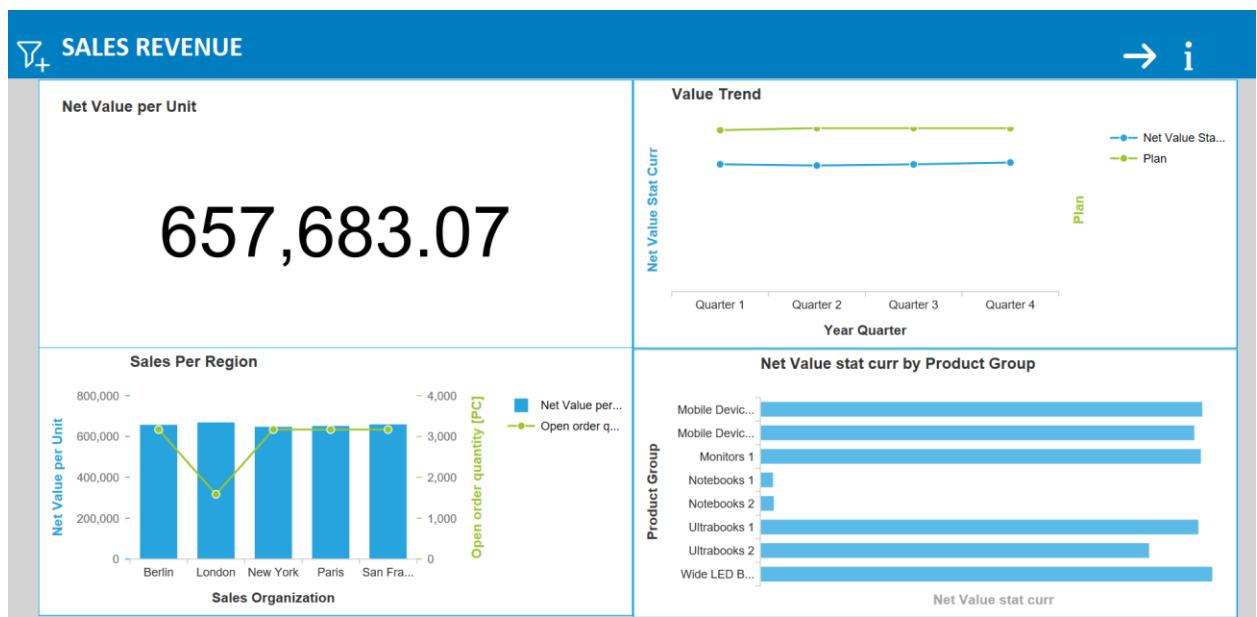
APPLIED_FILTERS.setText(Filter);
// This will set the set the String in the Text Component.
POP_INFO.open();

```

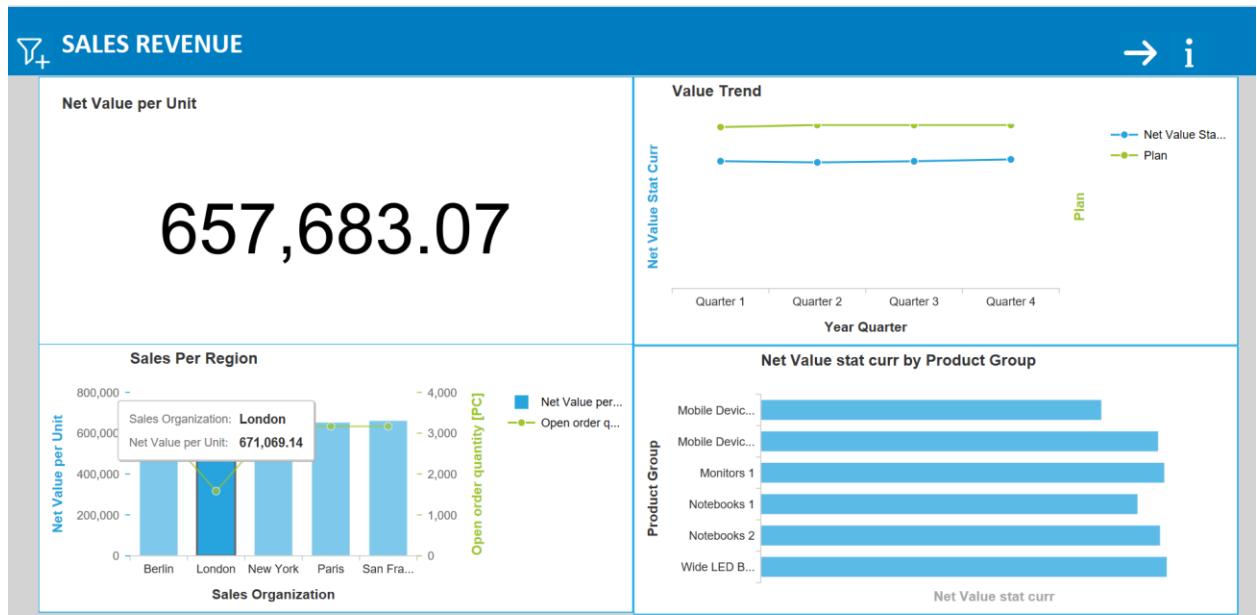


8 Testing the Application

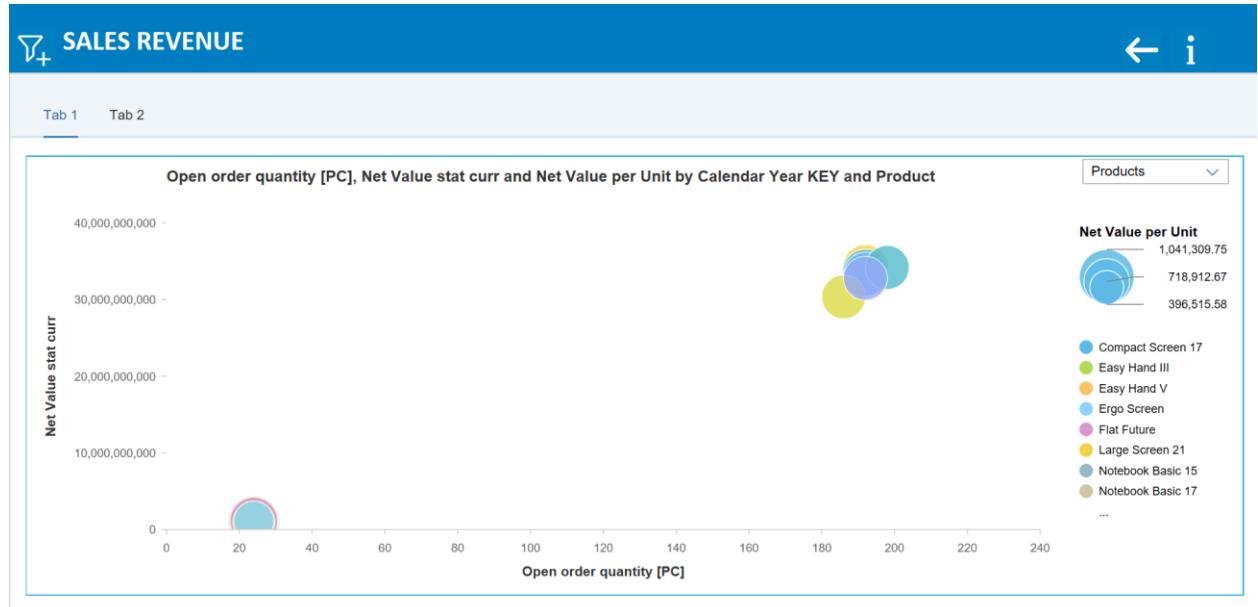
8.1 Right click on the Document from within the Documents tab. Save, synchronize and Execute on the BI Platform. Our application initially looks as shown:



8.3 On Page 1, selecting the London column in CHART Sales Per Region changes Net Value Curr Stat by Prod Group only for London



8.2 Click on the Right Navigator arrow to go to Storypage_2.



(**Note:** Arrow switches to a Left Arrow once on Storypage_2)

- 8.3 Go to the Bubble chart and click on the Swap Dimensions dropdown and select Sales Organization. Correspondingly the values in the Bubble Chart change based on the Sales Organization dimension.



- 8.4 Select the Tab2. Select the Sales Organization London from our CROSSTAB_1 component. We instantly notice our DS_DRILLDOWN data source is now loaded and shows us Net Value vs. Plan values for London region. (**Note:** The Text box now shows the title of the Chart)



8.5 Close the Popup using the Close button. Next, select the Info icon to get information on the Chart.

APPLICATION NAME :: DESIGNER_DEMO

CLOSE

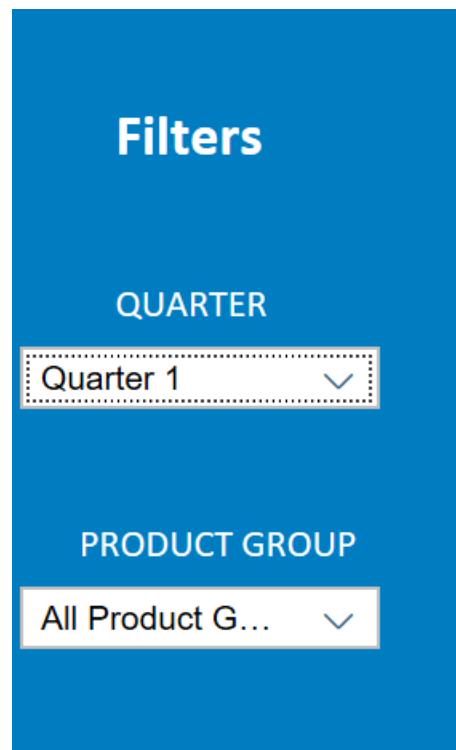
DATASOURCES NAME :: DS_1DS_2DS_3DS_4

QUERY DESCRIPTION :: 0D_FC_NW_C01_Q0005

LAST DATA UPDATED AT ::

All Quarters; All Product Groups; All Regions;

8.6 Close the Info panel using the Close button. Next, use the Filter Panel to set the Filter as Quarter 1.



8.7 Selecting the Info icon once again reflects the filter set on the application.

APPLICATION NAME :: DESIGNER_DEMO Close

DATASOURCES NAME :: DS_1DS_2DS_3DS_4

QUERY DESCRIPTION :: 0D_FC_NW_C01_Q0005

LAST DATA UPDATED AT ::

Quarter 1; All Product Groups; All Regions;

Exercise 6 – WORKING WITH MAP COMPONENT

Objective of the Exercise

- Introduction
- Adding and Configuring the Map Component
- Adding Offline Data Sources through Discovery
- Configuring Map and Chart
- Drilldown using scripting

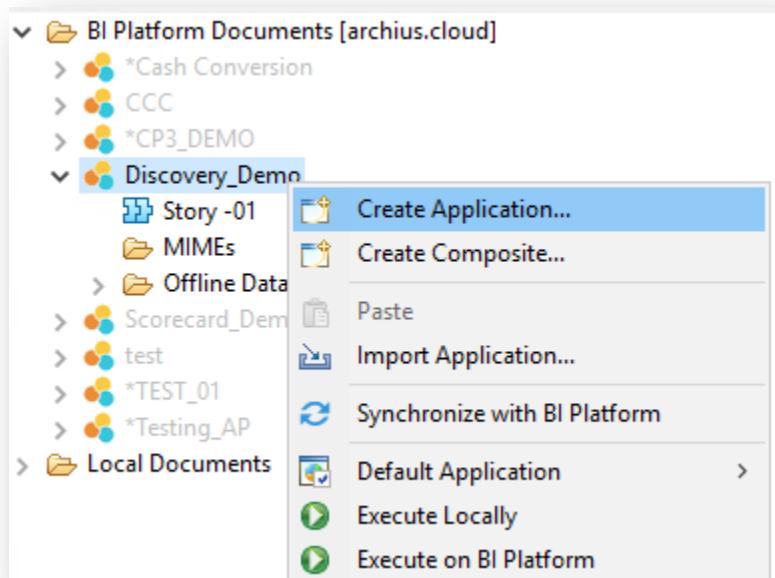
Introduction to the Map Component

In this Exercise we compare the Net Income from sales across multiple Countries and US States by using a Map component. We will add a double layered Map component to represent the Country and State layer separately. We will add a separate Chart component to represent additional data and use scripting to drill down on both components.

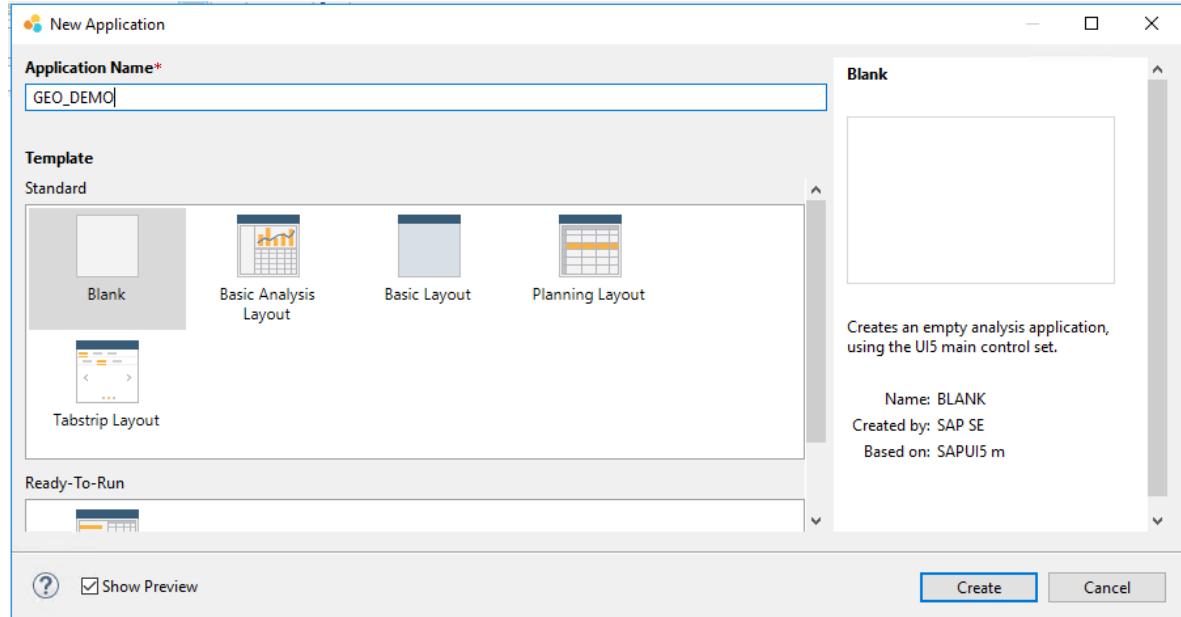
1. Adding a Map component to the Application

Let us first add a new Application to Discovery_Demo in the following manner:

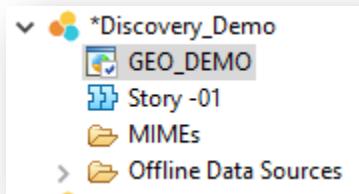
1.1 From the Documents view, right click on Discovery_Demo from under the BI Platform Documents folder and select **Create Application**.



1.2 The New Application dialog box opens up. Set the name as GEO_DEMO. Set the Template as Blank. Click Create.



1.3 We notice the new GEO_DEMO blank application has now been added under our Discovery_Demo files.



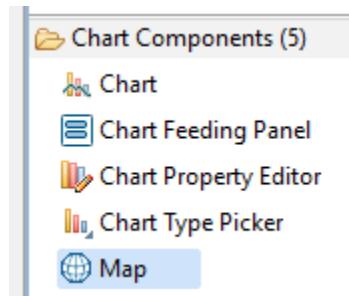
1.4 Select the GEO_DEMO application and from the Outline folder right click on Layout and select Create>>Container Components>>Grid Layout. Rename it to GEO_GRID_1. Click on Maximize selected component . Set No. of Rows, Columns and Column Widths as shown below:

Layout	
Top Margin	0
Left Margin	0
Bottom Margin	0
Right Margin	0
Width	auto
Height	auto
Number of Rows	1
Number of Columns	2
Column Width	2
Column Width	1

- 1.5 Add two Panel components to GEO_GRID_1 under Cell [0, 0] and Cell [0, 1] respectively.
 Rename the Panel components to GEO_PANEL_1 and GEO_PANEL_2 respectively. Set the
 CSS Class of both panels as panel-box. Click on Maximize selected component  . Set All 4
 margins = 3. (Top, Bottom, Left, Right)

Layout	
Top Margin	3
Left Margin	3
Bottom Margin	3
Right Margin	3
Width	auto
Height	auto

- 1.6 From under Components Panel>>Chart Components, add a Map Component to
 GEO_PANEL_1 under the new GEO_DEMO application.



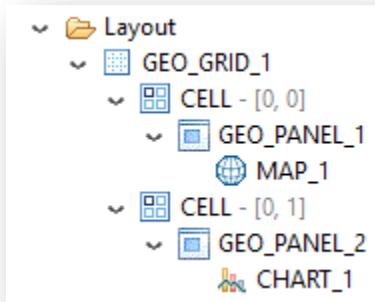
- 1.7 Click on Maximize selected component  . Following properties will be shown:

Layout		
Top Margin	0	+
Left Margin	0	+
Bottom Margin	0	+
Right Margin	0	+
Width	auto	+
Height	auto	+

1.8 Similarly, add a Chart component to GEO_PANEL_2. Click on Maximize selected component . Following properties will be shown:

Layout		
Top Margin	0	+
Left Margin	0	+
Bottom Margin	0	+
Right Margin	0	+
Width	auto	+
Height	auto	+

1.9 The layout of our new GEO_DEMO application now should appear similar to:

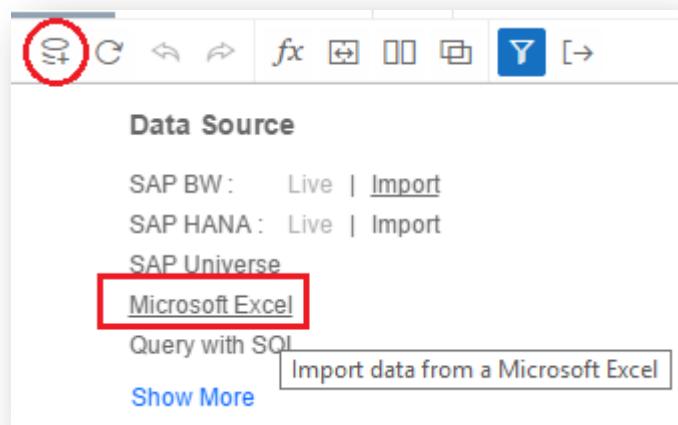


2. [Adding Offline Data Sources through Discovery](#)

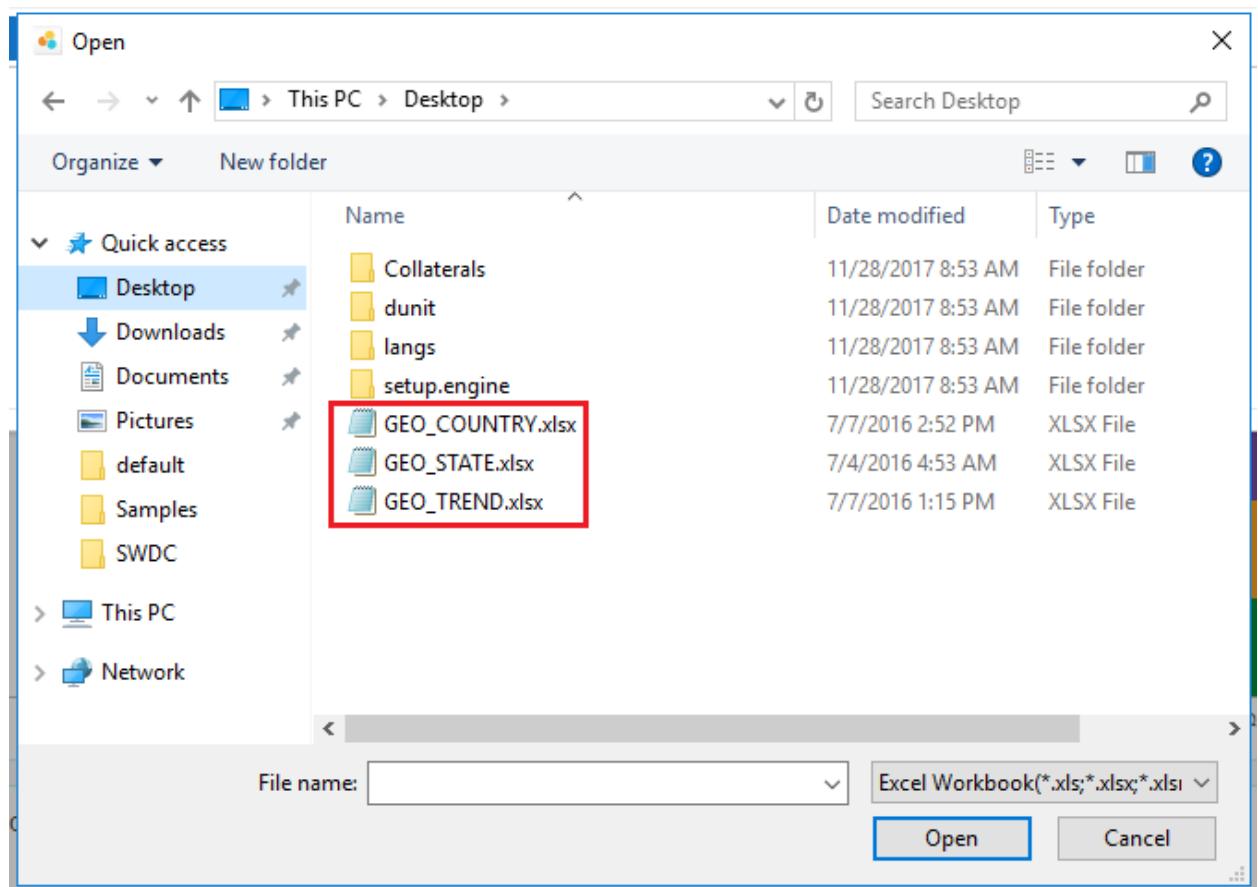
We need geo location data to be able to utilize the full potency of the Map component, let us add three Offline Data Sources to achieve this objective. We do this by adding the data sources in the form of Excel spreadsheets to our application through the Lumira Discovery tool.

We've already discussed how to open and log into the BI Platform through the Lumira Discovery tool in the Lumira Discovery Demo. Once logged in, open up the Discovery_Demo application.

2.1 From the storyboard, select New Dataset button and click Microsoft Excel.



2.2 The three data sets we want to use for our Map component are GEO_COUNTRY, GEO_STATE and GEO_TREND. Select GEO_COUNTRY and click Open. (**Note:** We have conveniently placed them on the Desktop of the local system for easy upload access. We will demonstrate how to upload GEO_COUNTRY in this demo, do the same for all three data sets.)



2.3 Under the Microsoft Excel dialog box, click Visualize.

Microsoft Excel

Dataset Name : GEO_COUNTRY.xlsx/Sheet1

Files : C:\Users\Administrator\Desktop\GEO_COUNTRY.xlsx;

Sheet : Sheet1 Append all Sheets Set first row as column names

Table Header Type : Standard Table (No Transformations)

Select All 10 / 10 columns - 42 rows

<input checked="" type="checkbox"/> Country	<input checked="" type="checkbox"/> State	<input checked="" type="checkbox"/> Product Gro...	<input checked="" type="checkbox"/> City	<input checked="" type="checkbox"/> Customer N...	<input checked="" type="checkbox"/> Month	<input checked="" type="checkbox"/> Latitude	<input checked="" type="checkbox"/> Longitude	<input checked="" type="checkbox"/> Gross Sales	<input checked="" type="checkbox"/> COGS
AE		Office Supplies		Muhammed ...	01/2016			19800000	18500000
AG		Office Supplies		Barry French	02/2016			19000000	17500000
BR		Office Supplies		Barry French	03/2016			18500000	16000000
CA		Technology		Clay Rozenda...	04/2016			19500000	17500000
CH		Office Supplies		Carlos Soltero	05/2016			19500000	17000000
CL		Furniture		Carlos Soltero	06/2016			18900000	16000000
CR		Office Supplies		Carl Jackson	07/2016			18400000	18500000
DE		Office Supplies		Carl Jackson	08/2016			19000000	18500000

Advanced Options

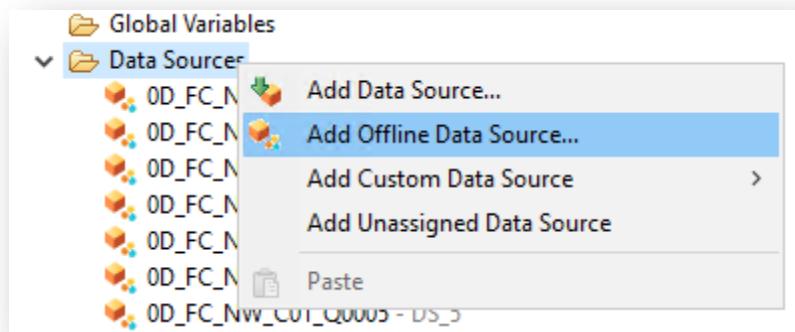
Visualize Cancel

2.4 The dataset has now been successfully added to our application, accessible as an Offline Data Source through Designer. Perform a similar operation on GEO_STATE and GEO_TREND.

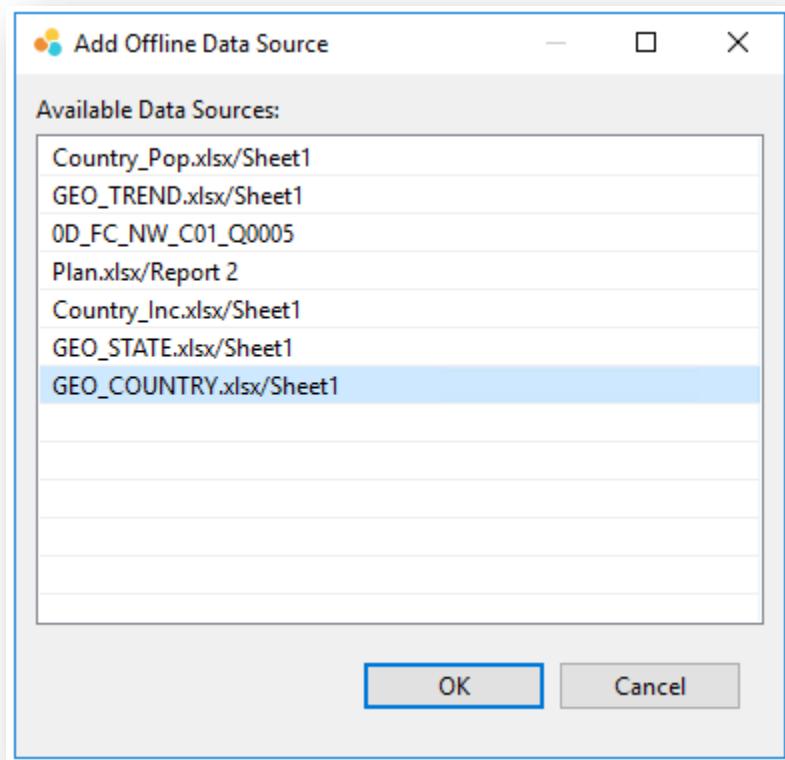
DATASET DESIGN

- > OD_FC_NW_C01_Q0005
- > Plan.xlsx/Report2
- > Country_Pop.xlsx/Sheet1
- > Country_Inc.xlsx/Sheet1
- > GEO_STATE.xlsx/Sheet1
- > GEO_TREND.xlsx/Sheet1
- ▽ GEO_COUNTRY.xlsx/Sheet1

2.5 Save and Exit Lumira Discovery. Reopen the Discovery_Demo application on Lumira Designer. Under the Outline View right click on Data Sources and select Add Offline Data Source.



2.6 From under the Add Offline Data Source dialog box, select GEO_COUNTRY.xlsx and click OK.



