Visual BI Value Driver Tree For SAP Lumira Designer

- User Guide -

1 Copyright

1.1 Trademark Information

1.2 Patent Information

1.3 SAP Trademarks

www.visualbi.com

solutions@visualbi.com

Phone: +1 888-227-2794

Fax: +1 888-227-7192

Plano, TX ■ Dallas, TX ■ Chennai, India

2 Definitions

Design Studio	SAP BusinessObjects™ Design Studio
Lumira Designer	SAP Lumira Designer
Value Driver Tree Extension	Value Driver Trees for Lumira Designer and Design Studio
Value Driver Tree	Mathematical or conceptual business models visualised in a way that links the business KPI's (what management or stakeholders care about) to the operational drivers (the things that can be influenced to change the KPI's).
Node	A generic reference to a unique KPI or Value Driver in a value driver tree
Node Widget	The visualization of a KPI or Value Driver in a value driver tree
Derived Node	Additional Information for a node but not directly related to the calculation hierarchy. This will often be key benchmarking KPI's such as cost/production unit.
Parent	A reference to the Node one level above in the Value Driver Tree
Children	A reference to the collection of nodes immediately below a Node

3 Introduction

3.1 Document History

This document is valid for the releases shown in the Table below (see Table 3.1):

Major Release	Minor Release	Patch	Date
2	**0**	**0**	**December 2017***
2	0	5	July 2018
3	**0**	**0**	**February 2019***

Table 3.1: Document History

3.2 Who should read this guide?

This User Guide for the Visual BI Value Driver Tree for SAP Lumira Designer (VDT) offered by Visual BI is meant for users that are going to use Apps leveraging the Value Driver Tree Extension to analyse value driver relationships, compare versions such as Forecast vs Current Budget and simulate what-if scenarios such a change in price, volumes and efficiencies real-time.

3.3 What is the Visual BI Value Driver Tree for SAP Lumira Designer (VDT) Suite?

The Visual BI Value Driver Tree for SAP Lumira Designer (VDT) suite is mathematical or conceptual business models visualized in a way that links the business KPI's (what management or stakeholders care about) to the operational drivers (the things that can be influenced to change the KPI's).

3.4 Traditional Approach

Spreadsheet models are often used to support planning and forecasting. They often contain a high number of parameters. Visualizing, simulating and sharing models is challenging and the models often become personal tools only understood by the accountant or engineer who created it. Integrating data real-time and dynamically in spreadsheets is difficult resulting in the models that often are static, point-in-time and for a single period.

4 Instructions

The Value Driver Tree Extension provides various components that the Designer can use in conjunction with standard SAP components or other extensions. The extension also offers many configuration options allowing the designer to tailor the final App to suit your specific needs.

In this guide, we will describe the common end user functionality using the demo App as an example. It is however important to understand that your specific implementation may differ.

Once your designer has implemented and configured the value driver tree, it will be accessed the same way you access your current Design Studio/Lumira Apps. This would typically be via the BI Launchpad or via an OpenDoc URL.

When you open a Value Driver Tree App, it will open in a similar manner at the screenshot below:

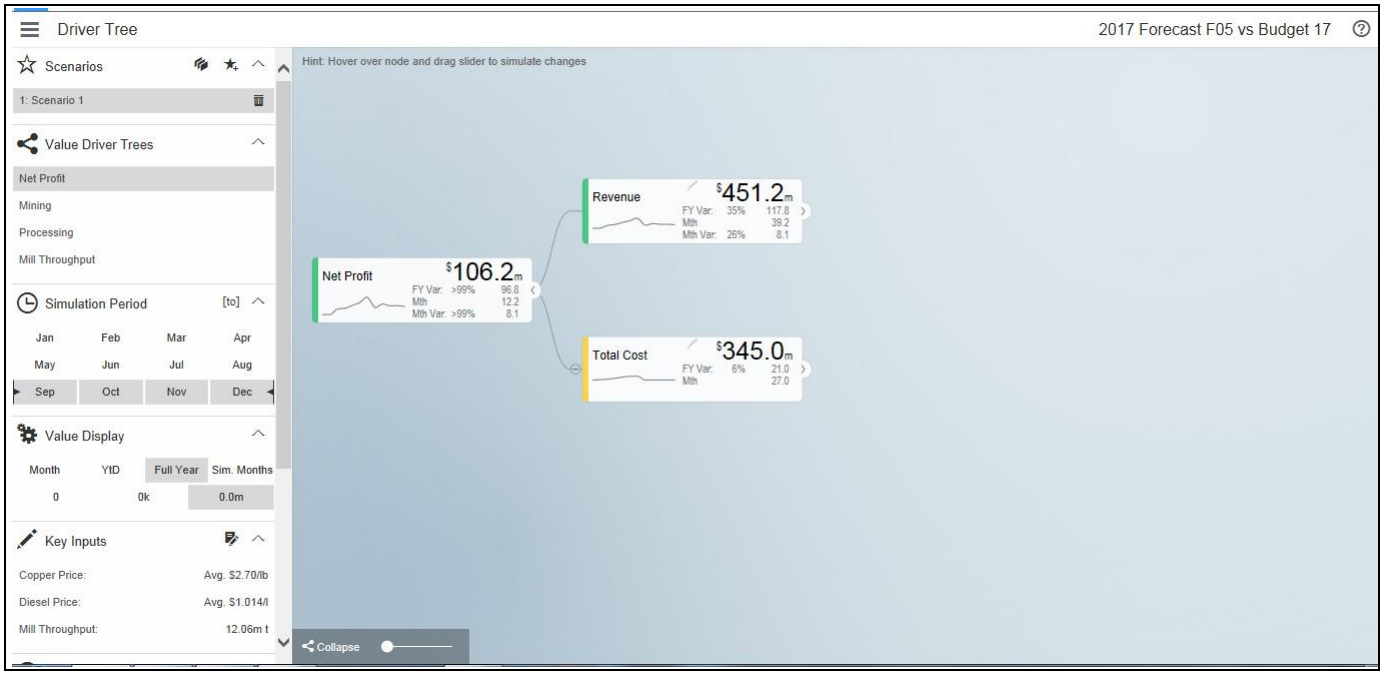


Figure 4.1: Value Driver Tree

The left panel is the navigation panel, you can hide this panel by clicking on the top left corner icon ☰. On the canvas you will find the value driver tree. By the default the tree is open on collapsed mode at a level defined by the designer. This is typically completely collapsed.

4.1 Input Data

The Value Driver Tree works with two input data series – the primary and the comparison data series. A common way of using this is to compare Forecast vs Budget, Budget vs Last Year Actuals etc. The data series is at a certain time grain. This would typically be 1 year by 12 months or a multiyear view like 5 years plan by year.

The selecting of the time series will often be based on Filters or Variables. A typical scenario is where the designer has defined 4 variables on the underlying data sources:

Primary:

Version: ie. Defaulted to Forecast

Year: ie. Defaulted to Current Year

Comparison:

Version	ie. Defaulted to Budget
Year	ie. Defaulted to Current Year

This allows the users to compare different versions and years in the value driver tree model.

The Value Driver Tree can display individual periods but can also apply time aggregation such as Full Year, Year-to-Date. You can see the currently selected Value Display aggregation on the Navigation Panel. In the screenshot above, Full Year is selected. The primary value displayed on the Widgets is therefore in this instance the Full Year aggregated data.

4.2 Tree Drill Down

Nodes that has children will have an expand '>' button to the right of the Widget as highlighted below:

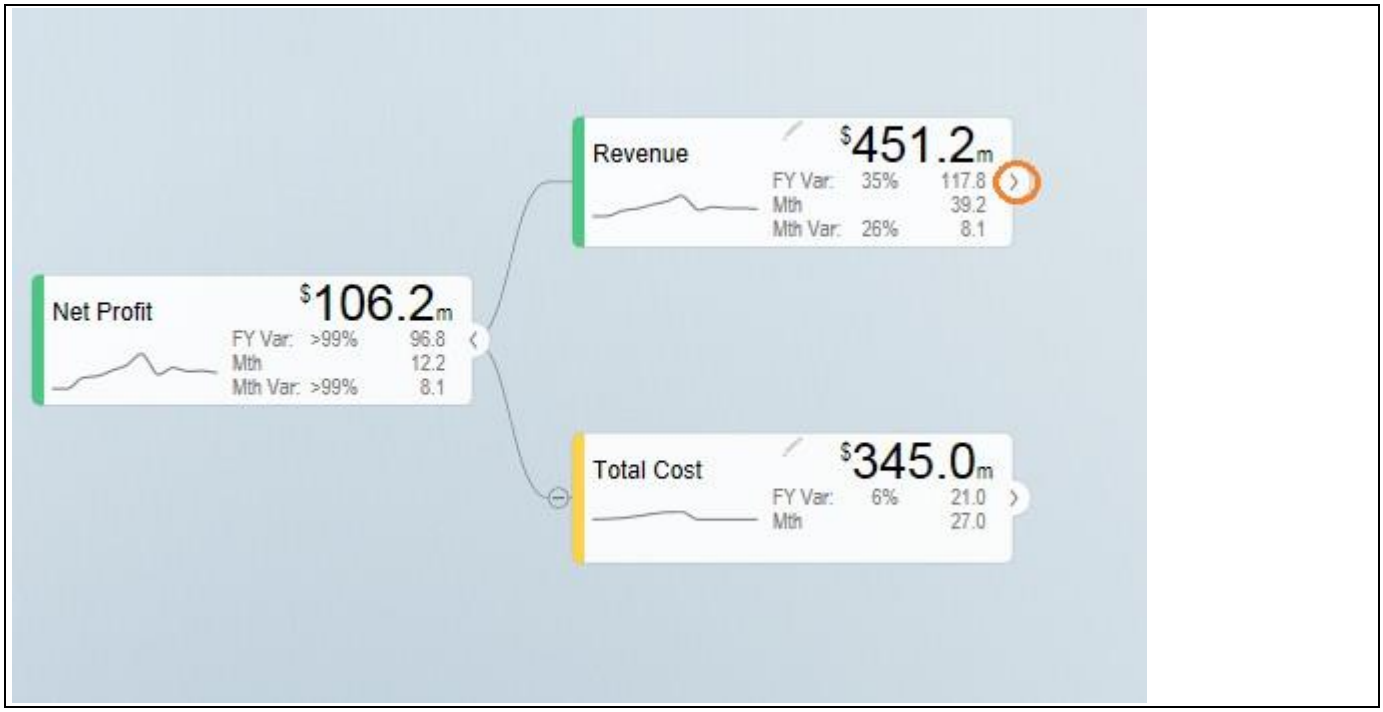


Figure 4.2: Nodes with children

You can drill down the tree by clicking the expand button.

The tree will focus on the branches that are being expanded. You can collapse any specific node by clicking the '<' collapse button on the right of a widget.

Value Driver Trees are in reality networks. The same driver may impact multiple aspects of the three. The component therefore contains a concept of linked nodes. Such nodes have a link icon that you can click which will jump to where the linked node is defined and highlight the node:

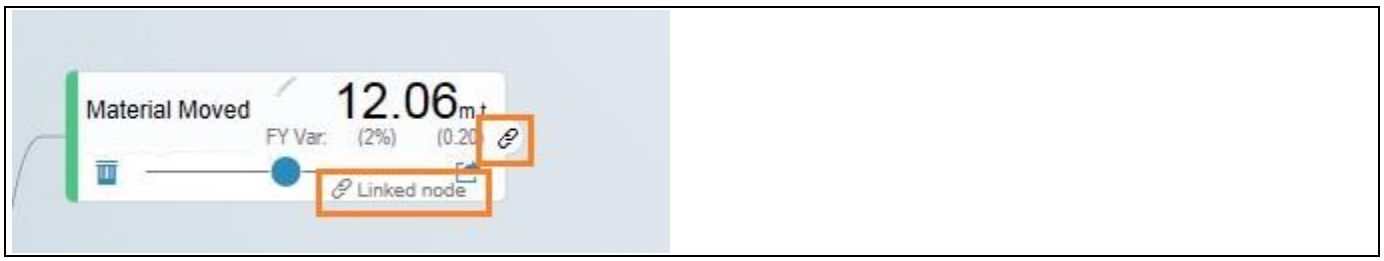


Figure 4.3: Link

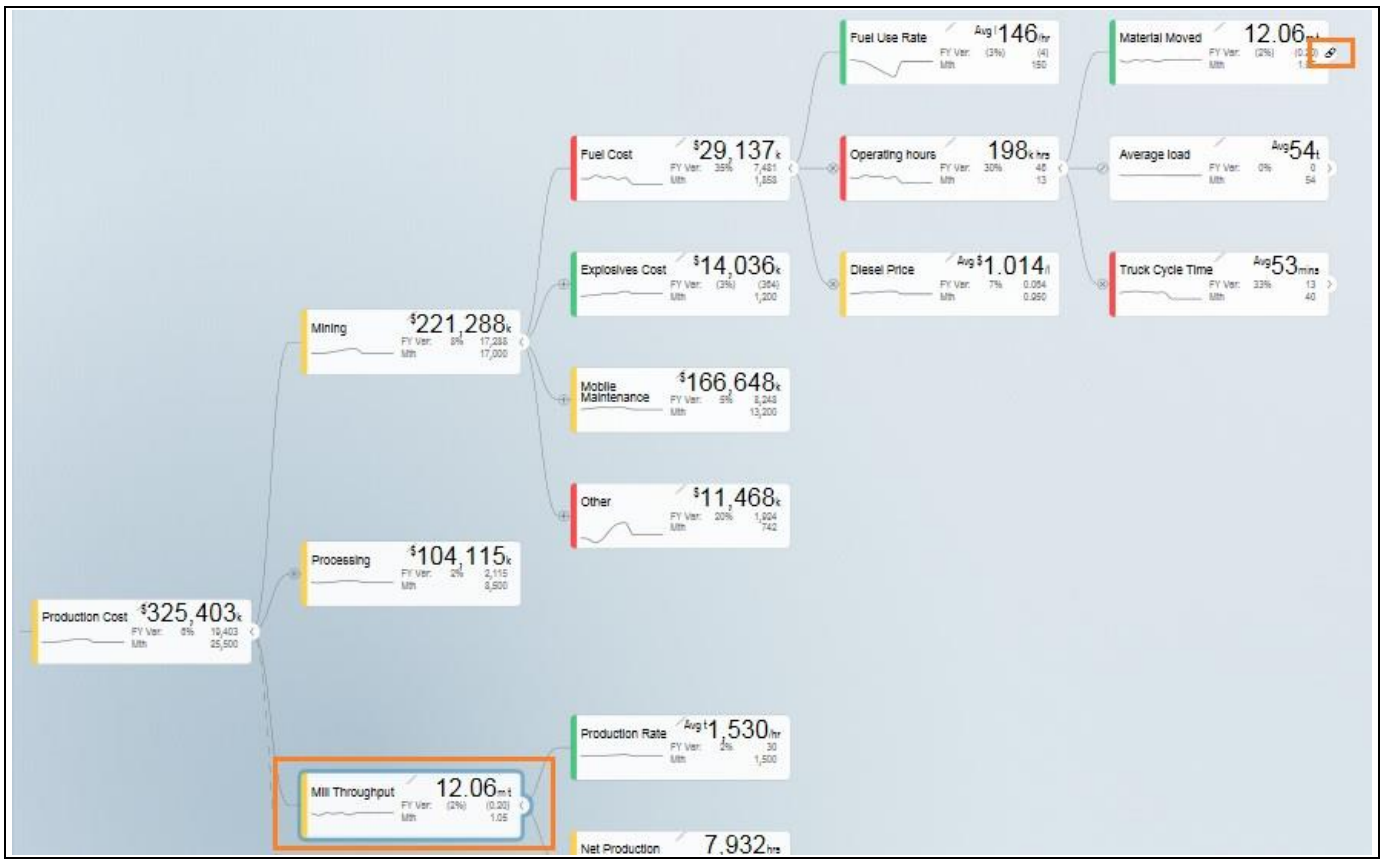
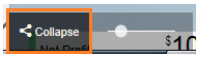


Figure 4.4: Linked Nodes

To collapse all nodes, click on the collapse icon on the bottom left of the canvas 

4.3 Zoom and Pan

Value Driver Trees can be large. You can zoom and pan the canvas view. If you are using an iPad or iPhone, you can zoom by pinching and pan by dragging the same way you would navigate a map.

If you are using a mouse with a scroll wheel, you can point your curser anywhere in the canvas and scroll the wheel to zoom in and out.

To pan, click and hold the left bottom of the mouse and drag the tree around.

You can also use the zoom button to zoom if your devise doesn't support other options 

4.4 Value Display and Simulation Period

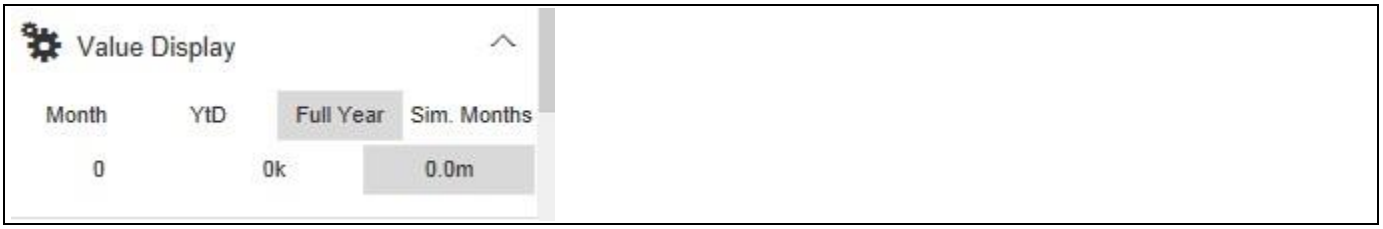


Figure 4.5: Value Display

With the 'Value Display' functionality you can choose the time aggregation you want to display as the primary value. The designer can link certain nodes to a user selected scaling. You can select the scale you wish to display by selecting this under the Value Display section. Note that the options here can be configured so they may differ. It is important to note that not all notes are scaled using this as the value driver tree typically contains both amounts, prices, rates and physical volumes.

In addition to the primary display value, the designer can also choose a secondary value to be displayed. I.e. if you are showing Full Year as primary, in the below example, it has been configured to visualize the Month data as secondary:

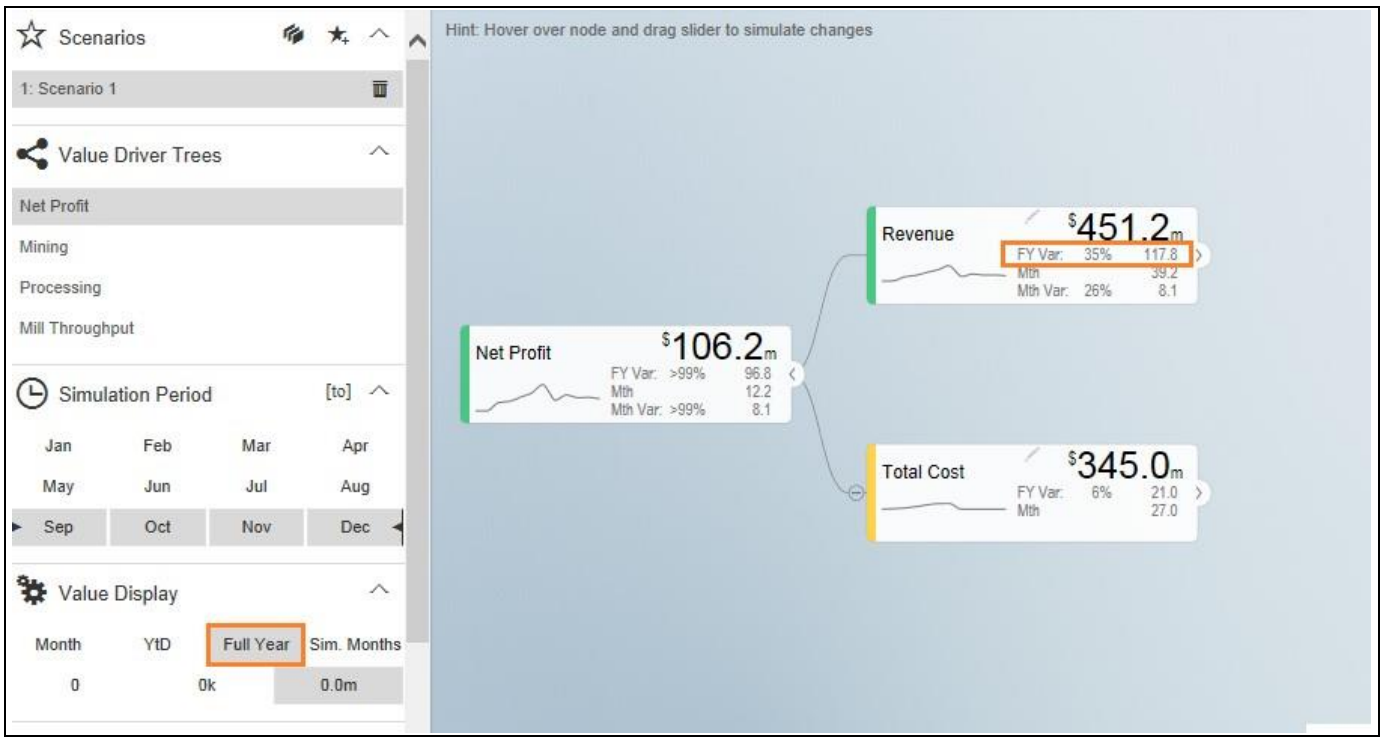


Figure 4.6: Value Display

The time aggregation and display relates to the Simulation Period selected.

When **Month** is selected, it shows the first month of the simulation interval selected. By default, the simulation period is the selected period and the future. This is an important concept as when we simulate, we generally don't want to apply a change to the past as we can't change it. It is possible to also set the 'to' period in a simulation, just click [to] and highlight the month you want as the end period of a simulation. This is however a special scenario.

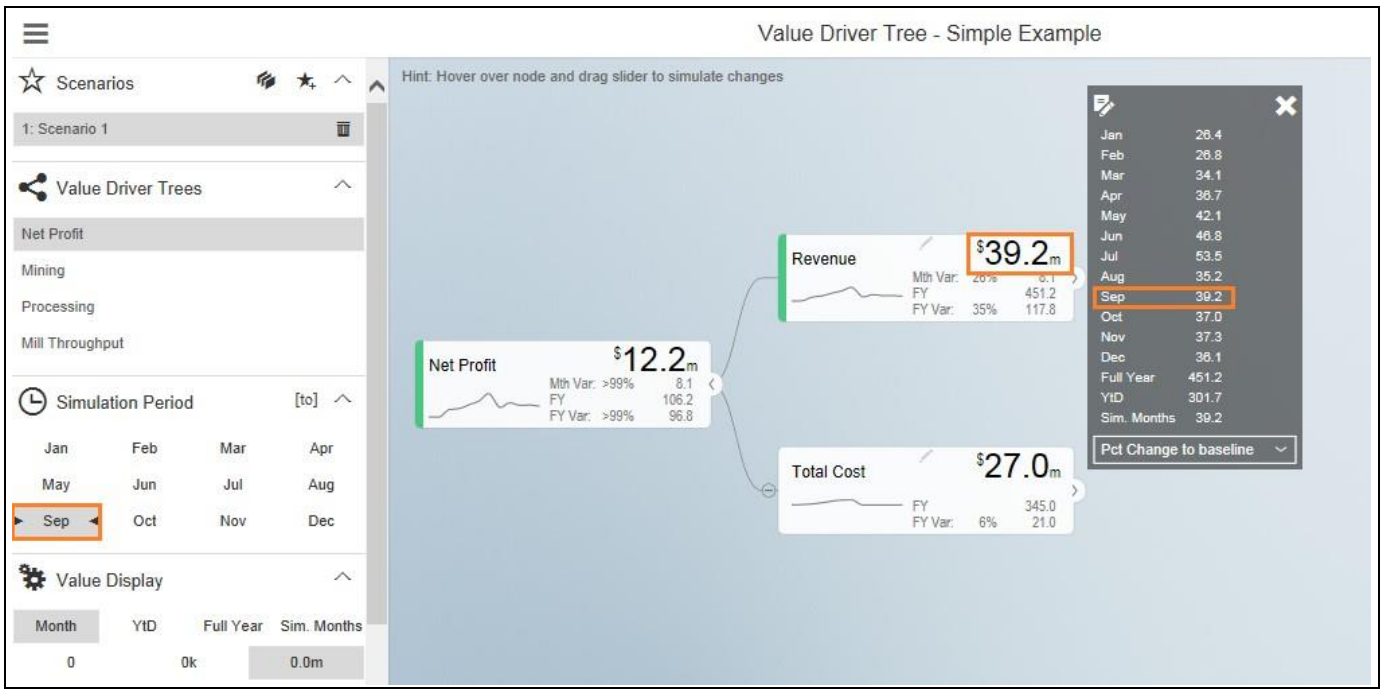


Figure 4.7: Simulation Period

YTD – the selected period -1 is the year to date data (eg. selected simulation period is September to December, the YTD data will be from Jan to August).

Note: Simulation of future periods will not impact the YTD data.

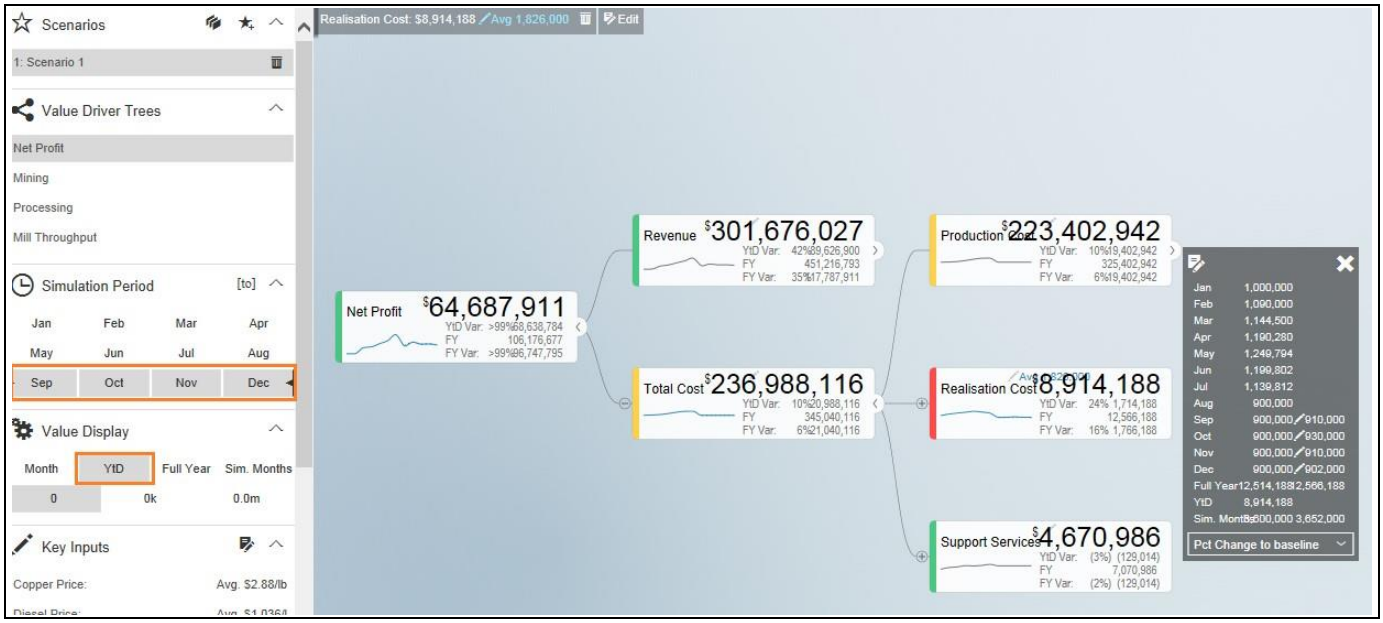


Figure 4.8: Value Display and Simulation Period

Full Year – Shows the full year cumulated data

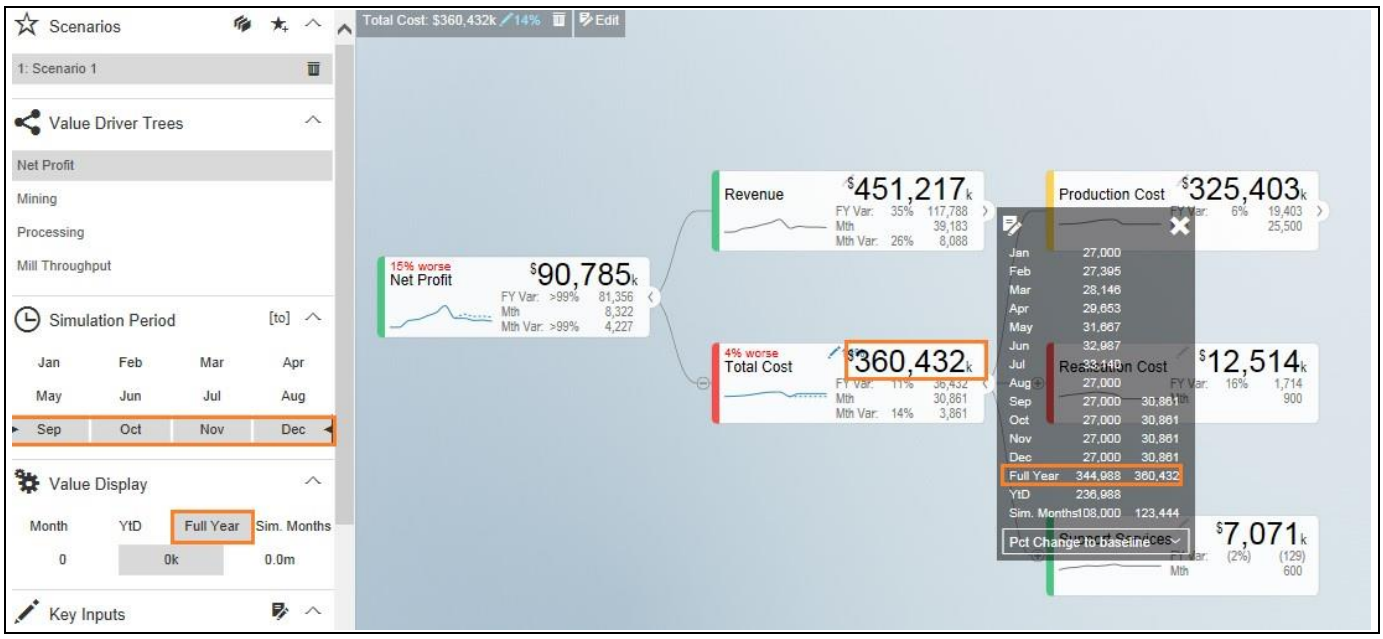


Figure 4.9: Value Display and Simulation Period

Sim. Period – to simulate a specific period, for example a 3 month rolling forecast from September to November, click the first month September then click [to] and click the last month November for simulation. You can then choose to only show the values aggregated for the selected simulation periods

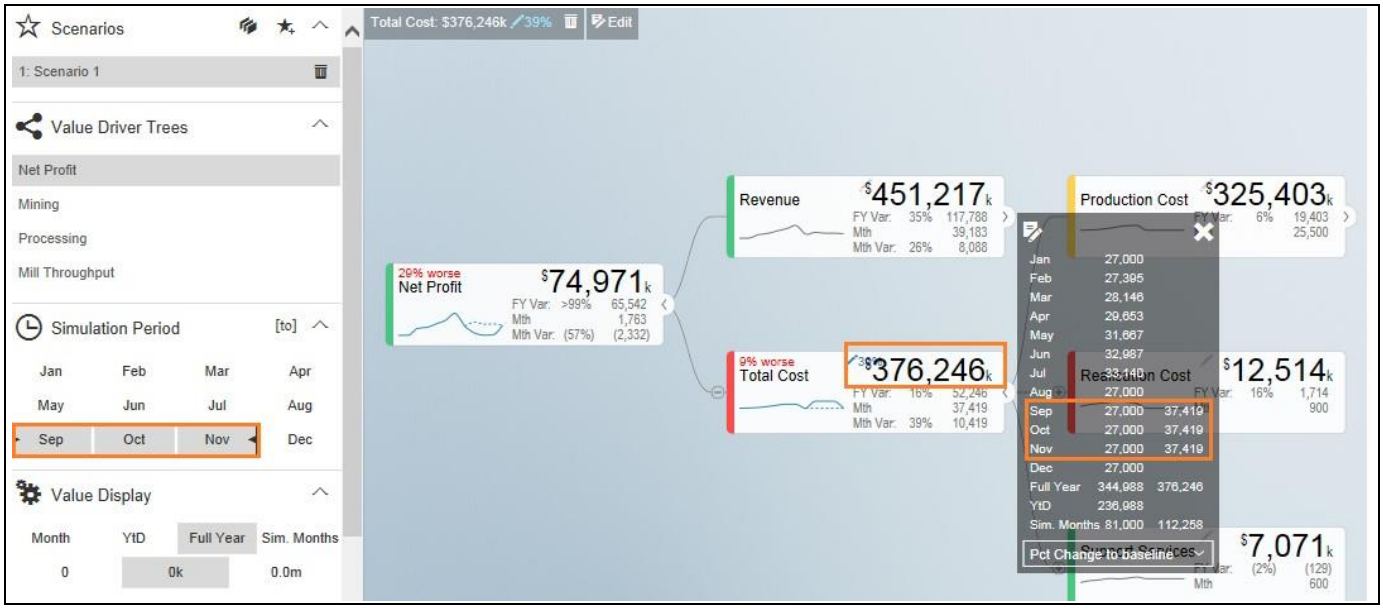


Figure 4.10: Value Display and Simulation Period

4.5 Value Driver Tree views - Filters

If your Value Driver Tree is large, the designer may have defined specific sub views. These would be listed under the Value Driver Trees Section of the Navigation Panel. When clicking on of these views, the tree will be filtered and the specific node will be made the top node. Calculation dependencies are still respected but this significantly improves the navigation as it focuses the tree on a specific area.

In this example, the options defined are:

- Open all company level: Net Profit
- Open per department: Mining, Processing or the operational tree for Mill Throughput

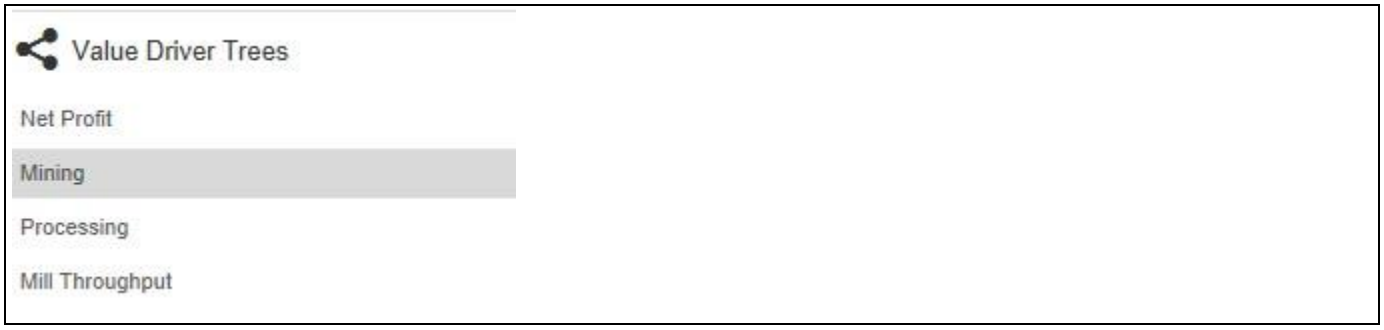


Figure 4.11: Value Driver Tree - Filters

4.6 KPIs

The Navigation Panel can be configured to list a set of KPI's from the value driver tree. You are able to navigate to each of them for a fast view on the simulation impact on a given KPI. Click on the KPI to navigate to the node.

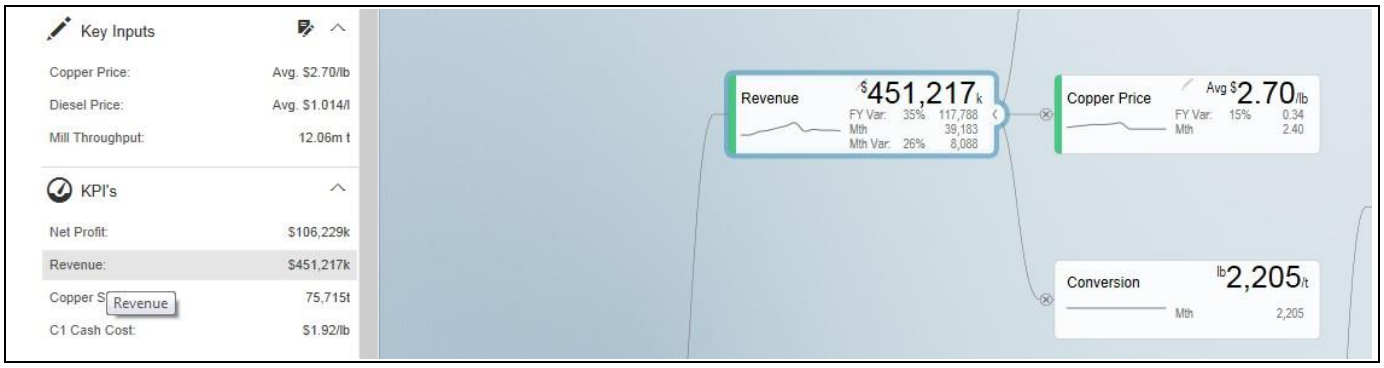
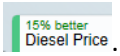


Figure 4.12: KPIs

4.7 Data Simulation

There are 2 options to simulate:

- Hover over a desired node to simulate, slide the scaling button to right for positive impact or left for negative impact. The simulation will be showed on the right hand side. A window will open displaying the changes. The pencil icon will display the change in percentage that has been made: 

, and a legend with percentage showing if the adjust has made the scenario better or worse 

.