2.7 Rename the newly added GEO_COUNTRY data source to COUNTRY_DETAILS. Edit the Initial View of the COUNTRY_DETAILS and set Gross Sales as the Measure under columns and Country dimension under Rows. Click OK.

Edit Initial View of 63 - DS_12

63 - DS_12
Live Preview 17 data cells
 Pause Refresh

Columns

- Measures
- City - DS:63,DIM:id_1E
- COGS - DS:63,DIM:id_
- Country - DS:63,DIM:i
- Customer Name - DS:
- Gross Sales - DS:63,DII
- Latitude - DS:63,DIM:i
- Longitude - DS:63,DIN
- Month - DS:63,DIM:id
- Product Group - DS:6:
- State - DS:63,DIM:id_1

Rows

- Country - DS:63,DIM:i

Background Filter

Global Data Source Settings
Display of Negative Value

-X
?
Undo
Redo
OK + Create Crosstab
OK
Cancel

Country	Gross Sales
AE	19,800,000
AG	19,000,000
BR	18,500,000
CA	19,500,000
CH	19,500,000
CL	18,900,000
CR	18,400,000
DE	19,000,000
FR	19,500,000
GB	20,000,000
IN	25,000,000
JA	21,000,000
MX	19,800,000
QA	19,000,000
SA	18,500,000
US	541,200,000
Overall Result	836,600,000

2.8 Similarly, add the GEO_STATE data source to the application and rename it to STATE_DETAILS. Edit the initial view and set Gross Sales and COGS under Measures in Rows and set State and Product Group under Columns. The data source appears as shown below:

Edit Initial View of 52 - STATE_DETAILS

52 - STATE_DET...

- > Measures
- > COGS - DS:52,DIM:id
- > Gross Sales - DS:52,D
- > Product Group - DS:5
- > State - DS:52,DIM:id_

Columns

- > Measures

Rows

- > State - DS:52,DIM:id_
- > Product Group - DS:5

Background Filter

Global Data Source Settings

Display of Negative Values

-X
?
Undo
Redo
OK + Create Crosstab
OK
Cancel

Live Preview 62 data cells Pause Refresh

State	Product Group	Gross Sales	COGS
AK	Office Supplies	37,900,000	35,500,000
	Technology	19,500,000	17,500,000
AZ	Technology	19,500,000	17,500,000
	Office Supplies	77,800,000	73,500,000
CA	Technology	21,000,000	20,200,000
	Office Supplies	19,000,000	18,500,000
FL	Office Supplies	37,400,000	37,000,000
	Furniture	18,900,000	16,000,000
IL	Office Supplies	18,500,000	16,000,000
	Technology	19,500,000	17,500,000
KS	Office Supplies	19,800,000	18,500,000
	Technology	21,000,000	20,200,000
LA	Office Supplies	25,000,000	20,000,000
	Office Supplies	20,000,000	19,000,000
MA	Technology	19,500,000	17,500,000
	Office Supplies	37,400,000	36,000,000
MD	Office Supplies	20,000,000	19,000,000
	Technology	77,800,000	70,500,000
MI	Office Supplies	19,000,000	18,500,000
	Office Supplies	64,300,000	55,500,000
MN	Technology	42,000,000	40,400,000
NJ	Office Supplies	20,000,000	19,000,000
NM	Office Supplies	19,000,000	18,500,000
NY	Technology	22,000,000	20,000,000

2.9 Finally, add the GEO_TREND data source to the application. Set Gross Sales and Gross Margin under Measures in Rows and set Month under Columns. Rename the data source to DS_TREND.

The screenshot shows the 'Edit Initial View' dialog for a report named '56 - DS_TREND'. The interface is divided into several sections:

- Left Panel:** A tree view of the data source structure under '56 - DS_TREND', including Measures, City, Country, Customer Name, Gross Margin, Gross Sales, Month, Net Sales, Office Supplies, and State.
- Columns Section:** Shows 'Gross Sales' and 'Gross Margin' selected as columns.
- Rows Section:** Shows 'Month' selected as the row dimension.
- Background Filter:** An empty section.
- Live Preview:** A table showing monthly sales data from January 2016 to December 2016, plus an 'Overall Result' row at the bottom.
- Buttons:** Includes standard dialog buttons like '?', Undo, Redo, OK + Create Crosstab, OK, and Cancel.

Month	Gross Sales	Gross Margin
01/2016	39,300,000	3,930,000
02/2016	39,000,000	3,900,000
03/2016	43,500,000	4,350,000
04/2016	60,000,000	6,000,000
05/2016	58,800,000	5,880,000
06/2016	56,800,000	5,680,000
07/2016	56,300,000	5,630,000
08/2016	38,000,000	3,800,000
09/2016	39,000,000	3,900,000
10/2016	40,000,000	4,000,000
11/2016	50,000,000	5,000,000
12/2016	40,000,000	4,000,000
Overall Result	560,700,000	56,070,000

3. Configuring the Map Component

We now add and configure the Layers on the map component utilizing the geo data.

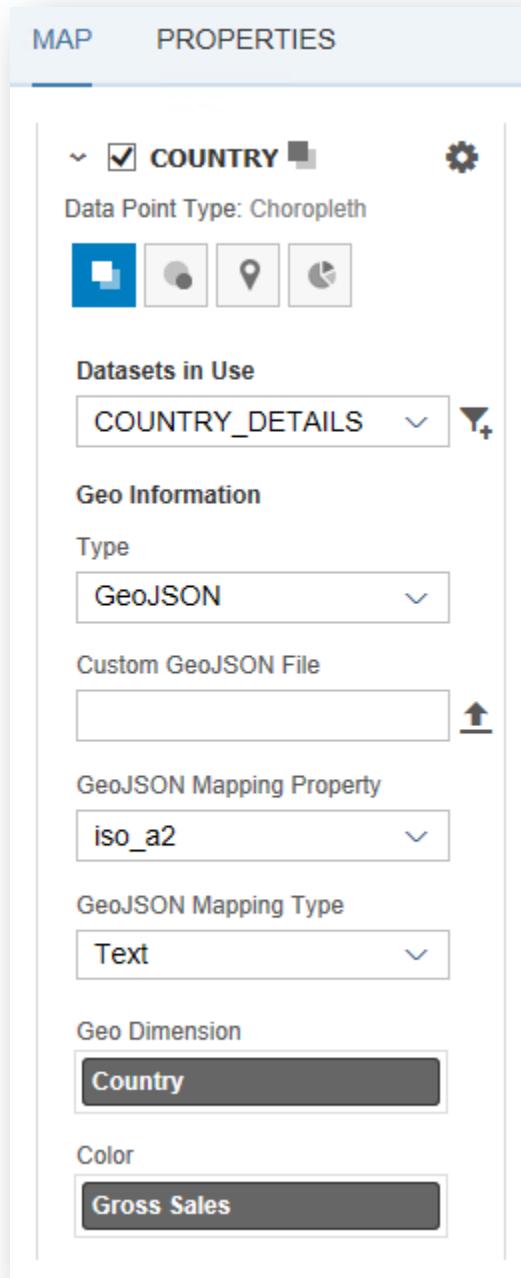
3.1 From under Outline view>>Layout select the Map component and go to Properties View>>Display>>Map Configuration.

The screenshot shows the SAP Fiori Properties dialog for a component named 'MAP_1'. The 'Display' section is expanded, showing the following properties:

Property	Value	Bind
Name	MAP_1	
Type	Map	
Vendor	SAP SE	
Visible	true	+
Display		
CSS Class		+
Map Configuration	(highlighted with a red circle)	
Show Scaling Factors	false	

3.2 The Configure Map dialog box opens up. We notice that by default a single un-configured layer is already added. Let us now configure the layer using our data.

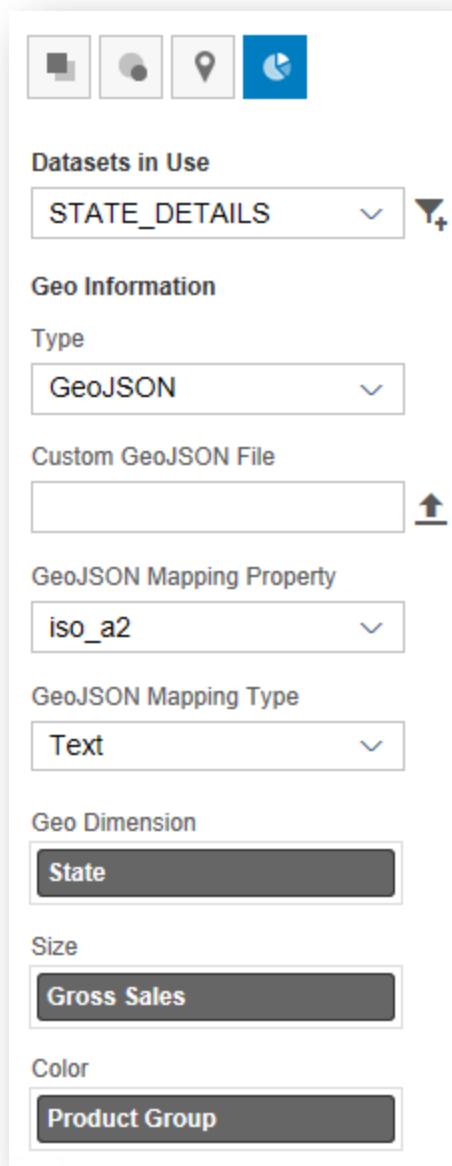
3.3 Rename the Layer to COUNTRY. Set the Type to Choropleth. Set the Datasets in Use option to COUNTRY_DETAILS. Set the Geo Dimension to Country and set the Color property to Gross Sales.



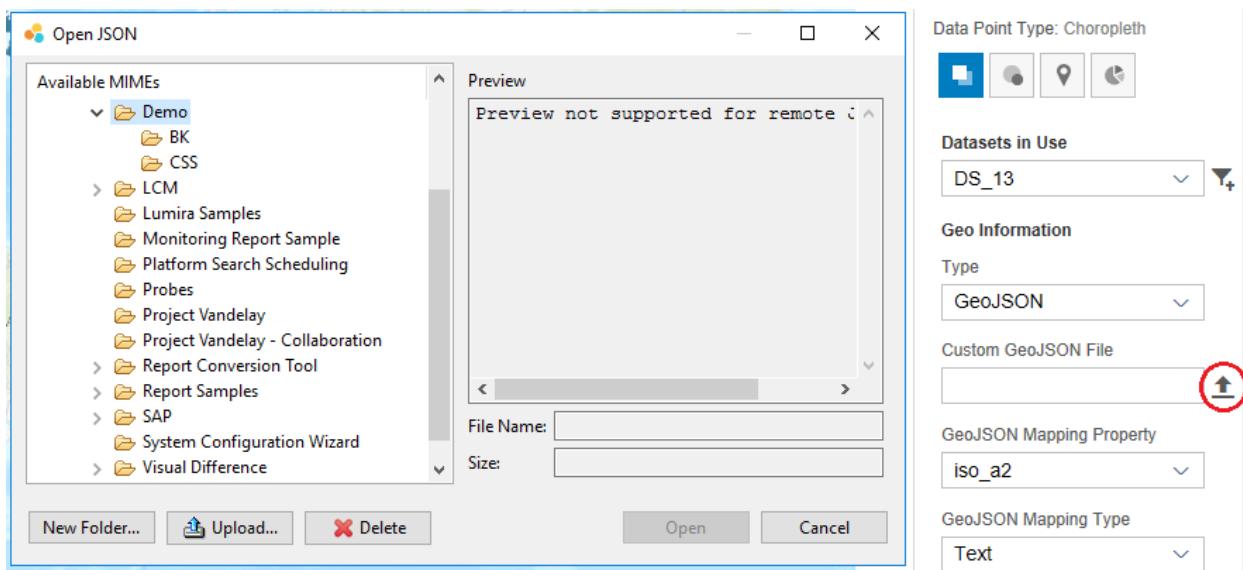
3.4 Next, let us add our second layer by selecting the Add Layer button from under the Configure Map dialog.



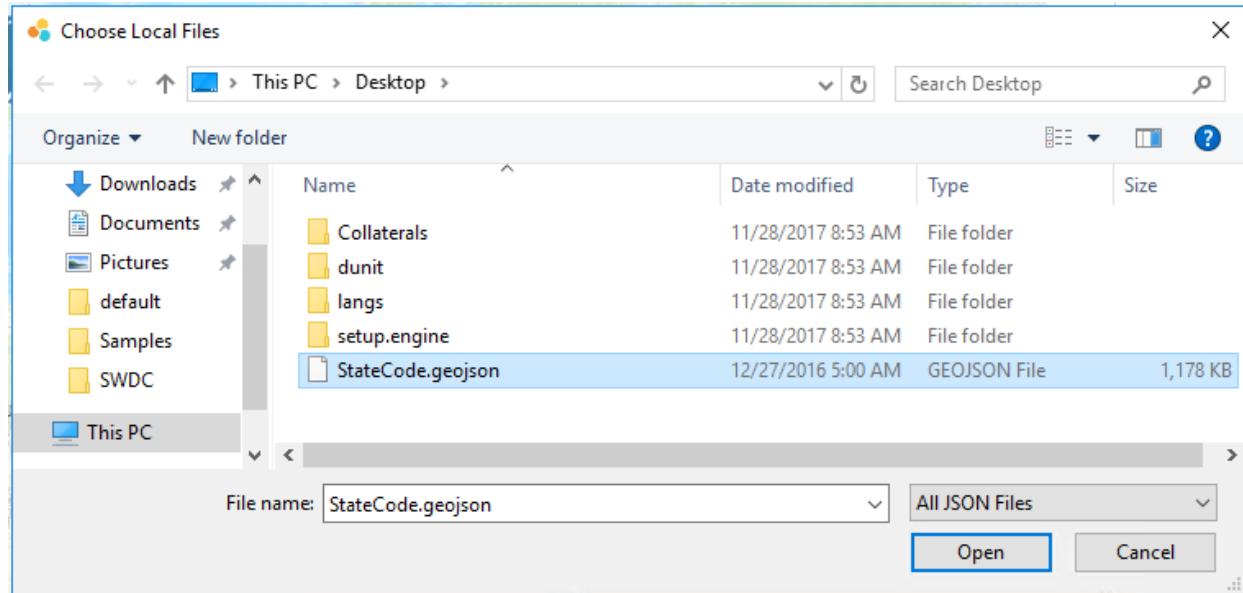
3.5 Rename the Layer to STATE. Set the Type as Pie. Set the Datasets in Use property to STATE_DETAILS. Set the Geo Dimension to State. Set the Size to Gross Sales and the Color to Product Group.



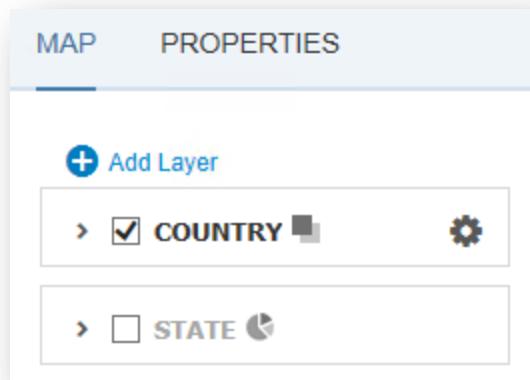
3.6 Now, to geocode the layer according to US States we add the custom GeoJSON file to the layer. Select the Custom GeoJSON option and from the Open JSON dialog box select the Public Folders>>Demo folder and select Upload... to upload the file from your local system.



3.7 Select the appropriate geojson file from the Local System. Click Open. After the file is successfully uploaded to the BI Platform select it from under the Demo folder and click Open to add it to the layer.



3.8 The STATE layer has now been fully configured. Drag and drop the STATE layer below the COUNTRY layer and uncheck the Visible checkbox to ensure that the STATE layer remains hidden initially. In the coming sections we will explain how to Show the STATE layer using scripting. Click OK to finish configuring the Map component.



4. [Configuring the Chart Component](#)

We now configure the Chart component utilizing the geo data.

4.1 From under Outline view>>Layout select the Chart component under GEO_PANEL_2 and go to Properties>>Data Binding>>Data Source and set it to DS_Trend. Next under Display>>Chart Configuration select the Configure Chart button.

▼ Data Binding	
Data Source	[DS_TREND]
Data Selection	[]
▼ Display	
CSS Class	[]
Chart Configuration	Column Chart [DATA] +
Allow Data Source Modification	[false]
Conditional Formatting Visible	[false]
Conditional Formatting Settings	[]

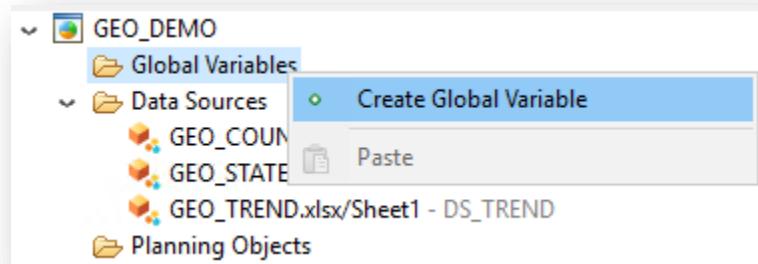
4.2 Under the Configure Chart dialog box, set it to a Line Chart. Click OK.



5. Adding Drilldown using Scripting

We now configure drill down on the Map and Chart components utilizing scripting.

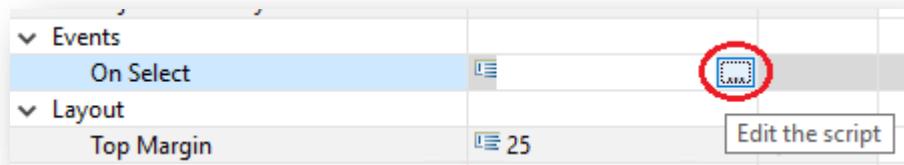
5.1 Let us first define a Global Variable. In the Outline View of the GEO_DEMO Application right click on the Global Variables folder and select Create Global Variable.



5.2 In the Properties View of the Global Variable set the Name to G_SELLAYER. Set the Type to String and Default Value as COUNTRY as shown.

Properties		Additional Properties
Property	Value	
General		
Name	G_SELLAYER	
Description		
Type	String	
Array	false	
Default Value	COUNTRY	

5.3 Next, go to the Properties View of the Map component. Under Events>>On Select click on the Edit the script button.



5.4 Under the Script Editor dialog box that opens up, enter the following script:

```

if (MAP_1.getSelectedLayer() == "STATE") {
    DS_TREND.setFilter("DS:56,DIM:id_145",
    MAP_1.getSelectedMember("DS:52,DIM:id_124"));
    // Filters the Data Sources of the Info Charts based on the STATE(DS:52,DIM:id_124) value
    selected in the Geo Map
}
else if (MAP_1.getSelectedLayer() == "COUNTRY" &&
MAP_1.getSelectedMember(("DS:63,DIM:id_174"))
//Filter only when Country is US
{
    MAP_1.setLayerVisible("STATE", true);
    MAP_1.setLayerVisible("COUNTRY", false);
    // Hides the COUNTRY(default) layer and shows the STATE(initally hidden) layer
    G_SELLAYER = "STATE";
    // Global Variable value set to STATE
}

```

Script Editor

Script for event "On Select" of component "MAP_1":

```
1 if (MAP_1.getSelectedLayer() == "STATE") {  
2  
3 DS_TREND.setFilter("DS:56,DIM:id_145", MAP_1.getSelectedMember("DS:52,DIM:id_124"));  
4 // Filters the Data Sources of the Info Charts based on the STATE value selected in the Geo Map  
5  
6 TEXT_1.setText("Filter Level: " + MAP_1.getSelectedMember("DS:52,DIM:id_124").text);  
7 // Sets the Text box to display the selected State in its output  
8 G_SELLAYER = "CITY";  
9 }  
10 else if (MAP_1.getSelectedLayer() == "COUNTRY" && MAP_1.getSelectedMember("DS:63,DIM:id_174").text == "US" )  
11 {  
12 //Filter only when Country is US  
13 // Filters the Data Sources of the Geo Map and Info Charts based on the COUNTRY value selected in the Geo Map  
14 MAP_1.setLayerVisible("STATE", true);  
15 MAP_1.setLayerVisible("COUNTRY", false);  
16 // Hides the COUNTRY(default) layer and shows the STATE(initially hidden) layer  
17 // MAP_1.centerMap("STATE");  
18  
19 |  
20 G_SELLAYER = "STATE";  
21  
22 }  
23 }
```

5.5 Next, add an Icon component to GEO_PANEL_1. Set its margins as shown:

Layout	
Top Margin	25
Left Margin	75
Bottom Margin	auto
Right Margin	auto
Width	42
Height	42

5.6 Under Properties View>>Display set the Background Color of the Icon to Black. Set the Icon URI to sap-icon://arrow-left and set the Tooltip property to Go Back.

Display	
CSS Class	
Custom Font	
Color	<input type="color"/> RGB: (255,255,255), HTML: white
Background Color	<input checked="" type="color"/> RGB: (0,0,0), HTML: black
Background Shape	Rectangle
Size Factor	0.8
Icon URI	sap-icon://arrow-left
Tooltip	Go Back

5.7 Under Events>>On Click add the following script:

```

if (G_SELLAYER == "COUNTRY") {
    MAP_1.centerMap("COUNTRY");
}
else if (G_SELLAYER == "STATE") {
    // This section deals with when we are displaying State data and want to look at all
    Countries
    COUNTRY_DETAILS.clearFilter("DS:63,DIM:id_174");
    // Clears the Filters set on the Country Data
    MAP_1.setLayerVisible("COUNTRY", true);
    MAP_1.setLayerVisible("STATE", false);
    // Hides the STATE layer and displays the COUNTRY layer
    G_SELLAYER = "COUNTRY";
    // Now that we are back on the COUNTRY layer our Global Script Object is set back to
    COUNTRY
}

```



Script for event "On Click" of component "ICON_1":

```

1 if (G_SELLAYER == "COUNTRY") {
2 MAP_1.centerMap("COUNTRY");
3 }
4 else if (G_SELLAYER == "STATE") {
5 // This section deals with when we are displaying State data and want to look at all Countries
6 COUNTRY_DETAILS.clearFilter("DS:63,DIM:id_174");
7 // Clears the Filters set on the Country Data
8 MAP_1.setLayerVisible("COUNTRY", true);
9 MAP_1.setLayerVisible("STATE", false);
10 // Hides the STATE layer and displays the COUNTRY layer
11
12 G_SELLAYER = "COUNTRY";
13 // Now that we are back on the COUNTRY layer our Global Script Object is set back to COUNTRY
14 }
15

```

6. [Adding a Text Box to Display Filter Level](#)

We now add a Text component to always display the level of the drilldown.

6.1 From the Basic Components add a Text Box component to the Application under GEO_PANEL_1 in GEO_GRID_1. Set its margins as:

Layout	
Top Margin	0
Left Margin	25
Bottom Margin	auto
Right Margin	auto
Width	200
Height	30

6.2 Set its Text property to 'Filter Level: World'. Set its CSS Class to panel-box and its CSS Style to:

font-weight: bold;

Display	
CSS Class	panel-box
Text	Filter Level: World
CSS Style	font-weight: bold;
Tooltip	

6.3 To configure the Text Box to show us the Filter Level we go back to the MAP_1 Geo Map component and under the Properties go to Events>>On Select and Edit the Script to add the following two lines:

```
TEXT_1.setText("Filter Level: " + MAP_1.getSelectedMember("DS:52,DIM:id_124").text);
```

```
TEXT_1.setText("Filter Level: " + MAP_1.getSelectedMember("DS:63,DIM:id_174").text);
```

Script Editor

Script for event "On Select" of component "MAP_1":

```

1 if (MAP_1.getSelectedLayer() == "STATE") {
2 TEXT_1.setText("Filter Level: " + MAP_1.getSelectedMember("DS:52,DIM:id_124").text);
3 DS_TREND.setFilter("DS:56,DIM:id_145", MAP_1.getSelectedMember("DS:52,DIM:id_124"));
4 // Filters the Data Sources of the Info Charts based on the STATE value selected in the Geo Map
5
6 TEXT_1.setText("Filter Level: " + MAP_1.getSelectedMember("DS:52,DIM:id_124").text);
7 // Sets the Text box to display the selected State in its output
8 //G_SELLAYER = "CITY";
9 }
10 else if (MAP_1.getSelectedLayer() == "COUNTRY" && MAP_1.getSelectedMember("DS:63,DIM:id_174").text == "US" )
11 {
12 //Filter only when Country is US
13 // Filters the Data Sources of the Geo Map and Info Charts based on the COUNTRY value selected in the Geo Map
14 MAP_1.setLayerVisible("STATE", true);
15 MAP_1.setLayerVisible("COUNTRY", false);
16 // Hides the COUNTRY(default) layer and shows the STATE(initially hidden) layer
17 // MAP_1.centerMap("STATE");
18
19 TEXT_1.setText("Filter Level: " + MAP_1.getSelectedMember("DS:63,DIM:id_174").text);
20 // Sets the Text box to display the selected Country in its output
21 G_SELLAYER = "STATE";
22 }
23

```

6.4 Similarly, select the ICON_1 component and add the following lines under the Script Editor:

```
TEXT_1.setText("Filter Level : World");
```

```
TEXT_1.setText("Filter Level : " + DS_TREND.getFilterExt("DS:56,DIM:id_143"));
```

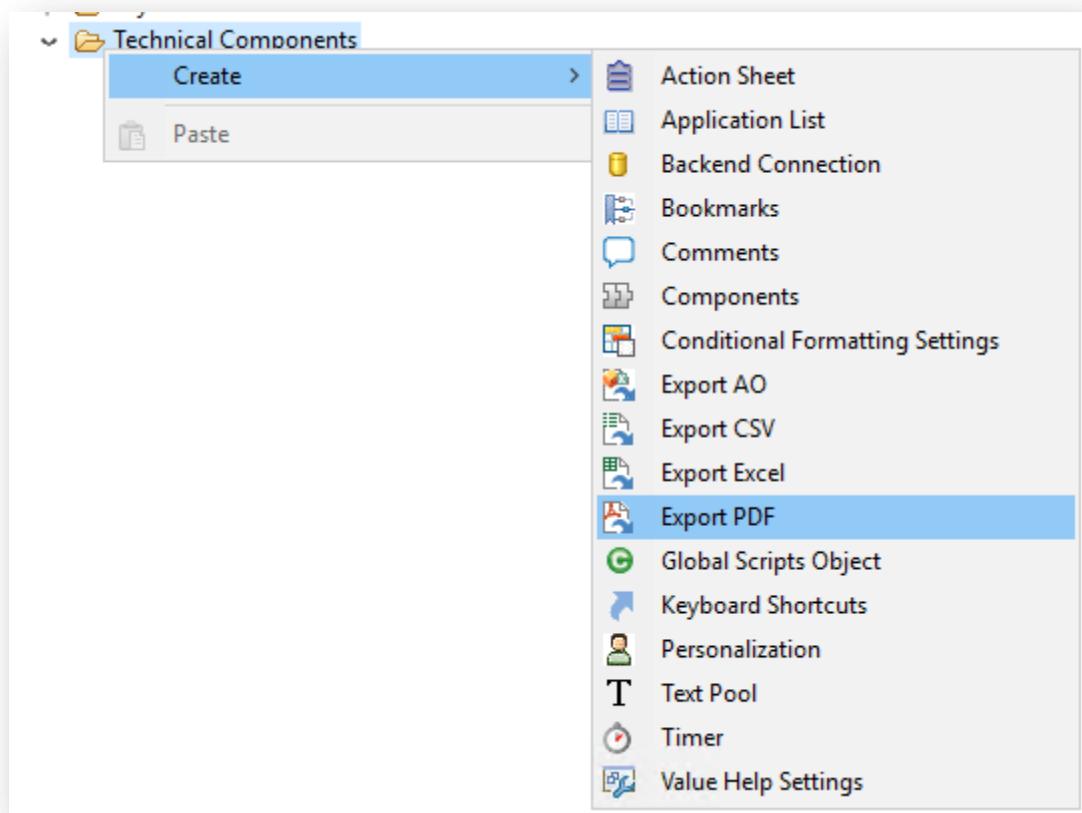
 Script Editor
Script for event "On Click" of component "ICON_1":