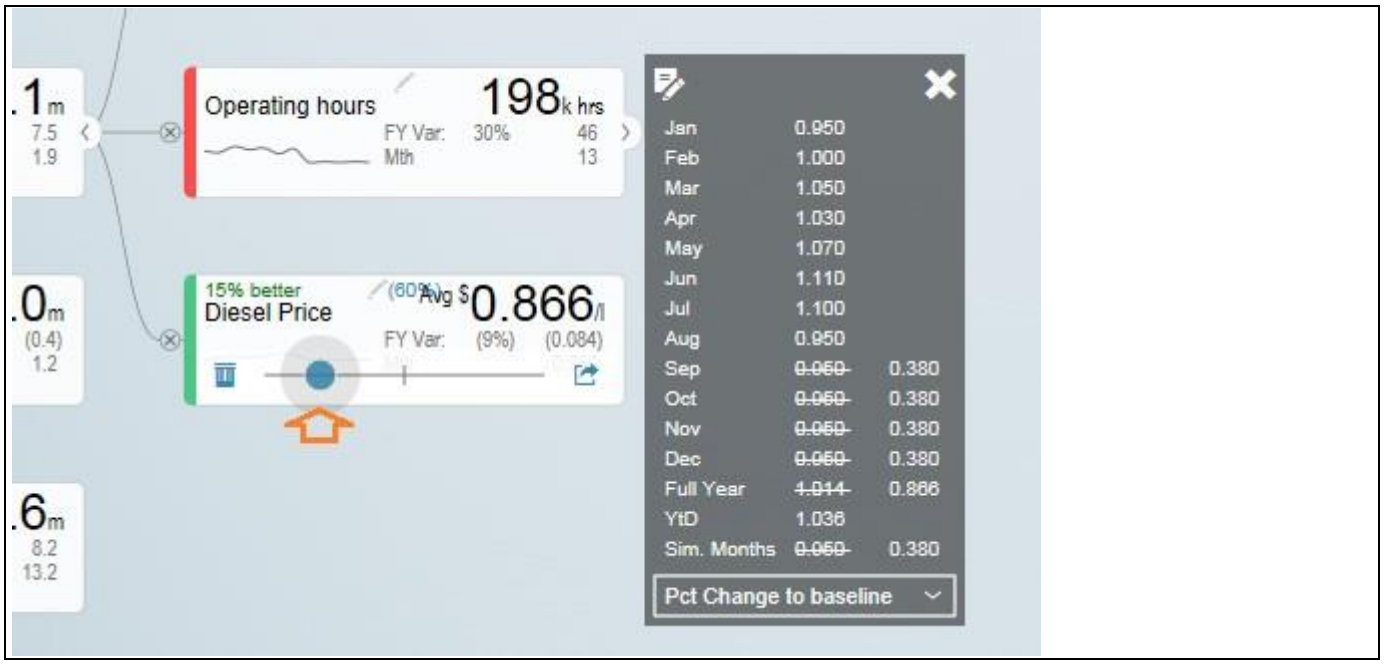


Figure 4.13: Data Simulation

- Click on the arrow icon of the node  then click 'Edit Key Input'  and manually edit the changes in simulation per period. Click X to close manual input table.

Notes

- You can update the periods highlighted in YELLOW based on the selected simulation period.

Figure 4.14: Simulation Periods

The edit icon will indicate the months that have been edited

Figure 4.15: Simulation Periods

- You can update the 'Variation' column which will overwrite the numbers in the selected simulation periods. Any manual updates in period will not be overwritten by the 'Variation' column. In the example below, in blue are the months which were manually edited and in orange are highlighted the variation and in this case the month to which the variation was applied.

Figure 4.16: Simulation Periods

Click reset icon to return to the original data .

4.8 Key Inputs

Key Inputs is available on the left side panel to directly simulate key cost drivers without going into the node details.

Click the 'Edit Key Input' icon and manually edit the changes in simulation per periods.

Figure 4.17: Key Inputs

Notes:

- Can update the periods highlighted in YELLOW based on the selected simulation period. All the changes in the Key Assumptions are highlighted in the left side panel side.

Figure 4.18: Simulation Periods

Click X to close manual input table.

- Click a Key Assumption or Key Performance Indicator and it will highlight the source nodes to further check the details.

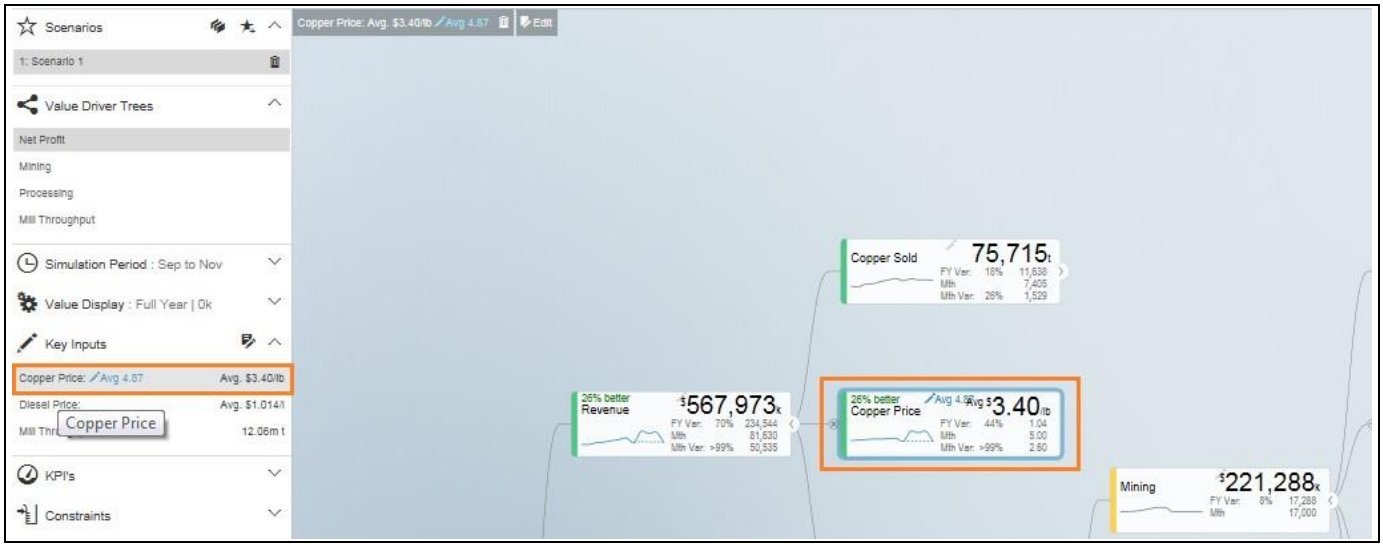
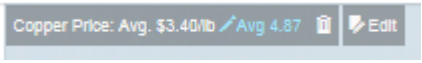


Figure 4.19: Key Inputs

All simulations are highlighted in the upper right corner.



Click  to return to the original selected version without any simulation.

4.9 Scenario Comparison

You can create multiple scenarios in parallel, compare them as well as sharing the collection with others.

This is how you do it.

- Update the default scenario 1.
- Click 'Create new Scenario' icon . This copies the active scenario.

Figure 4.20: Scenarios

Notes:

- The copied scenario is the replicate of the last scenario (e.g. Scenario 2 will have the same data from the copied Scenario 1.
- We have the option to make additional simulation in the new scenario or click to Reset all return to the original selected version without any simulation and then make a new simulation.
- Click 'Compare Scenario' icon to generate a report comparison format with the key Assumptions and KPI's.
- Comparison report compares all scenarios that have been created in the active collection. It also shows the Simulated Data Scenario (Baseline: Full Year), Comparison Version (Full Year) and Simulated Data Scenario Year to Date (Baseline: YTD).
- Highlighted in GREEN fonts the most favorable results and in RED fonts the least favorable result in the scenario comparison report.

Click Download to Excel located in the lower left corner to extract the report in excel format.

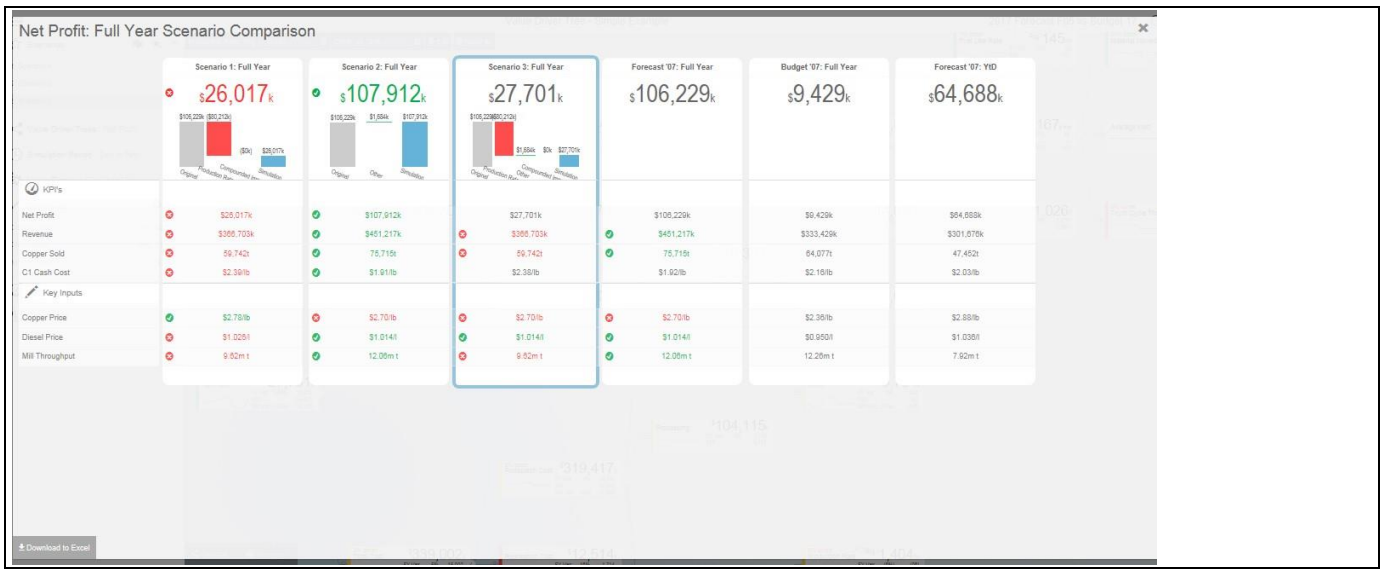
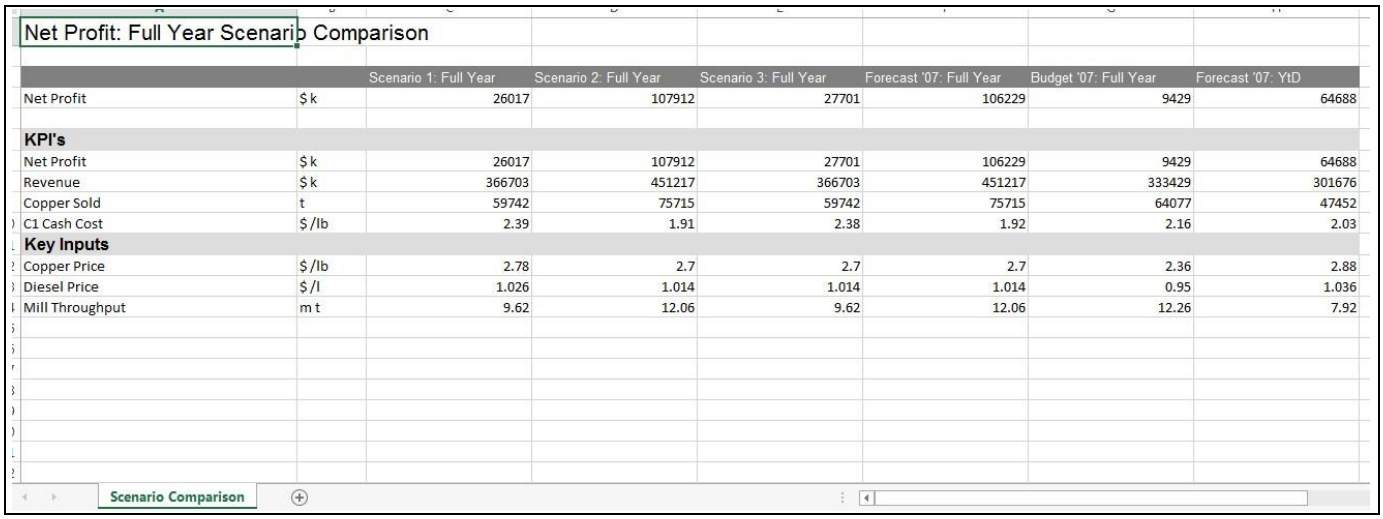


Figure 4.21: Full Year Scenario Comparison



Net Profit: Full Year Scenario Comparison

		Scenario 1: Full Year	Scenario 2: Full Year	Scenario 3: Full Year	Forecast '07: Full Year	Budget '07: Full Year	Forecast '07: YtD
Net Profit	$ k	26017	107912	27701	106229	9429	64688
KPI's							
Net Profit	$ k	26017	107912	27701	106229	9429	64688
Revenue	$ k	366703	451217	366703	451217	333429	301676
Copper Sold	t	59742	75715	59742	75715	64077	47452
C1 Cash Cost	$ /lb	2.39	1.91	2.38	1.92	2.16	2.03
Key Inputs							
Copper Price	$ /lb	2.78	2.7	2.7	2.7	2.36	2.88
Diesel Price	$ /l	1.026	1.014	1.014	1.014	0.95	1.036
Mill Throughput	m t	9.62	12.06	9.62	12.06	12.26	7.92

Scenario Comparison

Figure 4.22: Full Year Scenario Comparison

4.10 Constraints

Constraints are also available in the left side panel that identifies the metrics that are overcapacity (highlighted in RED fonts) or still have opportunities for further improvements (WHITE fonts).

Notes:

- RED fonts means over capacity as compared to the maximum limit. (e.g. Production Rate 1.691 tonnes per hour is over capacity as compared to the maximum limit of 1.600 tonnes per hour).
- WHITE fonts means still have opportunities for improvement as compared to the maximum limit.

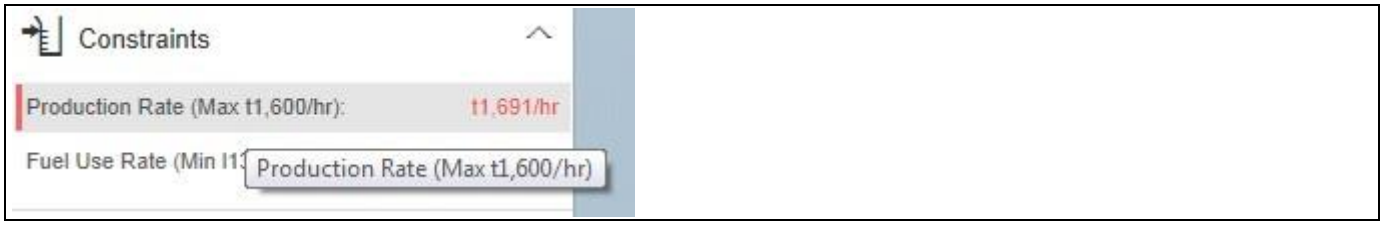


Figure 4.23: Constraints

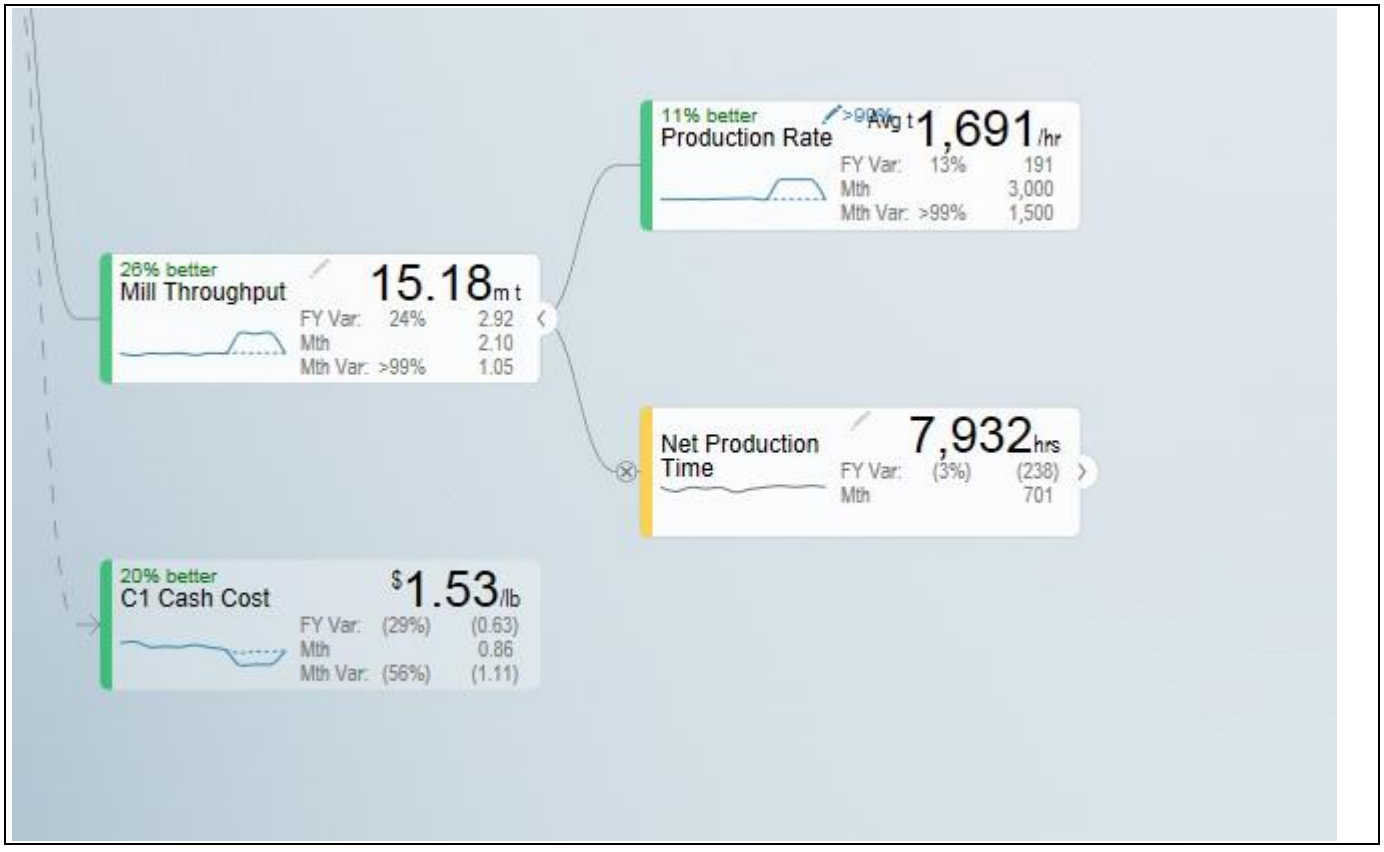


Figure 4.24: Constraints

5 Understanding Variances

The starting point for a simulation is a selection of two versions - **Primary Version** and a **Comparison Version** and a year. For instance a forecast vs current budget.

Once we start simulating, changes are applied to the primary version. We call this changing version the Simulation.

To understand the improvements we achieve, we also keep track of the original values from the primary version without simulation changes. We therefore have 3 data series in the model that we calculate and compare:

1. Simulation (The Primary Version including applied variations)
2. Original (The Primary Version without variations)
3. Target (The Comparison Version)

Based on the above, we calculate the following variances:

- Simulated Variance (Simulation vs Target)

Q: If we changed these things, would we hit target?

- Simulation Impact (Simulation vs Original)

Q: How much would we improve if we made these changes?

- Original Variance (Original vs. Target)

Q: What was our variance if we did nothing?

6 Understanding the Node Widget Information

The Node Widget for a Value Driver displays key information such as value, variance, simulation impacts and trend without having to navigate further.

Figure 6.1: Node Widget Information

7 Conducting what-if analysis

One of the main aspects of the Value Driver Tree is to be able to simulate the impacts changes to key drivers such as prices have on the full year forecast. As we can't change the past, such a simulated change should only be applied to future period. The App have this capability. When performing what-if analysis, the period that a simulation should be applied from is selected in the Side Panel (current period is default):

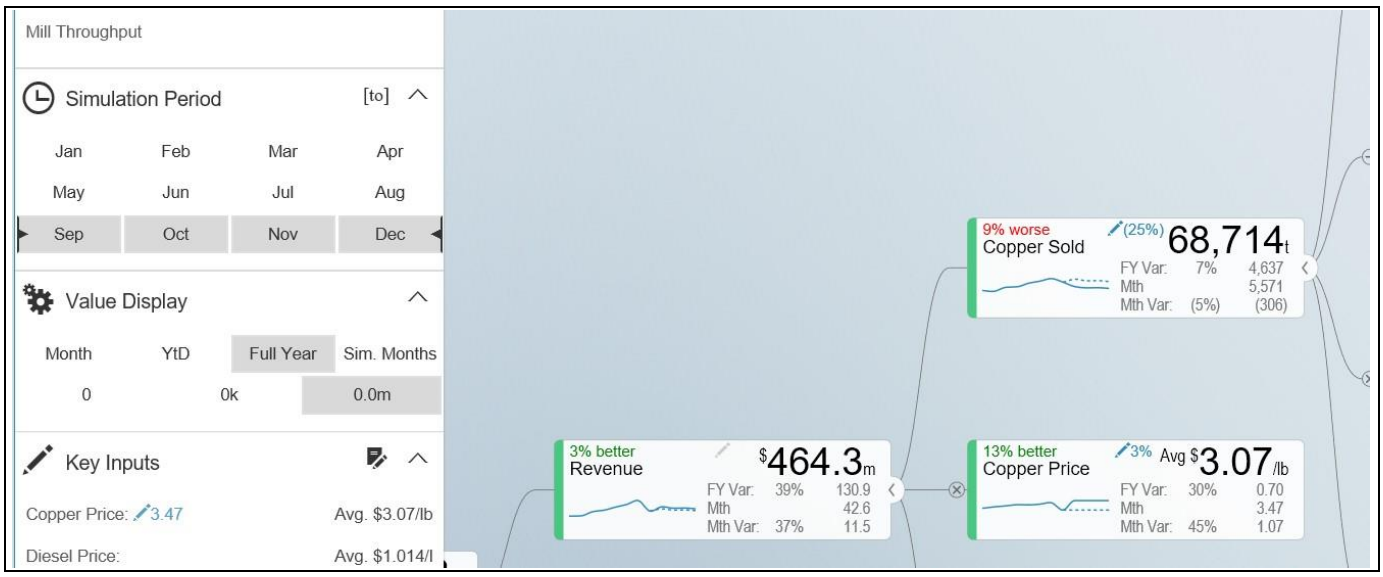


Figure 7.1: What-if-Analysis

Assuming that we calculate fuel cost based on litre per month and price as $/l and we are at the end of period 9, the simulated cost should be calculated first at the monthly level with the % change applied to period 9 and onwards. Once the individual months are calculated, they can then be aggregated based on the rule of the node (sum for diesel cost, weighted average for diesel price):

$$Simulated\ Forecast\ Fuel\ Cost = \sum_{i=1}^{6} Litres_i \times Price_i + \sum_{i=9}^{12} Litres_i \times Price_i \times Price\ Change\%$$

When hovering over a node, the bottom half of the node becomes a slider. Drag the slider left to vary the node by a negatively, drag right for positive. As you drag, the values for the selected node is dynamically recalculated and displayed. When releasing, the tree will immediately recalculate all dependent nodes and show the result.

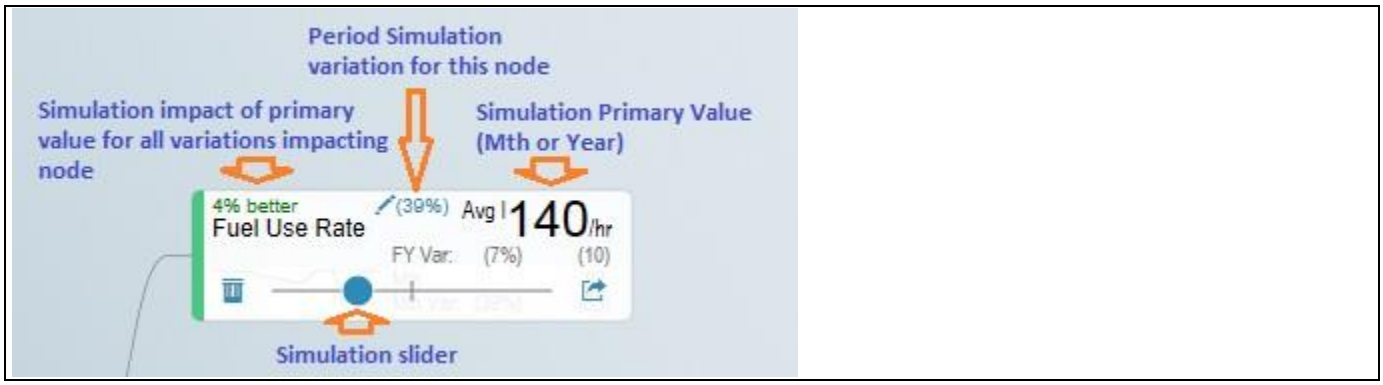


Figure 7.2: Simulation

A simulation change is either a pct. Change, a fixed future price or a pct. Growth depending on the simulation model defined for the value driver.

The default method for the driver can be seen on the simulation tooltip. You can also change the selected model by selecting at the bottom of the tooltip.

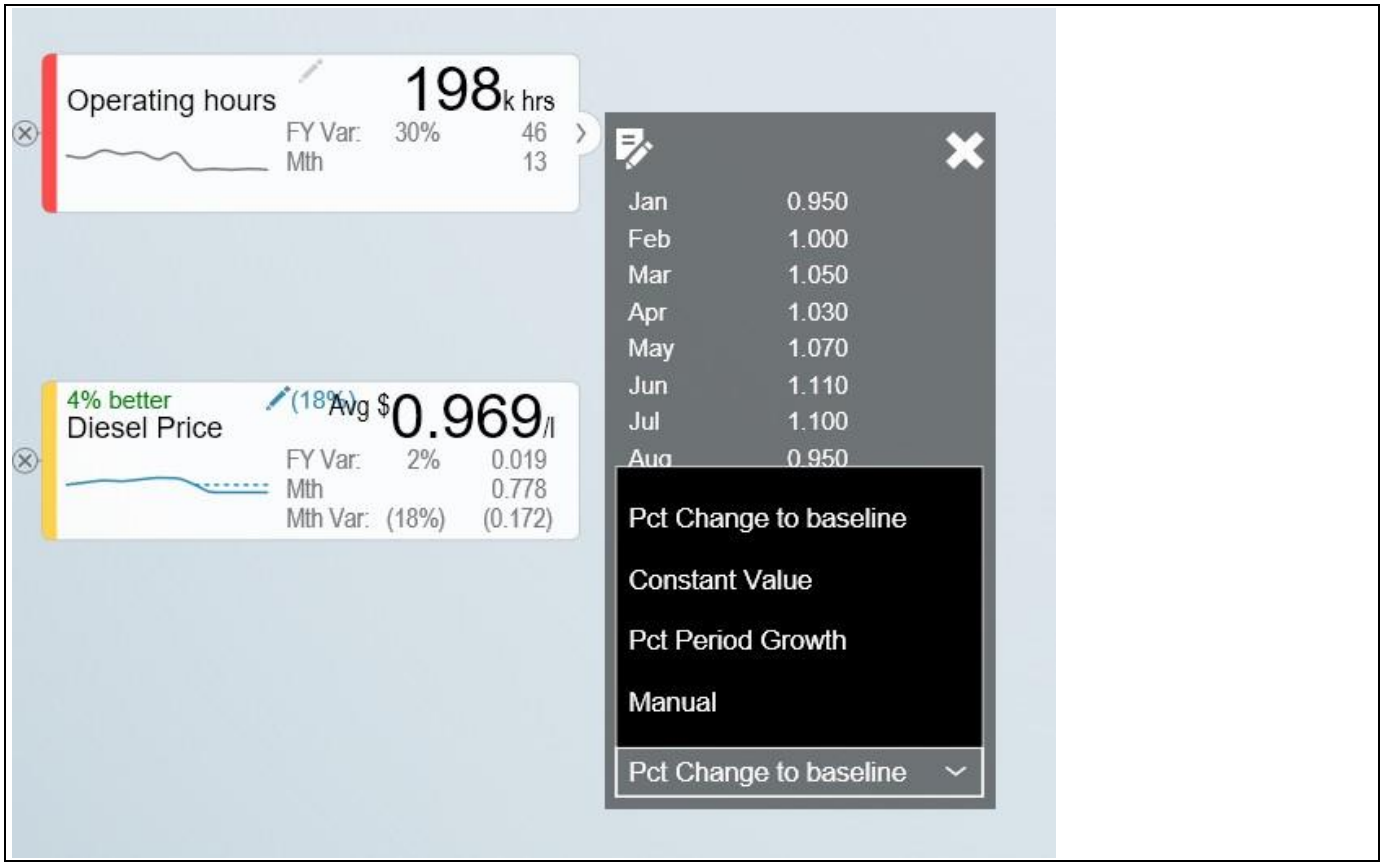


Figure 7.3: Simulation Tooltip

Understanding that a change is applied to the selected periods only is particularly important to understand when displaying the Full Year value and the node is using a weighted average. The displayed value is the weighted average value for the full year, not the value for future periods.

A simplified example will help understanding this critical concept. Let's assume that the diesel price is $1.014 for all periods and the monthly fuel consumption is constant as well at 1m litres and we have selected period 9 as we want to apply the simulation to the remaining periods of the year.

We now drag the slider for the fuel price to the right and the node will show the new weighted average value. Let's say we drag it to the right so the full year weighted average is $1.030. As the first 8 periods were $1.014, the weighted average of $1.030 means that the simulated price change is equivalent of a future fuel price of $1.20. To better understand the individual future period simulations when looking a Full Year aggregated number, it is useful to pay attention to the Month Data also shown on the node. The real world is a little more complex as the price is not necessarily the same every month and the fuel consumed is unlikely to be constant as well. Below is an example of such a simulation based on a real data set:

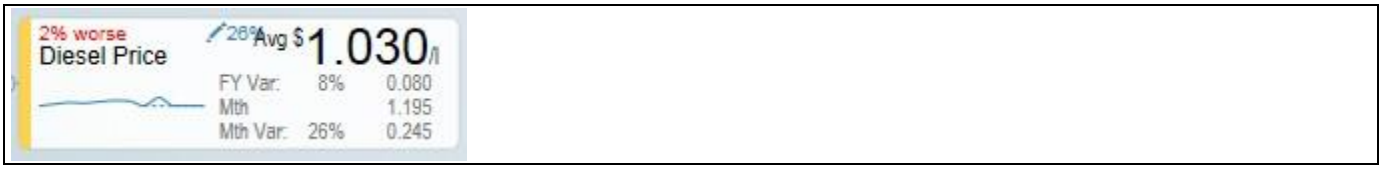


Figure 7.4: Simulation

The weighted average price was 1.014. We now simulated a 2% increase in the prices selected and future periods which equates to a new weighted average price of 1.030. If you look at the month Value, you can see for the selected month, this equates to the higher price of 1.195.

Multiple variations can be applied in parallel as they are expressed as pct. The top left % indicate the cumulative impact on this particular node of all simulation that impacts it.

Any node with a variation will have the blue pen icon in the top middle showing the percentage the node has been varied with. For quick what-if analysis at any level and a dynamic work process from the general to the specific, you can apply a simulation on any level.

Certain nodes may have been locked from changes in the model. This is generally when there is a specific reason not to vary this node directly for consistency purposes. A locked node will not have the grey pen icon or a slider when hovering.

Certain nodes may be displayed multiple times in the tree. An example is Material Moved as it affects Mining and Mill Throughput. In these scenarios, one node is linked to the other. If you simulate a change in a linked node, the node it is pointing to is instead changed so it applies both to the selected node and anywhere else where the driver is used. If a node is a linked node, it has the Linked node text in the bottom right corner:

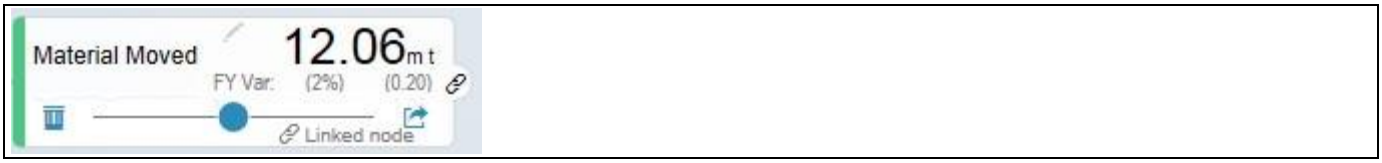


Figure 7.5: Simulation

8 Getting more information about a Value driver

If you click on a node Widget, a detail screen is displayed with additional information on the selected node.

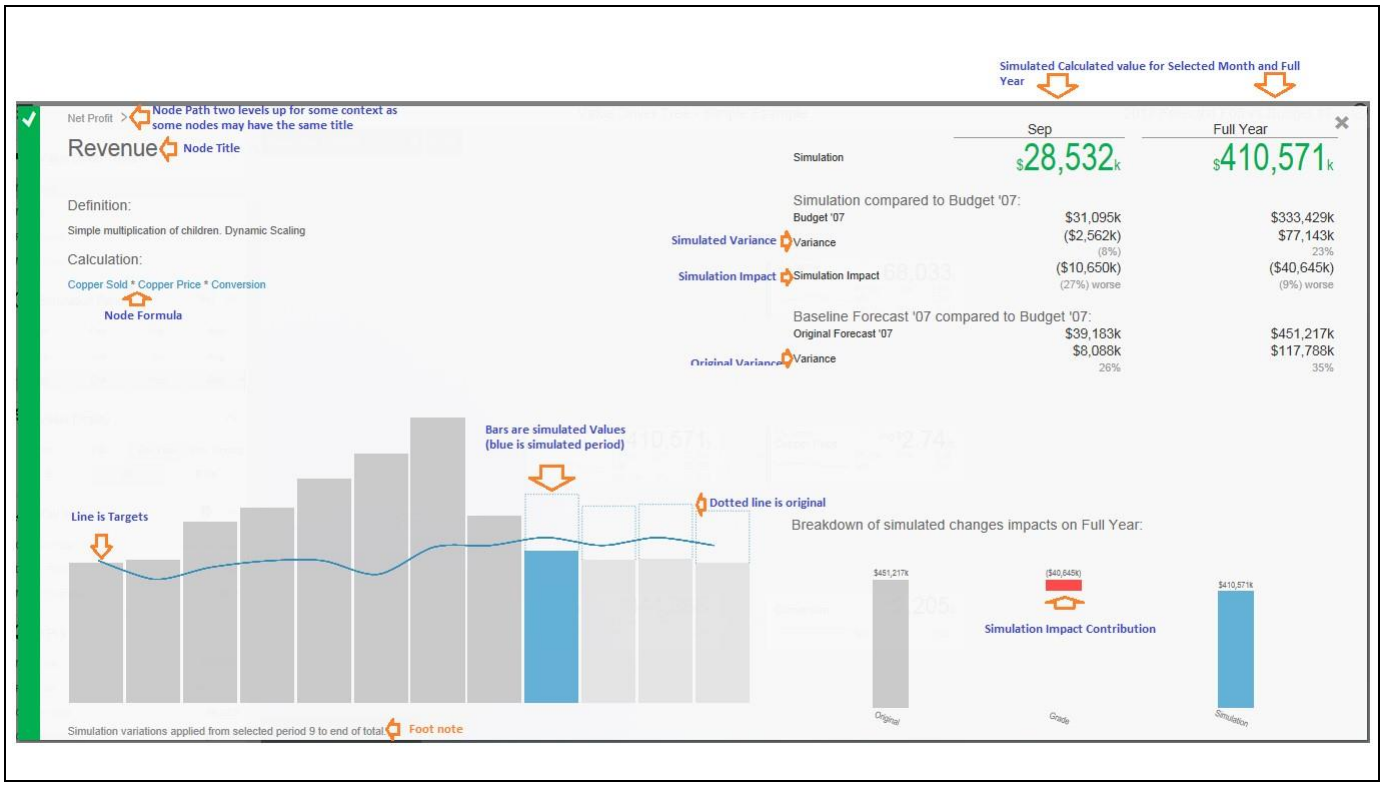


Figure 8.1: Information on Value Driver

For some nodes, the designer may have added a description beyond the title. If so, this will be displayed below the title.

If a node is calculated, a simplified version of the formula is displayed. This is the technical formula that is used to calculate individual period values. To calculate the real values, all simulation variations that affects the node are applied to selected and future periods and the aggregation rules for the node is applied (sum or weighted average).

The trend chart compares the simulated value to the comparison version as well as showing the original value as the dotted line on the bar if simulation variations are active. If you hover over the bar, you will the see the details for the period.

The table on the top right shows the period and full year values and the variance calculations as per the definition previously in this document.

The App also analyses all the active simulations to see if any impacts this particular node. If so, it then analyses how much each of the active simulation changes contributes to the overall impact on this node and generates a waterfall diagram with the break down which is displayed in the bottom right panel.

In some scenarios, you may vary multiple factors impacting the same node in a compound nature. I.e. you vary both sales volume and sales price and want to know the impact on revenue. As the sum of both changes is more than the impact of each individually, the waterfall chart breaks down each of the individual impacts and a compound impact bar.

9 Saving and Opening Scenarios

The standard Value Driver Tree App template contains functionality for saving and opening a scenario collection.

You can save a simulation for later reference by using the save button in the bottom right corner of the Side Panel:

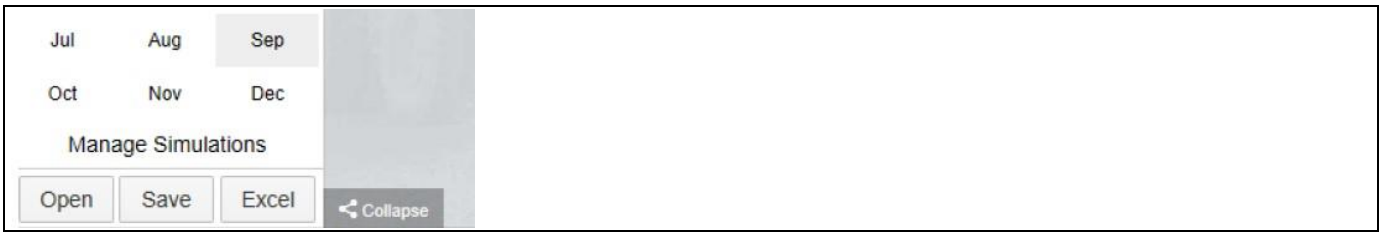


Figure 9.1: Manage Simulations

When clicking the button, you will be asked to provide a unique name.

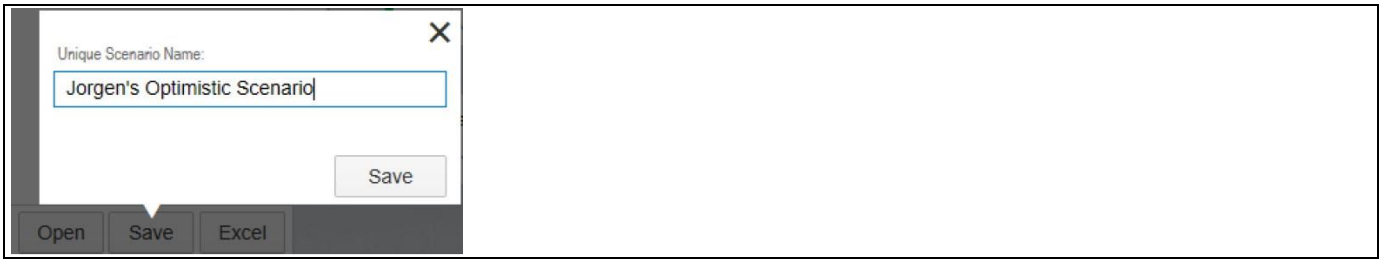


Figure 9.2: Unique Scenario Name

If you choose a name that already exists, you can will be prompted to either overwrite the existing scenario or choose a different name.

You can open existing scenarios by clicking the open button.

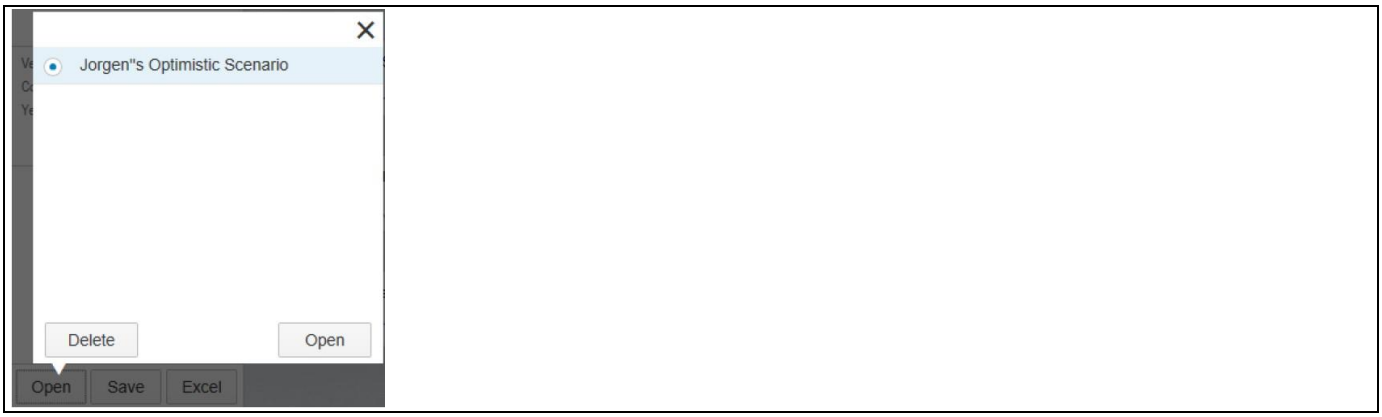


Figure 9.3: Unique Scenario Name

You are also able to delete existing scenario from there.

There is also an option to export the value driver tree data set to Excel. It will create an unformatted spreadsheet with the tree and all the node data.

10 Creating a Value Driver Tree

The steps below explains you on how to create a Value Driver Tree in SAP Lumira Designer using VBX Value Driver Tree Components.

1. In SAP Lumira Designer, navigate to the Outline panel as shown in the below Figure.

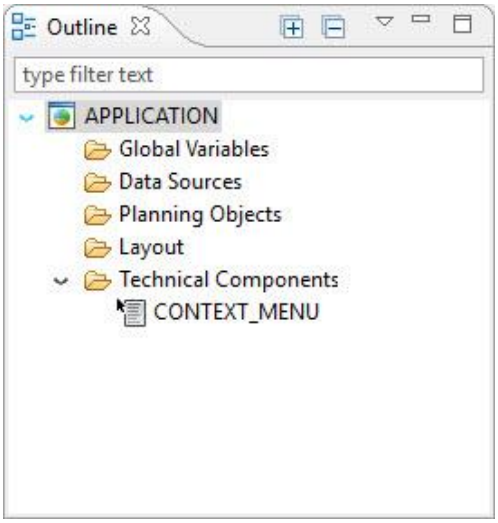


Figure 10.1: Outline Panel

2. Now navigate to the Layout • Create • Value Driver Tree Components • Core Value Driver Tree (see Figure below).

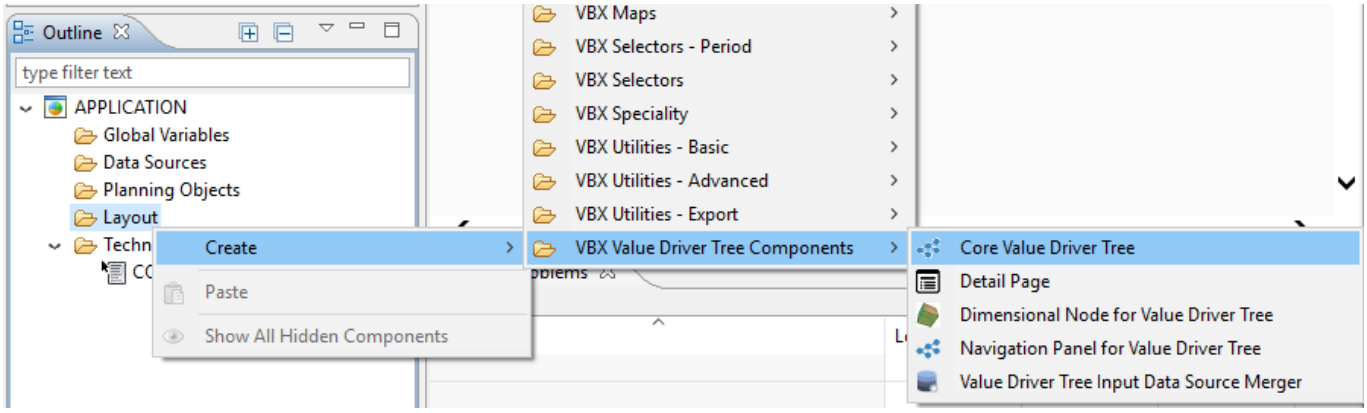


Figure 10.2: Selecting Core Value Driver Tree

3. Now navigate to the Additional properties of the Value Driver Tree. It has four different Tabs namely Tree, Settings, Data and About (see Figure below).

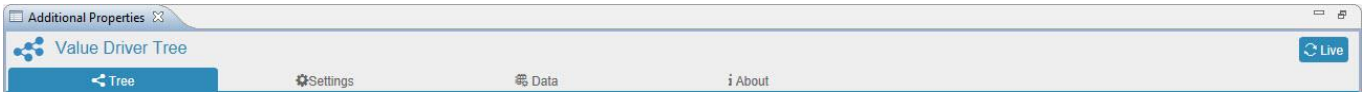


Figure 10.3: Additional Properties Sheet for Value Driver Tree

10.1 Tree Tab

By navigating to the Tree Tab, you will be able to view two different panels one is the Left Panel and the other is the Right Panel. In the Left Panel you will be able to create the Parent Node and the Child Nodes. In the Right Panel, you will be able to configure the properties for the created Parent Node and the Child Nodes.

10.1.1 Tree Tab – Left Panel

The steps below explains on how to create a Parent Node and Child Nodes in the Left Panel of the Tree Tab.

1. For our example, navigate to the left panel of the Tree Tab and you will find 4 different options as shown in the below Figure.

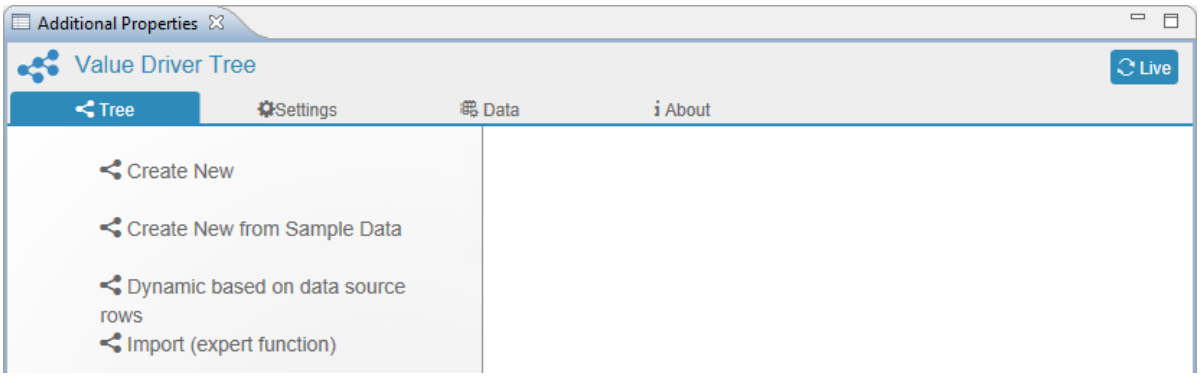


Figure 10.4: Tree Tab in Additional Properties

2. Click the option “Create New” and you will now observe that Node Structures will appear in the left panel and the properties to be configured for the Node will appear in the right panel (see Figure below).

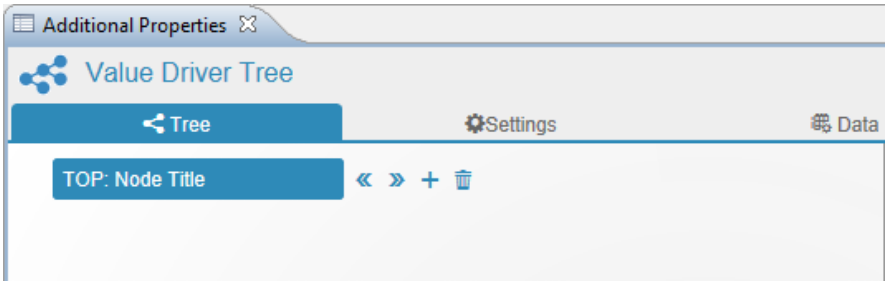


Figure 10.5: Node Configuration

3. In the above Figure you can observe that there is one Parent Node by default. Similarly you will be able to create multiple Child Nodes under the Parent Node. Click the “+” button near the Parent Node in order to create a Child Node. (see Figure below).

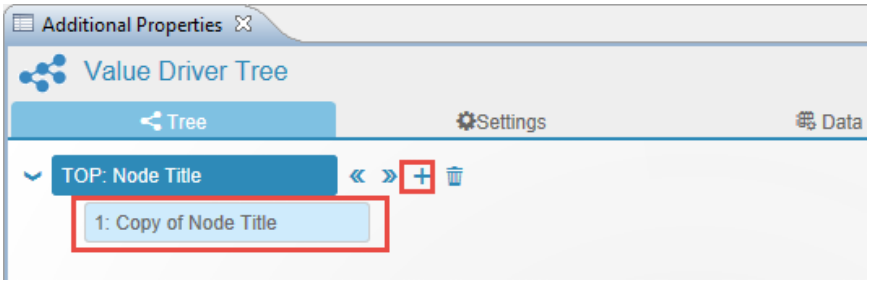


Figure 10.6: Creating a Child Node

4. You can delete the Parent Node and Child Nodes by clicking the Delete button as shown in the below Figure.

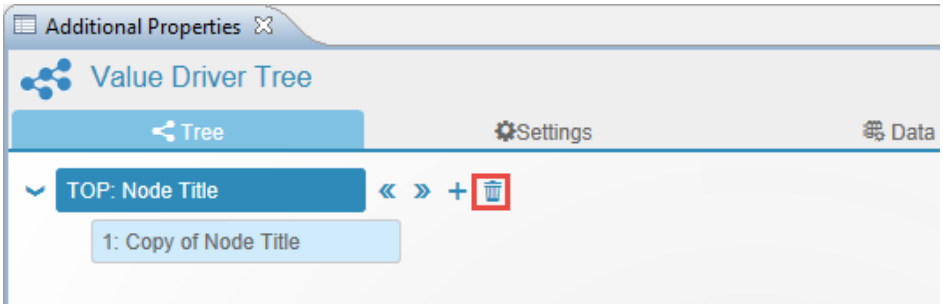


Figure 10.7: Deleting a Parent Node/Child Nodes

5. For our example we have created Node Structure in the left panel as shown in the below Figure.

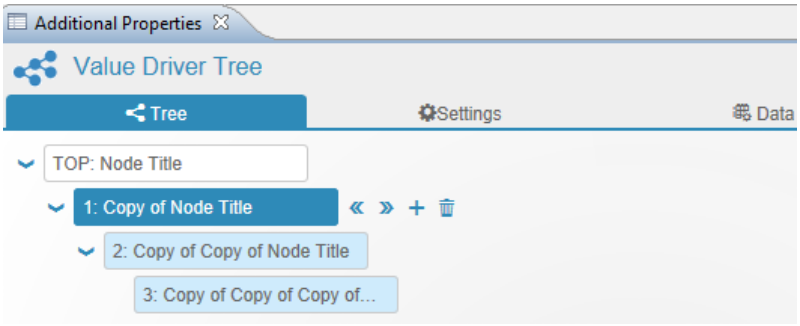


Figure 10.8: A Parent Node having different Child Nodes

6. Now click the "<<" button for the Node "1:Copy of Node Title" as shown in the below Figure.

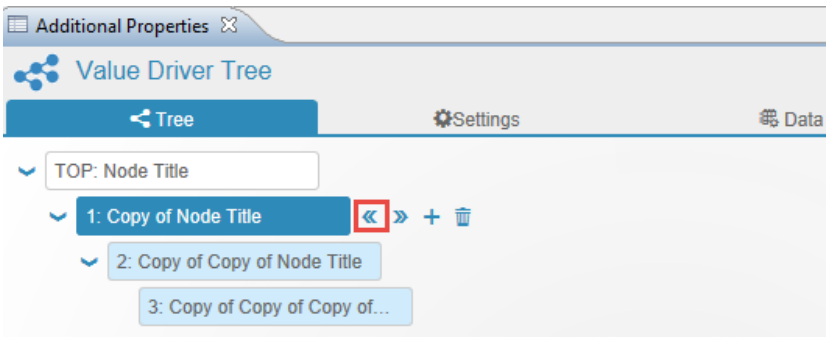


Figure 10.9: Select << button

7. Now you can observe that the Node "1:Copy of Node Title" is moved to the left position hence making a Node Sibling of the Parent Node "TOP:Node Title" (see Figure below).

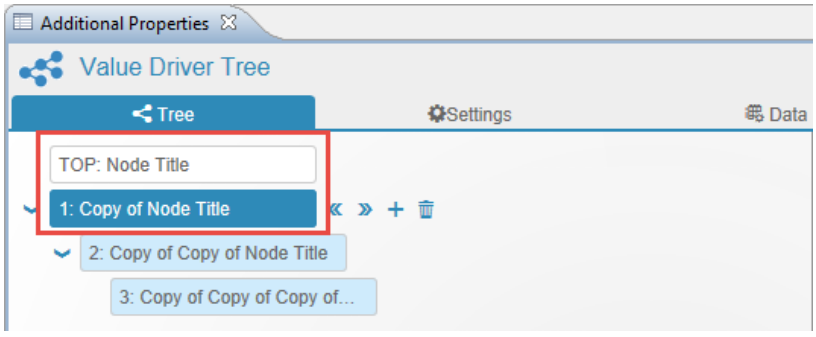


Figure 10.10: Making a Node Sibling of the Parent Node

8. Now click the “>>” button for the Node “1:Copy of Node Title” as shown in the below Figure.

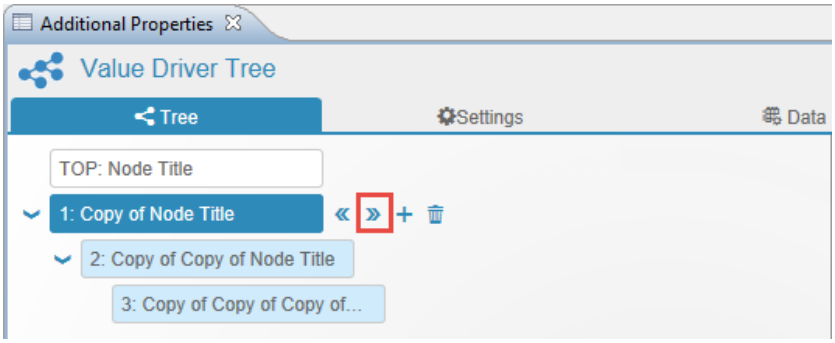


Figure 10.11: Select >> button

9. Now you can observe that the Node “1:Copy of Node Title” is moved to the right position hence making a Node a Child of previous Sibling “TOP:Node Title” (see Figure below).

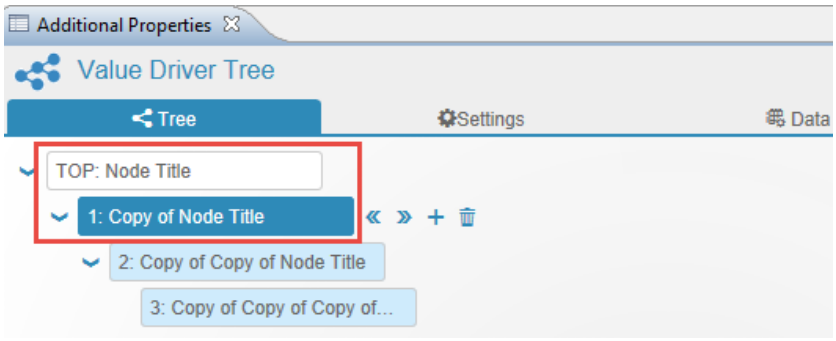


Figure 10.12: Making a Node Child of Previous Node

10.1.2 Tree Tab – Right Panel

In the Right Panel of the Tree Tab, you will be able to configure the properties for the Parent Node and the Child Nodes.

For our example, the steps below explains on how to configure the properties for the Nodes:

1. Now navigate to the right panel where you can view the properties to be configured for the Parent Node.
2. In the area TOP: Node Title, set the property Unique Id to the label TOP.
3. Enter the Title as “Savings” (see Figure below).

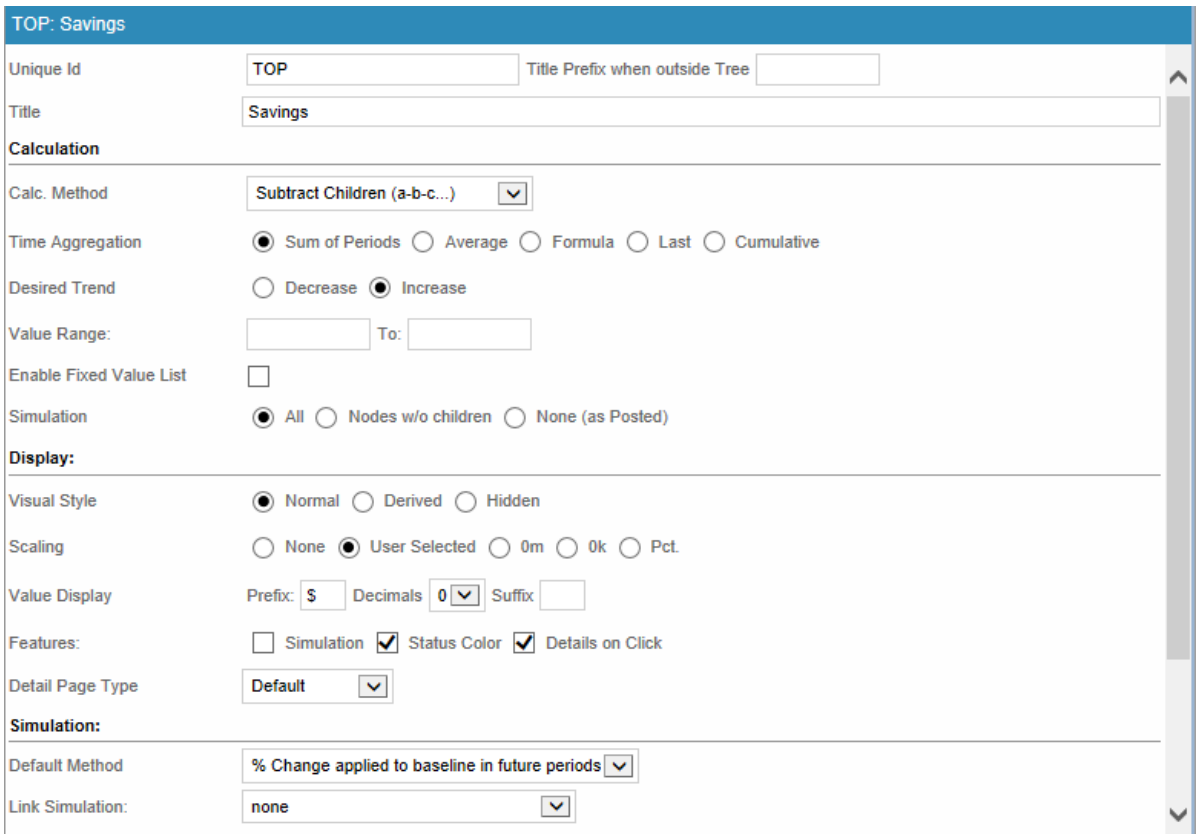


Figure 10.13: Node Properties

4. In the area Calculation, set the Calc. Method to the option Subtract Children (a-b-c…..) as shown in the above Figure .
5. Set the property Time Aggregation to the option Sum of Periods.
6. Set the property Desired Trend to the option Increase.
7. Set the property Simulation to the option All.
8. In the area Display, set the property Visual Style to the option Normal (see Figure above).
9. Set the property Scaling to the option User Selected.
10. For the property Value Display, set the sub property Prefix to the value “$”and set the sub property Decimals to the option “0”.
11. For the property Features, activate the options Status Color and Details on Click.
12. Set the property Detail Page Type to the option Default.
13. In the area Simulation, set the property Default Method to the option “% Change applied to baseline in future periods” (see Figure above).
14. Set the property Link Simulation to the option Default.
15. In the area Business Definition, set the values for the properties Description, Header, Footer and Technical Nodes based on your choice.

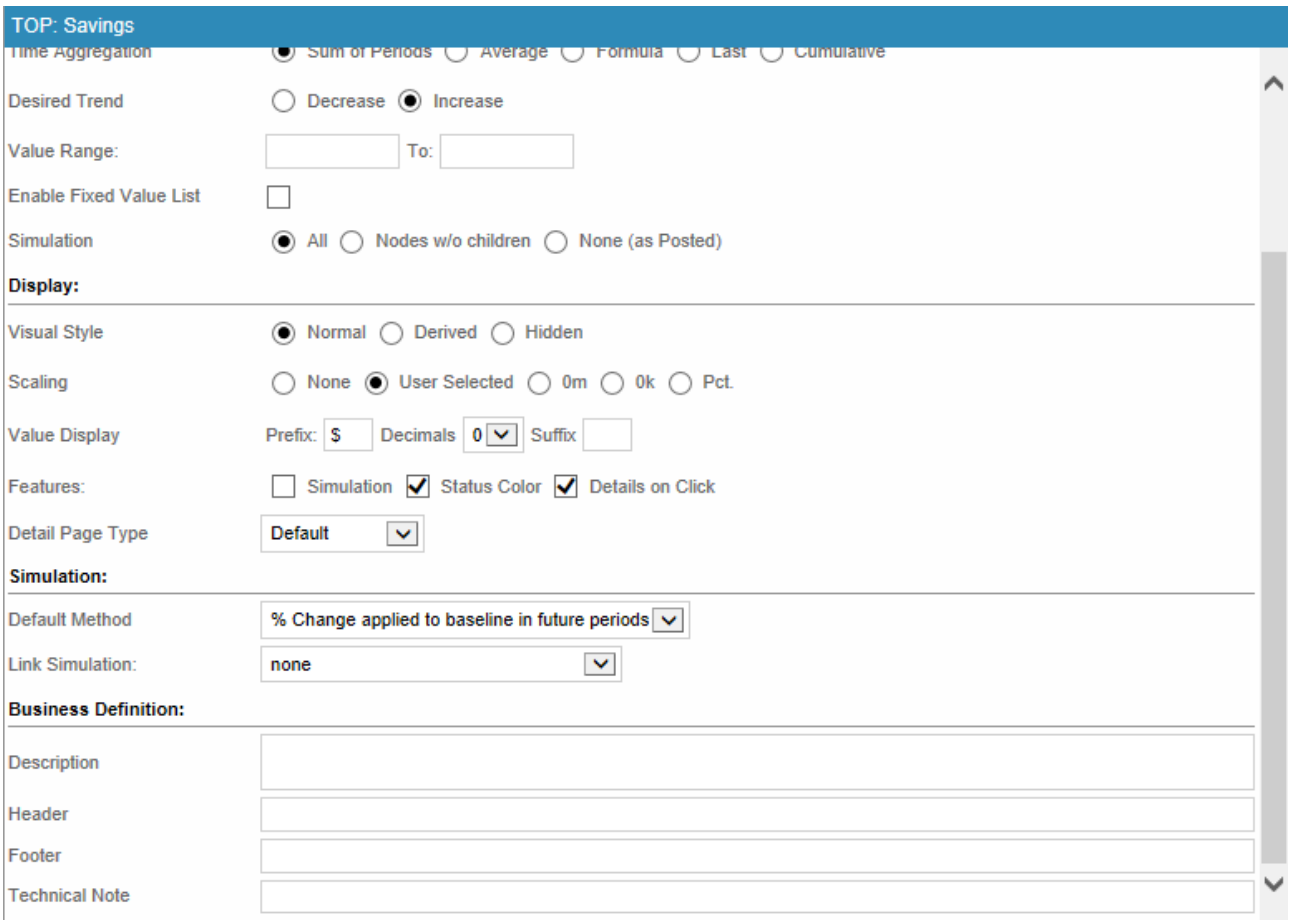


Figure 10.14: Node Properties

16. Based on the above configurations done in the Right Panel, the Parent Node in the Left Panel appears as shown in the below Figure.

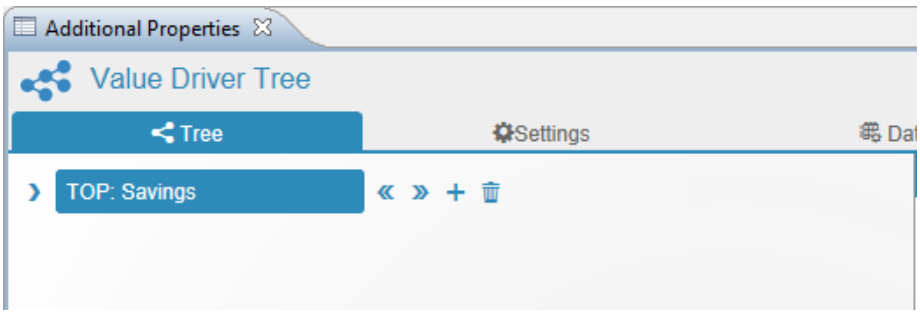


Figure 10.15: Tree with Parent Node

17. Similarly as part of our example, several Child Nodes have been created under the Parent Node with different configurations (see Figure below) and further each Node can be expanded by clicking the “>” button.

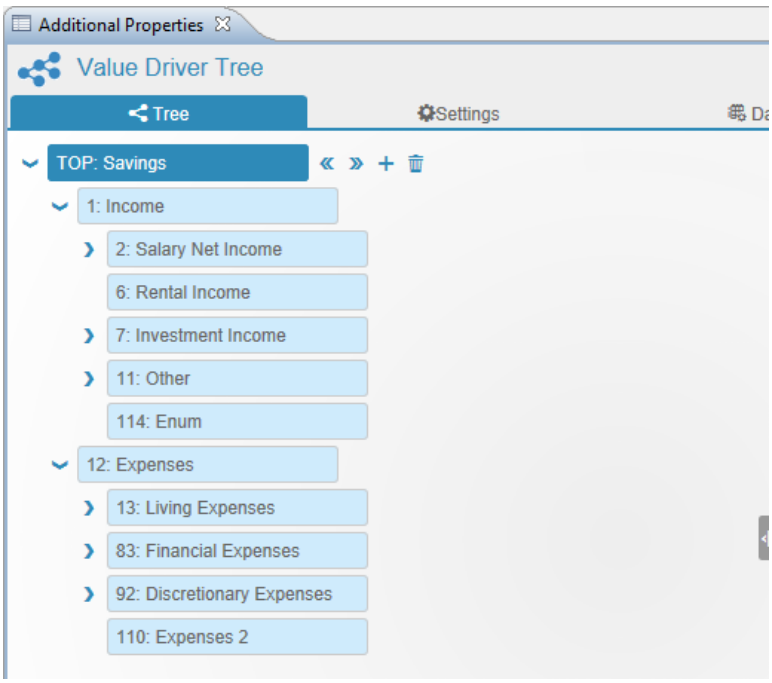


Figure 10.16: Tree created with different Child Nodes

10.2 Templated Nodes

As part of VDT Release 3.0, navigate to the Tab Tree in the Additional Properties of the VDT. You need to select the particular Node and add a Node under the selected Node for creating the Templated Node. Also you can create a Templated Node for the already existing Node. It is very specific that the Templated Nodes should not map the Parent Node.

The steps below explains on how you will be able to create the Templated Nodes:

1. For our example, after navigating to the Tab Tree, select the Node **Expenses** and click "+" to add a Node **Copy of Expenses** under the Node **Expenses** (see Figure below).

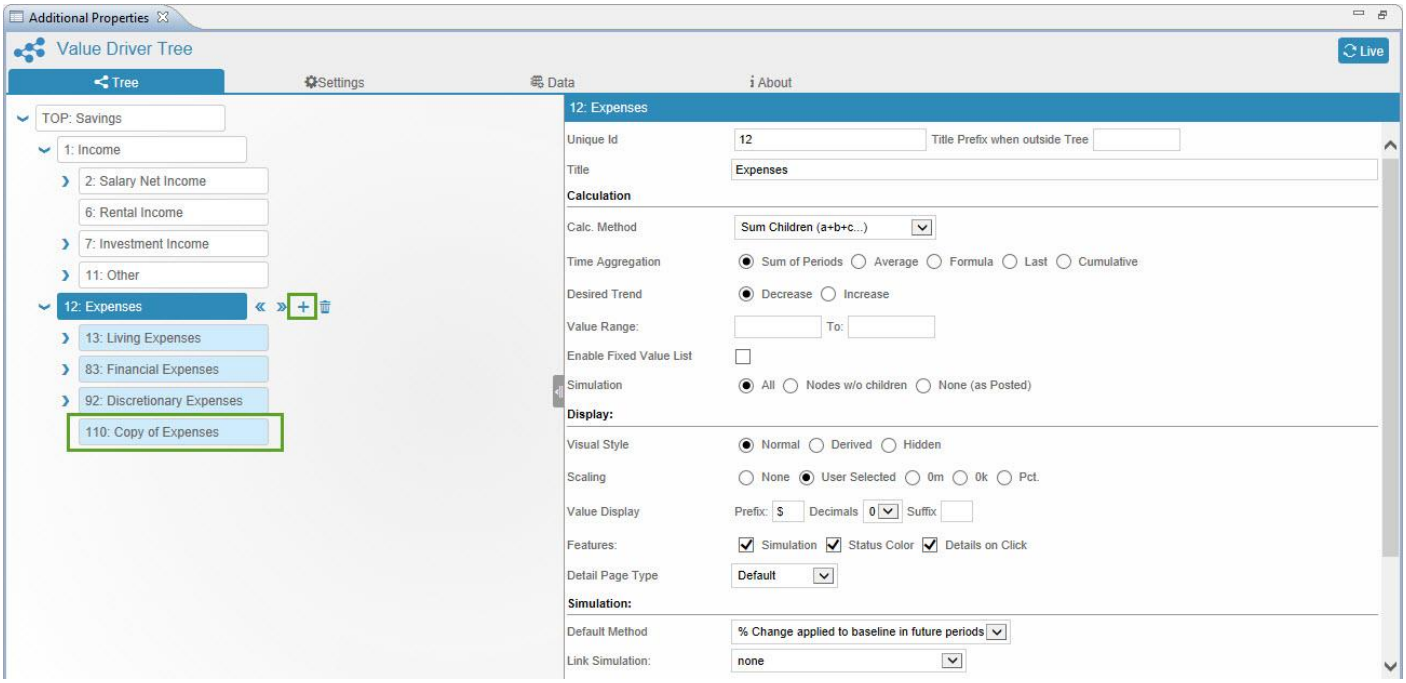


Figure 10.17: Tab Tree

2. For our example, enter the Title for the Node **Copy of Expenses** as Expenses 2 (see Figure below).

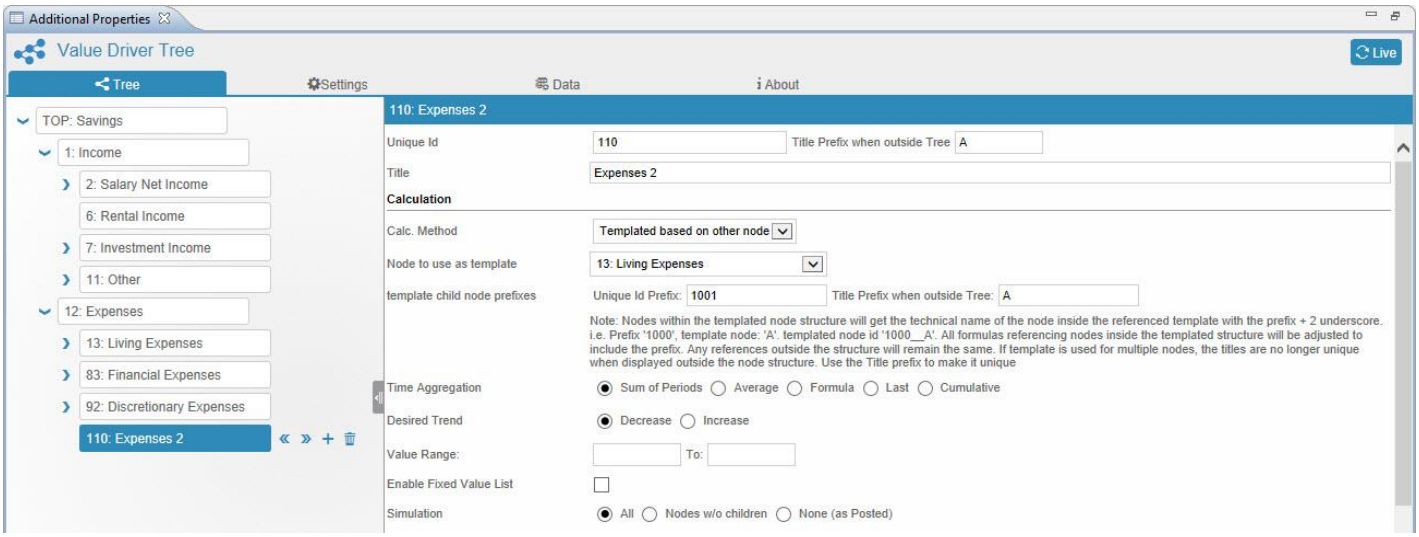


Figure 10.18: Tab Tree

3. In the area Calculation, set the property Calc. Method to the option Template based on the other Node.
4. Set the property Node to use as Template to the option 13: Living Expenses where all its child Nodes will get added into the Node Expenses 2.
5. Enter the details for the properties Template Child Node Prefixes and Title Prefix when outside Tree based on your choice.
6. Based on the above configuration, you will be able to view the Templated Node Expenses 2 having all its Nodes similar to the Living Expenses Node as per our example.

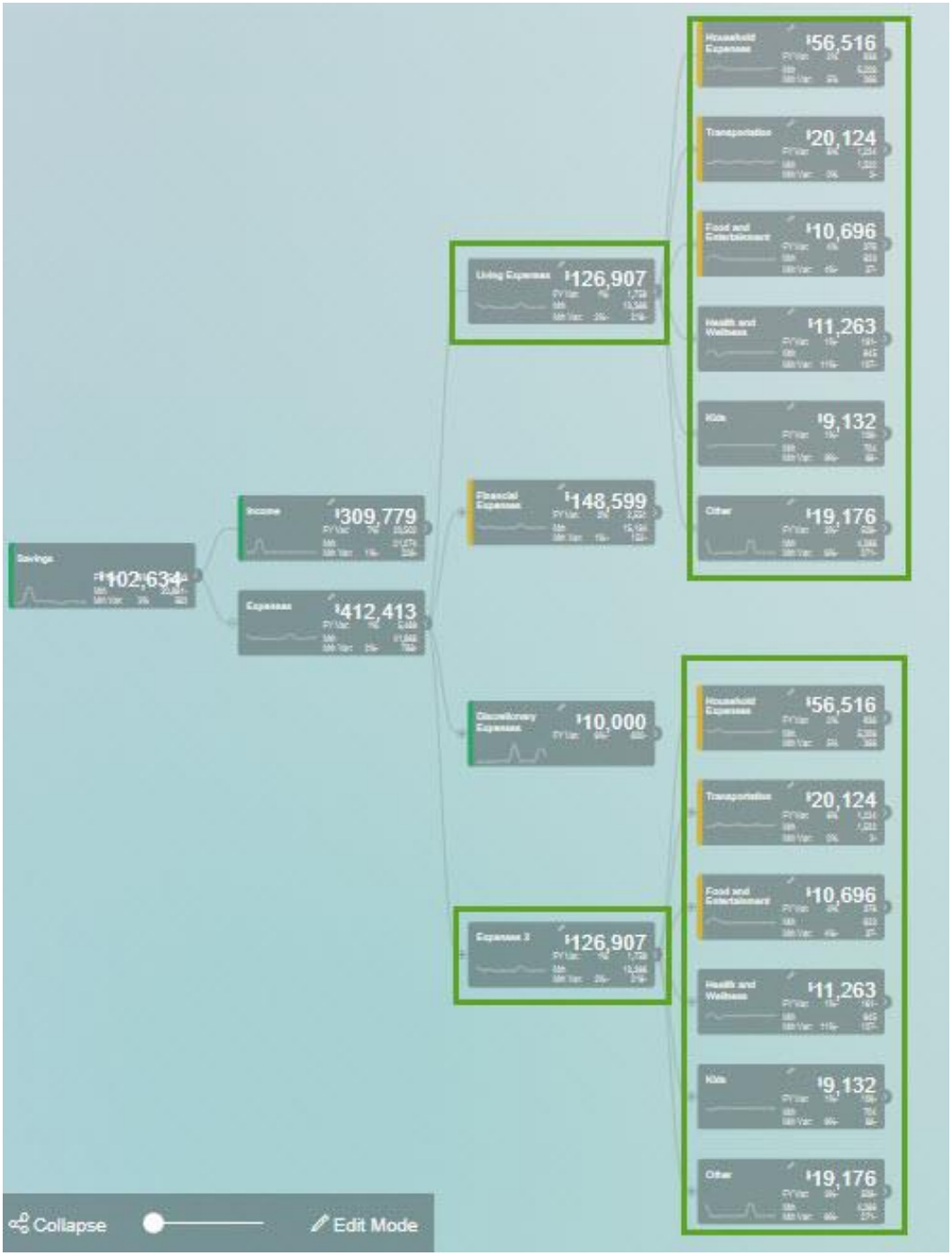


Figure 10.19: Templated Node at Runtime

Template Child Node Prefixes
Nodes within the templated node structure will get the technical name of the node inside the referenced template with the prefix + 2 underscore. i.e. Prefix '1000', template node: 'A'. templated node id '1000__A'. All formulas referencing nodes inside the templated structure will be adjusted to include the prefix. Any references outside the structure will remain the same. If template is used for multiple nodes, the titles are no longer unique when displayed outside the node structure. Use the Title prefix to make it unique.