```
1 if (G_SELLAYER == "COUNTRY") {  
2 MAP_1.centerMap("COUNTRY");  
3 }  
4 else if (G_SELLAYER == "STATE") {  
5 // This section deals with when we are displaying State data and want to look at all Countries  
6 COUNTRY_DETAILS.clearFilter("DS:63,DIM:id_174");  
7 // Clears the Filters set on the Country Data  
8 MAP_1.setLayerVisible("COUNTRY", true);  
9 MAP_1.setLayerVisible("STATE", false);  
10 // Hides the STATE layer and displays the COUNTRY layer  
11  
12 TEXT_1.setText("Filter Level : World");  
13 // Sets the text to what's set in between the parenthesis  
14 G_SELLAYER = "COUNTRY";  
15 // Now that we are back on the COUNTRY layer our Global Script Object is set back to COUNTRY  
16 }  
17
```

7. Adding Buttons to Export to PDF/CSV

We now two Button components to enable exporting Map data to PDF or CSV form.

7.1 To be able to successfully export components of our GEO_DEMO application in the form of either PDF or CSV files, we first need to add two Technical Components. From the Outline View right click on the Technical Components folder and select Export PDF.

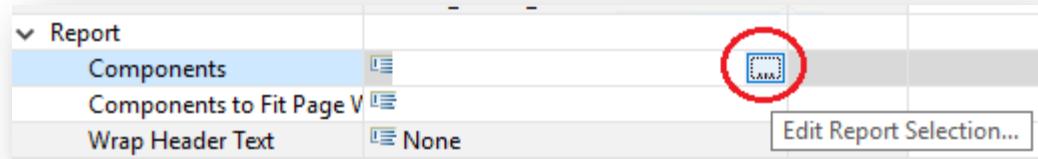


7.2 In a similar manner select Technical Components>>Export CSV to enable exporting to CSV files.

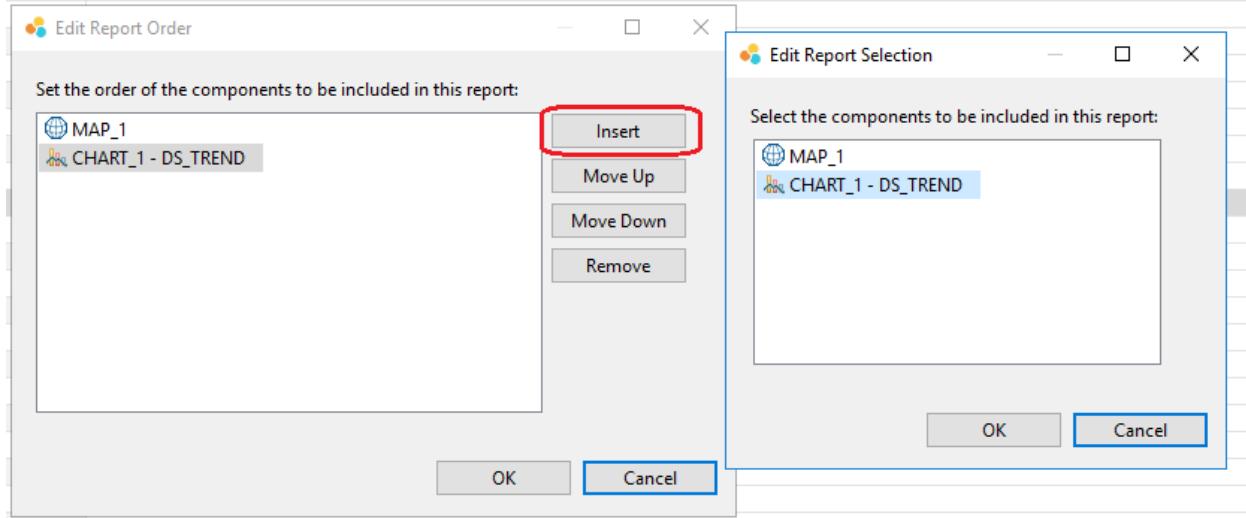
7.3 Select the Export PDF component and from the Properties View go to the Settings category. Set the Page Number Location to Footer. Set the File Name to GEO_DEMO_PDF.

Settings	
Orientation	Landscape
Paper Size	A4
Page Number Location	Footer
Date Location	None
File Name	GEO_DEMO_PDF

7.4 Under the Reports category go to Components and click the Edit Report Selection... button.



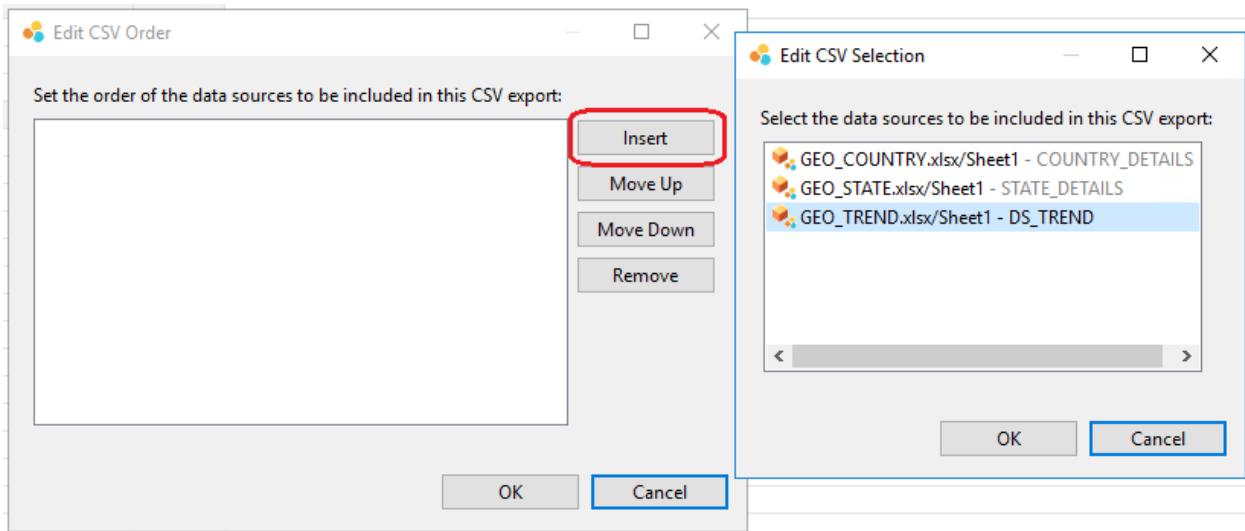
7.5 Add both the MAP_1 and CHART_1 components using the Insert button. Click OK.



7.6 Finally, under the Footer category set Footer Visible to true.

Footer	
Footer Visible	<input checked="" type="checkbox"/> true
Text	<input type="text"/>
Image	<input type="file"/>

7.7 Similarly, select the Export CSV component and from under Properties View go to Settings>>Data Sources. Add the DS_TREND data source by using the Insert option. Click OK.



7.8 Set the File Name as GEO_DEMO_CSV. Set the Separator option as Comma.

Settings	
Data Sources	DS_TREND
File Name	GEO_DEMO_CSV
Separator	Comma

7.9 From the Basic Components add a Button component to GEO_PANEL_2 under in GEO_GRID

_1. Click on Maximize selected component . Set Width = 120 and Height = 30. Following Layout should be shown.

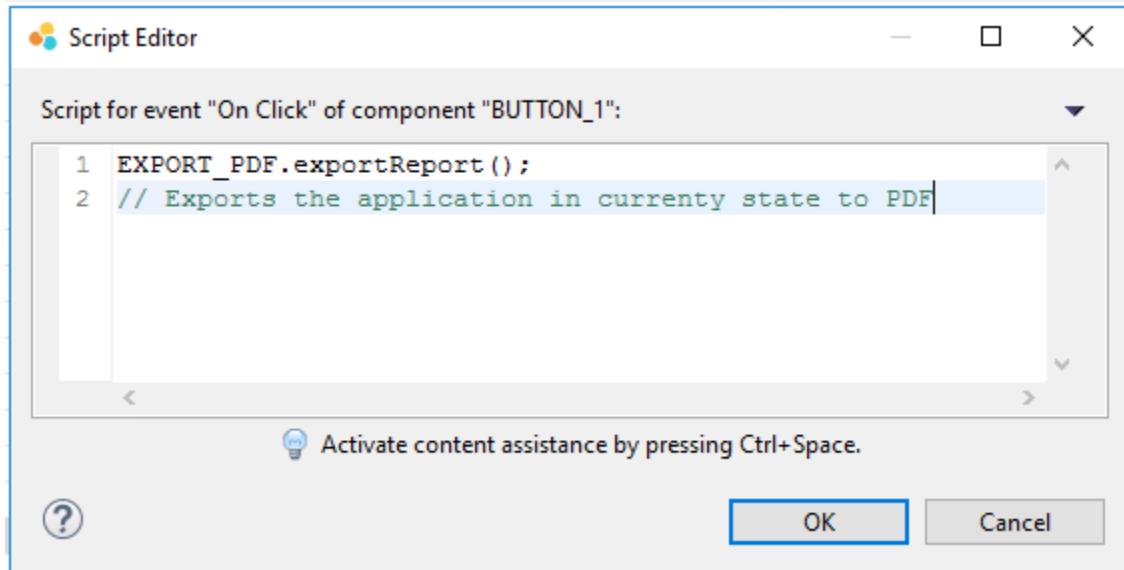
Layout	
Top Margin	0
Left Margin	0
Bottom Margin	auto
Right Margin	auto
Width	120
Height	30

7.10 Under Display set its CSS Class to panel-box. Set its Text property to Export PDF.

Display	
CSS Class	panel-box
Text	Export PDF
Icon	
Tooltip	
Button Type	Default

7.11 Under Events>>On Select set the following script:

```
EXPORT_PDF.exportReport();
```



7.12 Similarly, add another Button component to GEO_PANEL_2 under in GEO_GRID_1. Click on

Maximize selected component . Set Width = 120 and Height = 30. Following Layout should be shown.

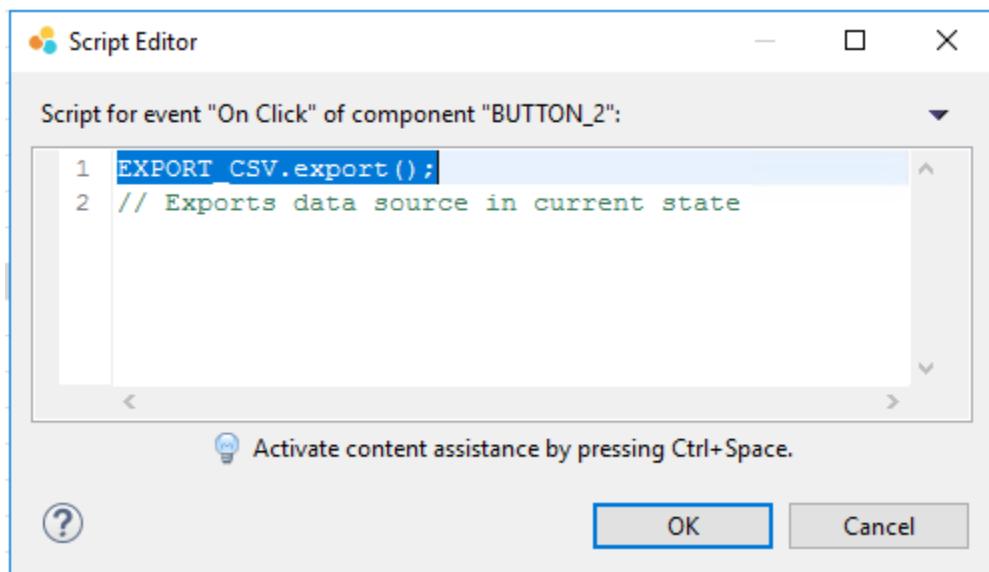
Layout	
Top Margin	0
Left Margin	auto
Bottom Margin	auto
Right Margin	0
Width	120
Height	30

7.13 Under Display set its CSS Class to panel-box. Set its Text property to Export CSV.

Display	
CSS Class	panel-box
Text	Export CSV
Icon	
Tooltip	
Button Type	Default

7.14 Under Events>>On Select set the following script. Click OK.

```
EXPORT_CSV.export();
```

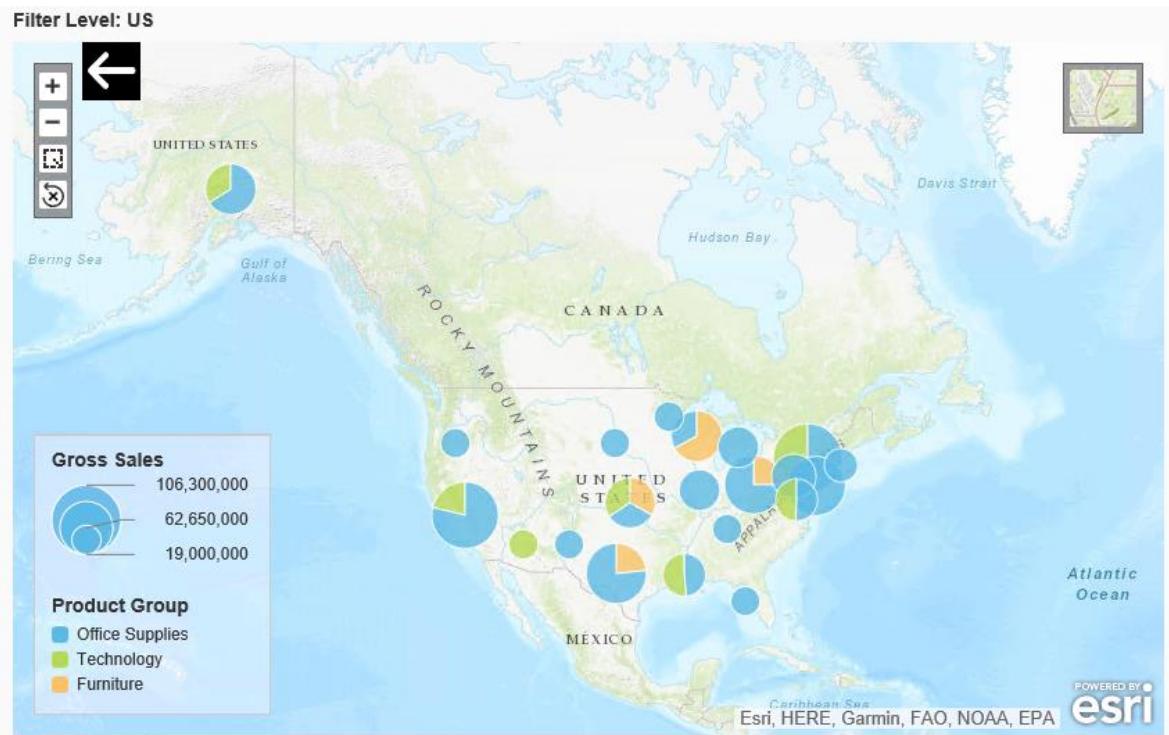


8. Testing the Map

8.1 Save the GEO_DEMO application. Synchronize it with the BI Platform and Execute it on the BI Platform. The application initially appears as shown below:



8.2 Click on the Country USA and we notice the Geo Map giving us a breakdown of GROSS SALES by PRODUCT GROUP (different Product Groups are represented in slices of the Pie chart) across states.

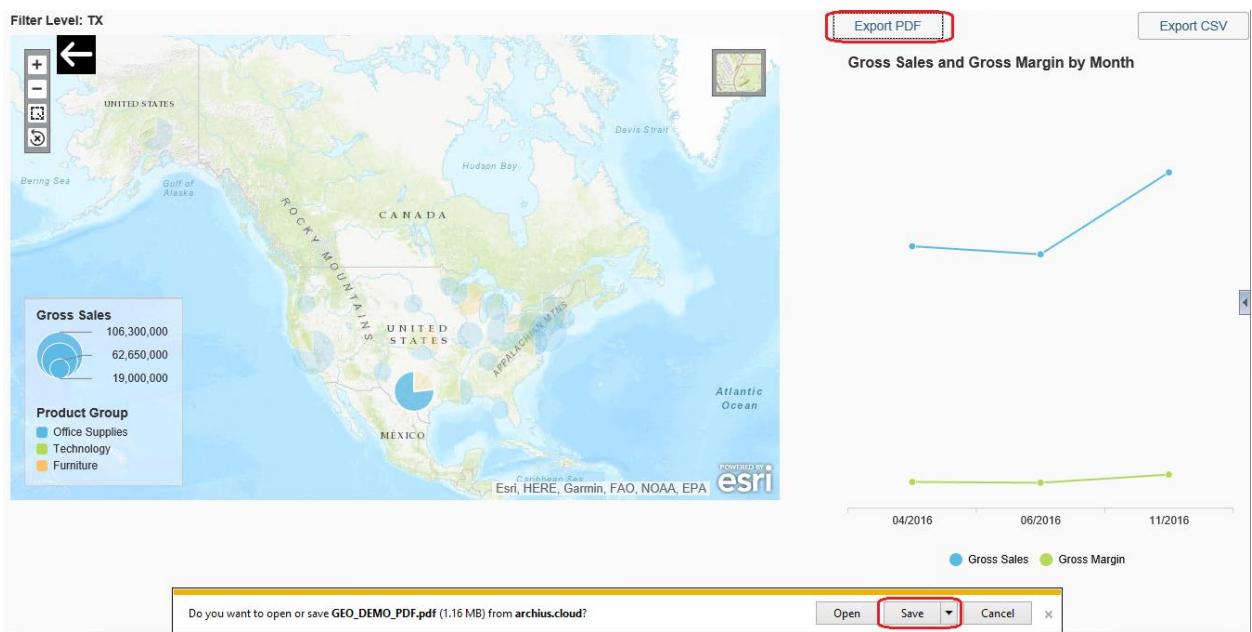


(Note: The change in the Filter Level)

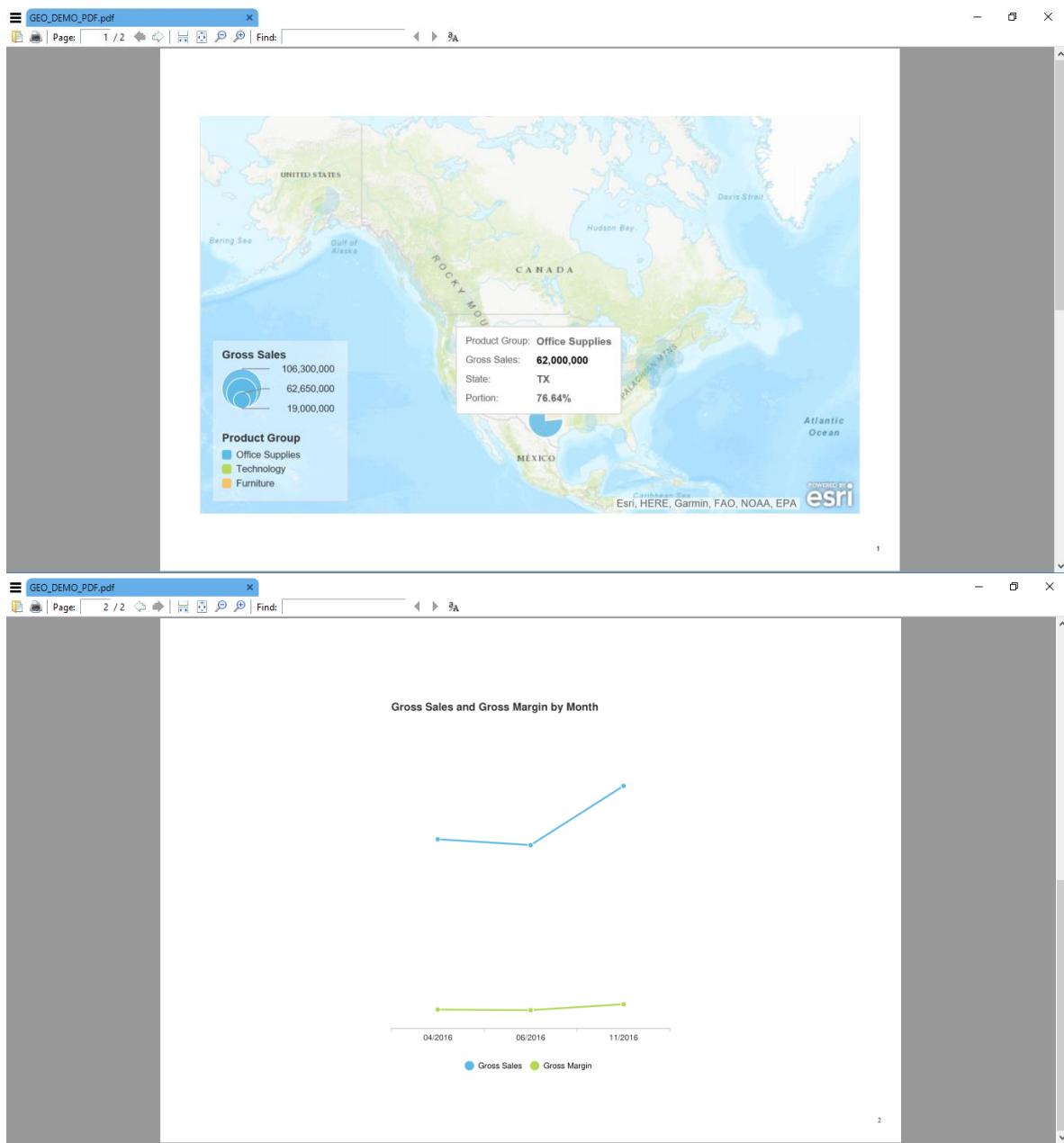
8.3 Clicking on the State of Texas enables drill down on the Info Chart. It now displays the Sales Trend details for Texas. The Filter Layer changes to Texas.



8.4 Select the Export PDF button. When prompted by your internet browser, select Save the PDF.



8.5 Opening up the GEO_DEMO_PDF created shows our Geo Map drilled down to the Texas state.



8.6 Next, select the Export CSV button. When prompted by your internet browser, select Save the CSV. Opening the DEMO_GEO_CSV file shows the filter applied on the DS_TREND data source chosen.

GEO_DEMO_CSV.csv - WordPad

View

Courier New 11 A A

B I U abe x A A

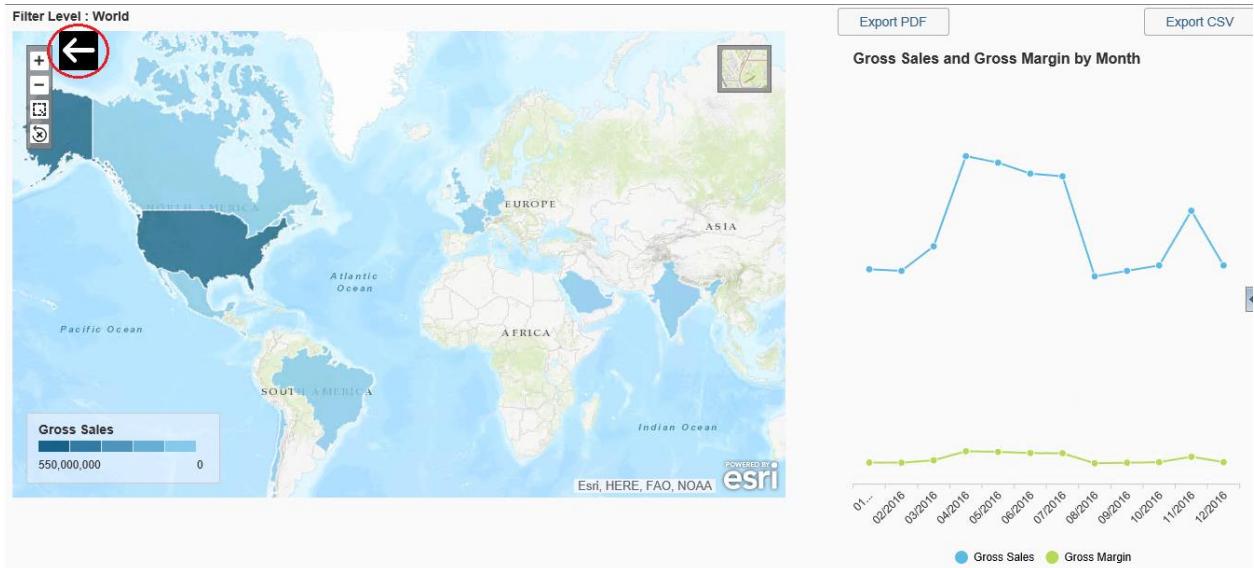
Font Paragraph Insert Editing

Picture Paint Date and time Insert object

Find Replace Select all