10.3 Enable Fixed Value List (Enum option)

As part of VDT Release 3.0, navigate to the Tab Tree in the Additional Properties of the VDT. Here you will have the option to use Enum option which contains a set of named constants and those named constants can be represented in the Node.

The steps below explains on how you will be able to use Enum option in VDT:

1. For our example, after navigating to the Tab Tree, select the Node **Income** and click "+" to add a Node **Copy of Income** under the Node **Income** as shown in the below Figure.

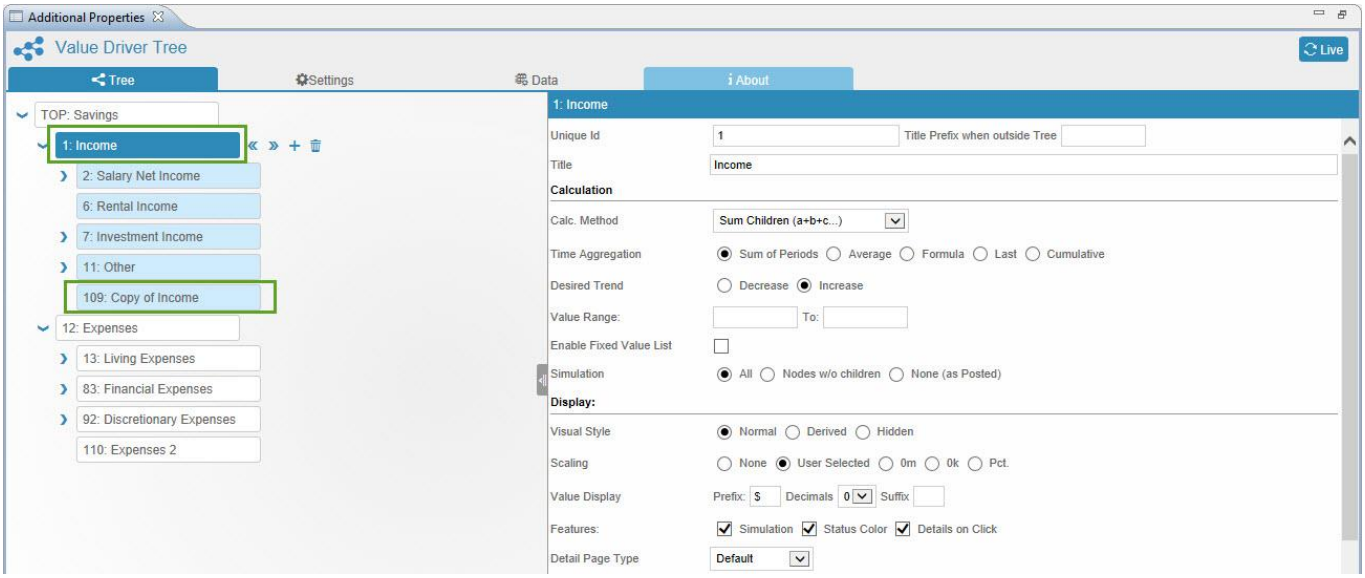


Figure 10.20: Tab Tree

2. Now click the Node **Copy of Income.** Enter the Title of the Node as "Enum" (see Figure below).

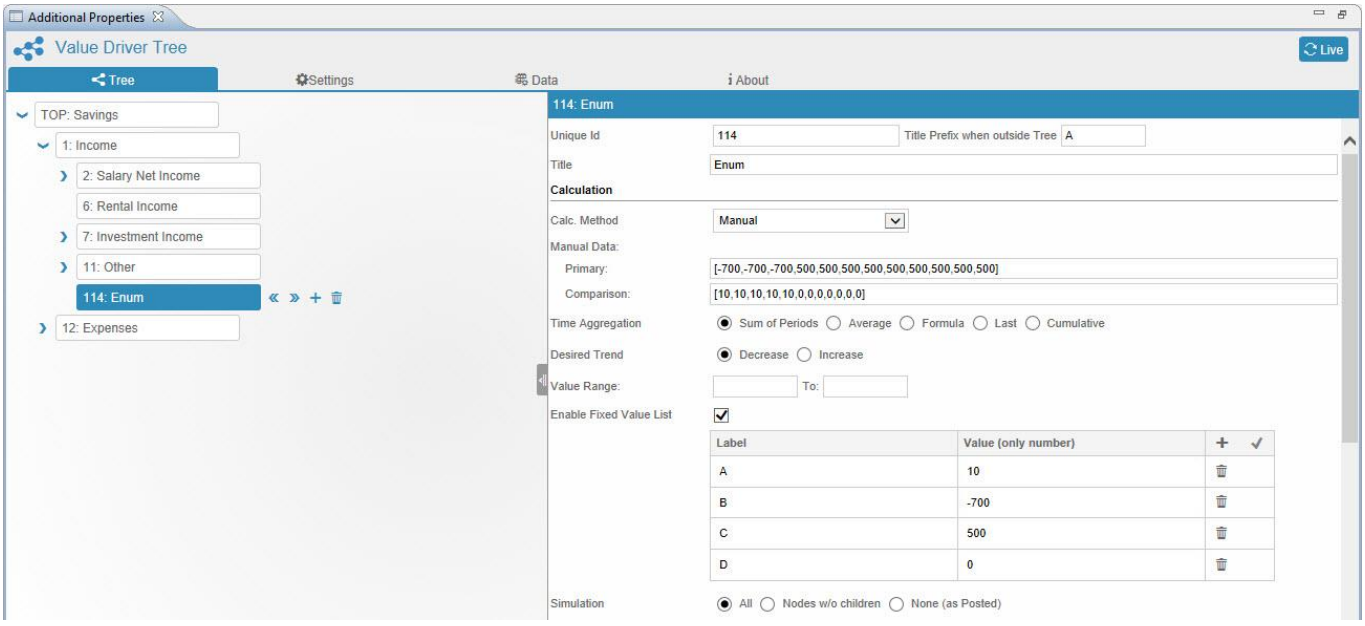


Figure 10.21: Primary and Comparison Data

3. For our example, set the property Calc. Method to the option Manual (see Figure above).

4. Set the values for the Primary Data as [-700,-700,-700,500,500,500,500,500,500,500,500,500]. In our example, we have set the Simulation Period as 12 months and we do have 12 values for the Primary Data.
5. Set the vales for the Comparison Data as [10,10,10,10,10,0,0,0,0,0,0,0]. In this case too we have 12 values for the Comparison Data.
6. Now activate the property Enable Fixed Value List.
7. Set the Label for the values as shown in the below table. It is to be noted that the labels should be only assigned to both Primary and Comparison Data values.

Label	Value (ony Number)
A	10
B	-700
C	500
D	0

Figure 10.22: Label and their assigned values

8. Now based on the above configuration you will able to view the Tree in the Runtime as shown below.

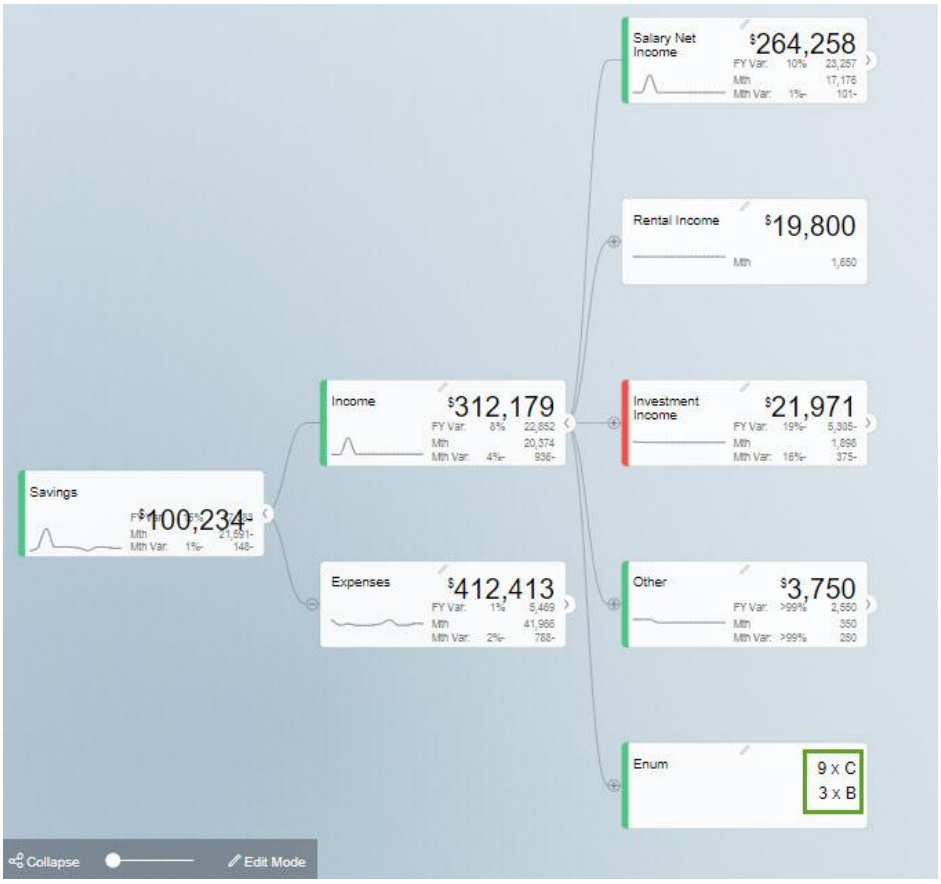


Figure 10.23: Label values in Runtime

9. From the above Figure you can observe that the primary data value -700 occurs thrice and it is labelled as 3 x B and the data value 500 occurs nine times and it is denoted as 9 X C. You can also select the specific labels from the drop down at the Node Level (see Figure below).

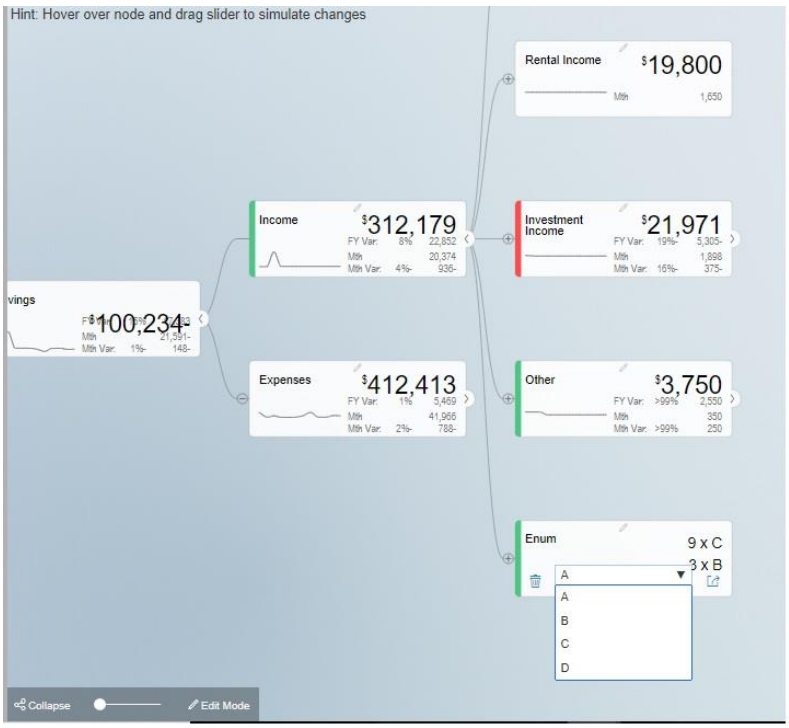


Figure 10.24: Select Labels from the Node

10. You can edit the Label values for the simulation period of 12 months using the options namely Constant and Manual. For editing the Label values in Constant option, click the icon as shown in the below Figure.

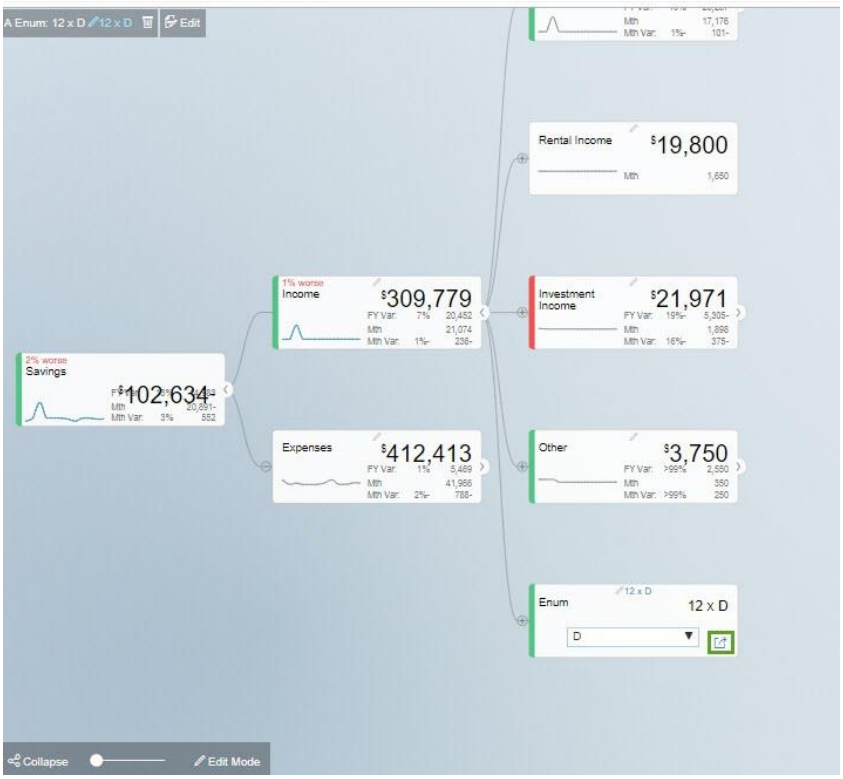


Figure 10.25: Edit Label values using Constant option

11. When the icon is clicked, it leads to the pop up window as shown below. Now select the Constant Value option from the drop down as shown in the below Figure.

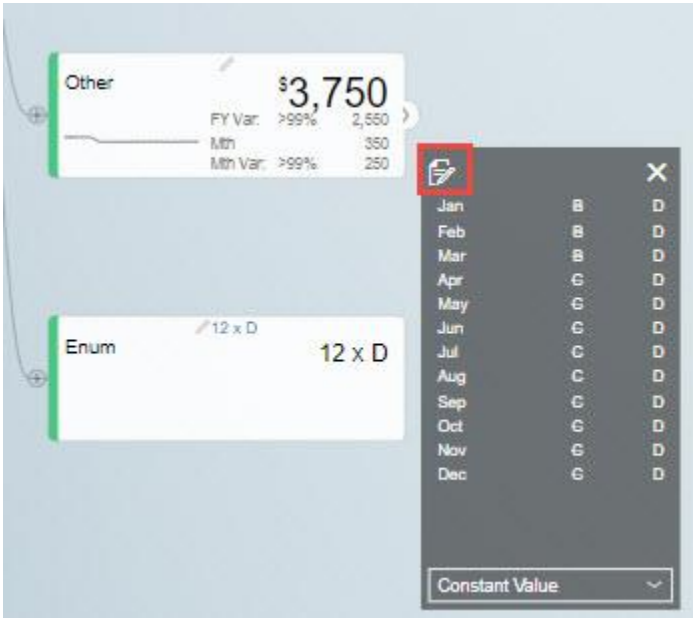


Figure 10.26: Pop Window with Constant Value option selected for Edit

12. Now click the Edit option as shown in the above Figure which will lead you to the Table Visualization page (see Figure below).

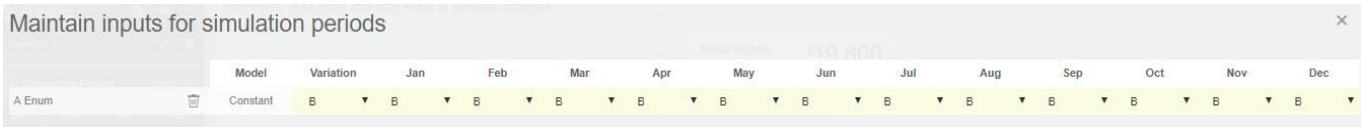


Figure 10.27: Table Visuals with Constant option

13. Here the constant value is selected as “B” and you can observe that all the Label values for the 12 months have been changed to the Label “B”.
14. For editing the Label values in Manual option, select the Manual option from the drop down in the pop up window as shown in the below Figure.

Figure 10.28: Pop Window with Manual option selected for Edit

After selecting the Manual option it will lead you to the Node **Enum** with Edit option (see Figure below).

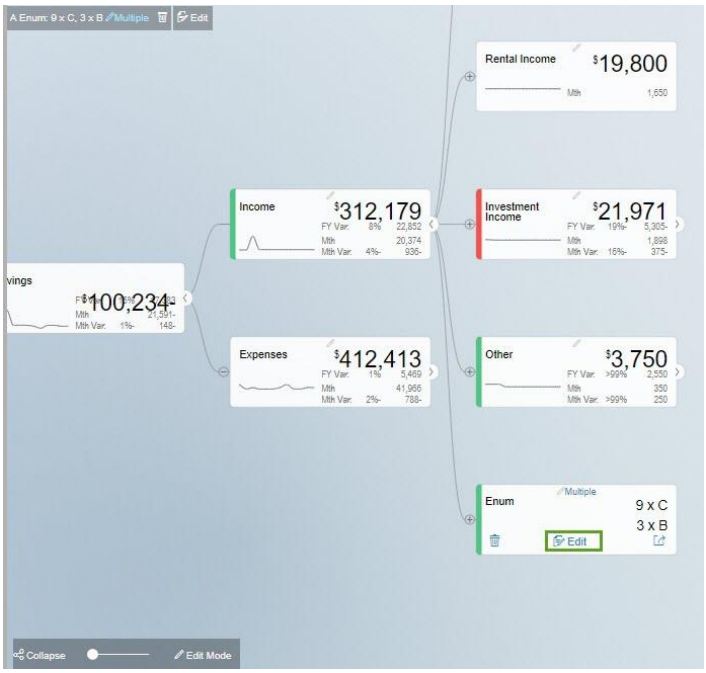


Figure 10.29: Enum Node with Edit option

15. Now click the Edit option as shown in the above Figure which will lead you to the Table Visualization page (see Figure below).

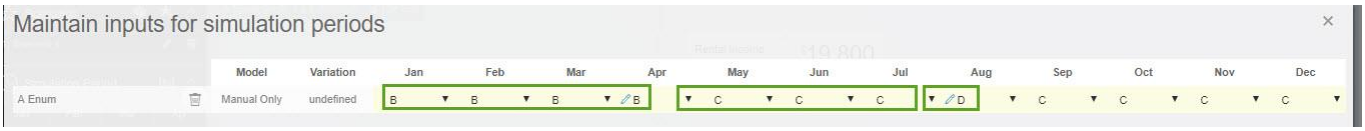


Figure 10.30: Table Visuals with Manual option

16. Here you will be able to edit the Label values individually for each month based on your need (see Figure above).

10.4 Dimensional Node Component

As part of VDT Release 3.0, as a Dimensional Node Component, now you have the option to assign the Data Source at the Node Level for further simulation.

The steps below explains on how you will be able to create the Dimensional Node Component in a Tree:

1. For our example, create a layout in SAP Lumira Designer as shown in the below Figure.

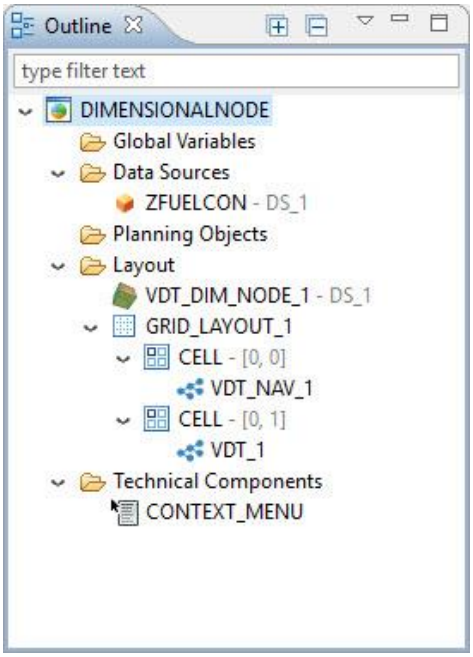


Figure 10.31: Layout

2. In our example, you can observe that in a grid layout cell (0,1) has been assigned with the Value Driver Tree and cell (0,0) has been assigned with the Navigation Panel for the Value Driver Tree.
3. A Dimensional Node needs to be assigned at the top of the Grid Layout as shown in the above Figure.
4. Data Source DS_1 has been assigned to the Dimensional Node.
5. In our example, navigate to the Node Revenue for which the Dimensional Node has been assigned (see Figure below).

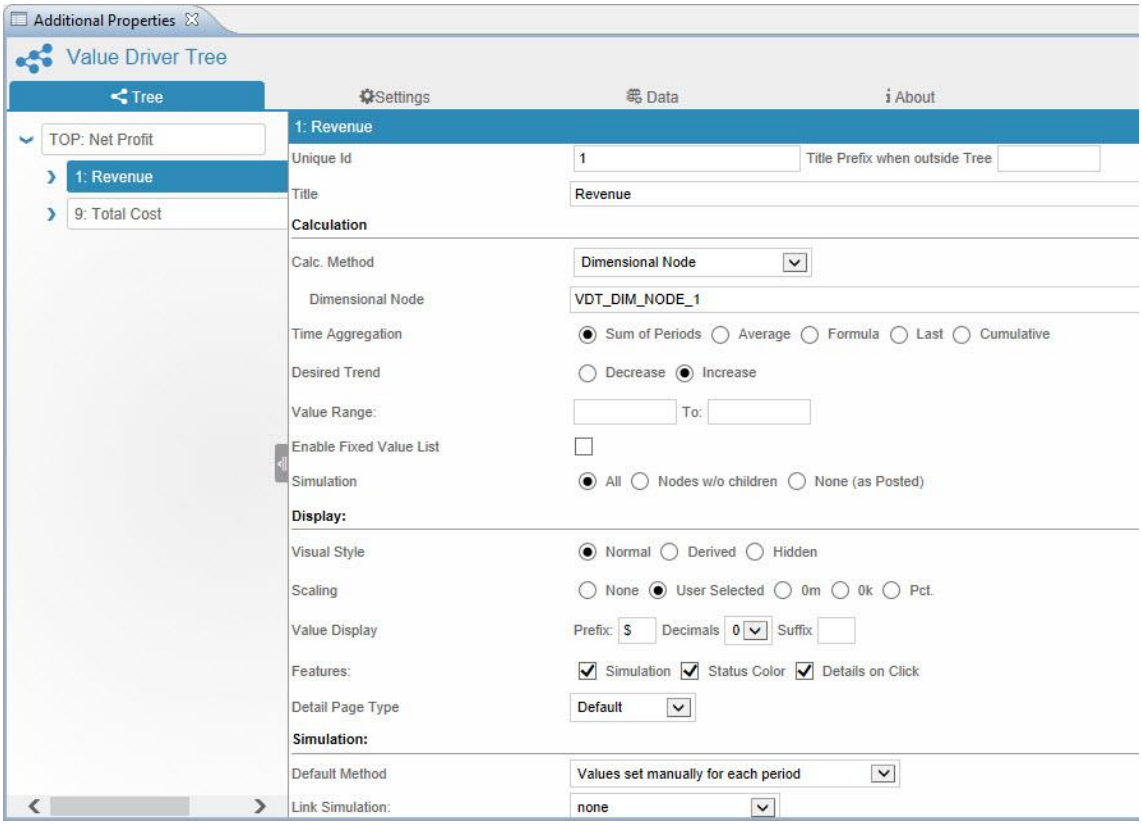


Figure 10.32: Tab Tree

6. In the area Calculation, set the property Calc. Method to the option Dimensional Node.
7. Set the property Dimensional Node to the text VDT_DIM_NODE_1.
8. In the area Simulation, set the property Default Method to the option Values set manually for each period. The other options are
    - % Change applied to baseline in future periods.
    - Constant Value for future periods.
    - % Growth applied to baseline in future periods.
9. Based on the above configuration, you will be able to observe the grid icon above the Node Revenue in the Runtime (see Figure below).

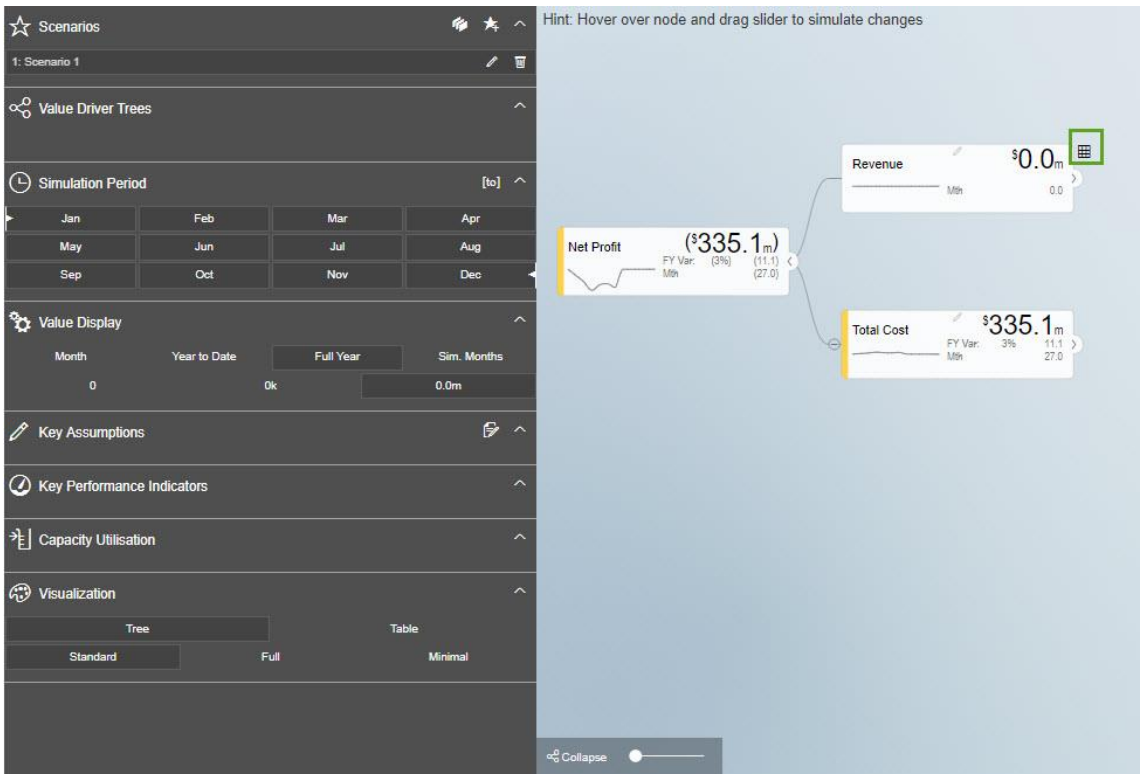


Figure 10.33: Revenue Node with Dimensional Node option

10.5 Additional Properties of Tree Tab

Area	Property	Description
TOP	Unique Id	This property sets the Unique Id for the Node.
	Title Prefix when outside Tree	This property sets the Title Prefix for the Node when outside Tree.
	Title	This property sets the Title for the Node.
Calculation	Calc. Method	This property sets the Calculation Method for the Node. The options are Data Source, Sum Children (a+b+c….), Subtract Children (a-b-c….), Multiply Children (a*b*c….), Divide Children (a/b/c….), Formula. Linked to Node, Dimensional Node, Template based on other Node and Manual. **Note:** When the Calc. Method is selected as Linked to Node option, then you can select the desired Node from the Nodes List.
	Source key	This property sets the Source key for the Node. If Auto-mapped, only fill if id in source differ from node id.
	Formula	This property sets the Formula for the Node.
	Dimensional Node	This property sets the Dimensional Node.
	Node to use as template	Using this property, you can select the desired Node from the Nodes List to assign it as a Template.
	Template child node prefixes - Unique Id Prefix	This property sets the Unique Id Prefix for the Template child node.
	Template child node prefixes - Title Prefix when outside Tree	This property sets the Title Prefix for the Template child node when outside the Tree.
	Manual Data - Primary	This property sets the Primary Data for the Node.
	Manual Data - Comparison	This property sets the Comparison Data for the Node.
	Time Aggregation	This property sets the Time Aggregation for the Node. The options are Sum of Periods, Average, Formula, Last and Cumulative.
	Desired Trend	This property sets the Desired Trend for the Node. The options are Decrease and Increase.
	Value Range	This property sets the Value Range for the Node.
	Enable Fixed Value List	This property enables/disables the Enable Fixed Value List option.
	Simulation	This property sets the Simulation for the Node. The options are All, Nodes without Children and None.
Display	Visual Style	This property sets the Visual Style for the Node. The options are Normal, Derived and Hidden.
	Scaling	This property sets the Scaling for the Node. The options are None, User Selected, 0m, 0k and Pct.
	Value Display	This property sets the Value Display for the Node. The options are Prefix,

Area	Property	Description
		Decimals starting from 0 to 7 and Suffix.
	Features	This property sets the Features for the Node. The options are Simulation, Status Color and Details on Click.
	Detail Page Type	This property sets the Detail Page Type for the Node. The options are Default and Node Specific.
Simulation	Default Method	This property sets the Default Simulation Method for the Node. The options are % Change applied to baseline in future periods, Constant value for future periods, % Growth applied to baseline in future periods and Value set manually for each period.
	Link Simulation	Using this property, you can select the desired Node from the Nodes List for Link Simulation.
Business Definition	Description	This property sets the Description for the Node.
	Header	This property sets the Header for the Node.
	Footer	This property sets the Footer for the Node.
	Technical Note	This property sets the Technical Note for the Node.

Table 10.1: Tree Tab

10.6 Settings Tab

In the Settings Tab, you will be able to configure the properties for the entire Tree.

The steps below explains on how to configure the properties for the entire Tree created with Parent Node and Child Nodes:

1. For our example, navigate to the right panel of the Settings Tab where you can view the properties to be configured for the entire Tree (see Figure below).

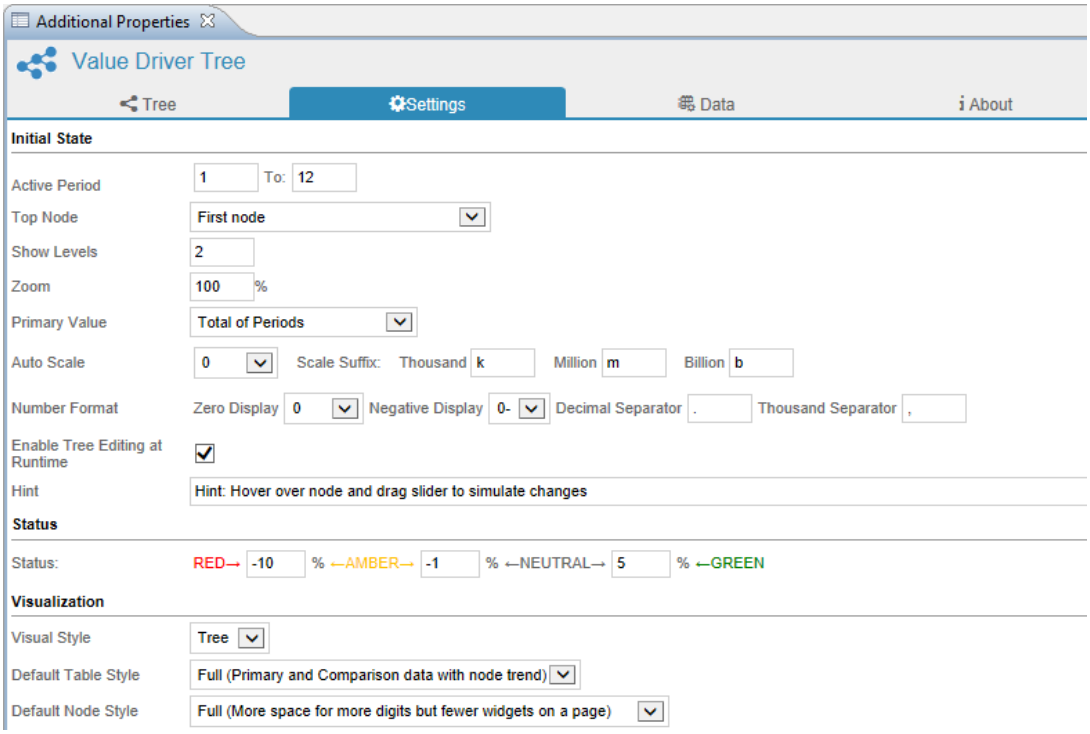


Figure 10.34: Settings Tab – Initial State, Status and Visualization

2. In the area Initial State, set the property Active Period to the values from 1 to 12 (see Figure above).
3. Set the property Top Node to the option First Node.
4. Set the property Show Levels to the value 2.
5. Set the property Zoom to the value 100%.
6. Set the property Primary Value to the option Total of Periods.
7. For the property Auto Scale, set the sub property Zero Display to the option 0, set the sub property Negative Display to the option 0-, set the sub property Decimal Separator to "." and set the Thousand Separator to ",".
8. Activate the property Enable Tree Editing at Runtime.
9. Enter the Hint Notes for the property Hint.
10. In the area Status and for the property Status, set the sub property RED to the value "-10%", set the sub property AMBER to the value "-1%", set the sub property GREEN to the value "5%" (see Figure above).
11. In the area Visualization, set the property Visual Style to the option Tree. The other option is Table.
12. For our example, set the property Default Node Style to the option "Full (More space for more digits but fewer widgets on a page)".

Visual Style

When the property Visual Style is set to the option Tree, you can select the property Default Node Style to the option based on your choice. When the property Visual Style is set to the option Table, you can select the property Default Table Style to the option based on your choice.

13. In the area Driver Widget Layout, set the property Status Bar to the option Variance (see Figure below).

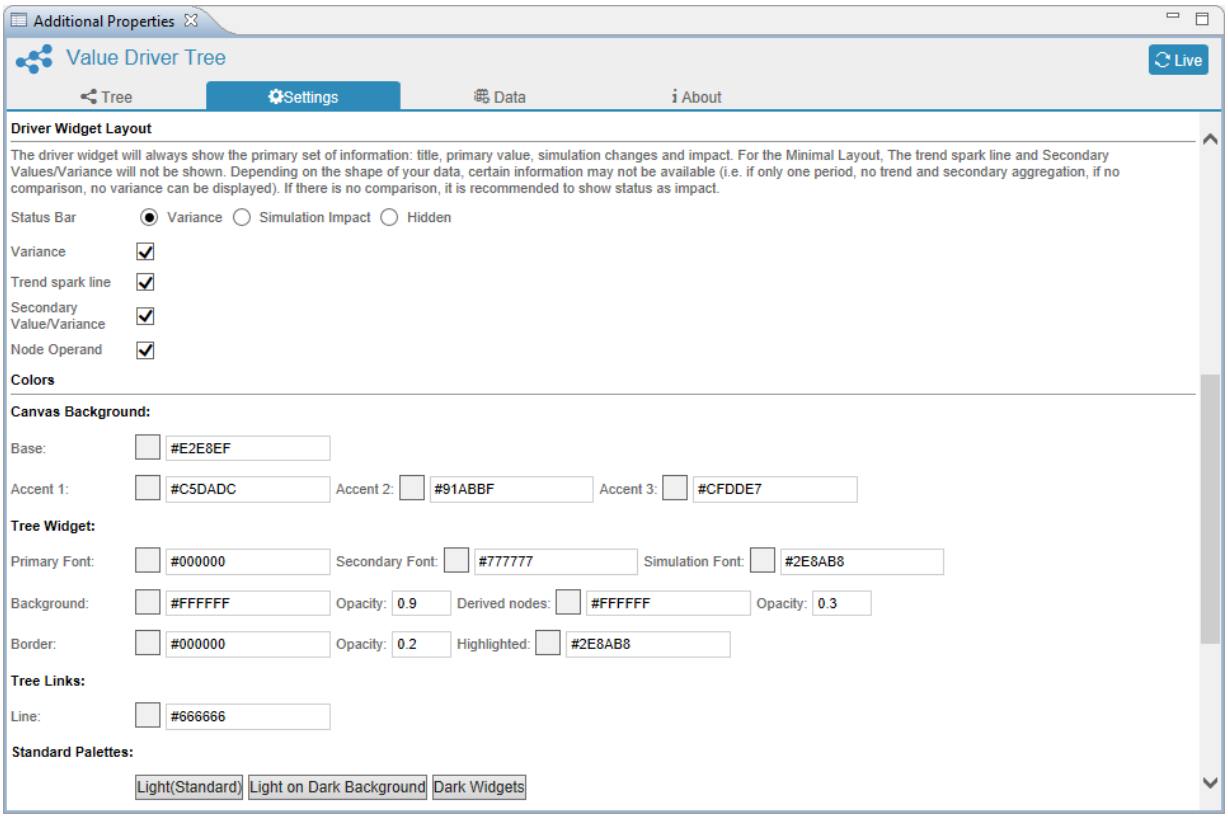


Figure 10.35: Settings Tab – Driver Widget Layout and Colors

14. Activate the property Variance.
15. Activate the property Trend spark line.
16. Activate the property Secondary Value/Variance.
17. Activate the property Node Operand.
18. In the area Colors we have four different properties namely Canvas Background, Tree Widget, Tree Links and Standard Palettes. It is to be noted that all the color selections will be represented in the Hexadecimal color formats.
19. For the property Canvas Background, set the sub property Base to the color code “#E2E8EF”.
20. Set the sub property Accent 1 to the color code “#C5DADC”, set the sub property Accent 2 to the color code “#91ABBF” and set the sub property Accent 3 to the color code “#CFDDE7 “.

21. For the property Tree Widget, set the following sub properties to the values as shown in the Table below:

Sub property	Color Code Values
Primary Font	#000000
Secondary Font	#777777
Simulation Font	#2E8AB8
Background	#FFFFFF
Derived nodes	#FFFFFF
Opacity	0.3
Border	#000000
Opacity	0.2
Highlighted	#2E8AB8

Table 10.2: Tree Widget Sub properties

22. For the property Tree Links, set the sub property Line to the color code "#666666".
23. In the area Events, set the property Default Action to the option "Show Embedded Standard Detail Popup" (see Figure below).