```
"Month","Gross Sales","Gross Margin"
"04/2016","19,500,000","1,950,000"
"06/2016","18,900,000","1,890,000"
"11/2016","25,000,000","2,500,000"
"Overall Result","63,400,000","6,340,000"
```

8.7 Click on the Go Back Icon (highlighted in Red) to return to the State layer. Similarly, clicking the button once more takes you back to the Country layer.



Exercise 7 - SCORECARD

Objective of the Exercise

- Introduction to the Exercise
- Adding and Configuring the Scorecard Component

1 Introduction to the Exercise

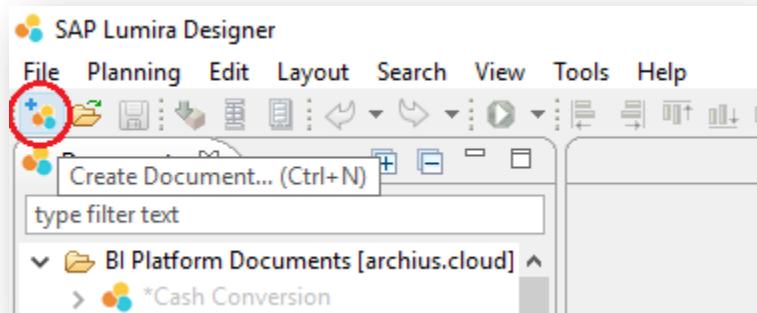
The Scorecard Component is a new feature that was introduced with the release of **Design Studio 1.6**. Lumira Designer continues to use this component that allows the user to define highly customizable tables. Users will have influence on all visualization properties for the content and more content types (small charts) as well. Binding any property of the content to the data view allows you to react on some values/master data and make any property value conditional.

The main idea behind the **Scorecard** component is the mapping between the data source and the visualization. The visualization is based on a table, so what you can define are columns of the table. A column has cell content with type, header and some general properties. A property has a value which can be maintained directly or bound to the data set using the known data selection model: this is the **binding concept**. Every property is independent of each other, but you can make them a kind of dependent if you bind them to the same data cell value.

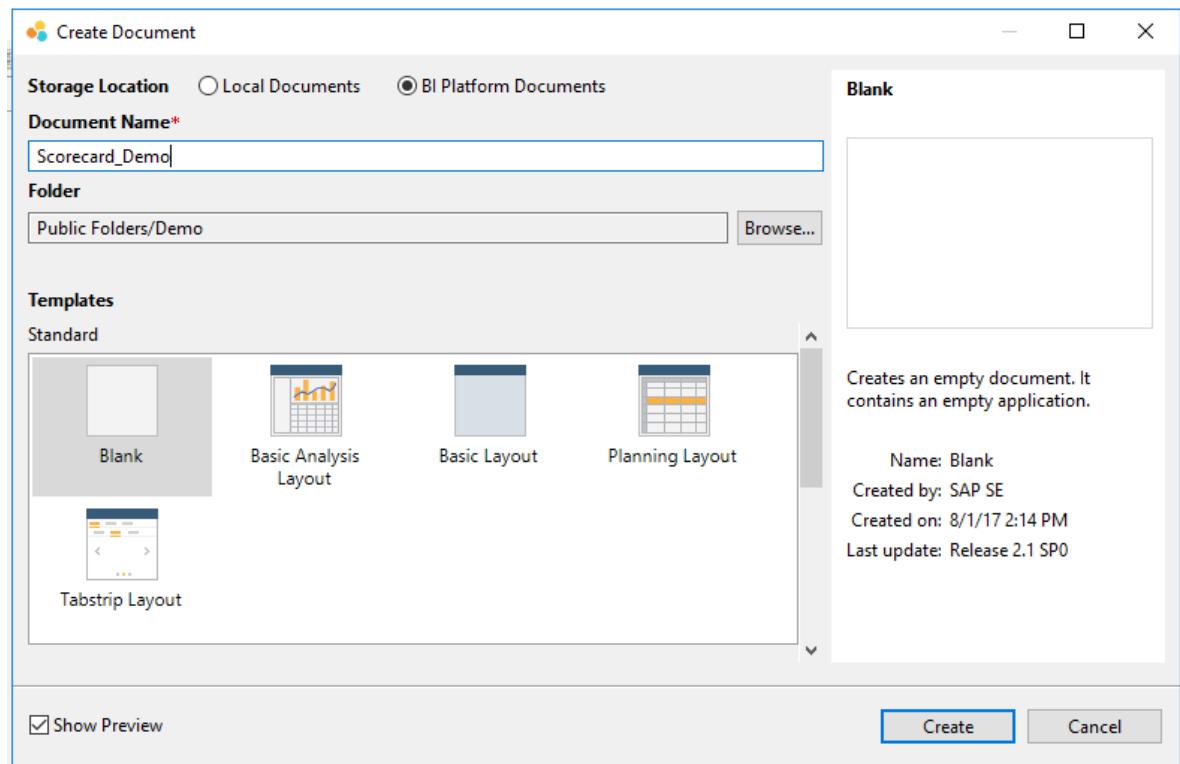
In this exercise we build a Scorecard component to display relationships between the State, Customer, Gross Sales and Plan. The steps below show us how:

2 Getting Started with the Application – The Backbone

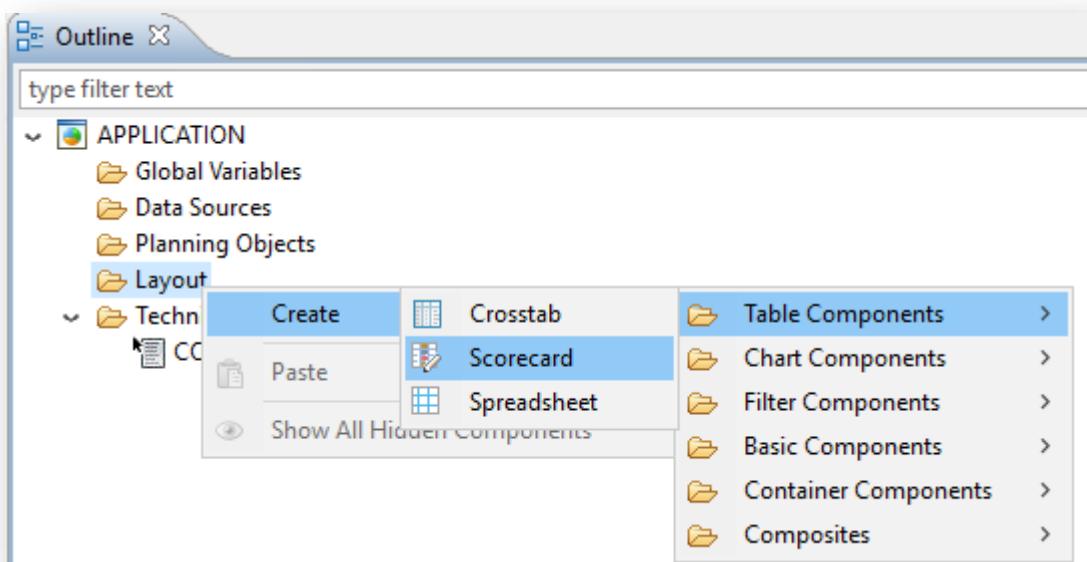
2.1 Let us create a new document by selecting the Create Document... button.



2.2 The Create Document dialog box opens up. Create it on the BI Platform under Public Folders>>Demo. Set the Name as Scorecard_Demo. Set the Template to Blank. Click Create.



2.3 Add a Scorecard component to the Layout by right-clicking on the Layout folder and selecting Create>>Table Components>>Scorecard.

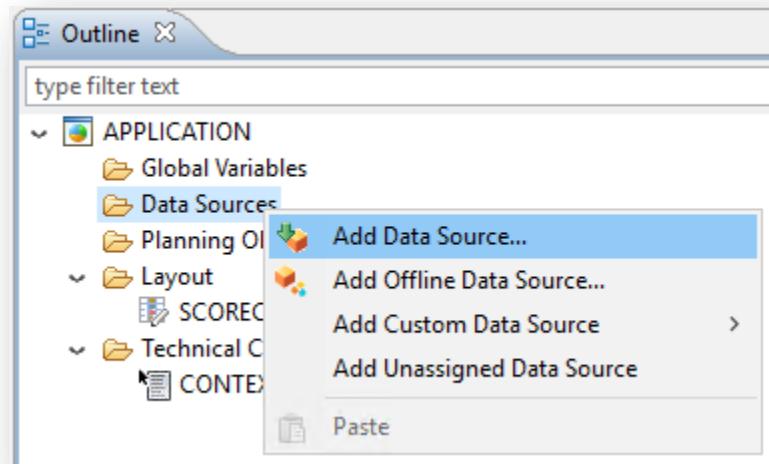


2.4 Click on Maximize Selected component .

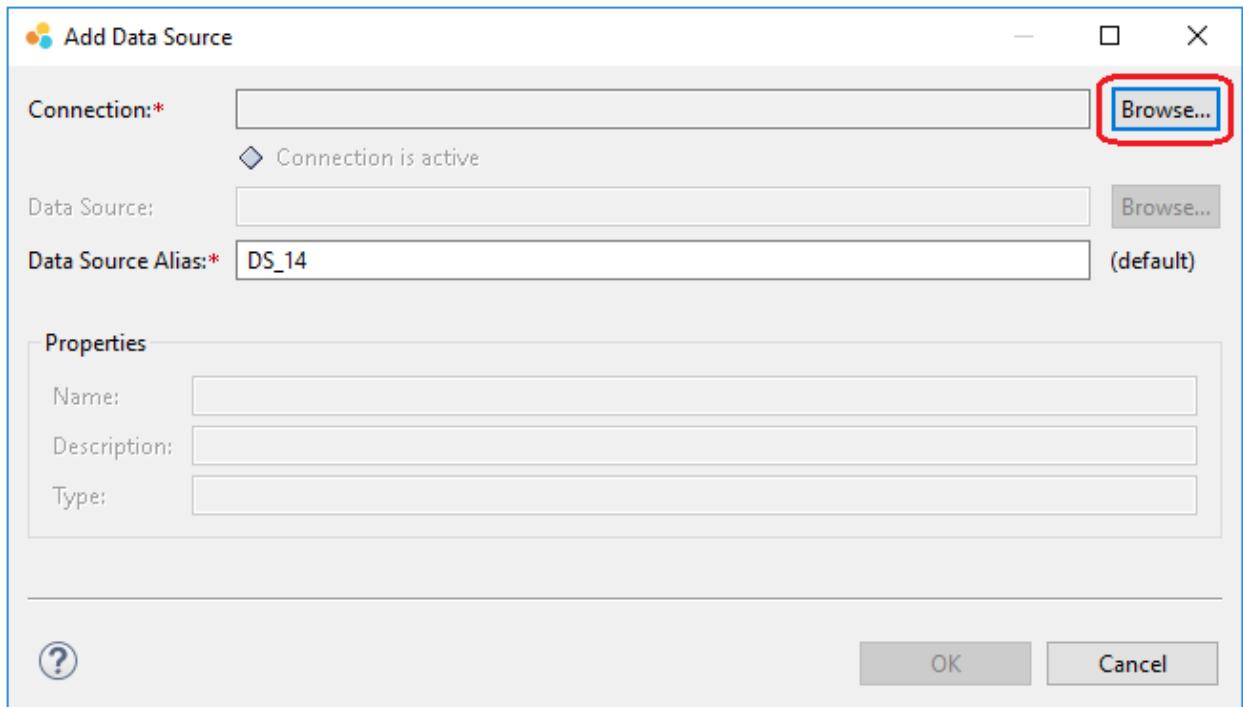
3 [Adding and Configuring the Data Source](#)

We now add the data source to be used by our Scorecard component from a HANA system. In this instance we choose the AV_SALES_ACTUALS data source. Here is how:

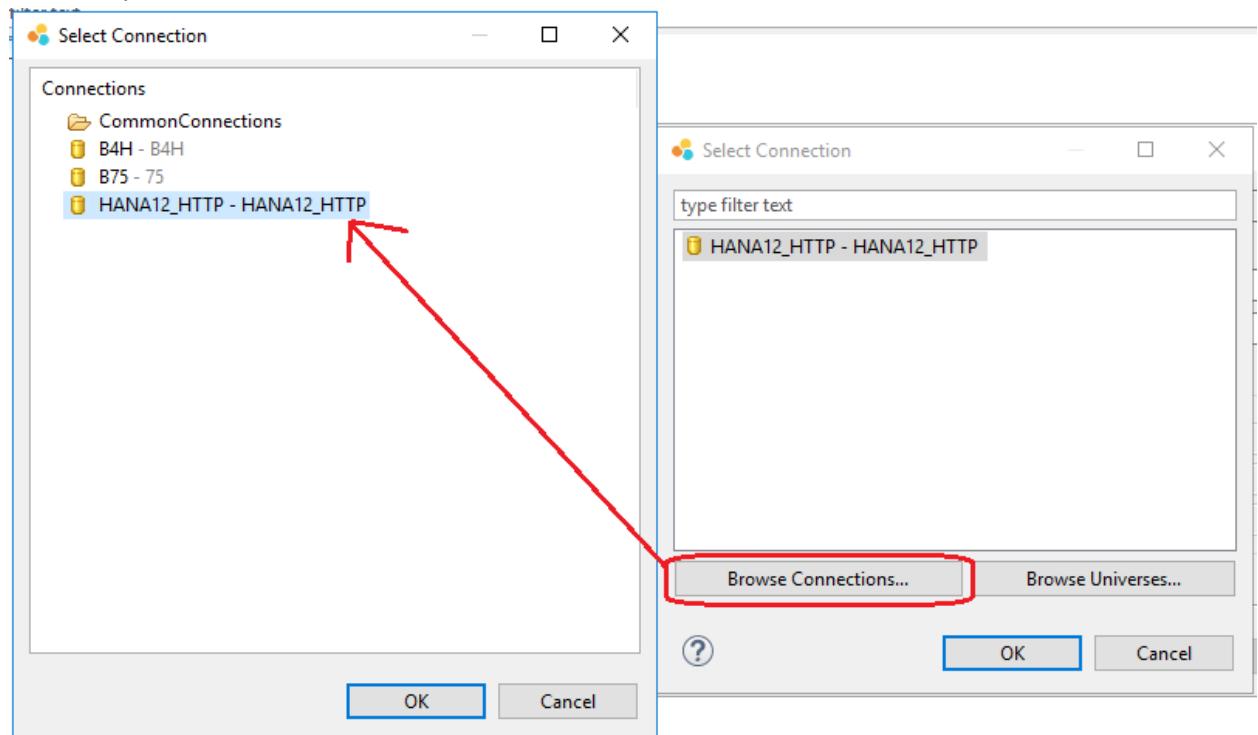
3.1 From under the Outline View, right click on the Data Sources folder and select Add Data Source.



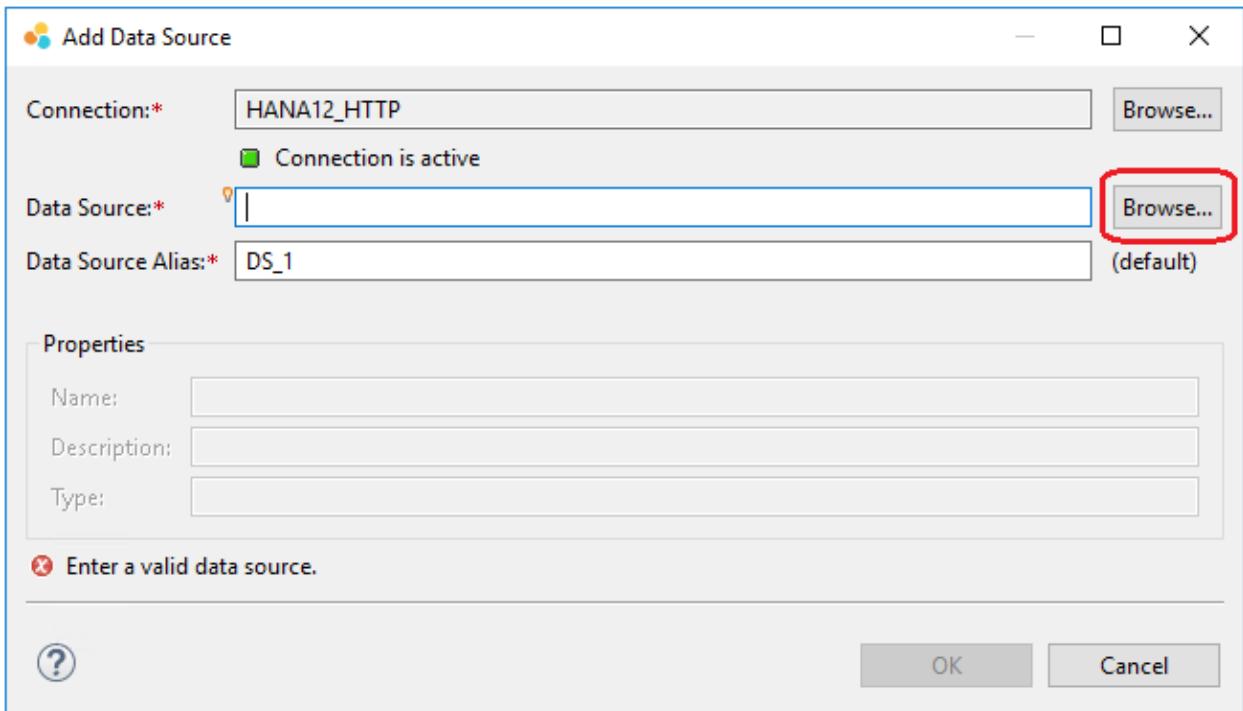
3.2 The Add Data Source dialog box opens up. First we must select the System in which our data source resides. To do this select the Browse... button beside the Connections option.



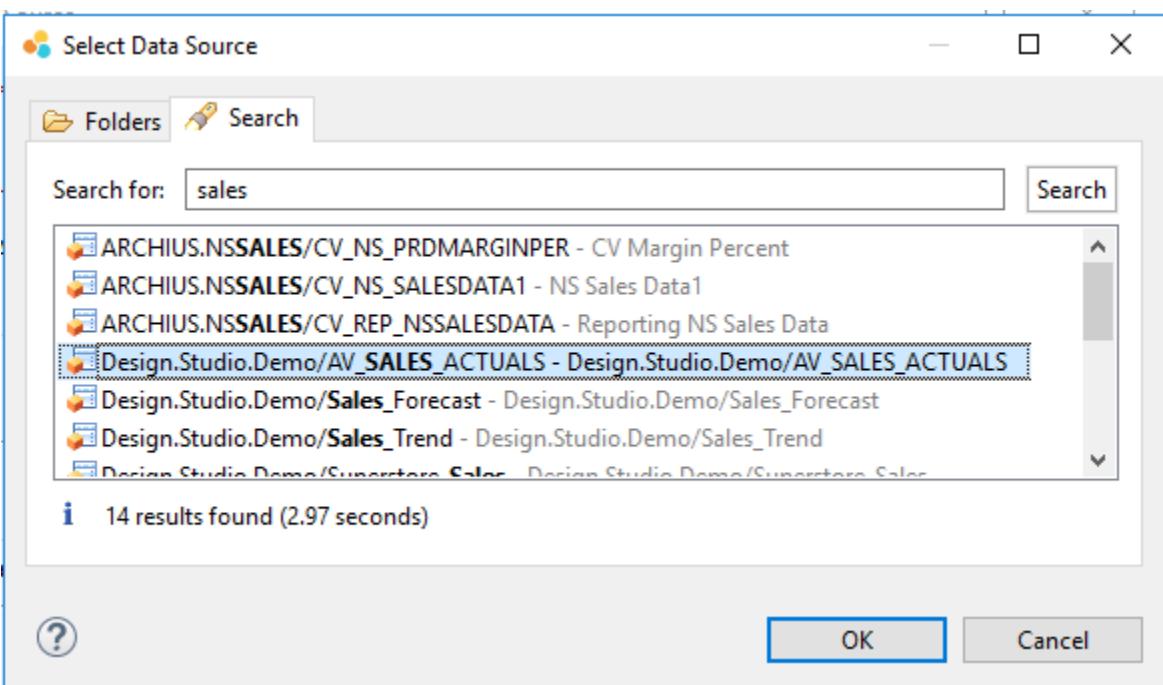
3.3 Under the Select Connections dialog box click on the Browse Connections option. Select the B75 system from the list. Click OK.



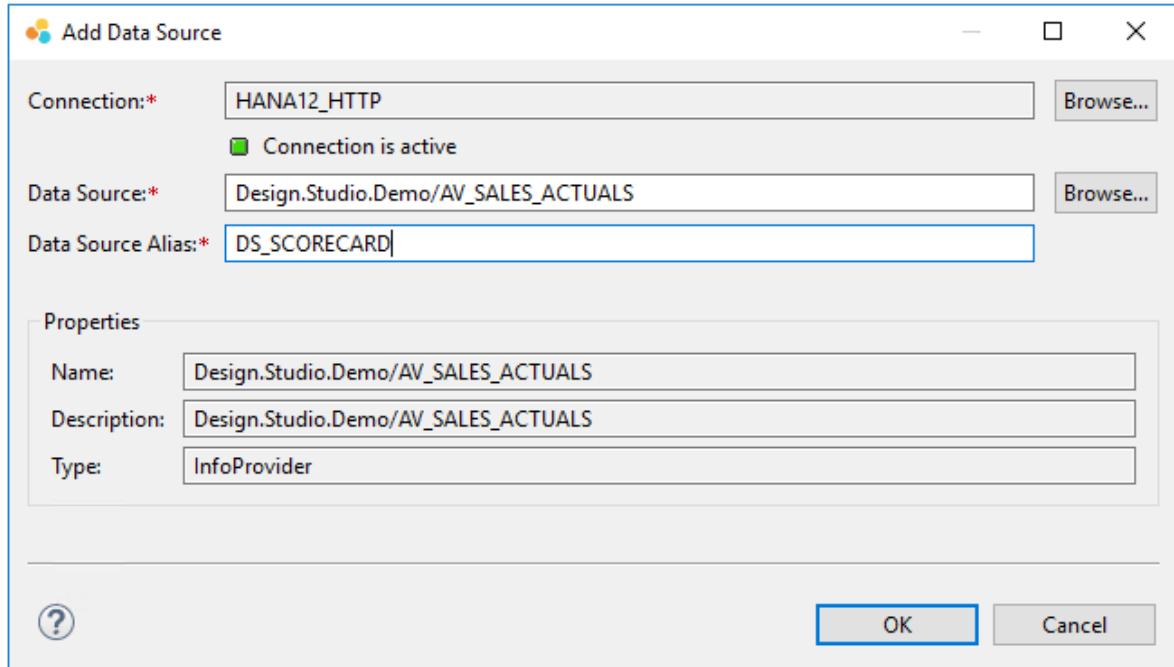
3.4 Next, click Browse... beside the Data Source option.



3.5 Under the Select Data Source folder go to the Search tab. Under there search for the term 'sales' and select Search. Select the resultant AV_SALES_ACTUALS data source and click OK to add it to our application.



3.6 Finally, set the Data Source Alias as DS_SCORECARD. Click OK.



3.7 Right click on the DS_SCORECARD data source from under the Data Sources folder and from the content menu select Edit the Initial View option. Under Columns>>Measures set the measures as Gross Sales, Plan, COGS, COGS Plan and Gross Margin(%). Also set the State and Customer Name dimensions under Rows. Click OK.

State	Customer Name	Gross Sales	Plan	COGS	COGS Plan	Gross Margin(%)
California	Carlos Soltero	18,900,000	19,000,000	17,000,000	16,500,000	0.53
	Dorothy Badders	20,000,000	19,500,000	17,500,000	17,000,000	0.89
	Muhammed MacIntyre	19,800,000	20,200,000	18,200,000	18,000,000	0.50
	Result	58,700,000	58,700,000	52,700,000	51,500,000	1.92
	Barry French	18,500,000	19,500,000	17,500,000	17,000,000	0.80
New Jersey	Carlos Soltero	19,500,000	19,000,000	17,000,000	16,500,000	0.78
	Result	38,000,000	38,500,000	34,500,000	33,500,000	1.58
	Carl Jackson	18,400,000	20,000,000	18,000,000	17,000,000	0.45
New York	Clay Rozendal	19,500,000	20,500,000	18,500,000	18,000,000	0.69
	Result	37,900,000	40,500,000	36,500,000	35,000,000	1.14
	Monica Federle	19,500,000	20,300,000	18,300,000	18,000,000	0.77
North Carolina	Neola Schneider	21,000,000	23,000,000	21,000,000	20,500,000	0.54
	Result	40,500,000	43,300,000	39,300,000	38,500,000	1.31
	Barry French	19,000,000	18,500,000	16,500,000	16,000,000	0.75
Texas	Carl Jackson	19,000,000	20,500,000	18,500,000	17,000,000	0.65
	Neola Schneider	25,000,000	27,000,000	25,000,000	24,500,000	0.90
	Result	63,000,000	66,000,000	60,000,000	57,500,000	2.30
Overall Result	238,100,000	247,000,000	223,000,000	216,000,000	8.25	

(Note: Do NOT Hide Totals in the data set as they will be used in the Scorecard component)

3.8 Assign DS_SCORECARD to SCORECARD_1 as shown below:

Properties		Additional Properties	
Property	Value		
General			
Name	SCORECARD_1		
Type	Scorecard		
Vendor	SAP SE		
Visible	true		
Data Binding			
Data Source	DS_SCORECARD		

4 Configuring the Scorecard Component

4.1 To initially configure the Scorecard, under Additional Properties go to **Dimension for Row Scope**. Click on the Select Dimension dropdown and select ‘**up to State**’. Following that, click on the **Generate Initial Scorecard** button.



Note: The row scope is defining how many rows you will have in the scorecard and which data is accessible to you in a single row. We need to select a dimension from the row dimension in our selected Data Source and the scorecard will get as many rows and as many combinations of the selected dimensions are available.

For example, if we had instead chosen ‘up to Customer Name’ we’d have rows for each customer value as well, increasing the number of rows of our scorecard substantially.

4.2 When the Row Scope is defined, we can view the Initial Scorecard that is generated with 4 Columns. We are free to configure this Scorecard in the ways explained in this exercise. Here is the initial Scorecard:

State	Gross Sales	Gross Sales for Customer Name	Gross Sales vs Plan
California	58,700,000.00	≤18,900,000.00 # 3 19,800,000.00⇒	58,700,000.00 0.00 58,700,000.00
New Jersey	38,000,000.00	≤18,500,000.00 # 2 19,500,000.00⇒	38,000,000.00 500,000.00 38,500,000.00
New York	37,900,000.00	≤18,400,000.00 # 2 19,500,000.00⇒	37,900,000.00 2,600,000.00 40,500,000.00
North Carolina	40,500,000.00	≤19,500,000.00 # 2 21,000,000.00⇒	40,500,000.00 2,800,000.00 43,300,000.00
Texas	63,000,000.00	≤19,000,000.00 # 3 25,000,000.00⇒	63,000,000.00 3,000,000.00 66,000,000.00
Overall Result	238,100,000.00		238,100,000.00 8,900,000.00 247,000,000.00

4.3 Under COLUMN_1 and COLUMN_2 in **Cell>>Cell Layout** set the **Cell Content Alignment to Center**.

COLUMN_1		
COLUMN_2		
COLUMN_3		
COLUMN_4		

Selected Column Definition [Previous](#) [Next](#)

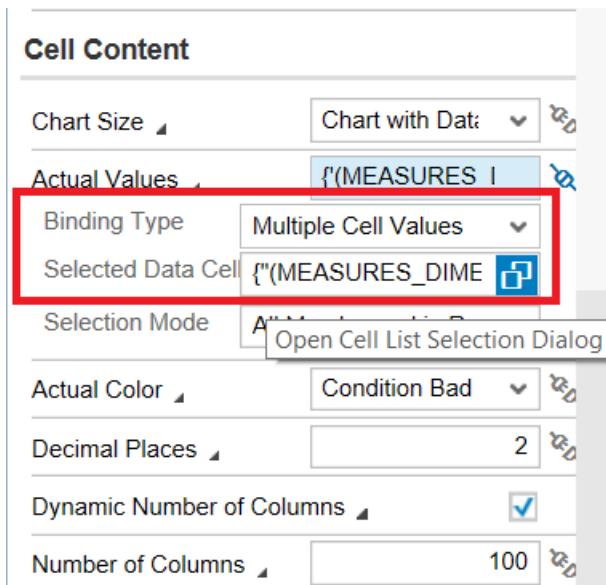
Column ID	COLUMN_1		
Cell Template	Content Display for Values, Texts		
Content	Cell	Header	Column

Cell Layout

Cell CSS Class	<input type="text"/>	
Cell Background	white	
Cell Content Alignment	Center	
Cell Padding	Left Center Right	
Cell Border Top		

Note: This sets the values shown in Column 1 to be aligned to the center of the column.

4.4 Select COLUMN_3 and toggle the **Bind Property** under **Cell Content>>Actual Values**. Set the **Binding Type** to *Multiple Cell Values*. Under Selected Data Cells click the Open Cell List Selection Dialog button and from under the Select Data dialog box that opens up, select the *Gross Sales* column to select all the values of *Gross Sales* per each Customer.



Click OK. (**Note:** The selected Column is highlighted):

Select Data

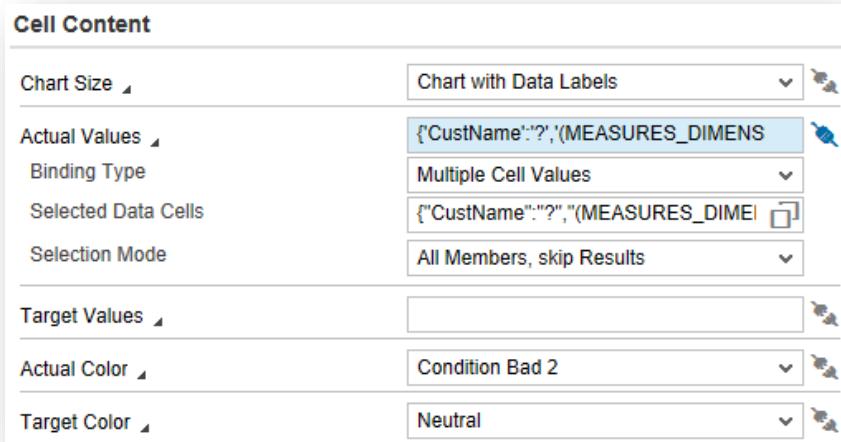
Select a row or column of data from the table below or enter a selection expression into the input field
You can either select a data cell or toggle members to change the effective selection.

Data Source: DS_SCORECARD {"MEASURES_DIMENSION": "DS:69,MEAS:id_242"}

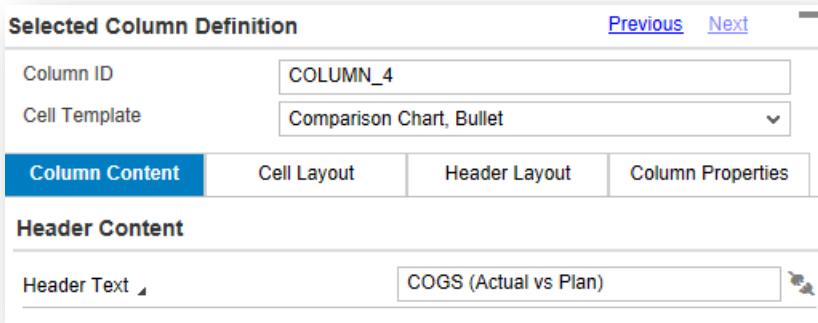
	Gross Sales	Plan	COGS	COGS Plan	Gross Margin(%)	
California	Carlos Soltero	18,900,000	19,000,000	17,000,000	16,500,000	
	Dorothy Badders	20,000,000	19,500,000	17,500,000	17,000,000	
	Muhammed MacIntyre	19,800,000	20,200,000	18,200,000	18,000,000	
	Result	58,700,000	58,700,000	52,700,000	51,500,000	
New Jersey	Barry French	18,500,000	19,500,000	17,500,000	17,000,000	
	Carlos Soltero	19,500,000	19,000,000	17,000,000	16,500,000	
	Result	38,000,000	38,500,000	34,500,000	33,500,000	
	New York	Carl Jackson	18,400,000	20,000,000	18,000,000	17,000,000
North Carolina	Clay Rozendal	19,500,000	20,500,000	18,500,000	18,000,000	
	Result	37,900,000	40,500,000	36,500,000	35,000,000	
	Texas	Monica Federle	19,500,000	20,300,000	18,300,000	18,000,000
		Neola Schneider	21,000,000	23,000,000	21,000,000	20,500,000
Result		40,500,000	43,300,000	39,300,000	38,500,000	
Overall Result		238,100,000	247,000,000	223,000,000	216,000,000	

OK Cancel

4.5 Set the Actual Color to Condition Bad 2.



4.6 Set the Cell Template of Column 4 as **Comparison Chart, Bullet**. Set its Header Text to *COGS (Actual vs Plan)*.



4.7 Toggle the **Bind Property of Actual Value**. Set the **Binding Type** to *Single Data Cell Value*. In the Selected Data Cell field click on the **Open Cell Selection Dialog** box and select the Total Result of the COGS measure(Highlighted in Blue). This is represented by the string:
`{"(MEASURES_DIMENSION)": "DS:69,MEAS:id_247", "DS:69,DIM:id_215": "(RESULT_MEMBER)", "DS:69,DIM:id_219": "(RESULT_MEMBER)"}`

Content Cell Header Column

Header Content

Header Text: Gross Sales fo

Cell Content

Chart Size: Chart with Data

Actual Value: {"(MEASURES_I

Binding Type: Single Data Cell Value

Selected Data Cell: {"(MEASURES_DIME

Content Type: Value as Number

Selection is Relative in Row:

Select Data

Select a single data cell from the table below or enter a selection expression into the input field
You can either select a data cell or toggle members to change the effective selection.

Data Source: DS_SCORECARD {"(MEASURES_DIMENSION": "DS:69,MEAS:id_247", "DS:69,DIM:id_215": "(RESU)

		Gross Sales	Plan	COGS	COGS Plan	Gross Margin(%)
California	Carlos Soltero	18,900,000	19,000,000	17,000,000	16,500,000	0.53
	Dorothy Badders	20,000,000	19,500,000	17,500,000	17,000,000	0.89
	Muhammed MacIntyre	19,800,000	20,200,000	18,200,000	18,000,000	0.50
	Result	58,700,000	58,700,000	52,700,000	51,500,000	1.92
New Jersey	Barry French	18,500,000	19,500,000	17,500,000	17,000,000	0.80
	Carlos Soltero	19,500,000	19,000,000	17,000,000	16,500,000	0.78
	Result	38,000,000	38,500,000	34,500,000	33,500,000	1.58
New York	Carl Jackson	18,400,000	20,000,000	18,000,000	17,000,000	0.45
	Clay Rozendal	19,500,000	20,500,000	18,500,000	18,000,000	0.69
	Result	37,900,000	40,500,000	36,500,000	35,000,000	1.14
North Carolina	Monica Federle	19,500,000	20,300,000	18,300,000	18,000,000	0.77
	Neola Schneider	21,000,000	23,000,000	21,000,000	20,500,000	0.54
	Result	40,500,000	43,300,000	39,300,000	38,500,000	1.31
Texas	Barry French	19,000,000	18,500,000	16,500,000	16,000,000	0.75
	Carl Jackson	19,000,000	20,500,000	18,500,000	17,000,000	0.65
	Neola Schneider	25,000,000	27,000,000	25,000,000	24,500,000	0.90
	Result	63,000,000	66,000,000	60,000,000	57,500,000	2.30
Overall Result		238,100,000	247,000,000	223,000,000	216,000,000	8.25

OK Cancel

4.8 Toggle the Bind Property of Benchmark Value. Set the Binding Type to Single Data Cell Value. In the Selected Data Cell field click on the Open Cell Selection Dialog Box and select the Total Result of the *COGS_Plan* measure. This is represented by the string:

```
{"(MEASURES_DIMENSION)": "DS:69,MEAS:id_248", "DS:69,DIM:id_215": "(RESULT_MEMBER)", "DS:69, DIM:id_219": "(RESULT_MEMBER)"}
```

Select Data

Select a single data cell from the table below or enter a selection expression into the input field
You can either select a data cell or toggle members to change the effective selection.

Data Source: DS_SCORECARD {"(MEASURES_DIMENSION)": "DS:69,MEAS:id_248", "DS:69,DIM:id_215": "(RESULT_MEMBER)", "DS:69, DIM:id_219": "(RESULT_MEMBER)"} Show Clear

		Gross Sales	Plan	COGS	COGS Plan	Gross Margin(%)
California	Carlos Soltero	18,900,000	19,000,000	17,000,000	16,500,000	0.53
	Dorothy Badders	20,000,000	19,500,000	17,500,000	17,000,000	0.89
	Muhammed MacIntyre	19,800,000	20,200,000	18,200,000	18,000,000	0.50
	Result	58,700,000	58,700,000	52,700,000	51,500,000	1.92
New Jersey	Barry French	18,500,000	19,500,000	17,500,000	17,000,000	0.80
	Carlos Soltero	19,500,000	19,000,000	17,000,000	16,500,000	0.78
	Result	38,000,000	38,500,000	34,500,000	33,500,000	1.58
New York	Carl Jackson	18,400,000	20,000,000	18,000,000	17,000,000	0.45
	Clay Rozendal	19,500,000	20,500,000	18,500,000	18,000,000	0.69
	Result	37,900,000	40,500,000	36,500,000	35,000,000	1.14
North Carolina	Monica Federle	19,500,000	20,300,000	18,300,000	18,000,000	0.77
	Neola Schneider	21,000,000	23,000,000	21,000,000	20,500,000	0.54
	Result	40,500,000	43,300,000	39,300,000	38,500,000	1.31
Texas	Barry French	19,000,000	18,500,000	16,500,000	16,000,000	0.75
	Carl Jackson	19,000,000	20,500,000	18,500,000	17,000,000	0.65
	Neola Schneider	25,000,000	27,000,000	25,000,000	24,500,000	0.90
	Result	63,000,000	66,000,000	60,000,000	57,500,000	2.30
Overall Result		238,100,000	247,000,000	223,000,000	216,000,000	8.25

?

OK Cancel

4.9 Set the **Content Type** of both **Actual Value** and **Benchmark Value** as *Value as number*.

4.10 Set the **Actual Color** dropdown to *Condition Critical 3*.

Header Content

Header Text	COGS (Actual vs Plan)
-------------	-----------------------

Cell Content

Chart Size	Chart with Data Labels
Actual Value	{"(MEASURES_DIMENSION)":"COGS","(
Binding Type	Single Data Cell Value
Selected Data Cell	{"(MEASURES_DIMENSION)":"COGS
Content Type	Value as Number
Selection is Relative in Row	<input checked="" type="checkbox"/>
Benchmark Value	{"(MEASURES_DIMENSION)":"COGS_F
Binding Type	Single Data Cell Value
Selected Data Cell	{"(MEASURES_DIMENSION)":"COGS
Content Type	Value as Number
Selection is Relative in Row	<input checked="" type="checkbox"/>
Forecast Value	0
Actual Color	Condition Critical 3

5 [Testing the Application](#)

5.1 Save the Application, Syncronize it and Execute it on the BI Platform. We see our **Scorecard** displayed as so:



(Note: By default the application comes with a scrollbar enabled to scroll down through the multiple rows of data in large data sets.)

Exercise 8 - COMPOSITE

Objective of the Exercise

- Introduction to Composite
- Creating a Composite
- Using Composite in Application

1. Introduction to Composite

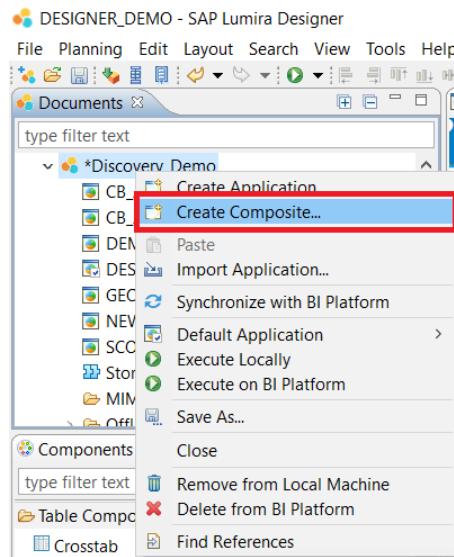
Composites are modular, reusable “app” blocks that can be plugged into other apps. When these apps are called by the user, the app is dynamically assembled, using the newest version of any composites in it. With composites, this is no longer the case. We create our header once, as a composite and add that as a component to each and every app. Then when the powers that be decide that a new logo is needed, you edit the logo in one place, the composite, and you are done. The change automatically takes effect everywhere!

Properties: Any properties defined in the interface show up in the properties pane of the composite, when it is in the application. These are properties that are globally visible within the composite but are a property of the composite from the outside.

Example: Our self-contained chart tile, from the last exercise, has an unwieldy name. Perhaps we want to allow the app designer to define the title of the chart tile in the properties pane, when they are editing the app.

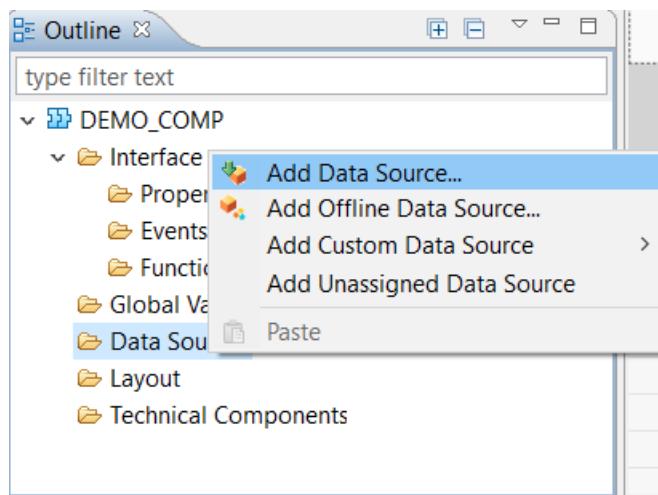
2. Creating a Composite

2.1 Right Click on the Application and elect Create Composite

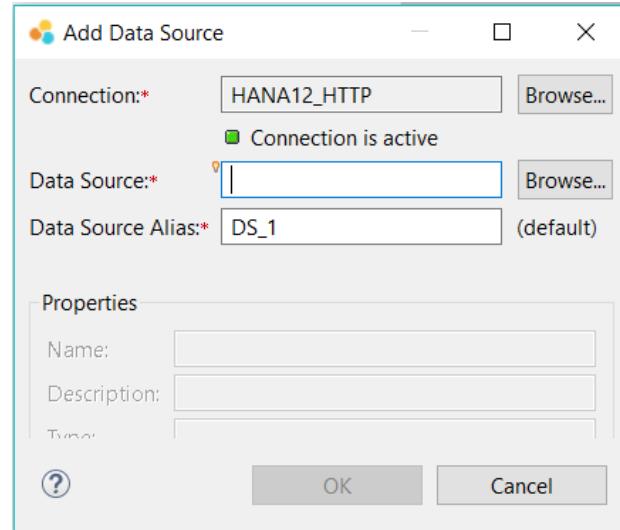


2.2 Name the Composite as DEMO_COMP.

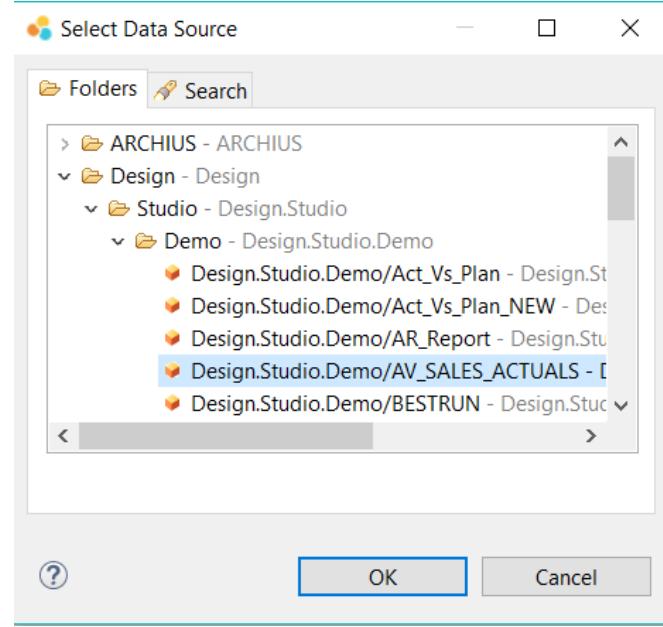
2.3 Right Click on Data Source and Select Add Data Source



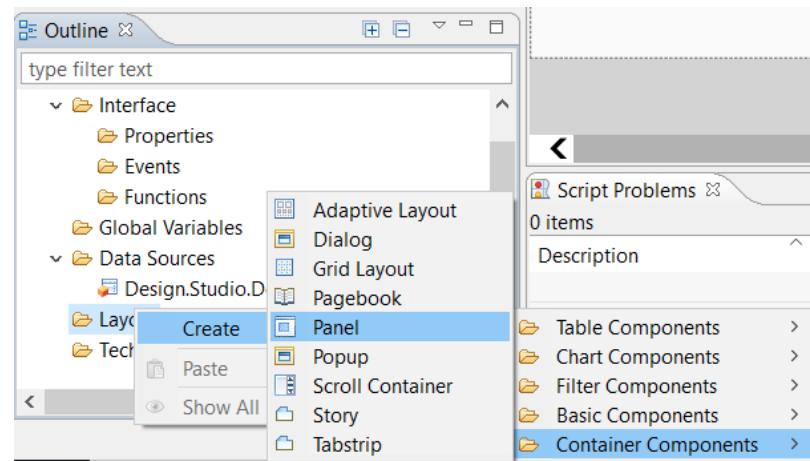
2.4 On connection Select Browse -> and choose connection as HANA12_HTTP



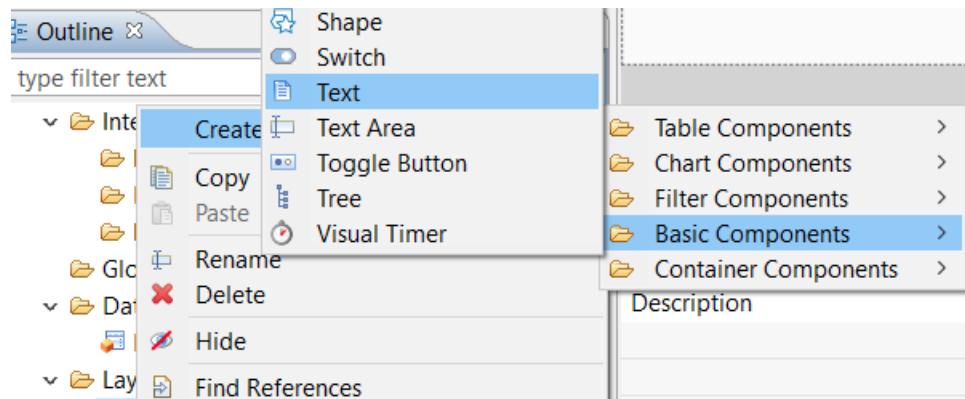
2.5 For Connection Click on Browse Choose the Connection As shown. Click on OK



2.6 Create a new Panel under Layout folder by Right Click -> Create Container Component -> Panel



2.7 Click on Maximize selected component . Next add a text component inside Panel by Right Click -> Basic Component -> Text.

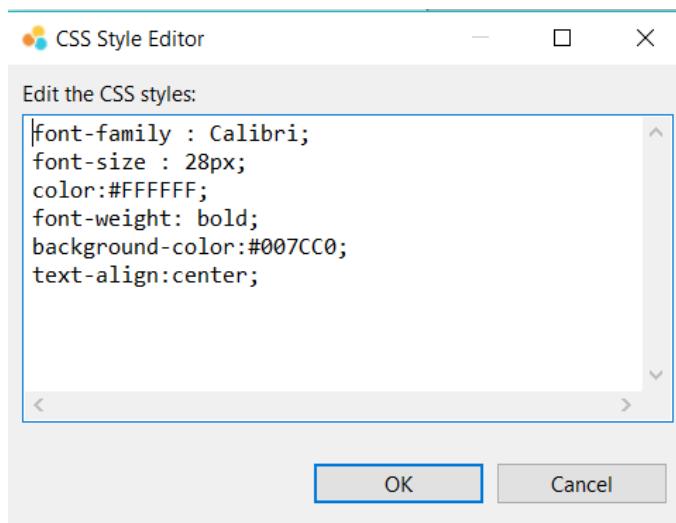


2.8 Click on Maximize selected component and set the height = 70.

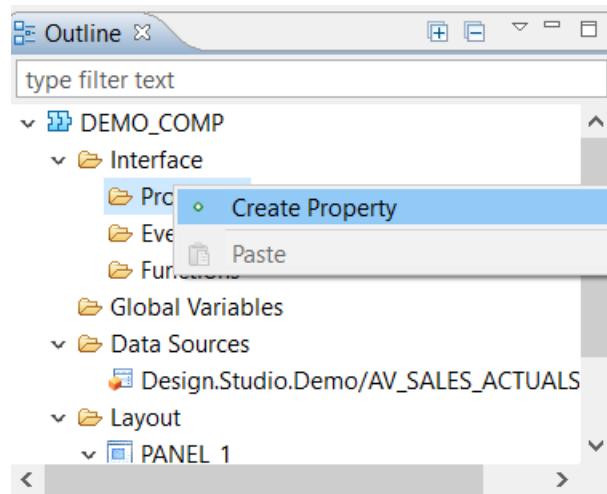
Layout	
Top Margin	0
Left Margin	0
Bottom Margin	auto
Right Margin	0
Width	auto
Height	70

2.9 Set the Name as **TITLE**, Text as **Demo Text**. Now click on CSS style and set the style as follows