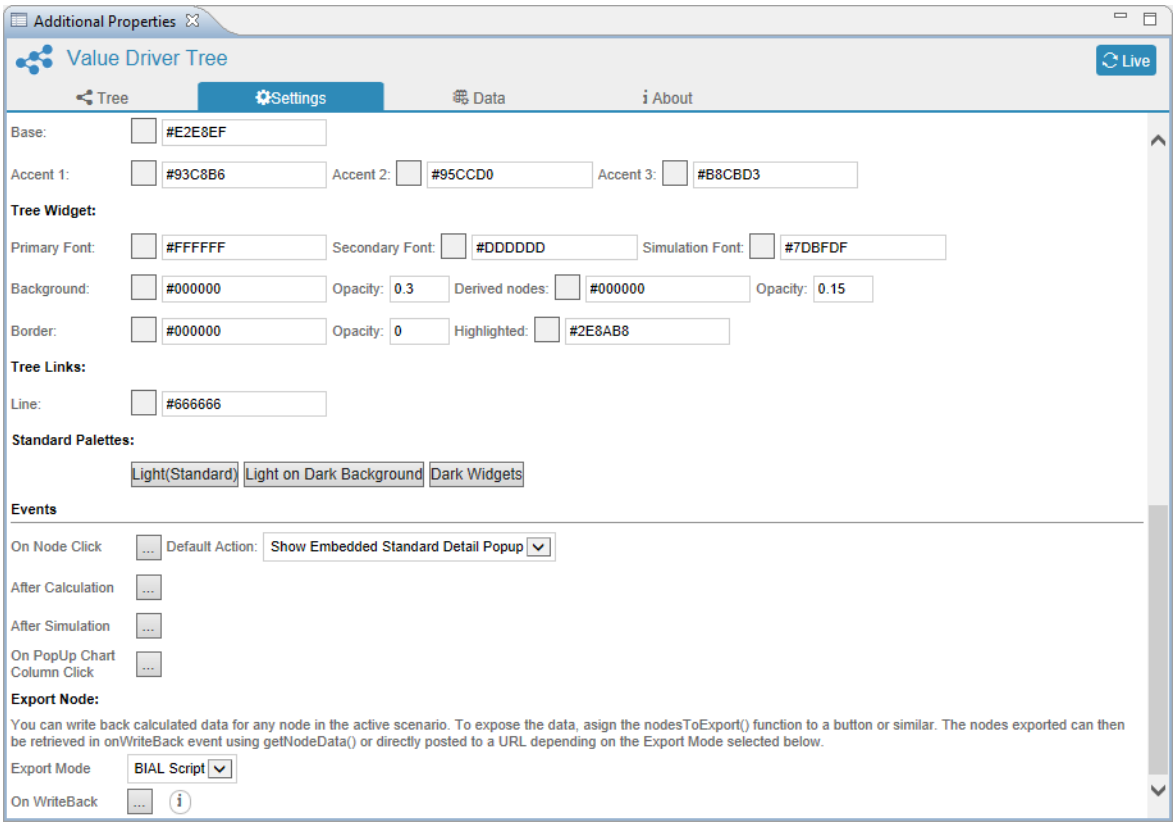


Figure 10.36: Settings Tab – Events

24. For the property Export Node, set the sub property Export Mode to the option "BIAL Script".

10.6.1 Scale Suffix

As part of VDT Release 2.0.5, the user will now be able to customize the Scale Suffix as shown below

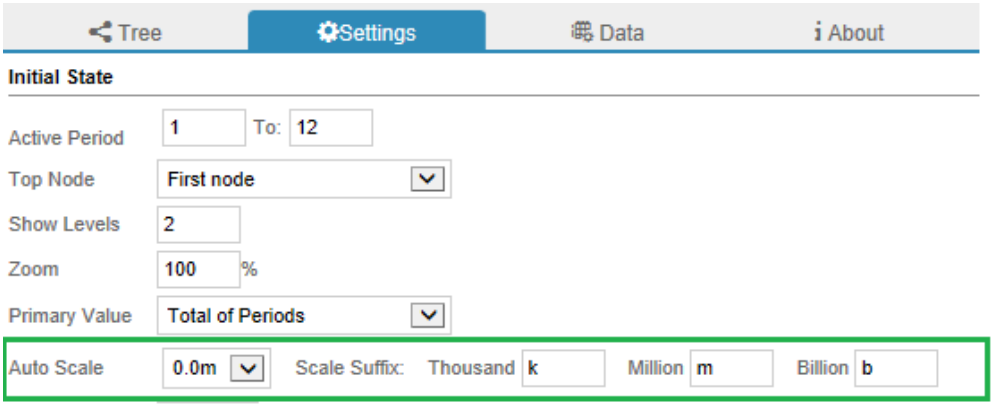


Figure 10.37: Settings Tab – Initial State

For our example, 2000 can be represented as 2K, 2k, 2Th by providing appropriate value in Thousand field of "Scale Suffix" (see Figure above).

10.6.2 Formatting of Numeric Values

As part of VDT Release 2.0.5, the user will be able to format the numeric values using custom "Decimal Separator" and "Thousand Separator" (see Figure below).

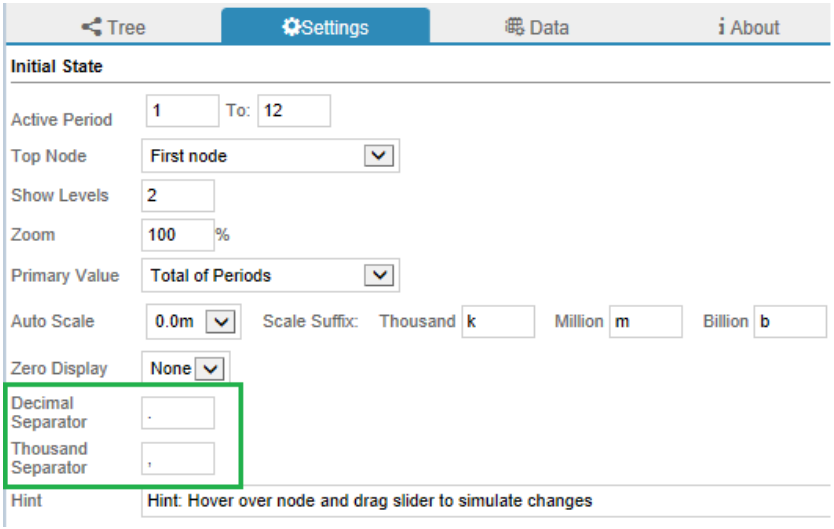


Figure 10.38: Settings Tab – Initial State

10.6.3 Write simulation results to underlying systems

As part of VDT Release 2.0.5, you can write back the calculated data for any node in the active scenario. To expose the data, assign the nodesToExport() function to a button or similar. The nodes exported can then be retrieved in onWriteBack event using getNodeData() or directly posted to a URL depending on the Export Mode selected below.

10.6.3.1 Steps to be followed for writing simulation results to underlying systems

1. The Node ID will be defined in the Tab Tree of the Additional Properties for the Value Driver Tree (see Figure below).

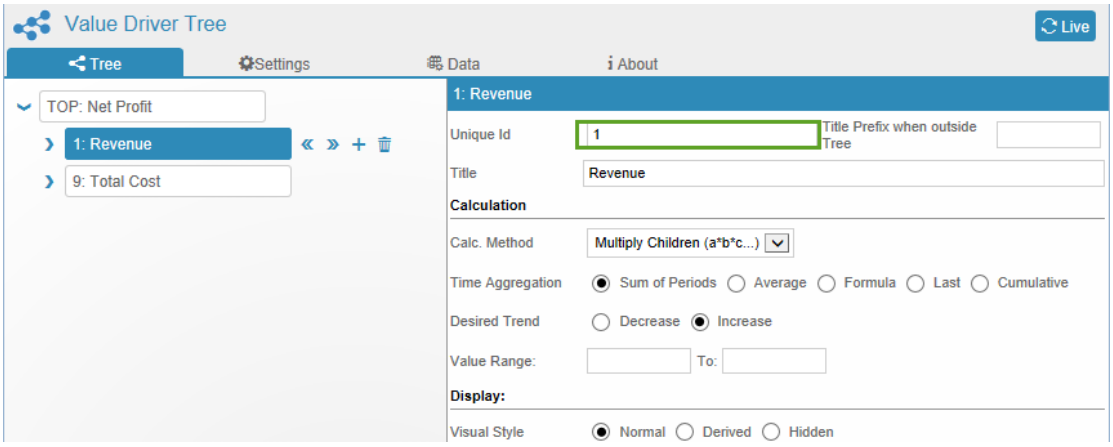


Figure 10.39: Node ID

2. On button click event, write the script as shown in the below screen. The script method “nodesToExport” takes Array of Node Ids or no parameter to export all the nodes.

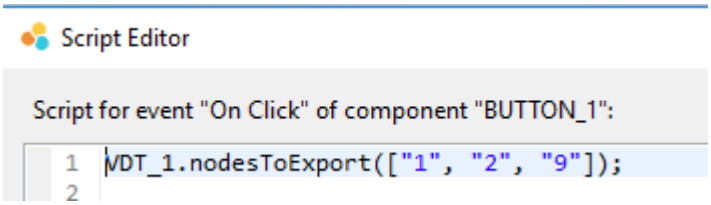


Figure 10.40: Script Editor

3. Navigate to Settings > Events > Export Node
4. Select “Export Mode”
5. Export Mode = “BIAL Script”
6. Provide the script in “On Writeback “

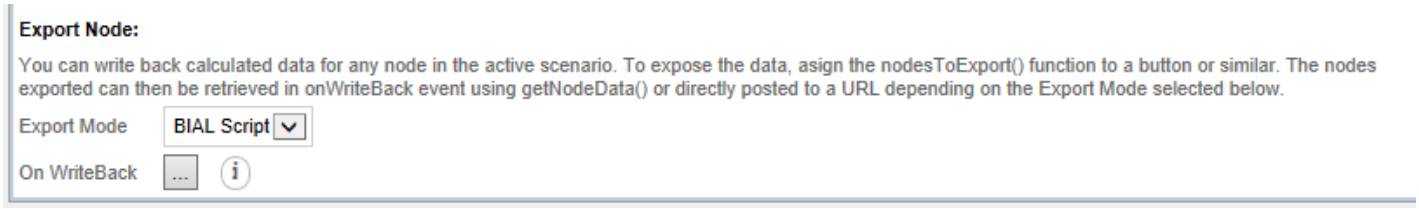


Figure 10.41: Export Node – BIAL Script

```
/***** Data Structure of Exported Data *****/
{
        "context": {
         "name": "1",
         "title": "Scenario 1",
         "descr": "Revenue: 51% \nTotal Cost: (44%)",
         "simVar": [{
                  "name": "1",
                  "value": "51.44",
                  "manSim": [],
                  "sMeth": "P"
                 }, {
                  "name": "9",
                  "value": "-44.30",
                  "manSim": [],
                  "sMeth": "P"
                 }
         ]
        },
        "data": [{
                 "name": "1",
                 "results": [{
                          "key": "1",
                          "text": "Jan",
                          "act": 39985500,
                          "base": 26403525.97,
                          "tgt": 26776114.98
                  }
                 ]
         }, {
                 "name": "9",
                 "results": [{
                          "key": "1",
                          "text": "Jan",
                          "act": 49985500,
                          "base": 36403525.97,
                          "tgt": 36776114.98
                  }
```

```
                    ]
                }
            ]
}

/***** Sample Code to demonstrate looping and extracting required information *****/

var exportedData = me.getNodeData(['1', '9']);

var acts = '';
exportedData.data.forEach(function(node, i) {
  acts = acts + ' ::: ' + node.name + ' => ';
  node.results.forEach(function(result, j) {
    acts = acts + ', ' + result.act;
  });
});

var context = exportedData.context.descr +
          ', ' + exportedData.context.name +
                              ', ' + exportedData.context.title;

exportedData.context.simVar.forEach(function(ele, i){
  context = context + ' ::: ' + ele.name + ', ' +
          ele.value + ' => ';
  ele.manSim.forEach(function(element, index) {
            context = context + ', ' + element;
  });
});

APPLICATION.log(acts);
```

7. Export Mode = “URL”
8. Provide the web service url / REST Endpoint in the “Export Url” field.

Figure 10.42: Export Node - URL

10.6.4 Negative Value Format

As part of VDT Release 3.0, navigate to the Tab Settings in the Additional Properties of the VDT. In the area Initial State, now you have the ability to set the option for the property Negative Display. The options are (0), -0 and 0-. For our example, the option 0- has been selected as shown in the below figure. When the option (0) is selected then you can view the whole number in the node.

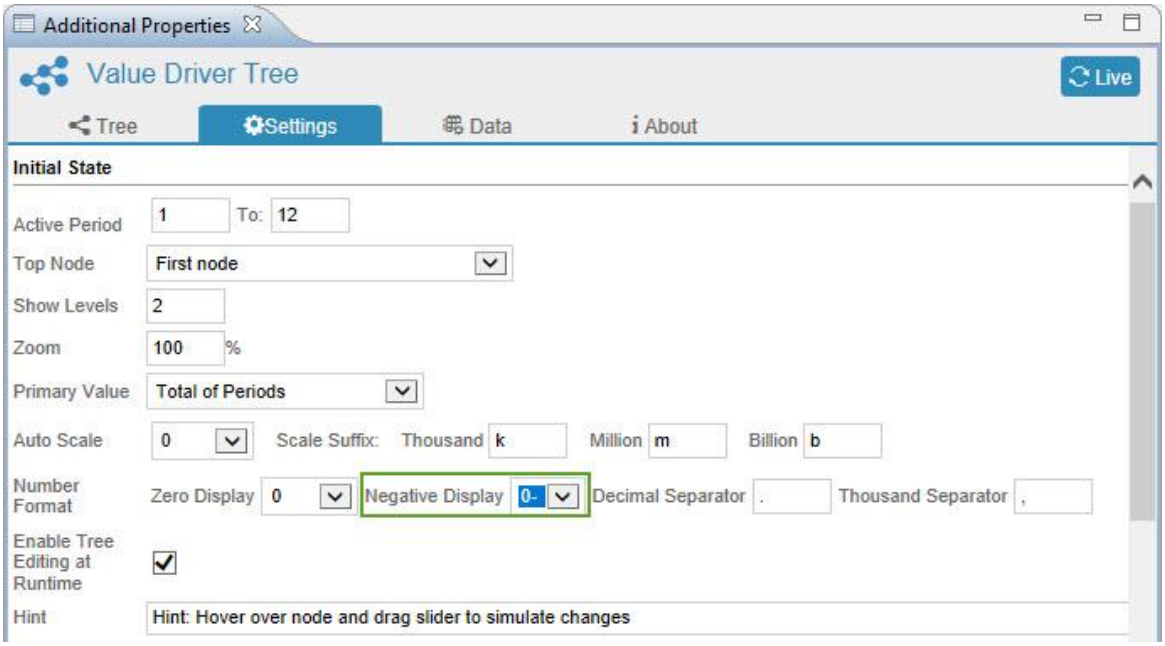


Figure 10.43: Negative Display

Based on the above configuration you will be able to observe that the values in the nodes will be highlighted with selected Negative Display option.

Figure 10.44: Negative Display in Runtime

10.6.5 Visualization

As part of VDT Release 3.0, navigate to the Tab Settings in the Additional Properties of the VDT. In the area Visualization, now you have the ability to configure the Table Style and Node Style using the properties Default Table Style and Default Node Style.

10.6.6 Tree Style

For our example, set the property Visual Style to the option Tree. Set the property Default Node Style to the option “Full (More space for more digits but fewer widgets on a page) as shown in the below Figure. The other options are “Standard (Good compromise between space and widget density)” and “Minimal : (no trend and secondary information).

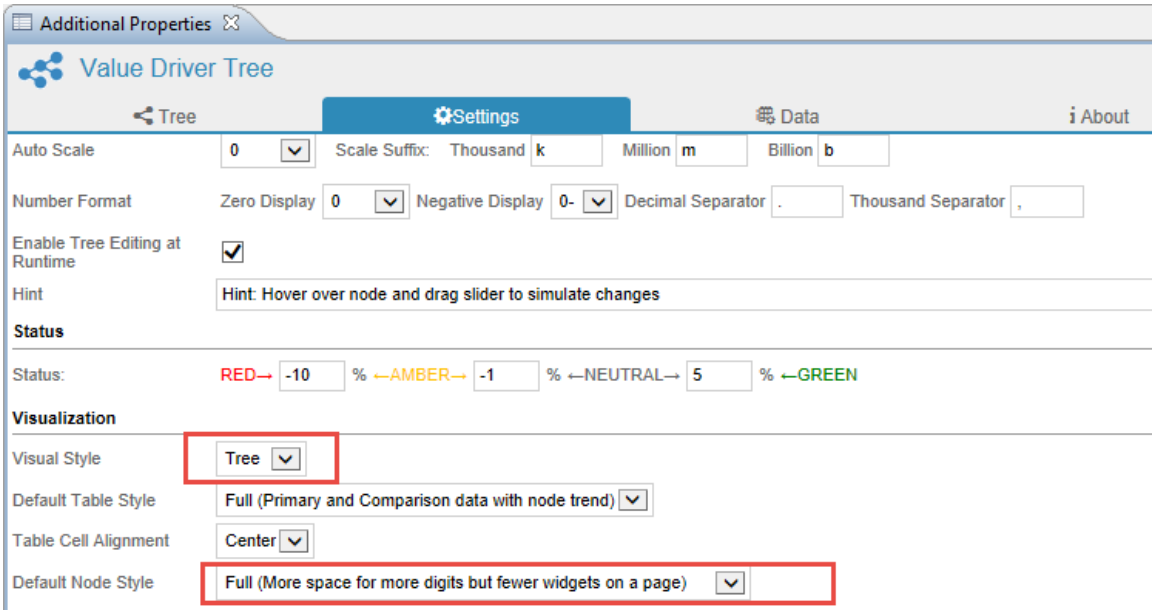


Figure 10.45: Default Node Style

Based on the above configuration, you will be able to observe that the Node will have more space for more digits but will have fewer widgets on a page as shown in the below Figure.

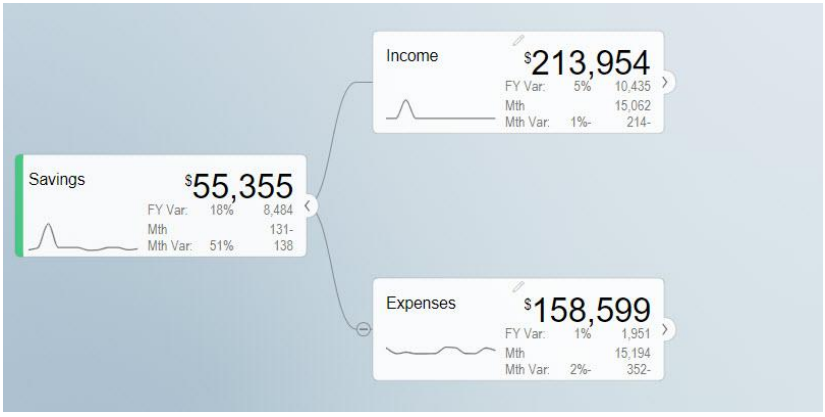


Figure 10.46: Node selected as Full Mode

10.6.7 Table Style

For our example, set the property Visual Style to the option Table. Set the property Default Table Style to the option “Full (Primary and Comparison data with node trend) as shown in the below Figure. The other options are “Standard (Primary data with node trend)” and “Minimal : (Primary data without node trend)”.

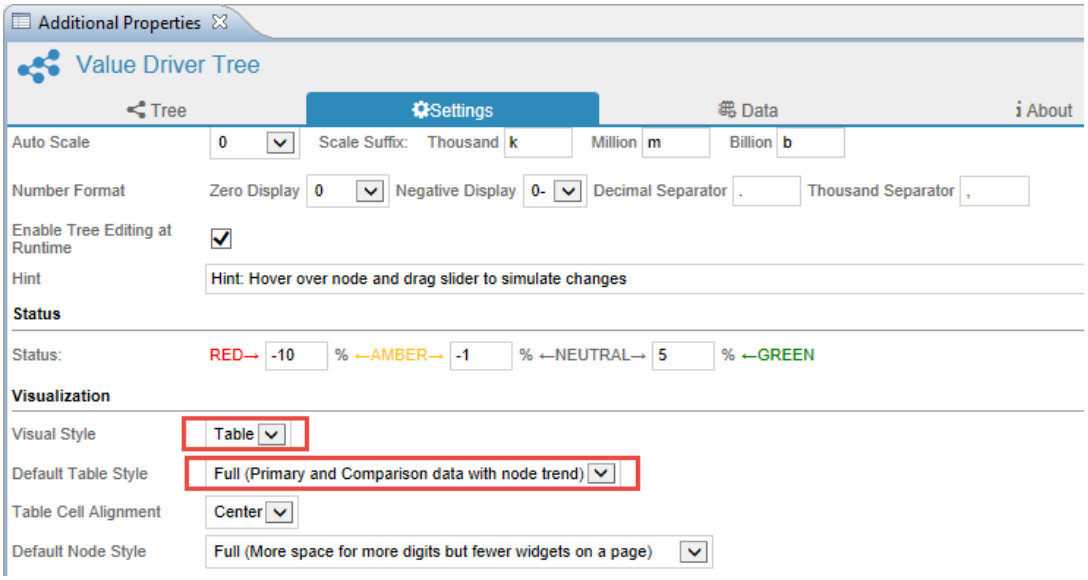


Figure 10.47: Default Table Style

Based on the above configuration, you will be able to observe that the Table will have the Primary data with node trend as shown in the below Figure.

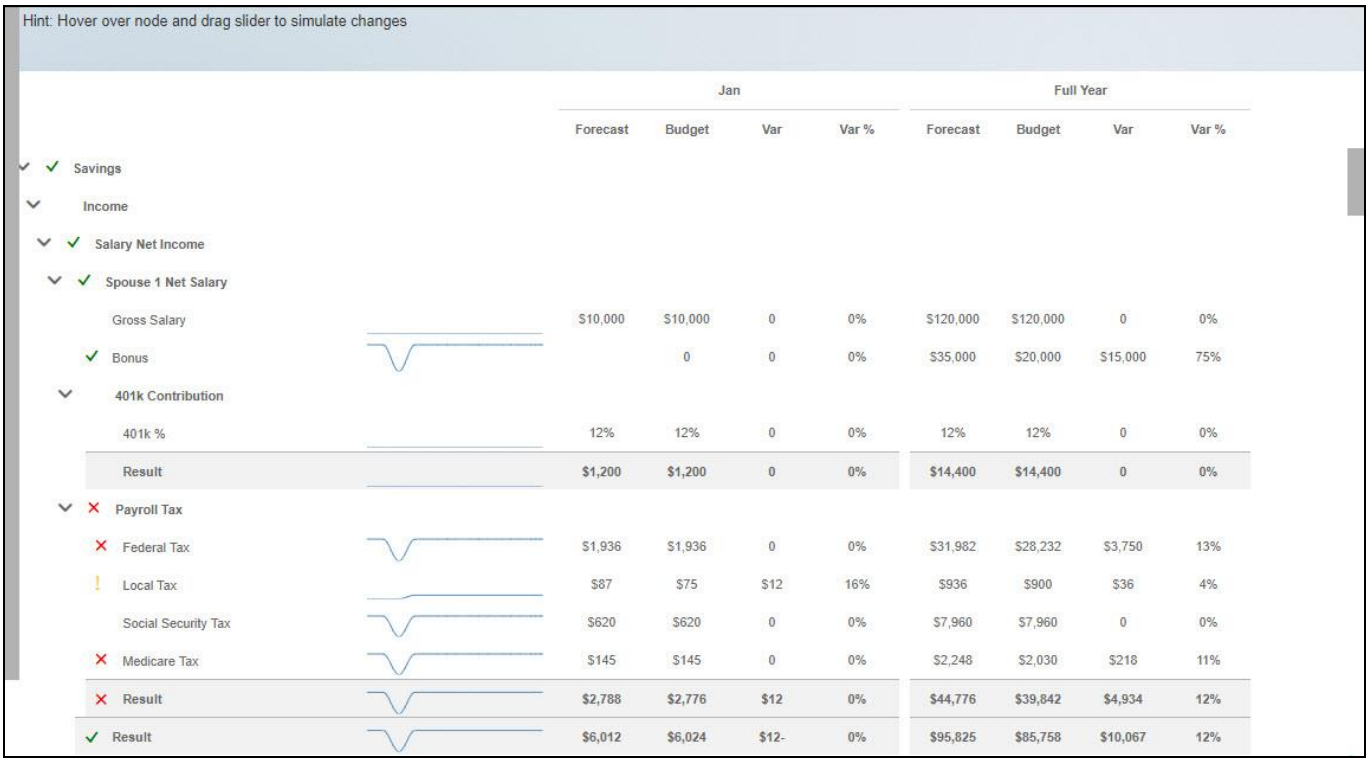


Figure 10.48: Table selected as Full Mode

10.6.8 Visualization in Runtime

In the VDT Runtime, there are two types of Visualizations namely Tree Visualization and Table Visualization (see Figure below).

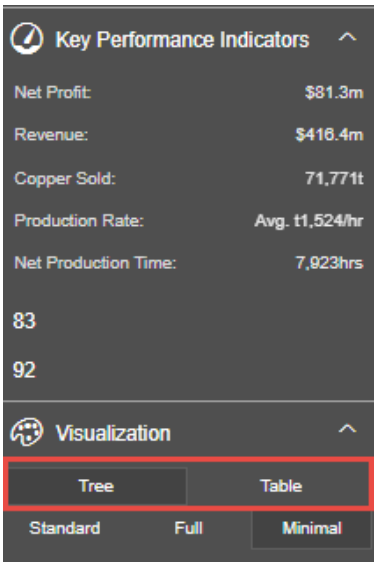


Figure 10.49: Visualization in Run time

10.6.8.1 Tree Visualization

In Tree Visualization, there are three different styles (see Figure below). The styles are listed as follows:

1. Standard
2. Full
3. Minimal

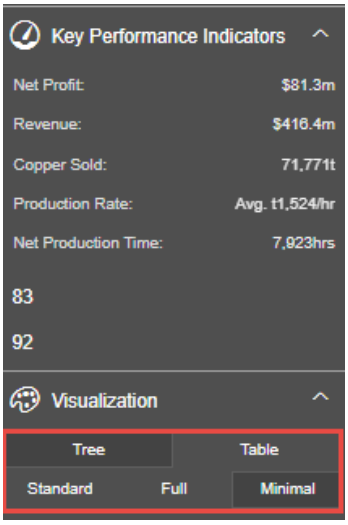


Figure 10.50: Tree Visualization in Run time

Standard Style:

In Standard Style, you will be able to view the Tree with the Node having good compromise between space and widget density (see Figure below).

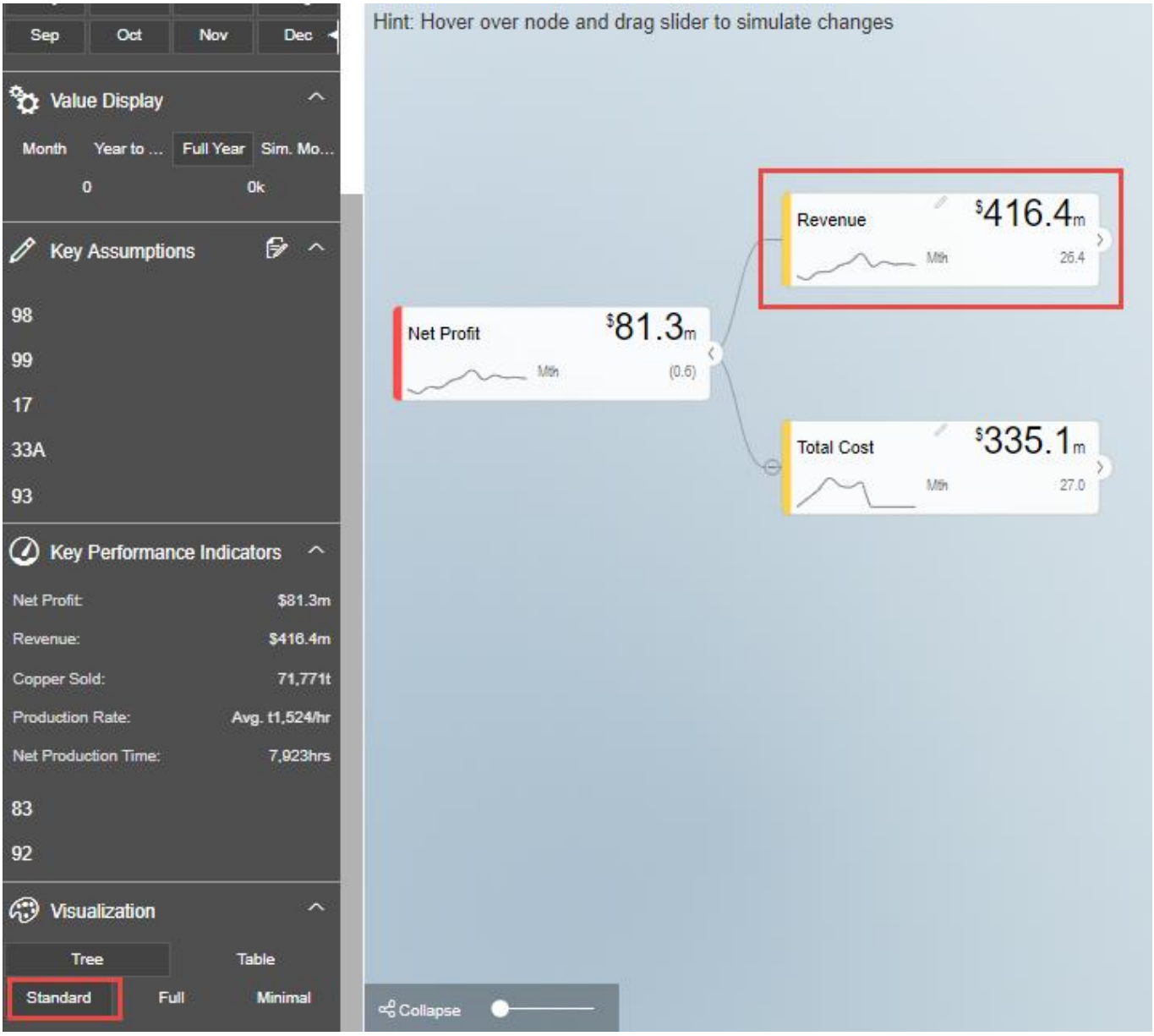


Figure 10.51: Tree Visualization in Standard Style

Full Style:

In Full Style, you will be able to view the Tree with the Node having more space for more digits but fewer widgets on a page (see Figure below).

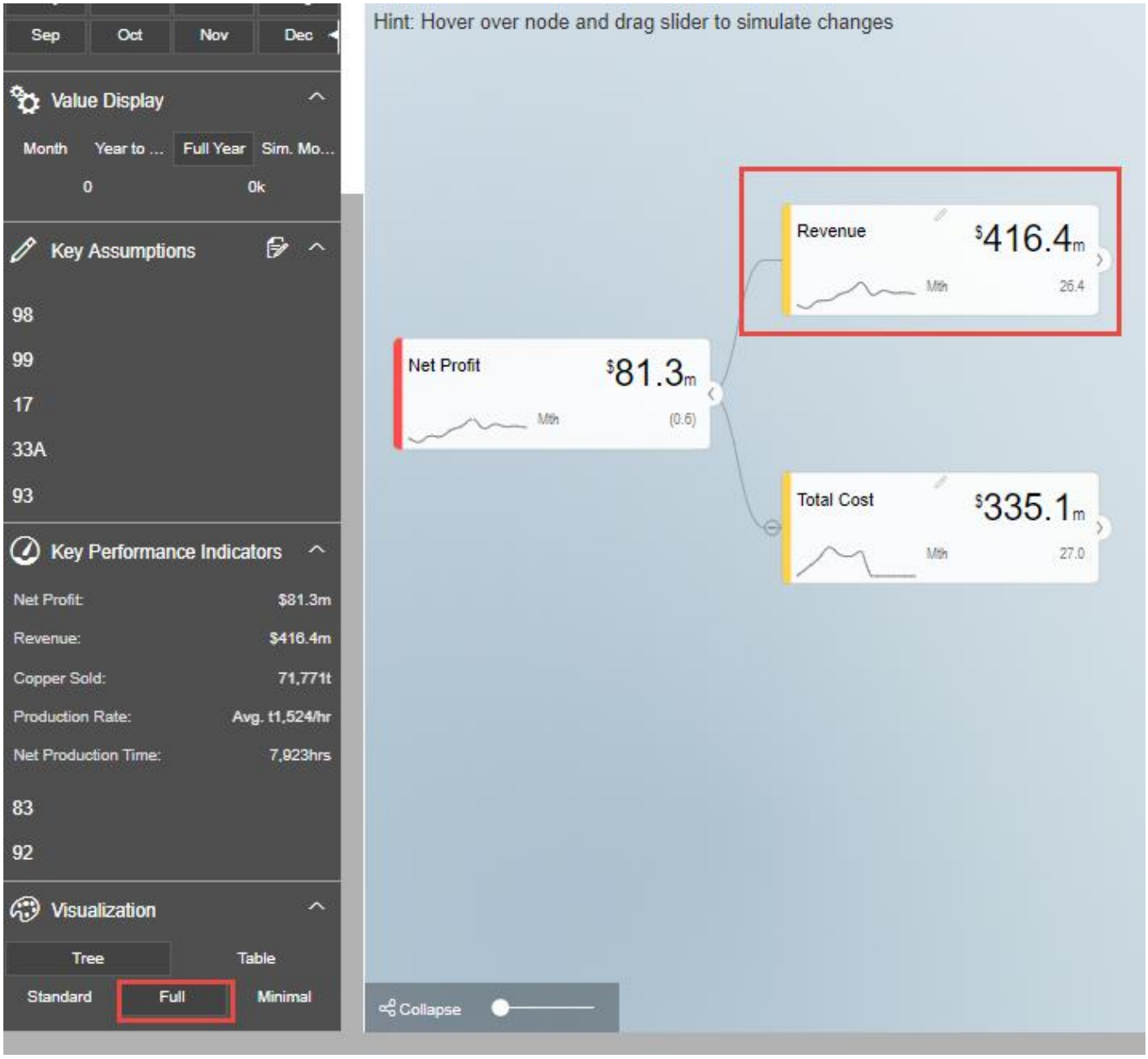


Figure 10.52: Tree Visualization in Full Style

Minimal Style:

In Minimal Style, you will be able to view the Tree with the Node without trend and secondary information (see Figure below).

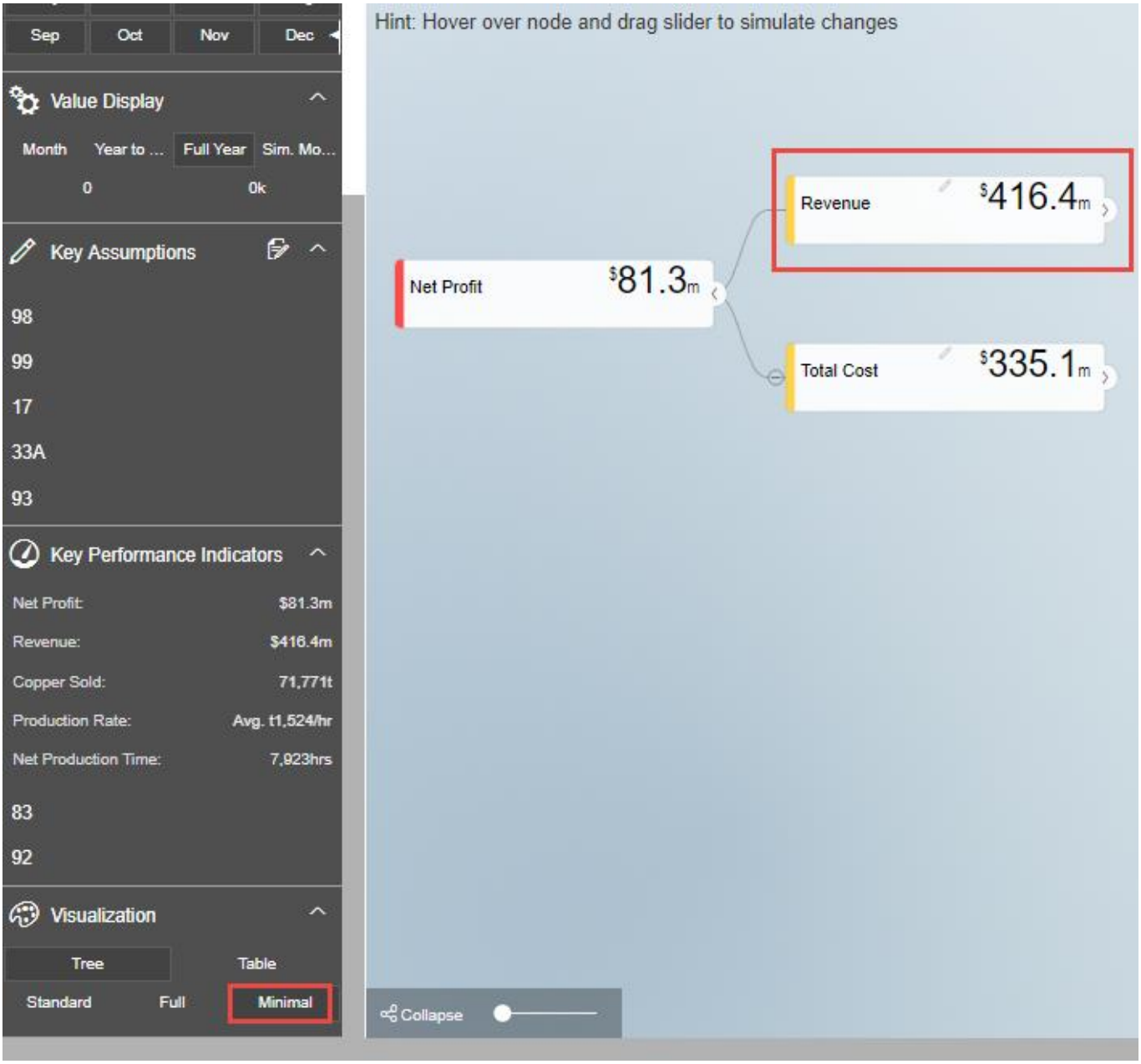


Figure 10.53: Tree Visualization in Minimal Style

10.6.8.2 Table Visualization

In Table Visualization, there are two different styles (see Figure below). The styles are listed as follows:

1. Standard
2. Full

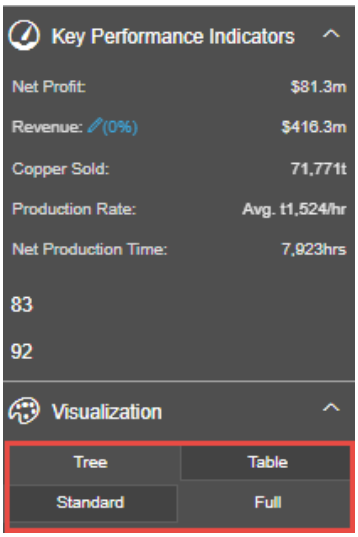


Figure 10.54: Table Visualization in Run time

Standard Style:

In Standard Style, you will be able to view the Table having Primary Data with the Node Trend (see Figure below).