```
font-family : Calibri;
font-size : 28px;
color:#FFFFFF;
font-weight: bold;
background-color:#007CC0;
text-align:center;
```



2.10 Right Click on Properties under Interface and select Create Property



2.11 Set the property name as **headerText** and default value as **Demo Text**

Property	Value	Bind
Name	headerText	
Title		
Description		
Type	String	
Bindable	true	
Default Value	Demo Text	

2.12 Go Back to the Title and under properties, Click on bind Property (+) next to Demo Text

Display	
CSS Class	+
Text	+

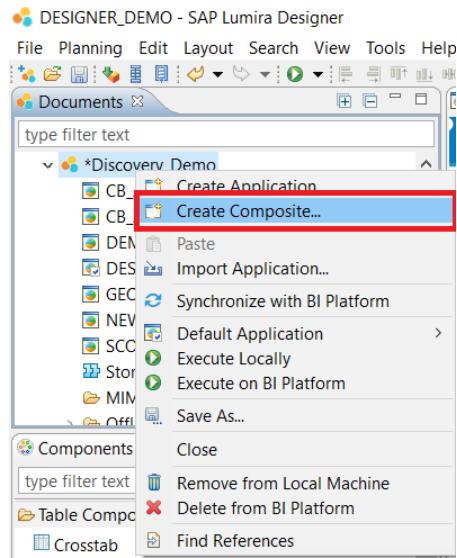
2.13 Set the properties as shown.

Type: Interface Binding Property

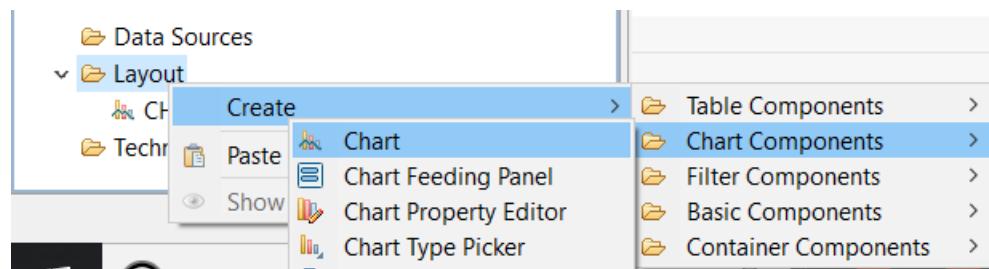
Interface Property: headerText

Text	<bound>	
Source		
Type	Interface Property Binding	
Interface Property	headerText	
Formatter Function	<none>	

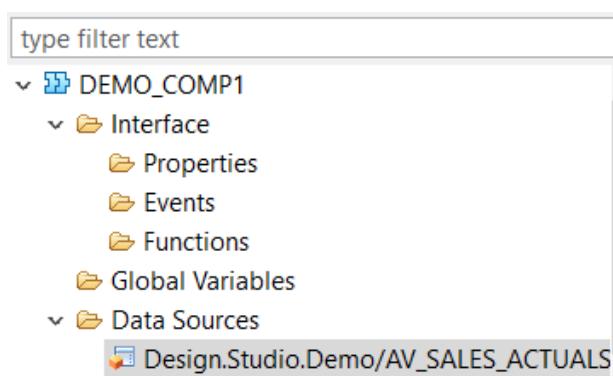
2.14 Next, let us create another **Composite** as **DEMO_Comp1**



2.15 Create a chart component under Layout by Right Click -> Create Chart Component -> Chart



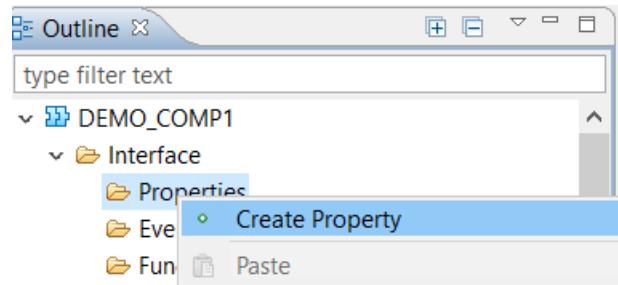
2.16 Assign a data source similar to the way assigner to DEMO_COMP. Data Source will be shown as below



2.17 Right Click on Data Source and Click on Edit Initial View. Set the initial view as shown. Click on OK

Product Sub-Category	Net Value
Appliances	38,500,000.00
Binders and Binder Accessories	36,900,000.00
Computer Peripherals	21,000,000.00
Office Furnishings	18,900,000.00
Paper	45,000,000.00
Storage & Organization	58,300,000.00
Telephones and Communication	19,500,000.00
Overall Result	238,100,000.00

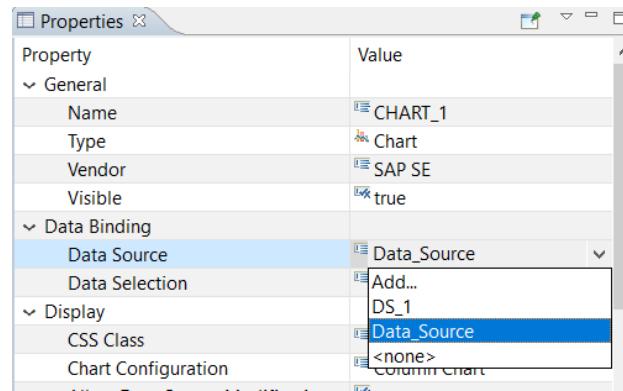
2.18 Right Click on Properties and select Create Property



2.19 Rename the property as Data_Source. And Type to **DataSourceAlias**

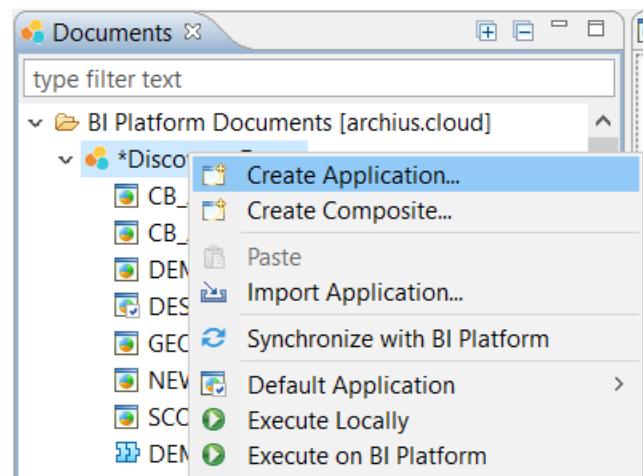
Property	Value	Bind
Name	Data_Source	
Title		
Description		
Type	DataSourceAlias	
Default Value	DataSourceAlias	

2.20 Next Go to Chart_1, and Assign Data Source to **Data_Source**

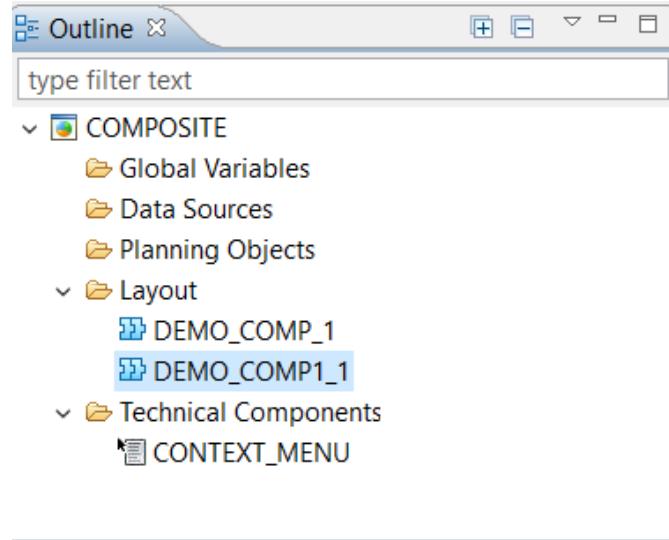


3 Using Composite in Application

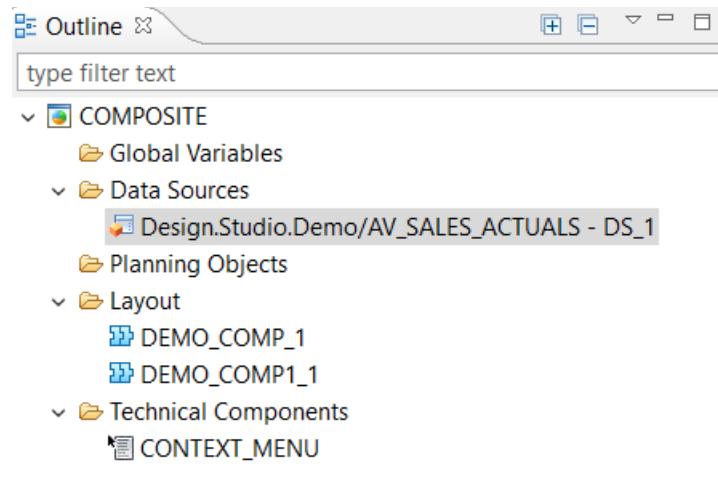
3.1 Right Click on the Document and click a new application



3.2 Name the application as COMPOSITE. Drag and Drop both the composites under Layout of Application COMPOSITE



3.3 Add a new Data Source as stated previously by using connection HANA12_HTTP



3.4 Click on Edit Initial View and set the view as follows

Edit Initial View of Design.Studio.Demo/AV_SALES_ACTUALS - DS_1

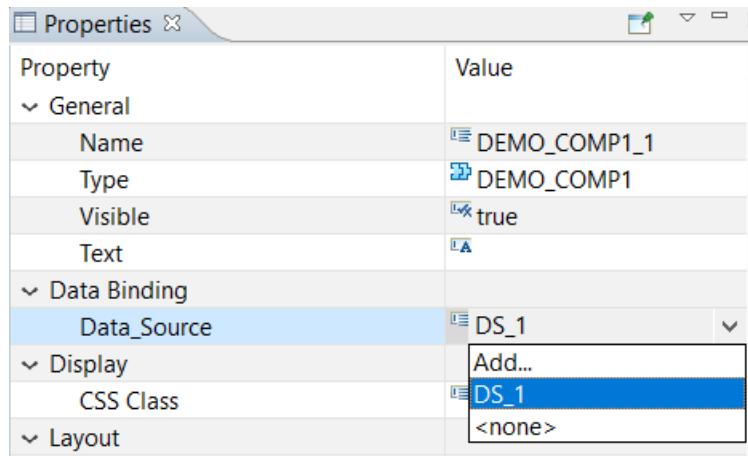
Product Sub-Category	Net Value
Appliances	38,500,000.00
Binders and Binder Accessories	36,900,000.00
Computer Peripherals	21,000,000.00
Office Furnishings	18,900,000.00
Paper	45,000,000.00
Storage & Organization	58,300,000.00
Telephones and Communication	19,500,000.00
Overall Result	238,100,000.00

Global Data Source Settings

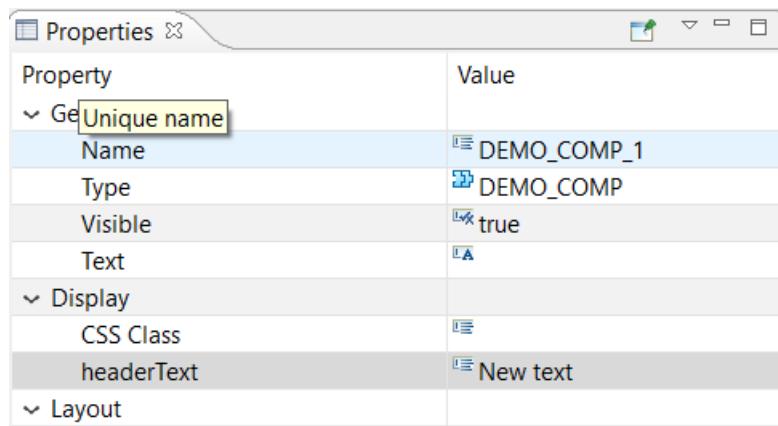
- Display of Negative Values:
- Display of Zero Values:

Buttons: Undo, Redo, OK + Create Crosstab, OK, Cancel

3.5 Click on DEMO_COMP1_1 and assign the data source as DS_1

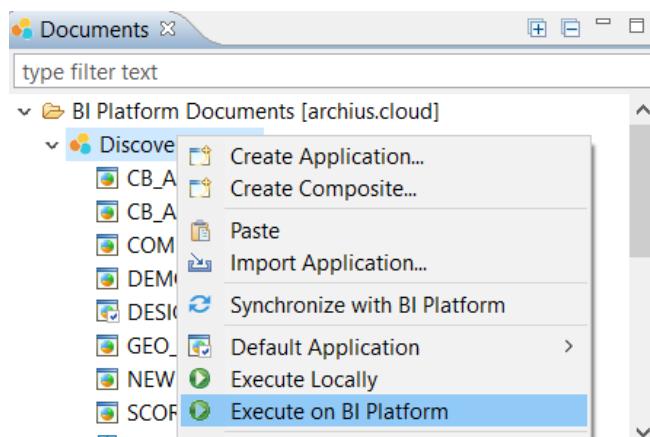


3.6 Click on DEMO_COMP1 and set the text as **New Text**. (This will overwrite the composite text and set **New Text** exclusively only for this application)

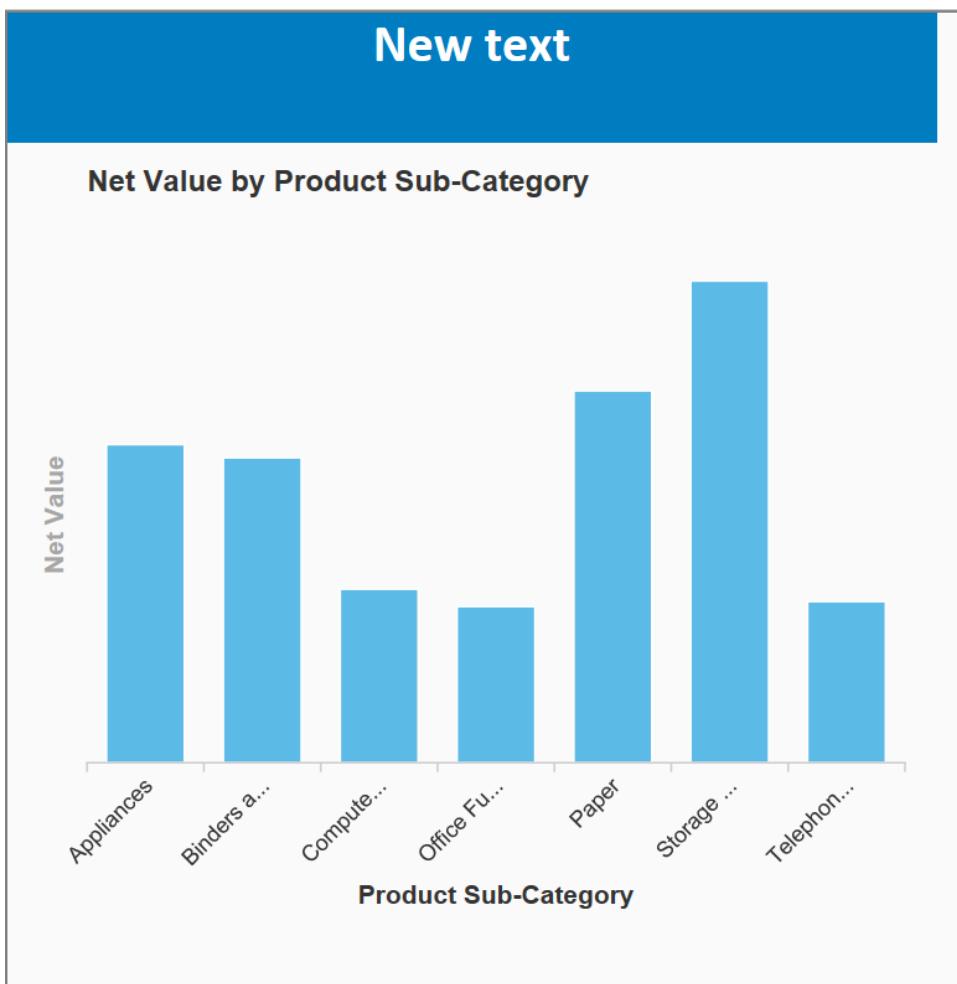


4 Running the Application

Right Click on Application COMPOSITE and set it as Default. Click on Synchronize with BI Platform. And then Execute on BI Platform



Following Output will be shown



Exercise 9 – ADAPTIVE LAYOUT

Objective of the Exercise

- Introduction to Adaptive Layout
- Creating a Adaptive Layout
- Running the Application

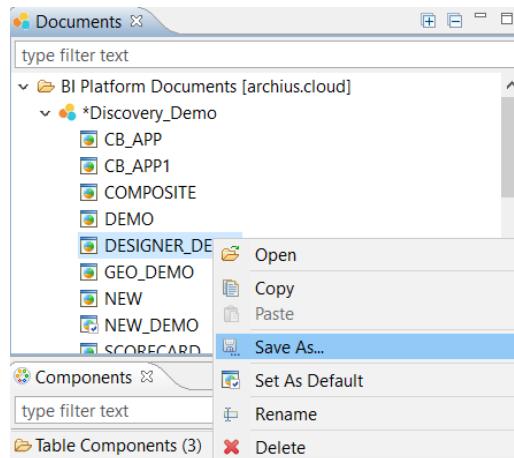
1. Introduction to Adaptive Layout

Adaptive Layout container components are used to create an adaptive application that can be viewed on multiple devices. While designing an application, use the different viewports available in the Adaptive Layout to get an idea of how the application displays on multiple devices. The viewports provide a preview of different size devices, determined by their width. This preview offers you the flexibility to create one application that can then be used across multiple devices.

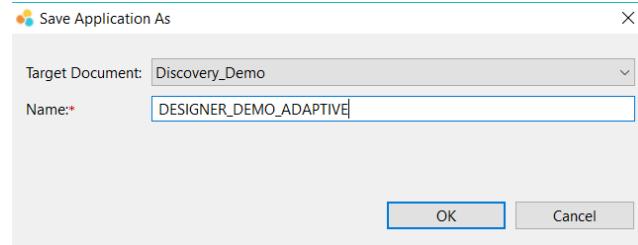
The Adaptive Layout contains a series of blocks. The blocks have a defined column span. The Adaptive Layout rearranges the layout (blocks) for each type of viewport. To visualize how an application looks when consumed by different viewports, preview the different viewports at design time.

2. Creating an Adaptive Layout

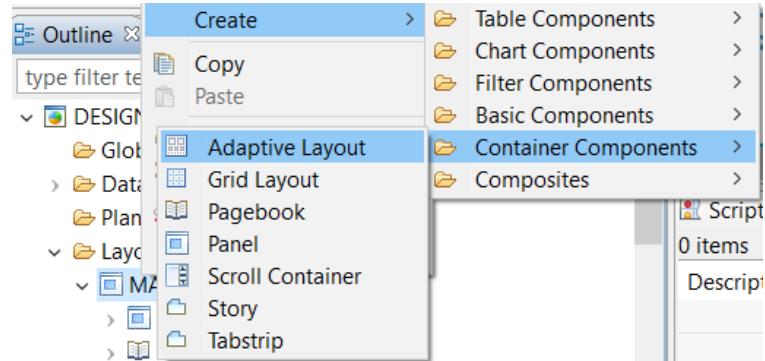
2.1 Right on the application created **DESIGNER_DEMO** and click on save as.



2.2 Save the application as Designer_Demo_Adaptive. Click on OK



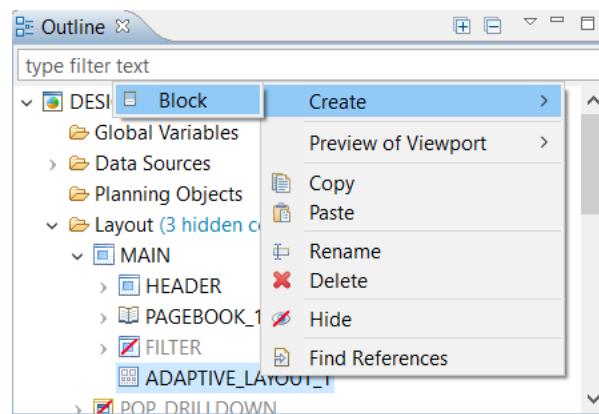
2.3 Right Click on Main Panel and select Create -> Container Components -> Adaptive Layout



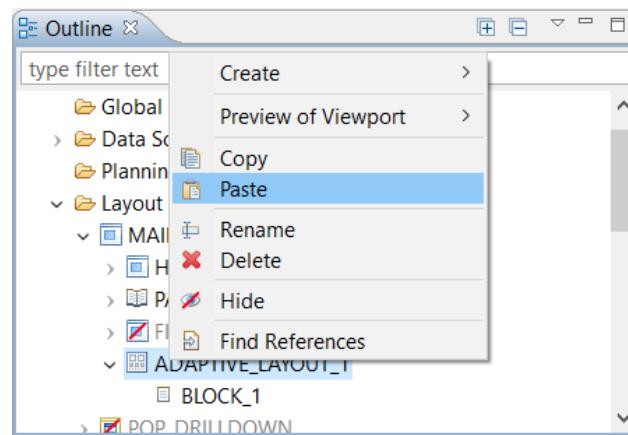
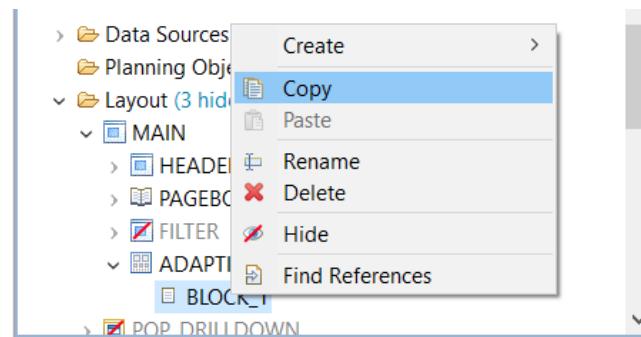
2.4 Click on Maximize selected component and Set the top Margin=70 so that it does not interfere with the header

Layout	
Top Margin	70
Left Margin	0
Bottom Margin	0
Right Margin	0
Width	auto
Height	auto

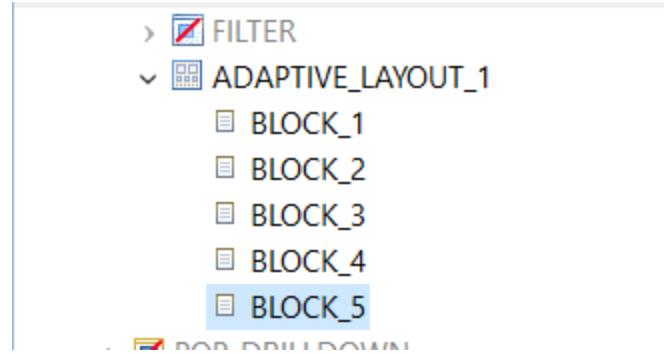
2.5 Right Click on ADAPTIVE_LAYOUT_1 and Select Create -> Block



2.6 A new Block **BLOCK1** will be created. Now copy the block and paste under ADAPTIVE_LAYOUT_1 to create another block **BLOCK_2**.



2.7 Similar copy the block and paste it under ADAPTIVE_LAYOUT_1 three more times to create a total of 5 blocks. Following layout should be seen



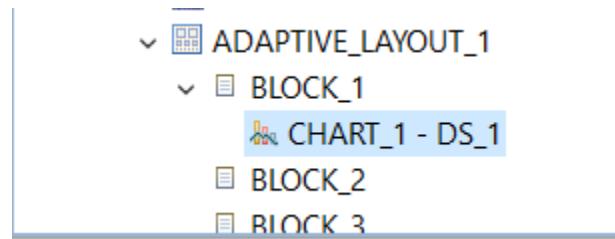
2.8 For **Block_1**, **Block_2** and **Block_5** set the Height = 225. And Colspan as follows

Display	
Small Colspan	12
Medium Colspan	12
Large Colspan	3
Extra Large Colspan	3
Layout	
Height	225

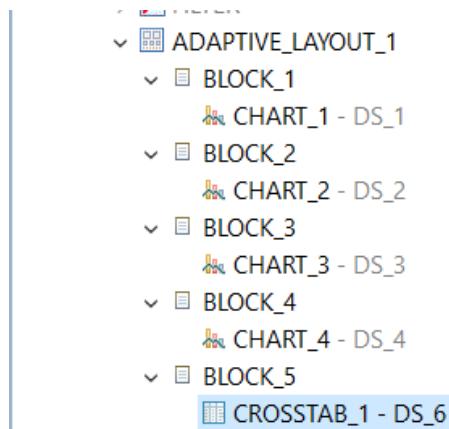
2.9 For **Block_2** and **Block_3** set the Height = 225 and Colspan as follows

Display	
Small Colspan	12
Medium Colspan	6
Large Colspan	3
Extra Large Colspan	3
Layout	
Height	225

2.10 Now Drag the Chart_1 component from PAGEBOOK_1-> PAGE_1 -> CONTAINER_GRID_1 -> Cell [0,0] to Block 1. Following layout should be seen



2.11 Similarly Drag **Chart_2** -> **BLOCK_2**, **Chart_3** -> **BLOCK_3**, **CHART_4** -> **BLOCK_4**, and **CROSSTAB_1** -> **BLOCK_5**



2.12 Click on All the 5 components one after the another and click on Maximize selected component . Also, Right Click on PAGEBOOK_1 and delete it.

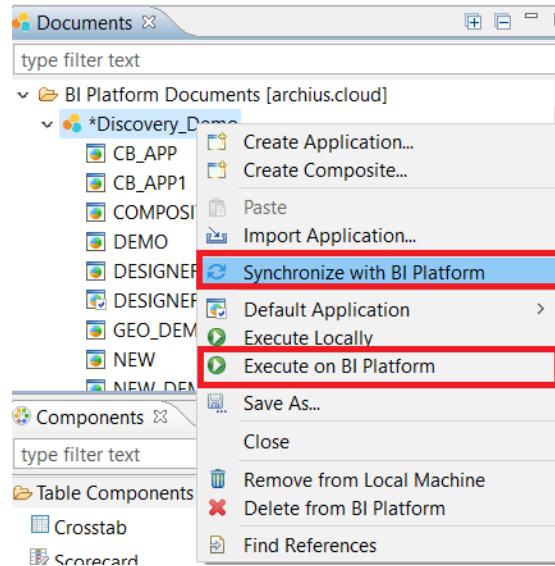
2.13 Again Go to Navigator Component and delete it. We will not require this as we have only 1 page.

2.14 Double click on Filter icon and change the script as shown

```
Script for event "On Click" of component "FILTER_ICON":  
1 if (FILTER.isVisible()== false)  
2 {ADAPTIVE_LAYOUT_1.setLeftMargin(190);  
3 FILTER.setVisible(true);}  
4 else  
5 {ADAPTIVE_LAYOUT_1.setLeftMargin(0);  
6 FILTER.setVisible(false);}  
7
```

[3. Running the Application](#)

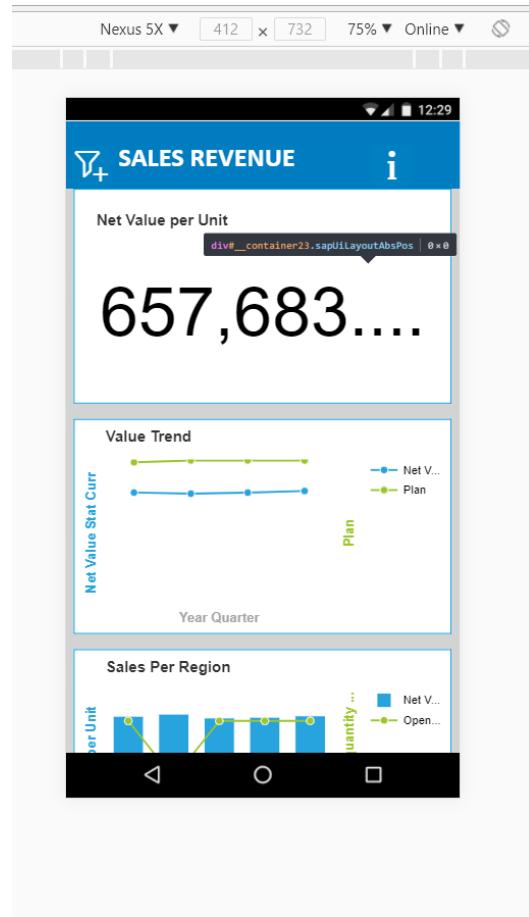
3.1 Right Click on DESIGNER_DEMO_ADAPTIVE and click on Save as default. Then Right Click on Document and click on Synchronize with BI Platform. After It Click on Run on BI Platform



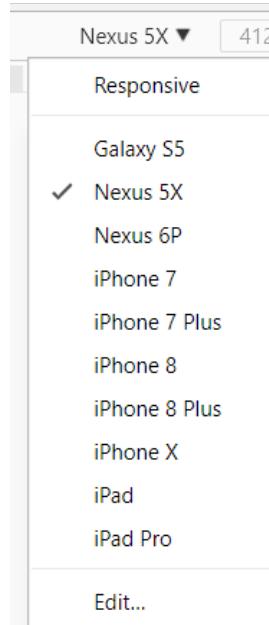
3.2 The Application will open in Browser. Right Click and select Inspect. Click on Toggle Device Bar



. The application will be seen as follows



3.3 Any options from the various available options can be selected.



3.4 For Example, iPad Pro.

iPad Pro ▾ 1024 x 1366 49% ▾ Online ▾

SALES REVENUE

Net Value per Unit

657,683.07

Value Trend

Net Value Stat. Curr. Plan

Sales Per Region

Net Value per Unit Sales Organization Open order quantity

Net Value stat curr by Product Group

Product Group Net Value stat curr

Sales Organization	Product Group	Net Value stat curr ↗	Net Value per Unit ↗	Qu. in base units [PC] ↗	Open order stat curr [E] ↗
Berlin	Mobile Devices 1	92,313,089.816	678,455.30	262.00	23,040 ↗
	Mobile Devices 2	90,171,178.020	631,068.39	264.00	21,656
	Monitors 1	91,544,515,788	681,026.21	263.00	23,284
	Ultrabooks 1	94,032,236,892	671,854.06	264.00	23,056
	Ultrabooks 2	85,243,047,516	567,408.12	264.00	19,473
	Wide LED Backlight 1	93,969,095,484	728,891.99	263.00	24,913
	Result	547,273,163,316	659,338.09	1,580.00	135,426
Mobile Devices 1	10,233,443,352	587,318.83	132.00	10,078 ↘	