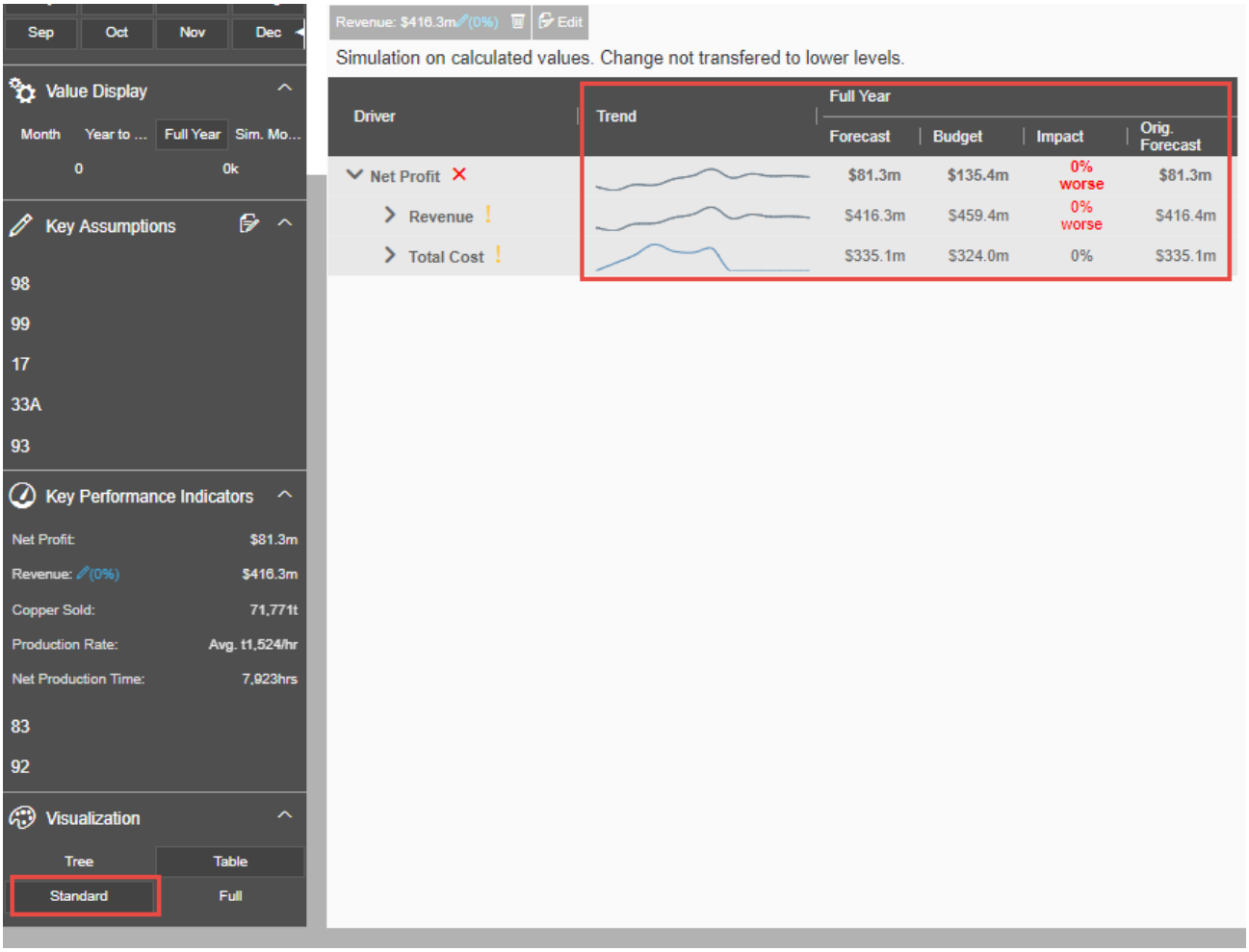


Figure 10.55: Table Visualization in Standard Style

Full Style:

In Full Style, you will be able to view the Table having Primary and Comparison Data with the Node Trend (see Figure below).

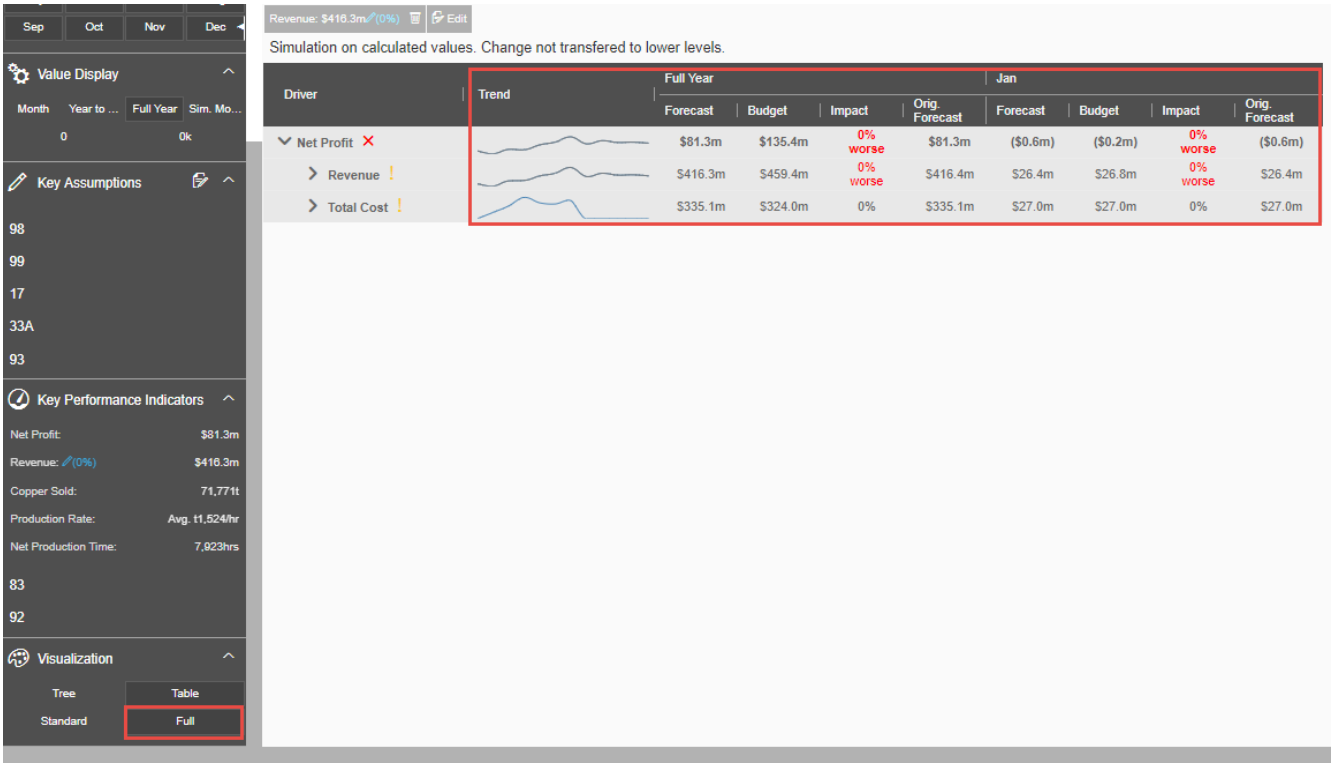


Figure 10.56: Table Visualization in Full Style

10.6.9 Enable Tree Editing in Runtime

As part of VDT Release 3.0, navigate to the Tab Settings in the Additional Properties of the VDT. In the area Initial State, you will be able to edit the Nodes in the Tree by enabling the property Enable Tree Editing at Runtime (see Figure below).

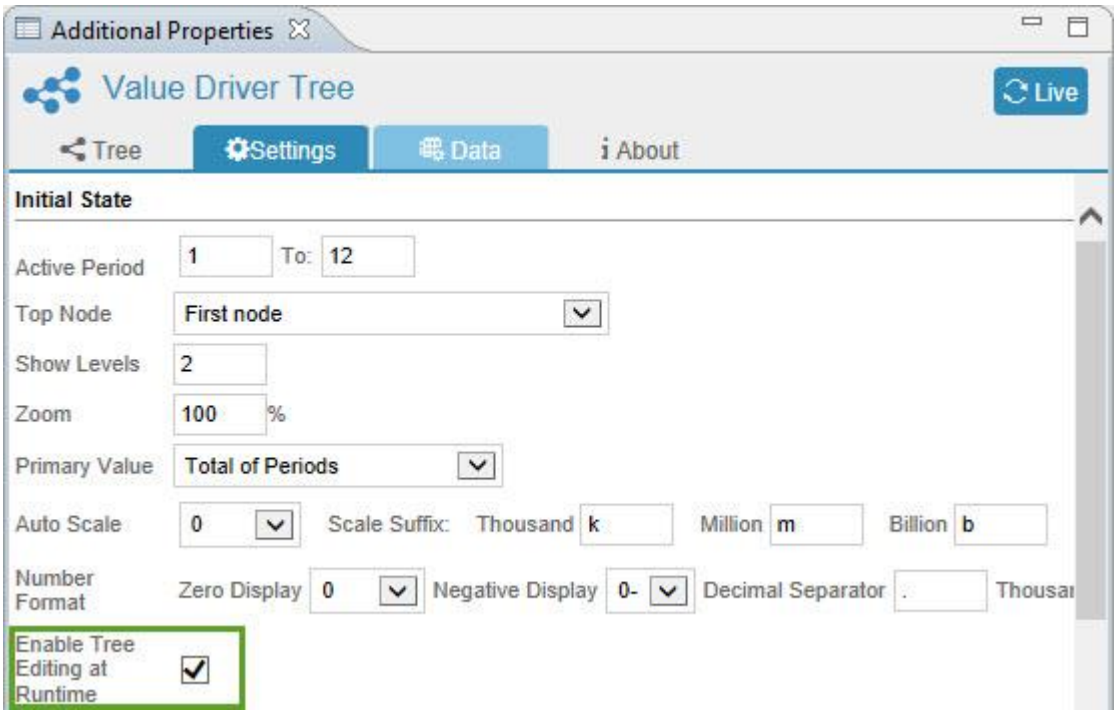


Figure 10.57: Enable Tree Editing at Runtime

Based on the above configuration you can observe that the Nodes in the Tree can be moved to the desired Node level by clicking the Edit Mode option at the bottom of the Runtime screen. At the same time number of child nodes can be also created under a respective node as its copy versions.

For our example, you can observe that the Node “Salary Net Income” in the Runtime is at the branch node level of the Node “Income” (see Figure below).

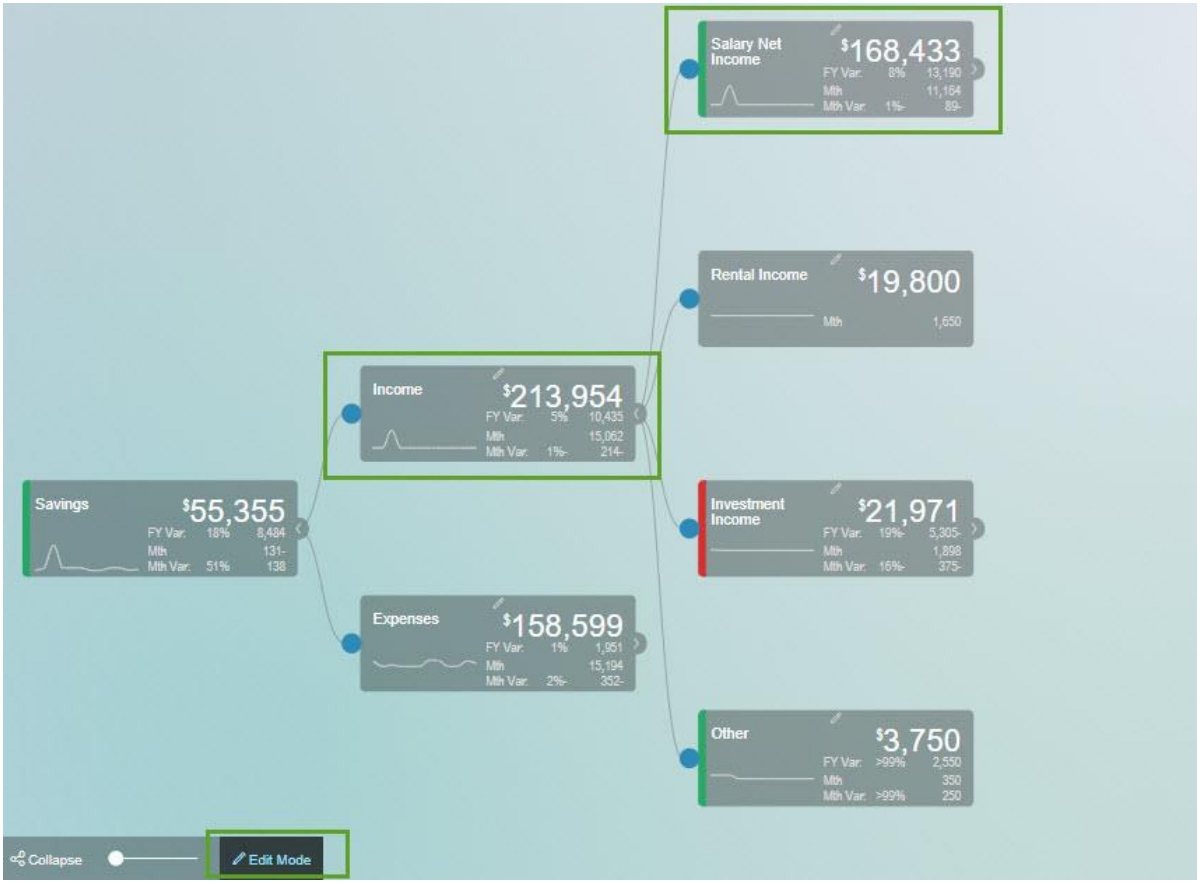


Figure 10.58: Tree with Nodes – Edit Mode

Now using the Edit Mode option as shown in the above Figure, you can drag/move the Node “Salary Net Income “ as a branch node for the Node “Expenses” (see Figure below).

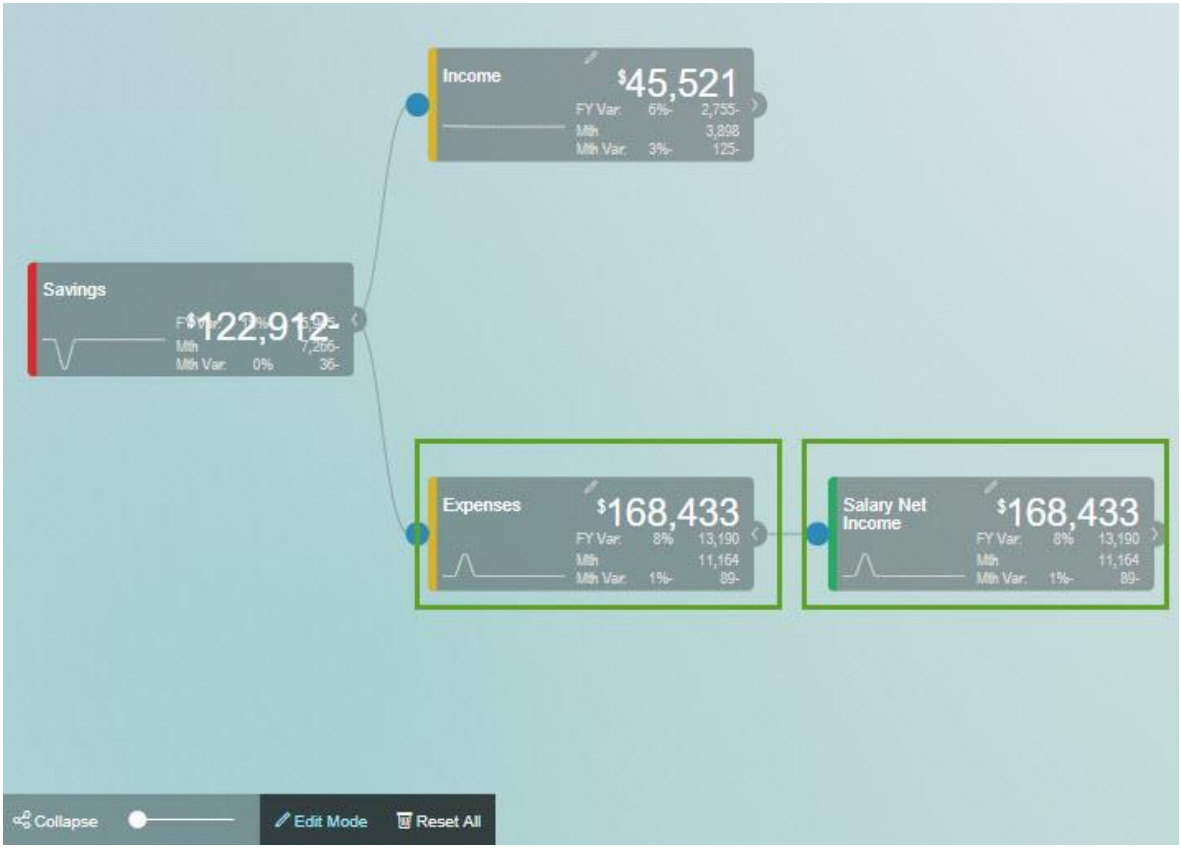


Figure 10.59: Tree with Nodes after Editing

Also you can create number of child nodes under a specific node as its copy versions. For our example, 3 child nodes have been created under the Node Savings (see Figure below).

Figure 10.60: Nodes created as copied versions

10.7 Additional Properties of Settings Tab

Area	Property	Description
Initial State	Active Period	This property sets the Active Period for the Tree.
	Top Node	Using this property, you can select the desired Node from the Nodes List to assign the Top Node for the Tree.
	Show Levels	This property sets the Levels for the Tree.
	Zoom	This property sets the Zoom Level for the Tree.
	Primary Value	This property sets the Primary Value for the Tree. The options are Active Period, Total to Period, Total of Periods and Selected Simulation Interval.
	Auto Scale	This property sets the Auto scale for the Tree. The options are 0, 0.0b, 0.00b, 0m, 0.0m, 0.00m, 0k and 0.
	Scale Suffix	This property sets the Thousand, Million and Billion suffix for the scale.
	Number Format	This property sets the Zero Display, Negative Display, Decimal Separator and Thousand Separator. The options for Zero Display are None, 0 and -. The options for Negative Display are (0), -0 and 0-.
	Enable Tree Editing at Runtime	This property enables/disables the Tree Editing at Runtime.
	Hint	This property sets the Hint for the Tree.
Status	RED	This property sets the RED color vertical bar in the Node based on the simulated value of the Node.
	AMBER	This property sets the AMBER color vertical bar in the Node based on the simulated value of the Node.
	NEUTRAL	This property sets NEUTRAL Status in the Node based on the simulated value of the Node.
	GREEN	This property sets the GREEN color vertical bar in the Node based on the simulated value of the Node.
Visualization	Visual Style	This property sets the Visual Style for the Tree. The options are Tree and Table.
	Default Table Style	This property sets the Default Table Style for the Tree. The options are Standard (Primary Data with Node Trend) and Full (Primary and Comparison data with Node Trend).
	Table Cell Alignment	This property sets the text alignment in the Table Cell. The options are Left, Center and Right.
	Default Node Style	This property sets the Default Node Style for the Tree. The options are Standard (Good compromise between space and widget density), Full (More space for more digits but fewer widgets on a page) and Minimal (no trend and secondary information).
Driver Widget Layout	Status Bar	This property sets the Status Bar. The options are Variance, Simulation Impact and Hidden.

Area	Property	Description
	Variance	This property enables/disables the Variance option.
	Trend spark line	This property enables/disables the Trend spark line option.
	Secondary Value/Variance	This property enables/disables the Secondary Value/Variance option.
	Node Operand	This property enables/disables the Node Operand option.
Colors - Canvas Background	Base	This property sets the Base color.
	Accent 1	This property sets the Accent 1 color.
	Accent 2	This property sets the Accent 2 color.
	Accent 3	This property sets the Accent 3 color.
Colors - Tree Widget	Primary Font	This property sets the Primary Font color.
	Secondary Font	This property sets the Secondary Font color.
	Simulation Font	This property sets the Simulation Font color.
	Background	This property sets the Background color.
	Opacity	This property sets the Opacity for the Background color.
	Derived Nodes	This property sets the color for the Derived Nodes.
	Opacity	This property sets the Opacity for the Derived Nodes Color.
	Border	This property sets the Border color.
	Opacity	This property sets the Opacity for the Border color.
	Highlighted	This property sets the Highlighted color.
Colors - Tree Links	Line	This property sets the Line color.
Colors - Standard Palettes	Light (Standard)	This property sets the Light Standard Palette.
	Light on Dark Background	This property sets the Light on Dark Background Palette.
	Dark Widgets	This property sets the Dark Widgets Palette.
Events	On Node Click	This Event will be triggered on Node Click.
	Default Action	This property sets the Default Action for the Node Click.
	Custom Detail PopUp Reference	This property sets the Custom Detail PopUp Reference.
	After Calculation	This Event will be triggered after calculation.
	After Simulation	This Event will be triggered after simulation.
	On PopUp Chart Column Click	This Event will be triggered on PopUp Chart Column Click.
Events - Export Node	Export Mode	This property sets the Export Mode for the Node. The options are BIAL Script and URL.
	On WriteBack	This Event will be triggered on WriteBack.

Table 10.3: Settings Tab

10.8 Data Tab

In the Data Tab, you will be able to configure the Data related properties for the entire Tree.

The steps below explains on how to configure the Data related properties for the entire Tree:

1. Now navigate to the Data Tab (see Figure below).

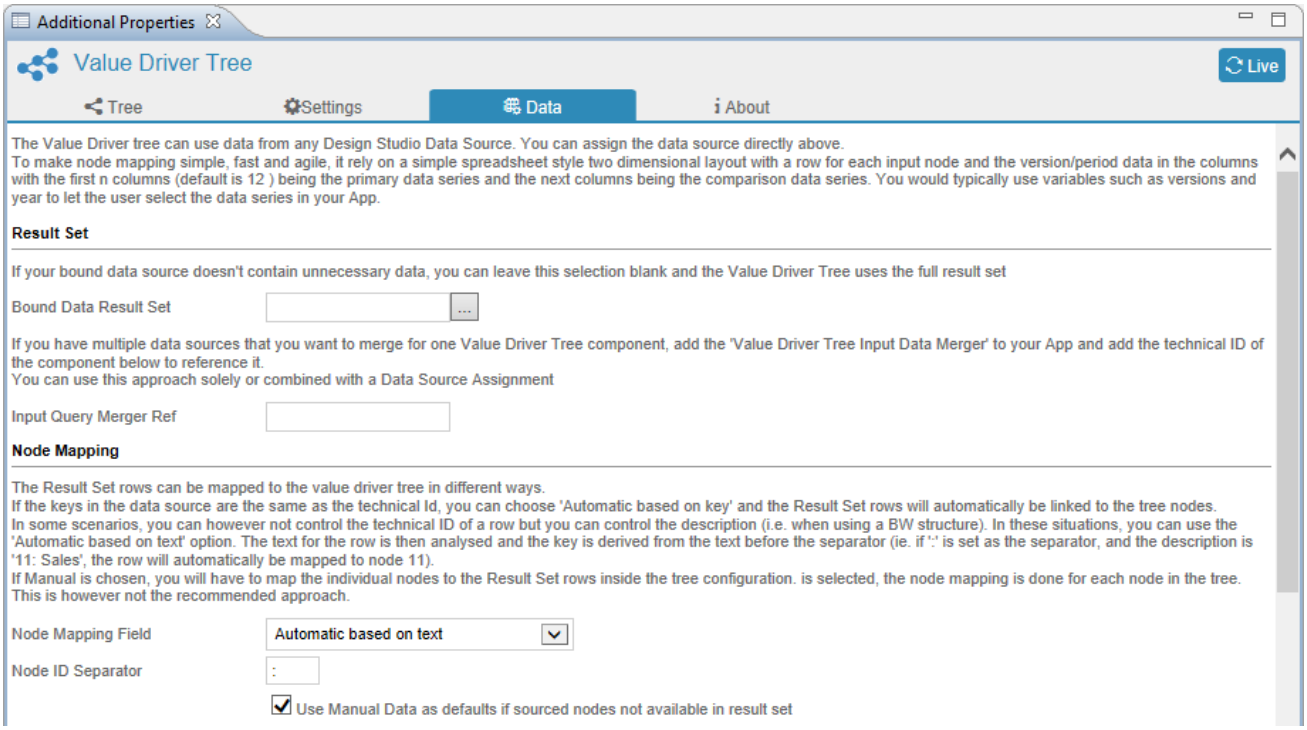


Figure 10.61: Data Tab – Node Mapping

2. In the area Node Mapping, set the property Node Mapping Field to the option “Automatic based on text”.
3. Set the property Node ID Separator to “:”.
4. Activate the property Use Manual Data as defaults if sourced nodes not available in result set.
5. In the area Data Series, activate the property “with Comparison data series” (see Figure below).

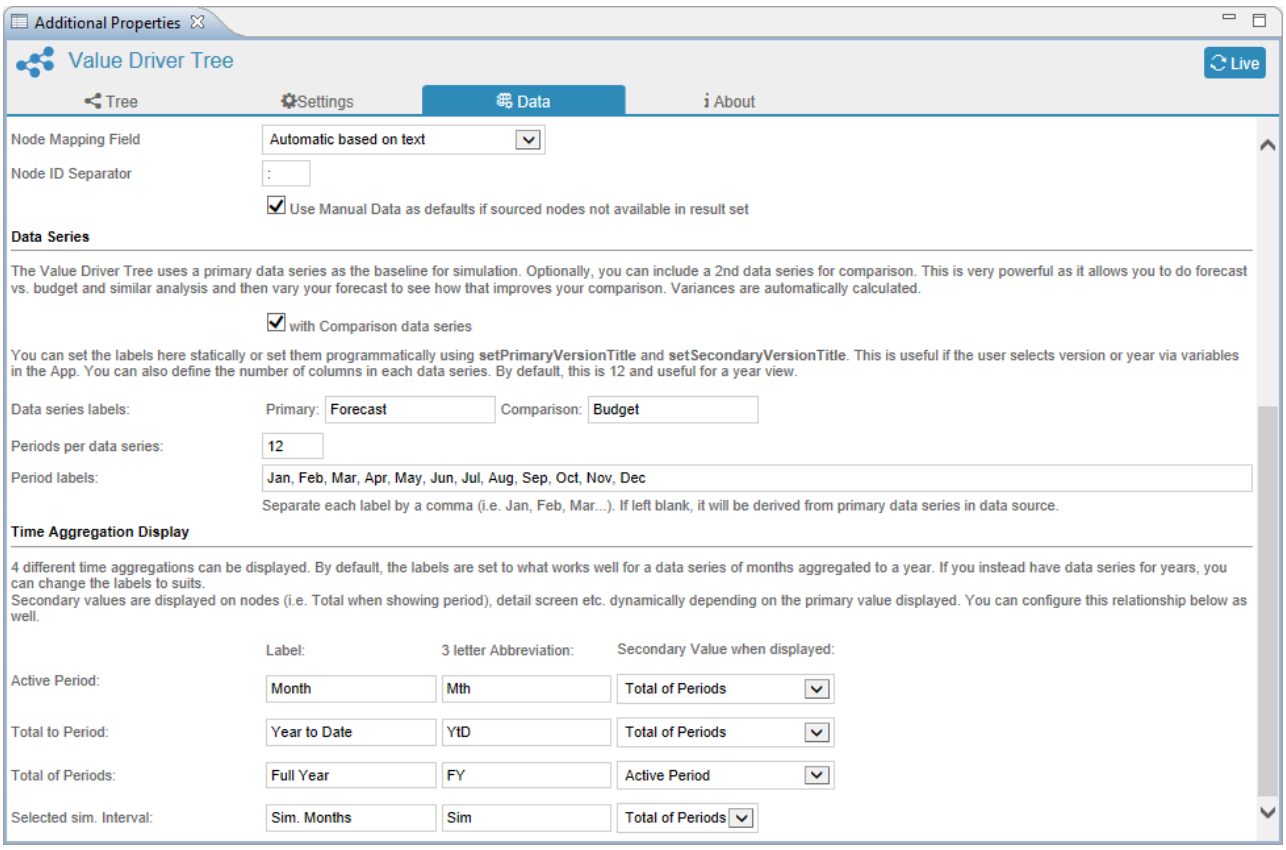


Figure 10.62: Data Tab – Data Series and Time Aggregation Display

6. For the property Data series labels, enter the label for the sub property Primary as “Forecast” and enter the label for the sub property Comparison as “Budget”.
7. Set the property Periods per data series to the value 12.
8. Set the property Period Labels as “Jan, Feb, Mar, Apr, May, Jun, Jul, Aug, Sep, Oct, Nov, Dec”.
9. In the area Time Aggregation Display set the properties and their sub properties to the values as shown in the Table below:

Property	**Sub properties**		
	Label	**3 letter abbreviation**	**Secondary value when displayed**
Active Period	Month	Mth	Total of Periods
Total to Period	Year To Date	YtD	Total of Periods
Total of Periods	Full Year	FY	Active Period
Selected sim. Interval	Sim. Months	Sim	Total of Periods

Table 10.4: Time Aggregation Display – Properties and sub properties

10. Now assign the Navigational Panel VDT_NAV_1 for Value Driver Tree VDT_1 as shown in the Outline Panel (see Figure below).

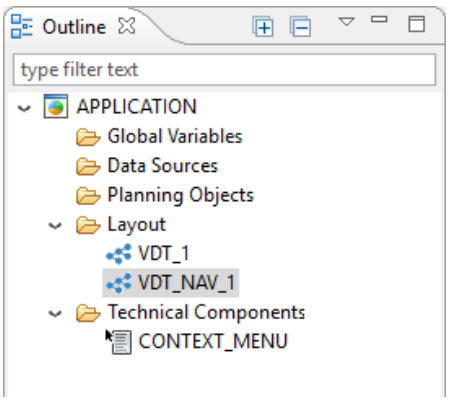


Figure 10.63: Outline

11. Navigate to the Standard Properties of the Navigational Panel for Value Driver Tree. Set the property Linked Value Driver Tree to the Value Driver Tree VDT_1 (see Figure below).

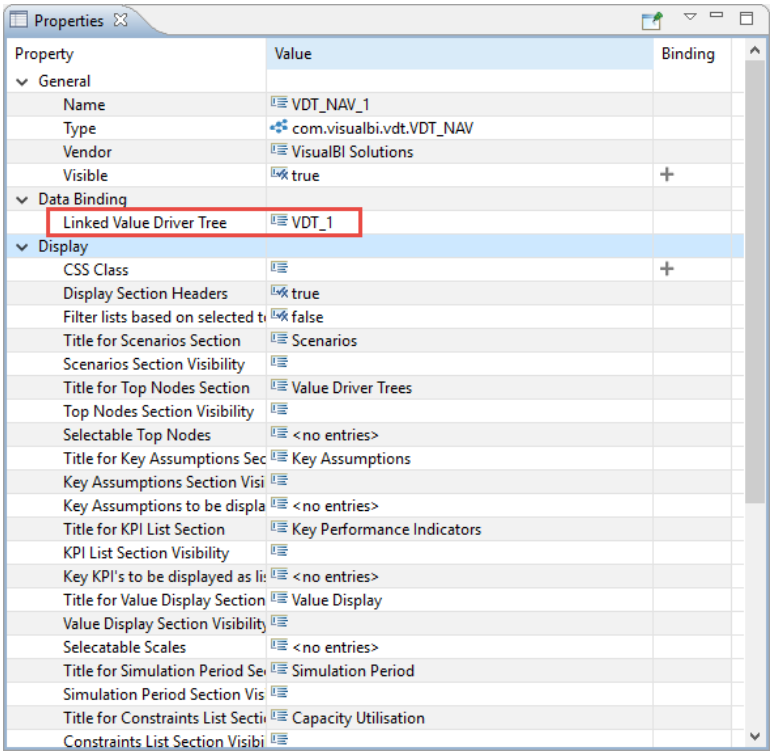


Figure 10.64: Standard Properties of Navigational Panel for Value Driver Tree

12. Based on the Standard Properties being configured for the Navigational Panel for Value Driver Tree and all the properties being configured from Tree, Settings and Data Tabs, you will be able to view the entire Tree with Parent Node and Child Nodes as shown below.

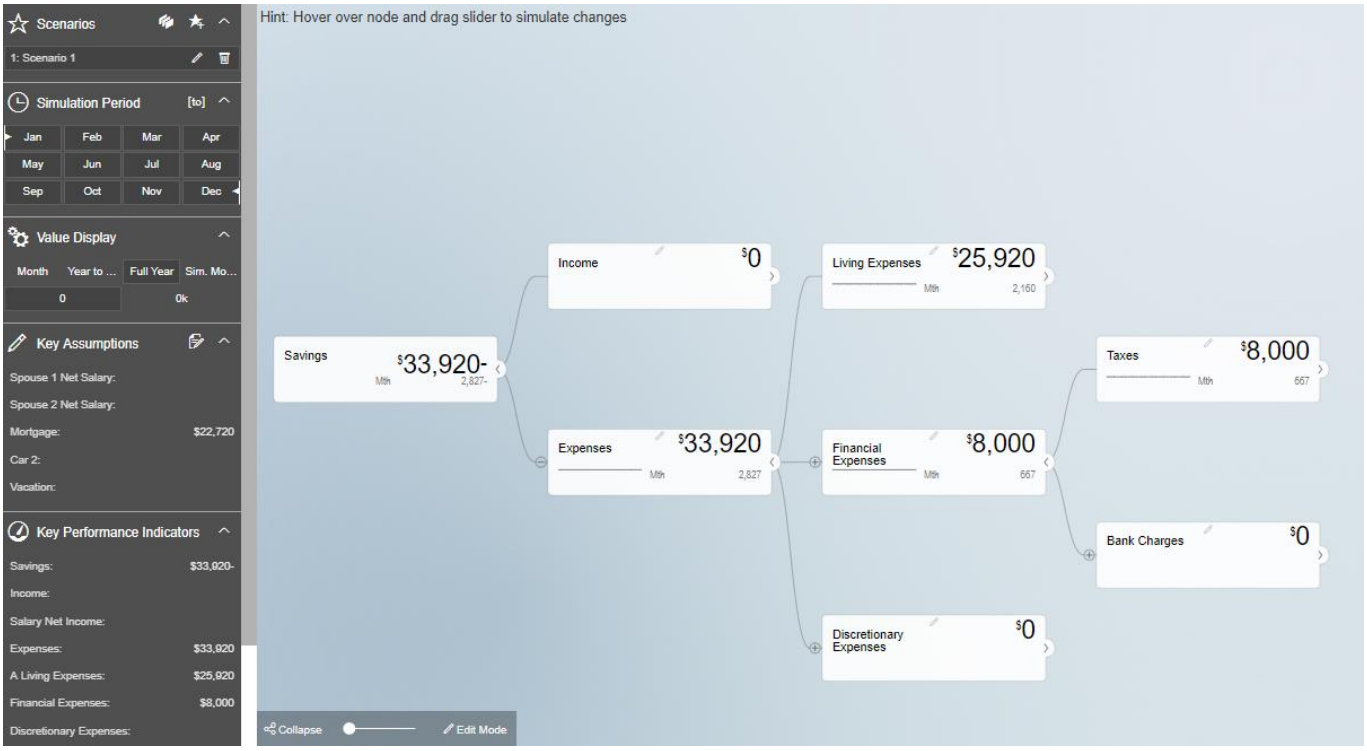


Figure 10.65: Value Driver Tree with Navigational Panel

10.9 Additional Properties of Data Tab

Area	Property	Description
Result Set	Bound Data Result Set	This property sets the Bound Data Result Set.
	Input Query Merger Ref	This property sets the Input Query Merger Reference.
Node Mapping	Node Mapping Field	This property sets the Node Mapping Field. The options are Automatic based on text, Automatic based on key and Manual for Node in Tree Configuration.
	Node ID Separator	This property sets the Node ID Separator.
	Use Manual Data as defaults if sourced nodes not available in result set	This property can be enabled/disabled based on the choice.
Data Series	with Comparison data series	This property enables/disables the option for Comparison data series.
	Data series labels - Primary	This property sets the Primary Data for the Data series labels.
	Data series labels – Comparison	This property sets the Comparison Data for the Data series labels.
	Periods per data series	This property sets the Periods per data series.
	Period labels	This property sets the Periods labels for the data series.

Area	Property	Description
Time Aggregation Display	Active Period - Label	This property sets the Label for the Active Period.
	Active Period - 3 letter Abbreviation	This property sets the 3 letter Abbreviation for the Active Period.
	Active Period - Secondary Value when displayed	This property sets the Secondary Value for the Active Period.
	Total to Period - Label	This property sets the Label for the Total to Period.
	Total to Period - 3 letter Abbreviation	This property sets the 3 letter Abbreviation for the Total to Period.
	Total to Period - Secondary Value when displayed	This property sets the Secondary Value for the Total to Period.
	Total of Periods - Label	This property sets the Label for the Total of Periods.
	Total of Periods - 3 letter Abbreviation	This property sets the 3 letter Abbreviation for the Total of Periods.
	Total of Periods - Secondary Value when displayed	This property sets the Secondary Value for the Total of Periods.
	Selected sim. Interval - Label	This property sets the Label for the Selected Simulation Interval.
	Selected sim. Interval - 3 letter Abbreviation	This property sets the 3 letter Abbreviation for the Selected Simulation Interval.
	Selected sim. Interval - Secondary Value when displayed	This property sets the Secondary Value for the Selected Simulation Interval.

Table 10.5: Data Tab

10.10 Scripting Functions in VDT Component

Below you will find the list of common scripting functions for the VDT Component.

Function/Method	Description
As(type)	This Function casts the component to the specified type. If the component is of the specified type then it is returned as an object of the specified type. If it is not of the specified type, then undefined is returned.
CreateScenario(title)	This Function creates a new Scenario and make it the active scenario.
DeleteAllScenarios()	This Function deletes all scenarios.
DeleteScenario(name)	This Function deletes the Scenario with the unique name from the scenario collection. If the scenario to be deleted is the active scenario, the subsequent scenario is set as the active scenario. There is always one scenario. You are not able to delete it if it is the last one.
EnableEditMode(isEditModeEnabled)	This Function enables or disables the Edit Mode feature.
EnableNodeSimulation(nodeIds, value)	This Function enables or disables Node Simulation.
ExportToExel()	This Function exports to Excel.
GetActiveScenario()	This Function retrieves Active Scenario Id.
GetBottomMargin()	This Function returns the bottom margin of the component.
GetCSSClass()	This Function returns the Cascading Style Sheet (CSS) class name of the component.
GetClickedPeriod()	This Function gets the Selected Period in on Click Event.
GetComparisonVersionTitle()	This Function gets the title for comparison version.
GetDataSource()	This Function returns the assigned data source.
GetExportUrl()	This Function gets the Export URL.
GetHeight()	This Function returns the height of the component.
GetLeftMargin()	This Function returns the left margin of the component.
GetName()	This Function returns the component name. The name is unique in an application or composite, but not in the running state when an application contains composite instances.
GetNodeData(nodes)	This Function gets the data for the nodes. If node ids are not specified then all the nodes will be exported.
GetPeriodColumn()	This Function gets the Period Column.
GetPeriodToColumn()	This Function gets the End Period Column for simulation.
GetPeriods()	This Function returns the number of periods per data series.

Function/Method	Description
GetPeriodsLabels()	This Function gets the Period Labels.
GetPrimaryDisplayValue()	This Function gets whether Selected Period or Full total is displayed as the primary value on nodes.
GetPrimaryVersionTitle()	This Function gets the title for primary version.
GetRightMargin()	This Function returns the right margin of the component.
GetScale()	This Function gets the Scaling for auto scale nodes.
GetSelectedNode()	This Function gets the Selected Node.
GetSourceRef()	This Function gets Referenced Input Data Source Merger.
GetTopMargin()	This Function returns the top margin of the component.
GetTopNode()	This Function gets the Top Node.
GetWidth()	This Function returns the width of the component.
GetconfigServerMap(clientURL)	This Function gets the external configuration Map URL.
HideLoadingState()	This Function hides the loading indicator on the component.
IsEditModeEnabled()	This Function returns whether the Edit Mode feature is enabled or disabled.
IsVisible()	This Function returns whether the component is visible.
NodesToExport(nodes)	This Function specifies the nodes to be exported. If node ids are not specified then all the nodes will be exported.
SetActiveDimensionalNodeData(dimensionalNode, vdt, node)	This Function sets the active Dimensional Node Config.
SetBottomMargin(bottomMargin)	This Function sets the bottom margin of the component.
SetCSSClass(className)	This Function sets the Cascading Style Sheet (CSS) class name of the component.
SetComparisonVersionTitle(title)	This Function sets the Title to be shown for comparison version. Often the title of a variable such as Budget 18.
SetConfigServerMap(clientURL, serverURL)	This Function sets the external configuration URL (expert function), sets the target url called to retrieve external configuration when this mode is selected. The URL can be a relative or fully qualified URL. Sometimes, you may want to parse parameters to the url such as user selected or dynamically parsed parameters. These can be added to the URL and parsed to the receiving server as part of the query string.
SetDataSource(dataSourceAlias)	This Function assigns a data source.
SetExportUrl(url)	This Function sets the target url called on export event if URL is chosen as the export mode. The URL can be a relative or fully qualified URL. Sometimes, you may

Function/Method	Description
	want to parse parameters to the url such as user selected or dynamically parsed parameters. These can be added to the URL and parsed to the receiving server as part of the query string.
SetFocusNode(nodeName)	This Function sets the Focus to a specific node.
SetHeight(height)	This Function sets the height of the component.
SetLeftMargin(leftMargin)	This Function sets the left margin of the component.
SetNodeStyle(style)	This Function sets the Node Visual Style.
SetNodeVisualStyle(nodeIds, value)	This Function sets the Node Visual Style : Normal -> 'N', Derived -> 'D', Hidden -> 'H'
SetPeriodColumn(PeriodColumn)	This Function sets the Period Column.
SetPeriodToColumn(PeriodColumn)	This Function sets the End Period Column for simulation.
SetPeriods(periods)	This Function sets the Total number of periods per data series. (i.e. 12 for 12 months).
SetPeriodsLabels()	This Function sets the Period Labels.
SetPrimaryDisplayValue(primaryDisplayValue)	This Function sets whether selected period value or full total value is displayed as primary value on nodes.
SetPrimaryVersionTitle(title)	This Function sets the Title to be shown for primary version. Often the title of a variable such as Forecast 18.
SetRightMargin(rightMargin)	This Function sets the right margin of the component.
SetScale(scale)	This Function sets the Scaling for auto scale nodes.
SetScenario(name)	This Function sets a scenario from the scenario collection to the active scenario.
SetSourceRef(component_id)	This Function sets the Input Data Source Merger.
SetTimeAggregationLabels(valTextSper, valTextLper, valTextSptd, valTextLptd, valTextStotal, valTextLtotal, valTextSsel, valTextLsel)	This Function sets the Time Aggregation labels and 3 letter abbreviations.
SetTopMargin(topMargin)	This Function sets the top margin of the component.
SetTopNode(nodeName)	This Function sets the Top Node.
SetVisible(isVisible)	This Function shows or hides the component.
SetWidth(width)	This Function sets the width of the component.
ShowCompareScenarios(topNode, assumpTitle, assumpList, kpiTitle, kpiList)	This Function shows the Scenario Comparison Popup. Calling this functions calculates all scenarios in the collection and shows a popup comparison view.
ShowComponent(component)	This Function shows the Component.
ShowDetail(nodeName)	This Function shows the Detail Panel.
ShowEditInputs(assumpList)	This Function shows the Input Edit Popup for the list parsed. Calling this functions shows a tabular dialog where you can maintain the period values for the

Function/Method	Description
	parsed list of Drivers You can add nodes to the list by adding their technical name. You can also add group headings by adding a simple text item.
ShowLoadingState(text)	This Function shows a loading indicator on the component.

Table 10.6: Scripting Functions for VDT Component

10.11 VDT Navigation Panel Component

The scenario based functionalities of the VDT Navigation Panel has been explained in the previous Sections (Section 4 to Section 9).

10.12 Scripting Functions in VDT Navigation Panel Component

Below you will find the list of common scripting functions for the VDT Navigation Panel Component.

Function/Method	Description
ShowEditInputs(assumpList)	This Function shows the Input Edit Popup for the list parsed. Calling this functions shows a tabular dialog where you can maintain the period values for the parsed list of Drivers You can add nodes to the list by adding their technical name. You can also add group headings by adding a simple text item.
As(type)	This Function casts the component to the specified type. If the component is of the specified type then it is returned as an object of the specified type. If it is not of the specified type, then undefined is returned.
CreateScenario(title)	This Function creates a new Scenario and make it the active scenario.
DeleteScenario(name)	This Function deletes the Scenario with the unique name from the scenario collection. If the scenario to be deleted is the active scenario, the subsequent scenario is set as the active scenario. There is always one scenario. You are not able to delete it if it is the last one.
ExportToExel()	This Function exports to Excel.
GetAssumpList()	This Function gets the Assumptions List (The list of node id's or Simple text for inline headers).
GetAssumpTitle()	This Function gets the Assumptions Section Title.
GetAssumpVisibility()	This Function gets the Assumptions Section visibility.
GetBottomMargin()	This Function returns the bottom margin of the component.
GetCSSClass()	This Function returns the Cascading Style Sheet (CSS) class name of the component.
GetConstraintsList()	This Function gets the Constraints List (List of node id's or Simple text for inline headers).
GetConstraintsTitle()	This Function gets the Constraints Section Title.
GetConstraintsVisibility()	This Function gets the Constraints Section visibility.
GetFilterLists()	This Function gets the Filter Lists.
GetHeight()	This Function returns the height of the component.
GetKPIList()	This Function gets the KPI's List (List of node id's or Simple text for inline headers).
GetKPITitle()	This Function gets the KPI's Section Title.
GetKPIVisibility()	This Function gets the KPI's Section visibility.

Function/Method	Description
GetLeftMargin()	This Function returns the left margin of the component.
GetName()	This Function returns the component name. The name is unique in an application or composite, but not in the running state when an application contains composite instances.
GetPeriodsTitle()	This Function gets the Simulation Period Section Title.
GetPeriodsVisibility()	This Function gets the Simulation Period Section visibility.
GetRightMargin()	This Function returns the right margin of the component.
GetScalesList()	This Function gets the Selectable Scales List (List of selectable scales currently configured).
GetScenariosTitle()	This Function gets the Scenarios Section Title.
SetScenariosVisibility(visibility)	This Function sets the Scenarios Section visibility. Calling this functions, you can set the section visibility.
SetShowTitles(show)	This Function sets the Display Section Headers. Calling this functions, you can change the display of section headers.
SetTopMargin(topMargin)	This Function sets the top margin of the component.
SetTopNode(nodeName)	This Function sets the Top Node.
SetTopNodesList(list)	This Function sets the Top Nodes List. Calling this functions, you can set the selectable top node list for the panel You can add nodes to the list by adding their technical name. You can also add group headings by adding a simple text item.
SetTopNodesTitle(title)	This Function sets the Top Nodes Section Title. Calling this functions, you can set the section header title.
SetTopNodesVisibility(visibility)	This Function sets the Top Nodes Section visibility. Calling this functions, you can set the section visibility.
SetValueDisplayList(list)	This Function sets the Selectable Primary Value Displays. Calling this functions, you can set the list of Primary Time Aggregations that users can select. Must be from the following list: 'per', 'ptd', 'total', 'sel'.
SetValueDisplayTitle(title)	This Function sets the Value Display Section Title. Calling this functions, you can set the section header title.
SetValueDisplayVisibility(visibility)	This Function sets the Value Display Section visibility. Calling this functions, you can set the section visibility.
SetValueDriverTree(component_id)	This Function sets the Linked Value Driver Tree. Calling this functions, you can change the linked value driver tree.
SetVisible(isVisible)	This Function shows or hides the component.
SetVisualStyleList(list)	This Function sets the Selectable Visual Style. Calling this functions, you can set the list of Visual Styles that users can select. Must be from the following list: 'S'=Standard, 'F'=Full, 'M',=Minimal.
SetVisualization(style)	This Function sets the Node Visual Style.
SetVisualizationStyle(style)	This Function sets the Node Visual Style.

Function/Method	Description
SetWidth(int width)	This Function sets the width of the component.
ShowCompareScenarios(topNode, assumpTitle, assumpList, kpiTitle, kpiList)	This Function shows the Scenario Comparison Popup for the list parsed. Calling this functions calculates all scenarios in the collection and shows a popup comparison view. The lines in the comparison report is defined by the list parsed as an input. You can add nodes to the list by adding their technical name. You can also add group headings by adding a simple text item.
ShowLoadingState(text)	This Function shows a loading indicator on the component.

Table 10.7: Scripting Functions for VDT Navigation Panel Component

11 Value Driver Tree result as Data Source

As part of VDT Release 2.0.5, the simulation result can be consumed in the dashboard in the form of “Value Driver Tree Output Data Source” and configures the component to get values for desired nodes.

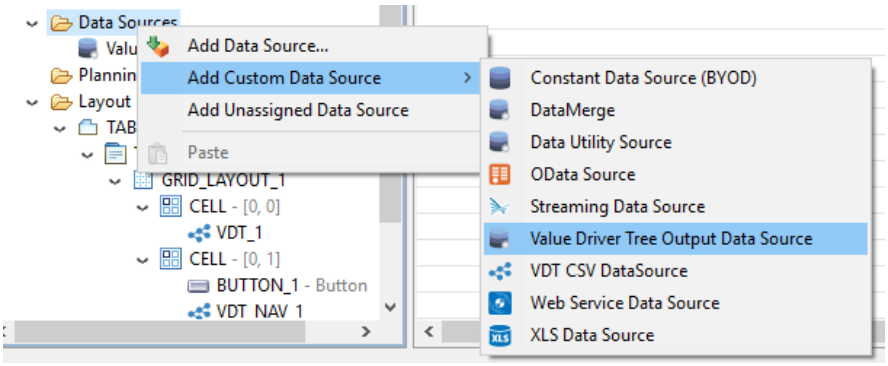


Figure 11.1: Value Driver Tree Output Data Source

The “Value Driver Tree Output Data Source” component should be associated with a “Value Driver Component” using the property “Linked Value Driver Tree”.

In the Standard Properties, provide the ID of the nodes whose data that should be available in the Data Source in the property “KPI’s and Drivers to be exported”

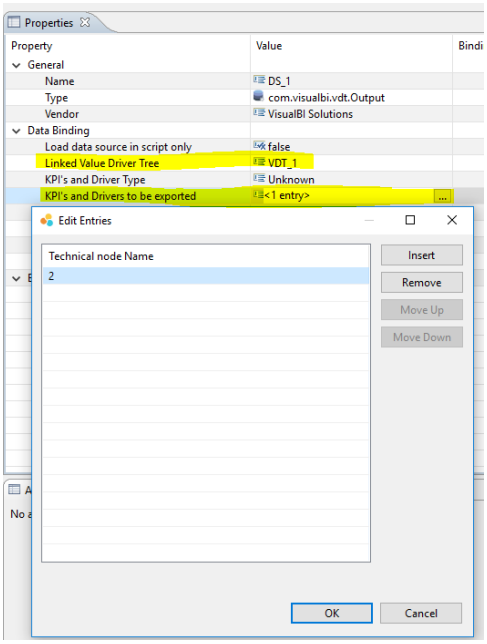


Figure 11.2: Standard Properties

12 Excel as Data Source

As part of VDT Release 2.0.5, you will be able to use Excel Sheet as Data Source. For our example, we have assigned Google Spreadsheet and it exists in the below format.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R
1		Primary	Primary	Primary	Primary	Primary	Primary	Primary	Primary	Primary	Primary	Primary	Primary	Comparison	Comparison	Comparison	Comparison	Comparison
2		Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Sep	Oct	Nov	Dec	Jan	Feb	Mar	Apr	May
3	1																	
4	2																	
5	3																	
6	4	0.1	0.1	0.1	0.1	0.05	-0.05	-0.05	-0.05	-0.05	-0.05	-0.05	-0.05	0	0	0	0	
7	5	0.8123	0.6123	0.6307	0.6875	0.6669	0.7203	0.7851	0.77	0.83	0.81	0.79	0.79	0.62	0.64	0.69	0.75	0.8
8	6	82	80.36	81.16	81.97	84.43	86.96	86.09	81	81	81	81	81	81	81	81	81	8
9	7																	
10	8	2.6	2.68	2.84	2.78	2.95	2.98	2.98	2.4	2.4	2.4	2.4	2.4	2.3	2.3	2.3	2.3	2.
11	9																	
12	10																	
13	11																	
14	12	1510	1510	1540.2	1524.8	1540.05	1540.05	1540.05	1500	1500	1500	1500	1500	1500	1500	1500	1500	150
15	13																	
16	14																	
17	15	744	672	744	720	744	720	744	720	744	720	744	720	744	672	744	720	74
18	16																	
19	17																	
20	18																	
21	19	20	20	22	21	22	40	50	10	10	10	10	10	10	10	10	10	1
22	20	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	
23	21	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	
24	22																	
25	23																	
26	24	152	152	153.52	156.59	151.89	153.41	148.81	150	150	150	150	150	150	150	150	150	15
27	25																	
28	26																	
29	27																	
30	28	64	64	64	64	64	64	64	64	64	64	64	64	64	64	64	64	6
31	29	85	84.15	82.47	80	79.2	79.99	78.39	84	84	84	84	84	84	84	84	84	8

Figure 12.1: Sample Excel Data Source

Note: In the above Google Spreadsheet example, the index starts from zero.

In the Outline panel, set the Custom Data Source as the XLS Data Source (assuming that the above sample excel data source is being assigned) as shown in the Figure below.

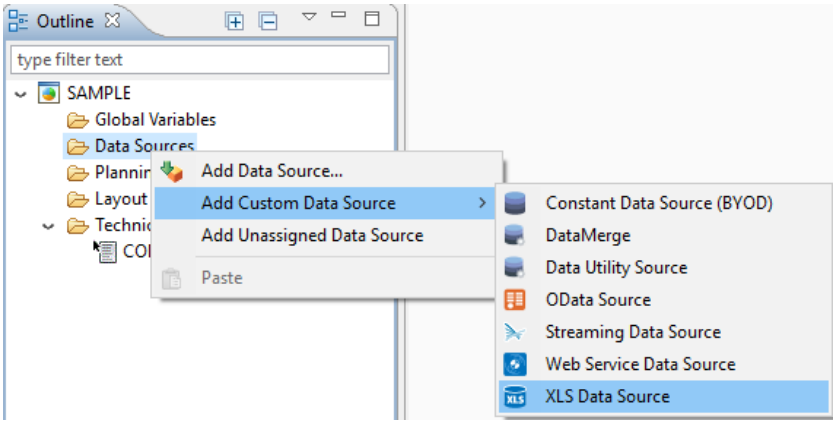


Figure 12.2: Outline

Now navigate to the category General and to the sub category General Settings of the Custom Data Source. Now set the property Type of Spreadsheet to the option Google Spreadsheet and set the File Location (see Figure below).

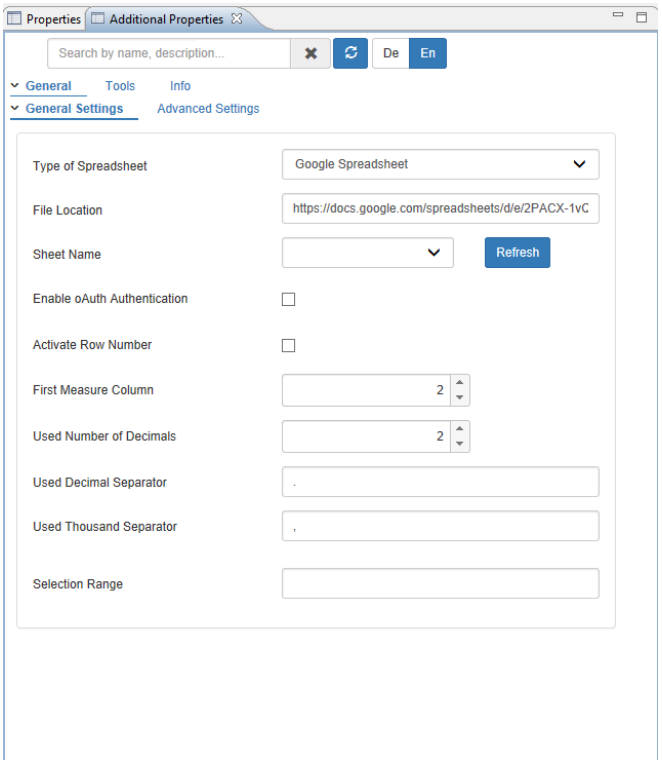


Figure 12.3: Category General and Sub category General Settings

Now navigate to the category General and to the sub category Advanced Settings of the Custom Data Source. For our example, activate the property Enable Column Dimensions. Set the property First Data Cell Column Index to the value 1, First Data Cell Row Index to the value 2 and Measure Index in Column to the value 2 (Here for the Measure, the index starts from the value 1. The dashboard will be displayed with the data assigned from the Google Spreadsheet.

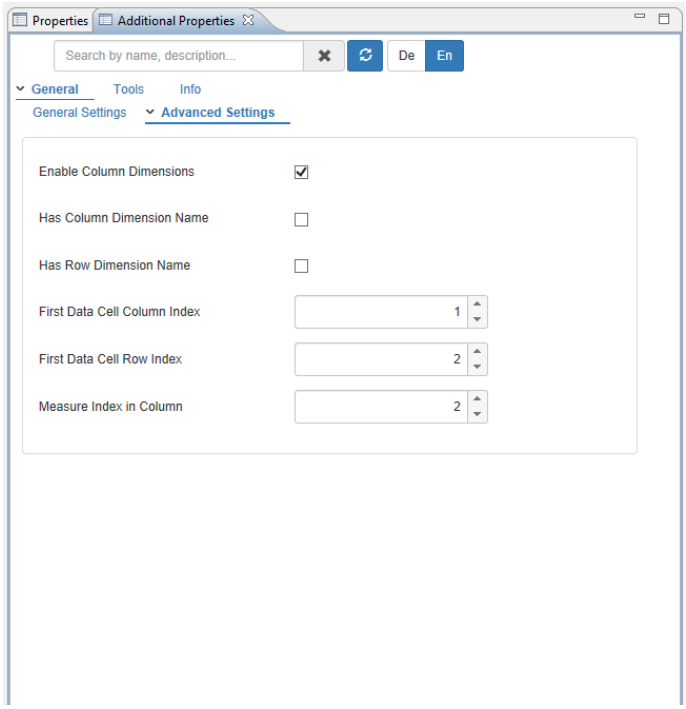


Figure 12.4: Category General and Sub category Advanced Settings

13 CSV Data Source

As part of VDT Release 3.0, you will be able to use CSV Data Source for the Value Driver Tree as part of the Data Source configuration.

The below steps explains on how to configure the CSV Data Source for the Value Driver Tree.

1. Create a new project in SAP Lumira Designer.
2. For our example, create a Layout as shown in the below Figure.

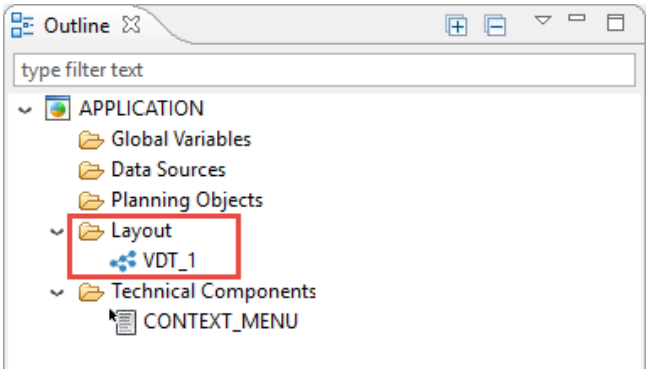


Figure 13.1: Layout

3. Assign the VDT Component to the Layout as in the Figure above.
4. Now assign the Custom Data Source by navigating to the Data Source • Add Custom Source • VDT CSVDataSource (see Figure below).

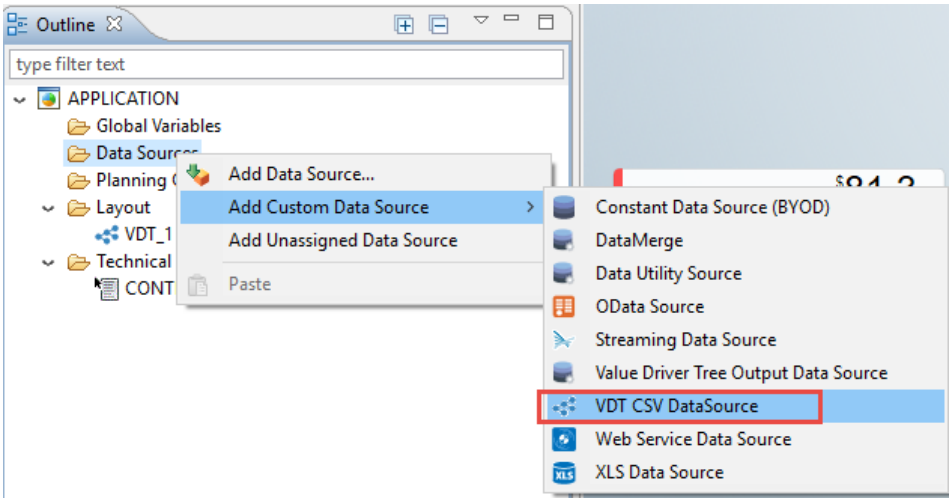


Figure 13.2: Assigning VDT CSV Data Source

5. Now the Layout looks similar to the Figure shown below.

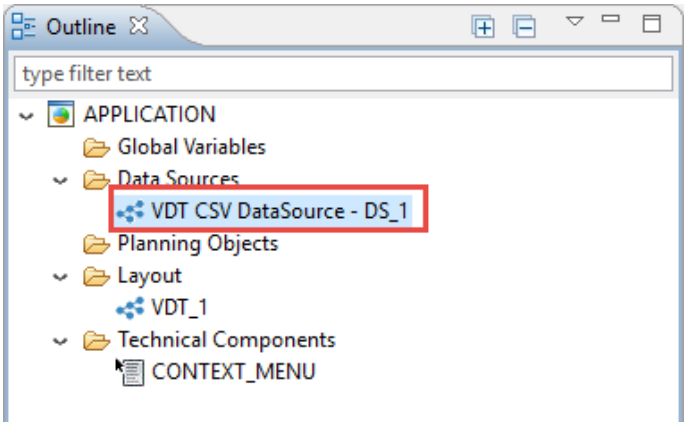


Figure 13.3: Assigned VDT CSV Data Source

6. Now navigate to the Standard Properties of the VDT CSV Data Source (see Figure below).

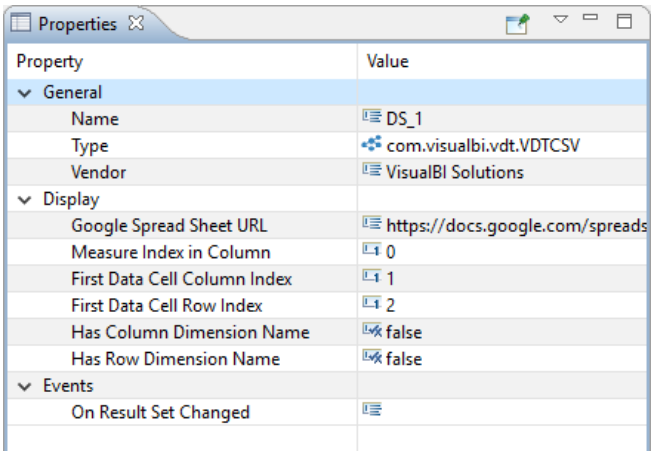
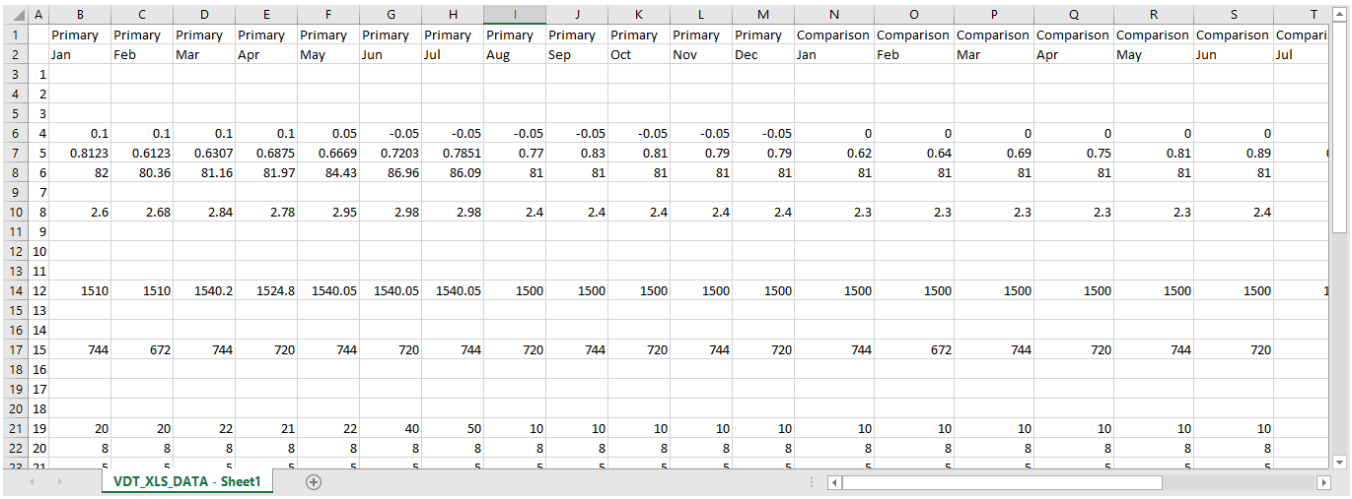


Figure 13.4: Standard Properties of the VDT CSV Data Source

7. For our example, set the property Google Spread Sheet URL to the respective URL Link based on our choice and the sample Data Source will be similar to the data shown below.



	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T
1		Primary	Primary	Primary	Primary	Primary	Primary	Primary	Primary	Primary	Primary	Primary	Primary	Comparison	Comparison	Comparison	Comparison	Comparison	Comparison	Compari
2		Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Sep	Oct	Nov	Dec	Jan	Feb	Mar	Apr	May	Jun	Jul
3	1																			
4	2																			
5	3																			
6	4	0.1	0.1	0.1	0.1	0.05	-0.05	-0.05	-0.05	-0.05	-0.05	-0.05	-0.05	0	0	0	0	0	0	
7	5	0.8123	0.6123	0.6307	0.6875	0.6669	0.7203	0.7851	0.77	0.83	0.81	0.79	0.79	0.62	0.64	0.69	0.75	0.81	0.89	
8	6	82	80.36	81.16	81.97	84.43	86.96	86.09	81	81	81	81	81	81	81	81	81	81	81	
9	7																			
10	8	2.6	2.68	2.84	2.78	2.95	2.98	2.98	2.4	2.4	2.4	2.4	2.4	2.3	2.3	2.3	2.3	2.3	2.4	
11	9																			
12	10																			
13	11																			
14	12	1510	1510	1540.2	1524.8	1540.05	1540.05	1540.05	1500	1500	1500	1500	1500	1500	1500	1500	1500	1500	1500	1
15	13																			
16	14																			
17	15	744	672	744	720	744	720	744	720	744	720	744	720	744	672	744	720	744	720	
18	16																			
19	17																			
20	18																			
21	19	20	20	22	21	22	40	50	10	10	10	10	10	10	10	10	10	10	10	
22	20	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	
23	21	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	

VDT_XLS_DATA - Sheet1

Figure 13.5: Sample CSV Data Source

8. Set the property Measure Index in Column to the value 1 (see Figure 13.4).
9. Set the property Measure First Data Cell Column Index to the value 2 (see Figure 13.4).

10. Set the property Measure First Data Cell Row Index to the value 2 (see Figure 13.4).
11. Set the property Has Column Dimension Name to the option True (see Figure 13.4).
12. Set the property Has Row Dimension Name to the option False (see Figure 13.4).
13. Now assign the VDT CSV Data Source to the VDT Component (see Figure below).

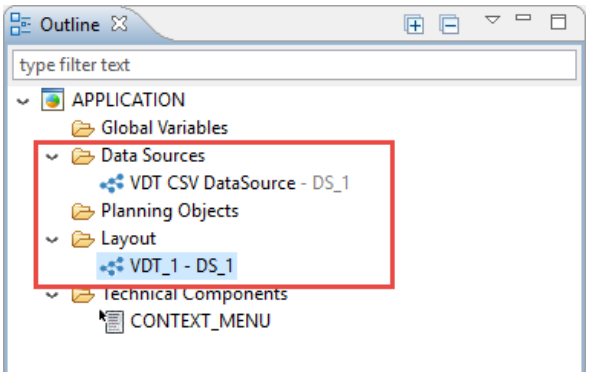


Figure 13.6: VDT CSV Data Source assigned to VDT Component

14. Now you will be able to view the VDT in Run time with the Nodes having the values based on the Data Source as shown in step 7. (see Figure below).

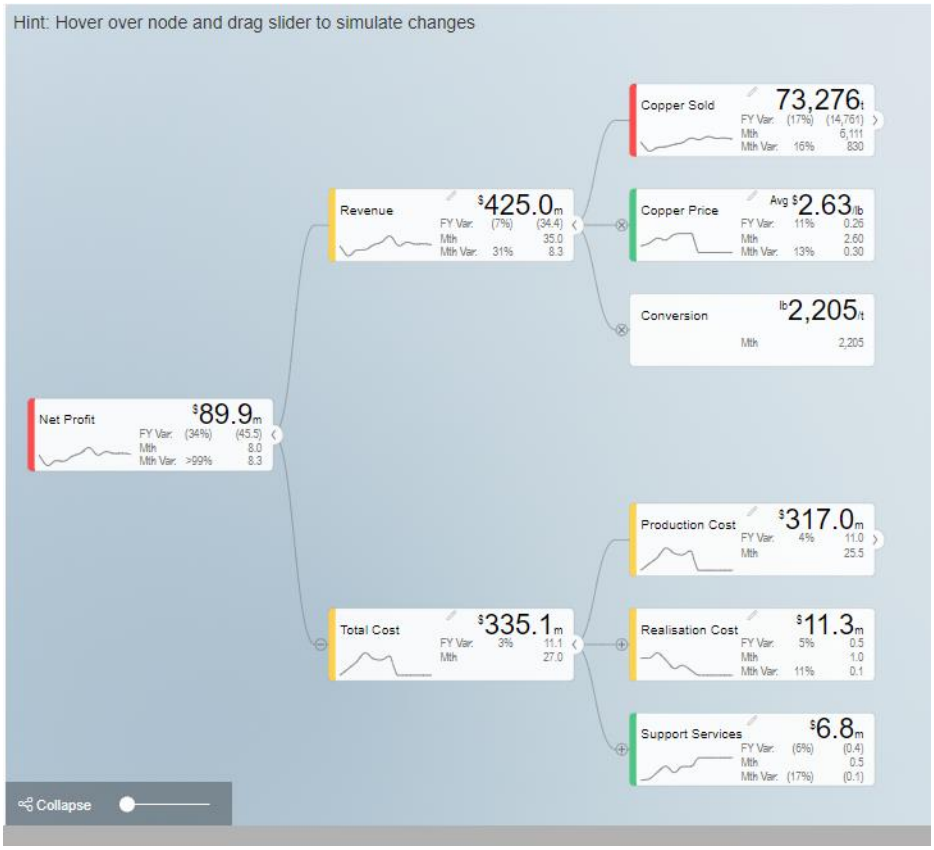


Figure 13.7: VDT with Nodes having the CSV Data Source values

Note: If the Calc. Method in the Tree Tab is selected as "Data Source" option then the VDT can be visualized based on the assigned Data Source. If the Calc. Method is selected as "Manual" or if there is no Data Source, then the VDT can be visualized based on the Primary and Comparison Data.

14 Edit Scenario and Description

As part of VDT Release 2.0.5, by clicking the Pencil icon in the Navigation Panel, Edit Scenario window will get displayed allowing the user to edit the “Scenario Title” and “Scenario Description”.

By default, the “Scenario Description” will be populated based on the simulation executed in the application.

When the user manually enters the comment in the “Scenario Description” then the auto populated simulation message will not get displayed further but the user entered text can be edited at any point of time (see Figure below).

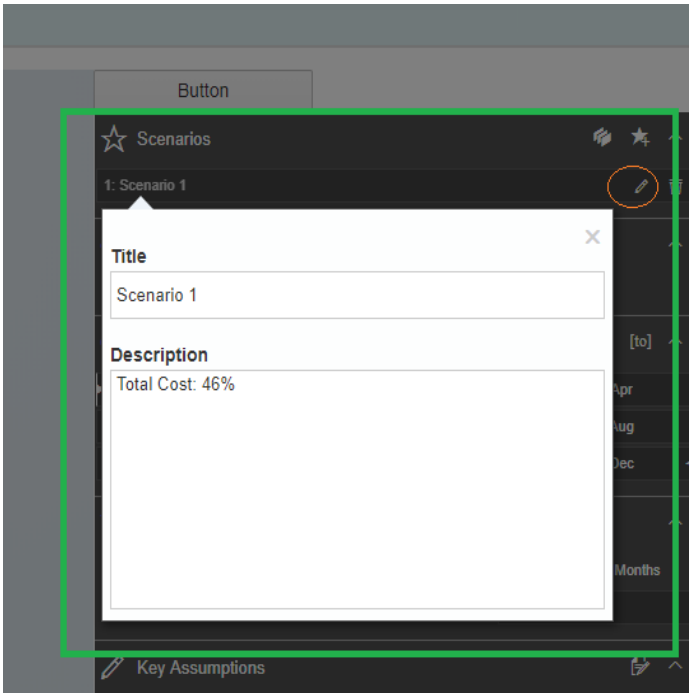


Figure 14.1: : Edit Scenario Window

15 Export Config

As part of VDT Release 2.0.5, after the dashboard completes loading, press “Ctrl + Alt + Shift + C” key combination to download configuration file related to VDT and Navigation Panel (see Figure below).

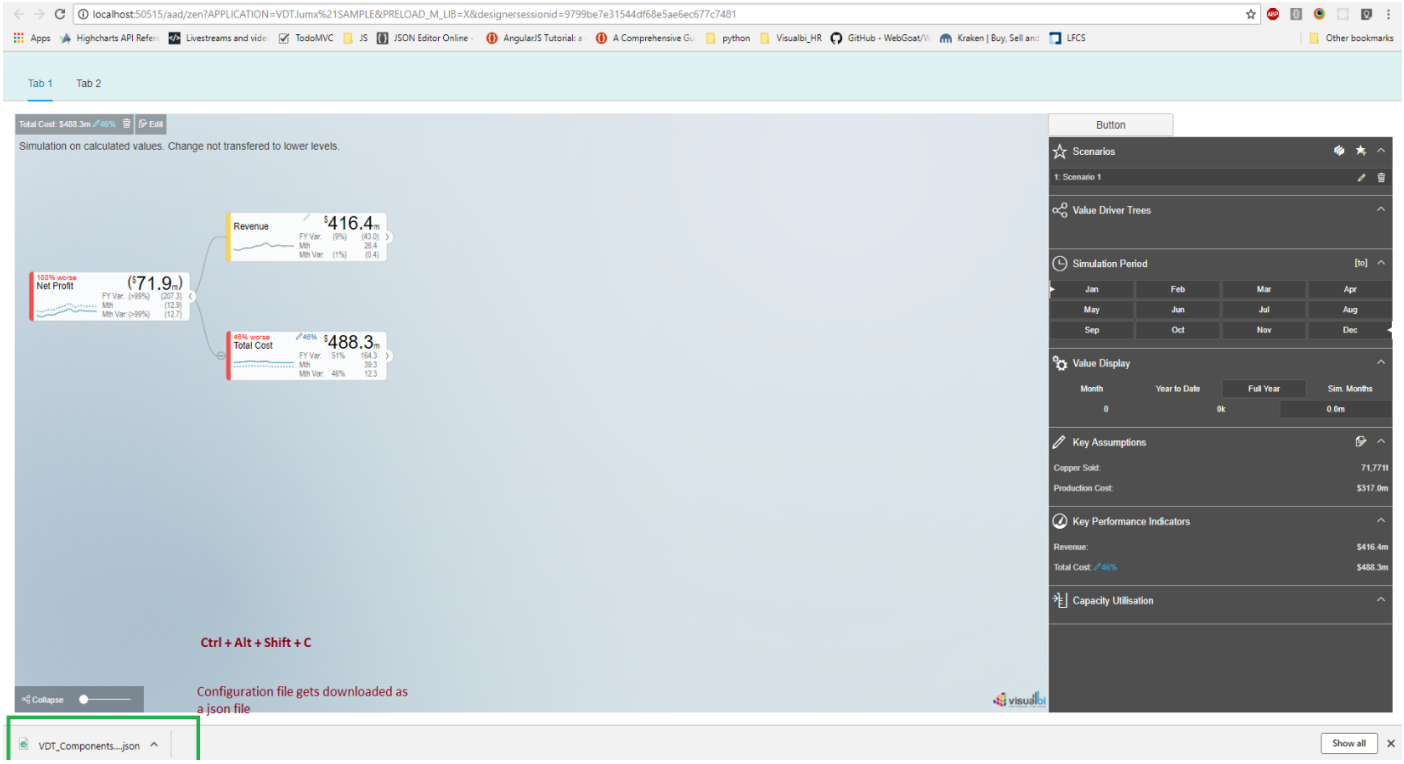


Figure 15.1: Downloading Configuration File

After the dashboard completes loading, use “Ctrl + Alt + Shift + N” key combination to download Node Snapshot (see Figure below).

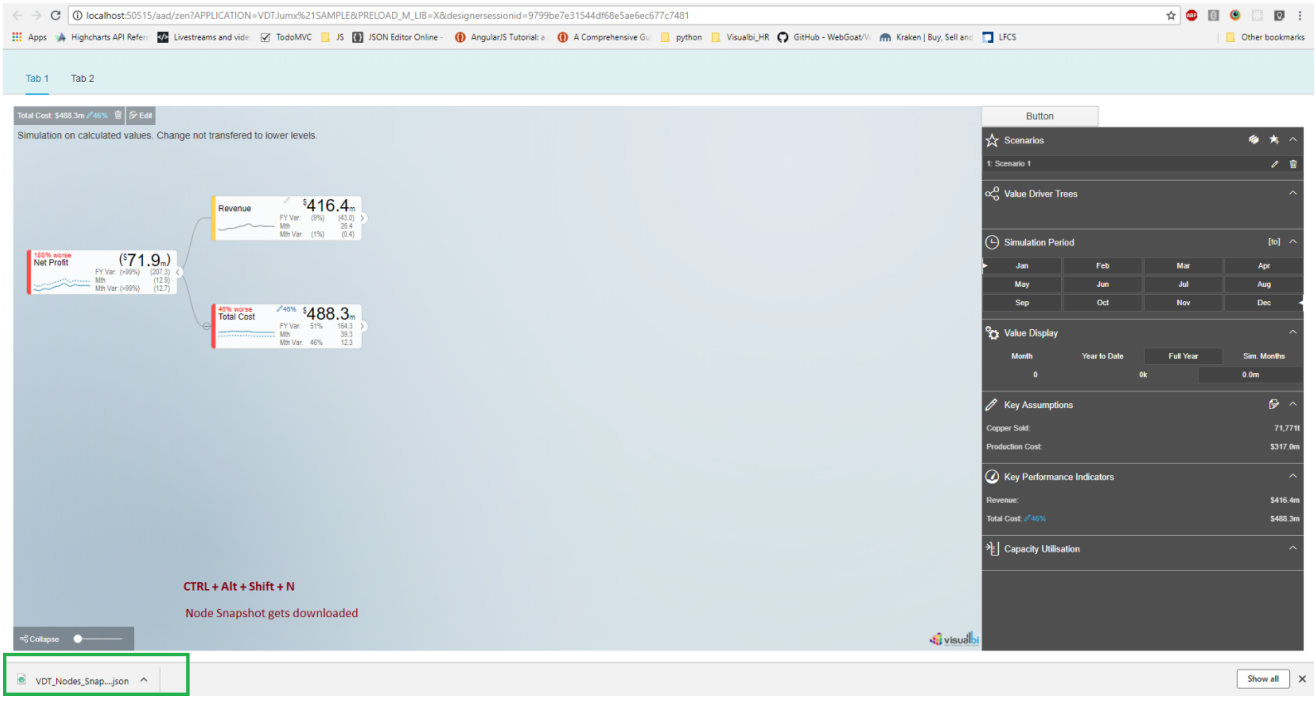


Figure 15.2: Downloading Node Snapshot

16 Detail Page Component

As part of VDT Release 3.0, a new Detail Page Component has been included which is used to show a Custom Detail Page in the Run time instead of the Standard Page.

The steps below explains on how you will be able to create the Custom Detail Page Component in VDT:

1. For our example, create a layout in SAP Lumira Designer by assigning the Detail Page Component as shown in the below Figure.

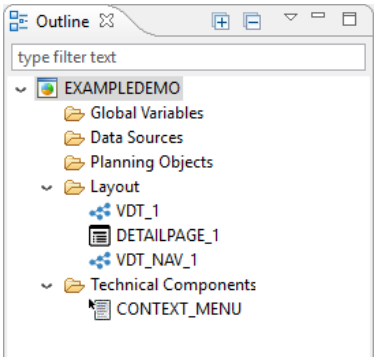


Figure 16.1: Layout

2. Now navigate to the Settings Tab of the Value Driver Tree. In the area Events, set the property Default Action to the option Show Custom Detail Pop up (see Figure below).

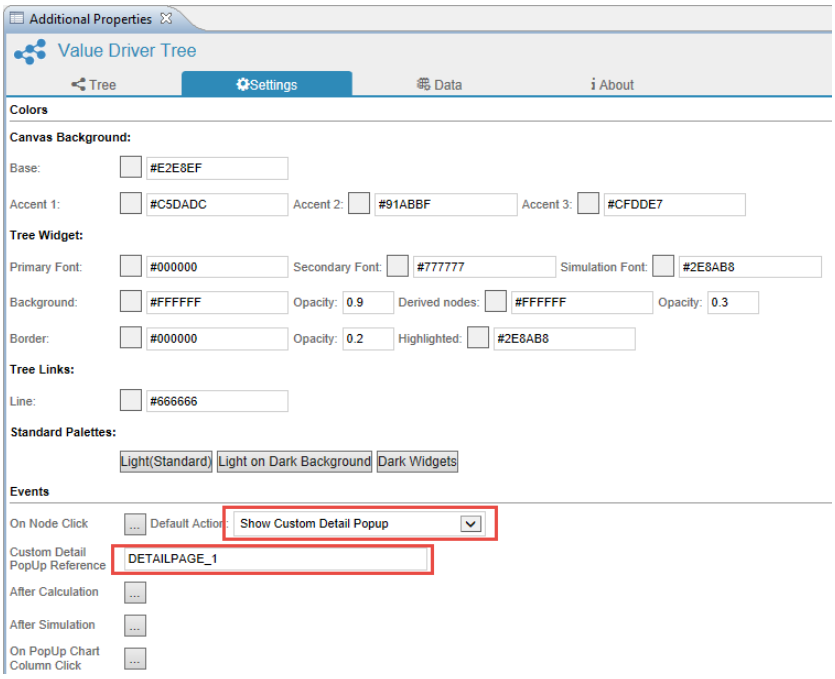


Figure 16.2: Settings Tab - Events

3. Set the property Custom Detail PopUp Reference to the Detail Page as “DETAILPAGE_1”.
4. Navigate to the Additional Properties of the DETAILPAGE Component (see Figure below).

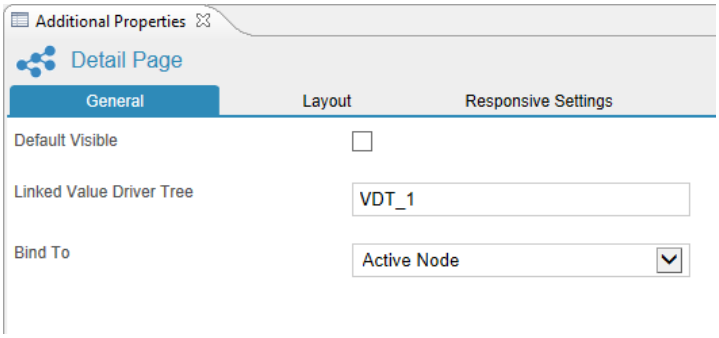


Figure 16.3: General Tab

5. In the General Tab, the property Default Visible is not activated in our example.
6. Set the property Linked Value Driver Tree to the Value Driver Tree as “VDT_1”.
7. Set the property Bind To the option Active Node.
8. Navigate to the Layout Tab as shown in the below Figure.

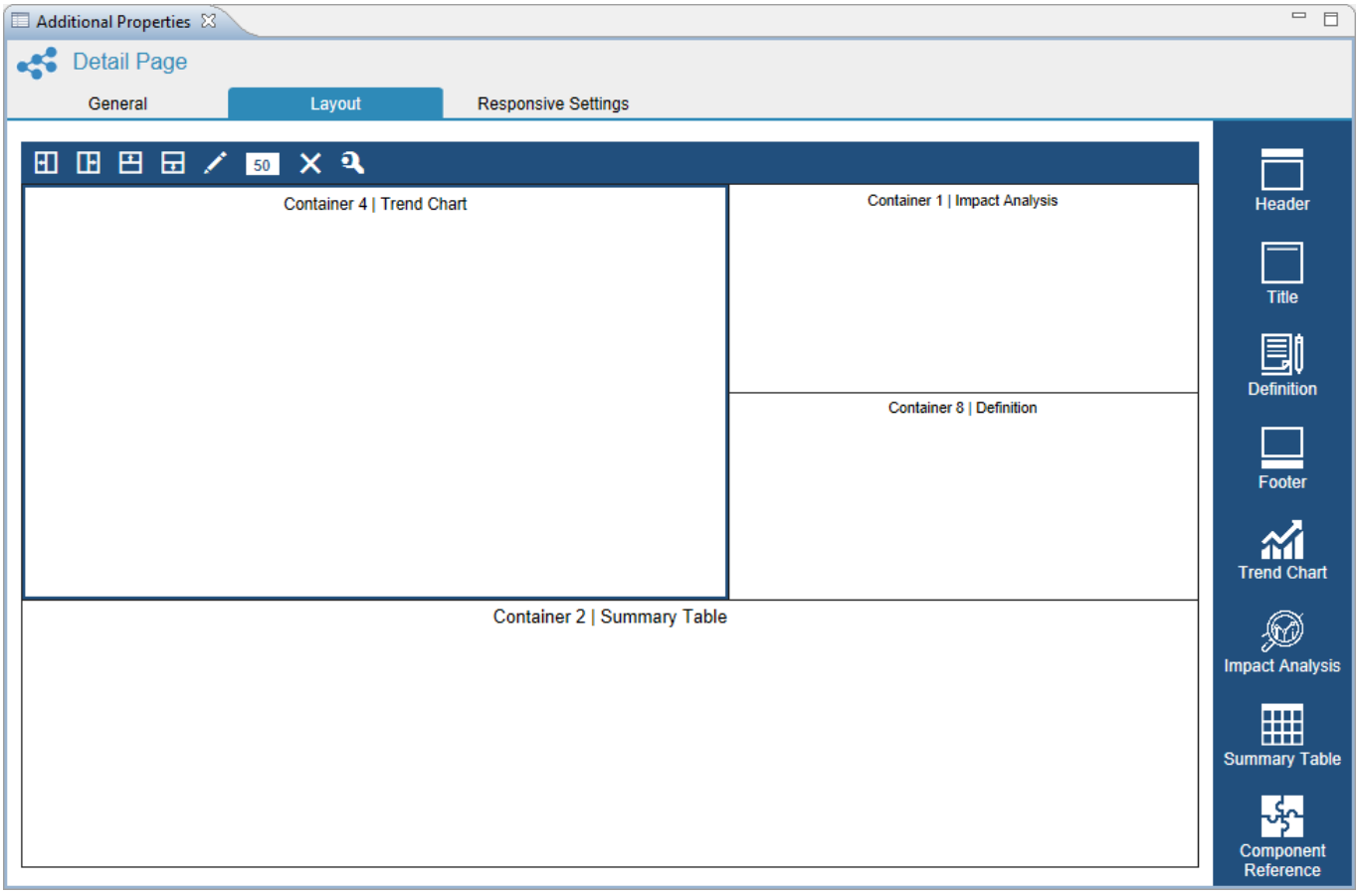


Figure 16.4: Layout Tab

9. In the Layout Tab, create four containers and assign the container components as shown in the above Figure.
10. Navigate to the Responsive Settings Tab. Now you can observe the responsive settings of the Containers and their components (see Figure below).

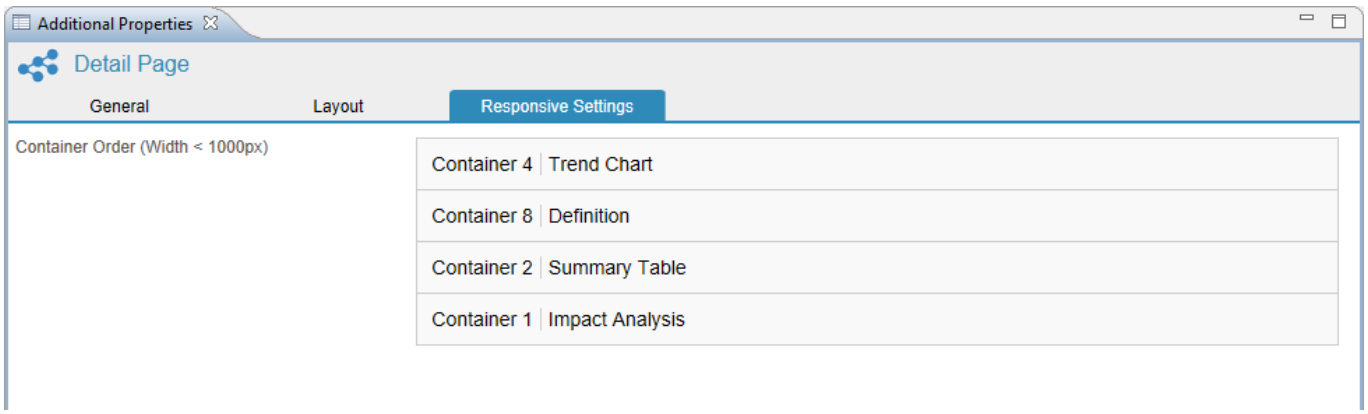


Figure 16.5: Responsive Settings Tab

11. In the Run time you will be able to view the configured Tree with Nodes as shown in the below Figure.

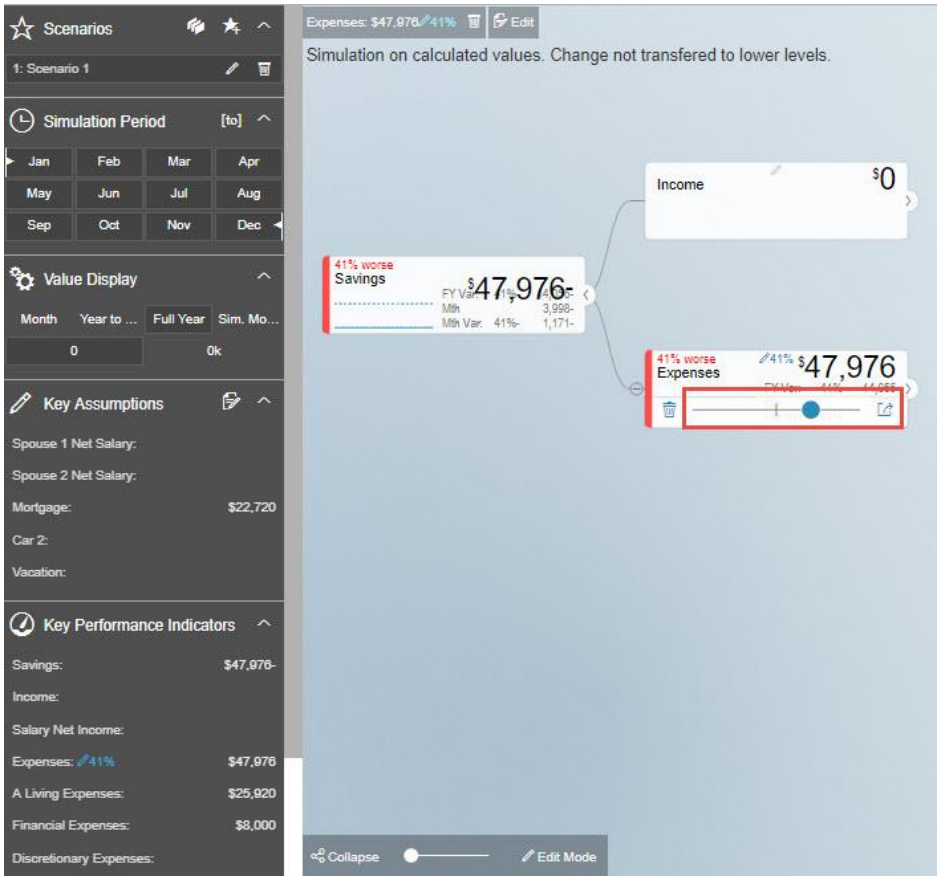


Figure 16.6: Simulated Node

12. For our example, you have simulated the Node as indicated in the above Figure. Now when you click that Node in the Runtime, you will be able to observe that a Custom Detail page for the Node will be displayed. The Custom Detail Page will be displayed based on the Layout being assigned with the containers (see Figure below).

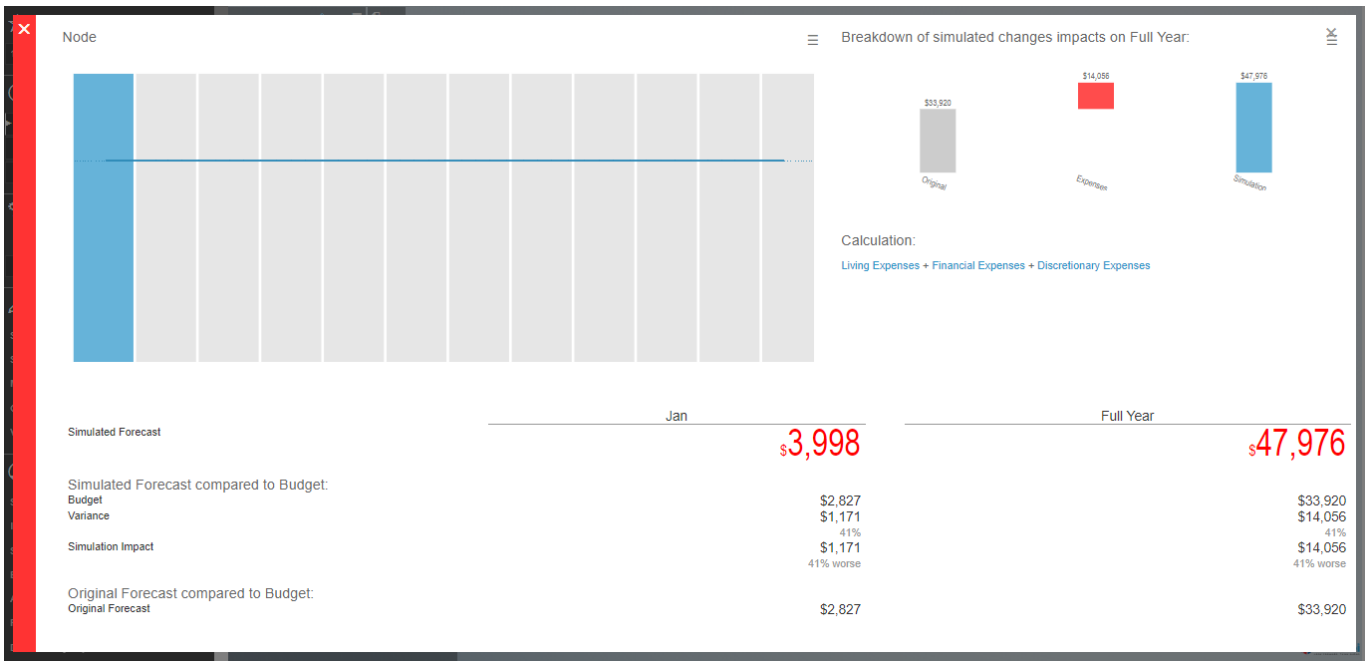


Figure 16.7: Custom Detail Page

16.1 Additional Properties of Detail Page Component

Tab	Property	Description
General	Default Visible	By enabling this property, the component will be visible by default otherwise the component will be hidden and it will be visible only when some event is triggered.
	Linked Value Driver Tree	This property sets the Value Driver Tree that links the Detail Page.
	Bind To	This property sets the Node from the Linked VDT that should be bound to the Detail Page.

Table 16.1: Detail Page Component

17 Single Data Series

As part of VDT Release 3.0, you will be able to configure the VDT Tree with the different set of Data Series options as listed below:

- Primary Data Series only
- Primary Data Series with Comparison Data Series
- Single Data Point
- Single Data Point with Comparison

The following sections will explain the four different configurations implemented for the Data Series Feature in VDT Tree.

17.1 Primary Data Series

The steps below explains on how you will be able to configure the Primary Data Series in VDT:

1. For our example, create a layout in SAP Lumira Designer as shown in the below Figure.

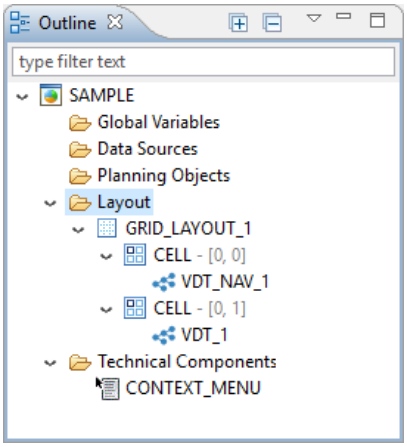


Figure 17.1: Layout

2. For our example, a Grid Layout has been created with two cells having one cell assigned with the VDT Tree and the other cell assigned with VDT Navigation Panel (see Figure above).
3. Navigate to the Additional Properties of the Value Driver Tree and create a VDT Tree with the Parent Node labelled as “Sum” and two Child Nodes “Value A” and “Value B” as shown in the below Figure.

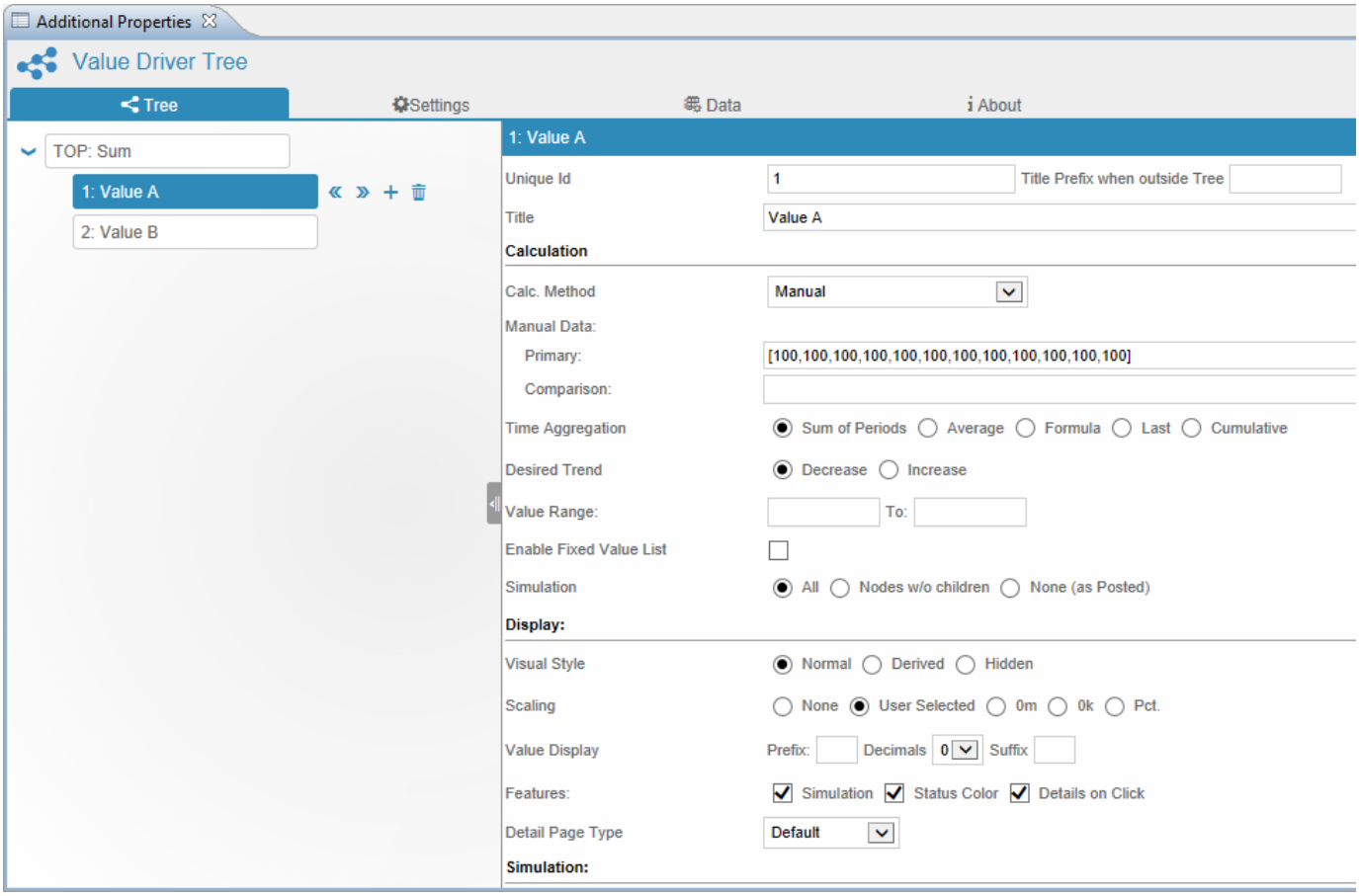


Figure 17.2: Properties Sheet for Node Value A

4. Now navigate to the Properties Sheet for the Node “Value A”. In the area Calculation, set the property Calc. Method to the option Manual.
5. Set the Primary Data to the value [100, 100, 100, 100, 100, 100, 100, 100, 100, 100, 100, 100].
6. Set the property Time Aggregation to the option Sum of Periods.
7. Now navigate to the Properties Sheet for the Node “Value B”. In the area Calculation, set the property Calc. Method to the option Manual (see Figure below).

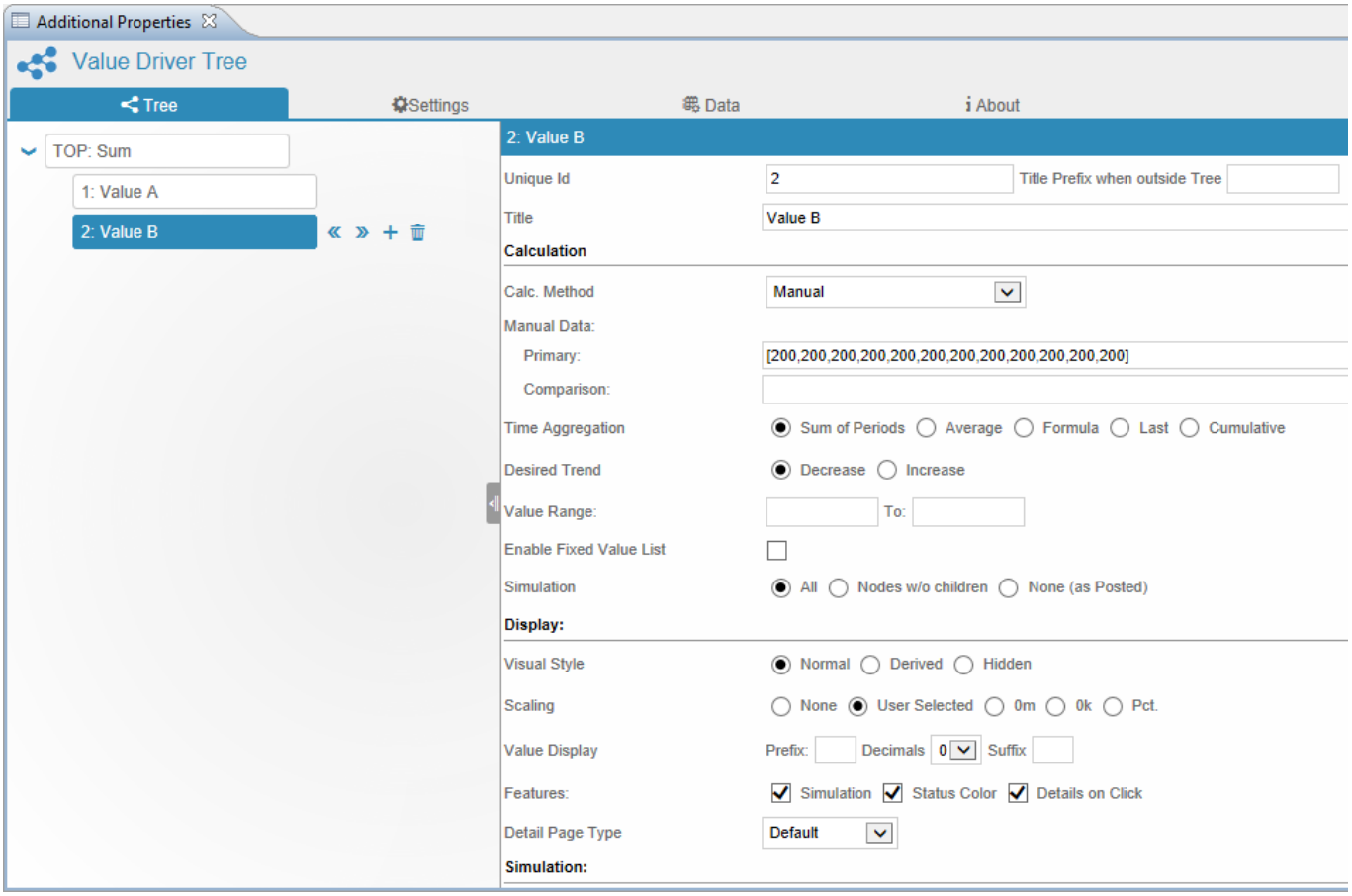


Figure 17.3: Properties Sheet for Node Value B

8. Set the Primary Data to the value [200, 200, 200, 200, 200, 200, 200, 200, 200, 200, 200, 200].
9. Set the property Time Aggregation to the option Sum of Periods.
10. Navigate to the Settings Tab. In the area Initial State, set the property Active Period from value 1 to value 12 as shown in the below Figure.

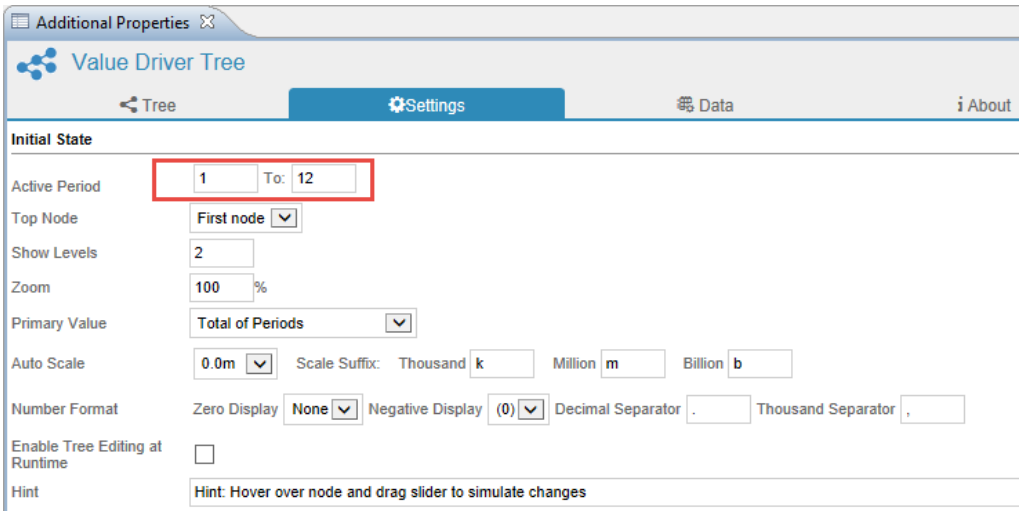


Figure 17.4: Settings Tab

11. Based on the above configuration you will be able to view the Value Driver Tree with the simulated values as shown in the below Figure.

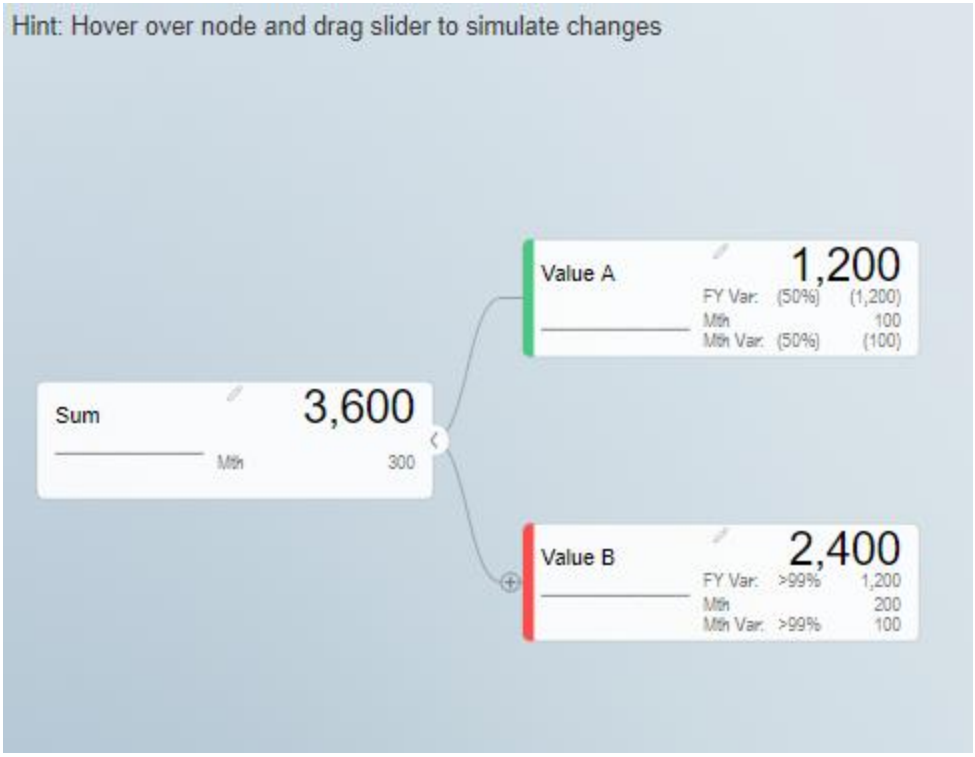


Figure 17.5: Simulated Data at Run time

12. From the above Figure, you can observe that the simulated value for Node “Value A” will be 100 for each month where the total value will be 1200 for 12 months and the simulated value for Node “Value B” will be 200 for each month where the total value will be 2400 for 12 months. The Parent Node “Sum” indicates the sum of the Primary values for Nodes Value A and Value B which is 1200+2400 → 3600.

17.2 Primary Data Series with Comparison Data Series

The steps below explains on how you will be able to configure the Primary Data Series with Comparison Data Series in VDT:

1. As an initial step, proceed with the similar set of steps for creating the Layout as explained in Section 17.1.
2. In this scenario consider that you have only one Node “Value A” with both the Primary and Comparison Data (see Figure below).

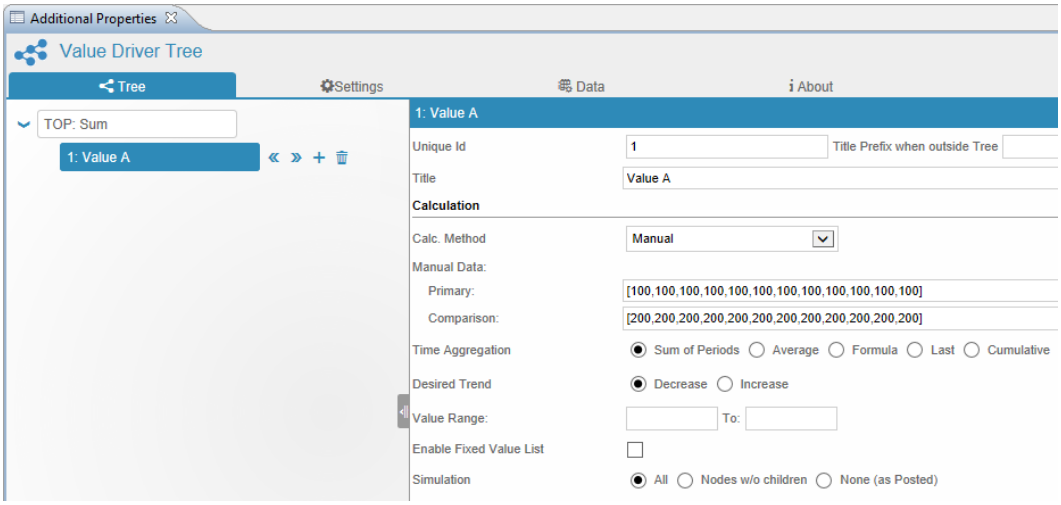


Figure 17.6: Properties Sheet for Node Value A

3. Navigate to the Additional Properties of the Tree Tab and to the area Calculation as shown in the Figure above.
4. For our example, set the property Calc. Method to the option Manual.
5. Set the Primary Data to the value [100, 100, 100, 100, 100, 100, 100, 100, 100, 100, 100, 100].
6. Set the Comparison Data to the value [200, 200, 200, 200, 200, 200, 200, 200, 200, 200, 200, 200].
7. Set the property Time Aggregation to the option Sum of Periods.
8. Navigate to the Settings Tab. In the area Initial State, set the property Active Period from value 1 to value 12 as shown in the below Figure.

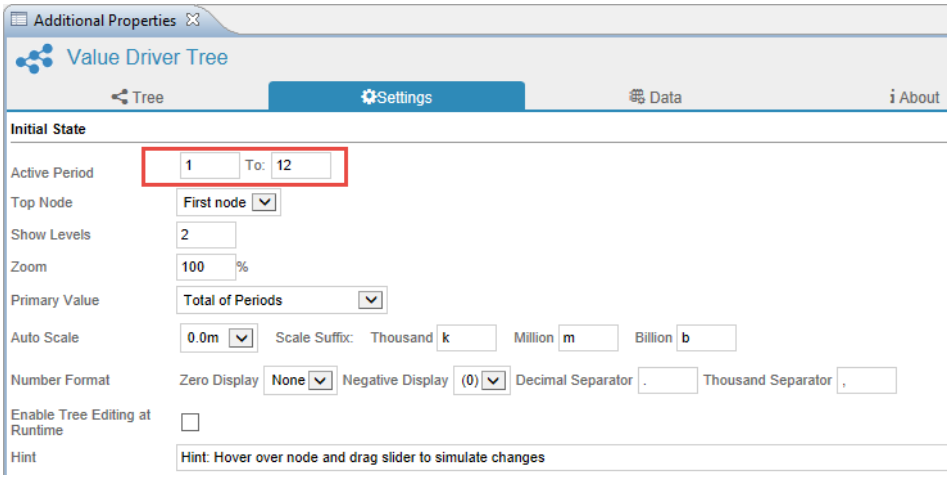


Figure 17.7: Settings Tab

9. Based on the above configuration you will be able to view the Value Driver Tree with the simulated values as shown in the below Figure.

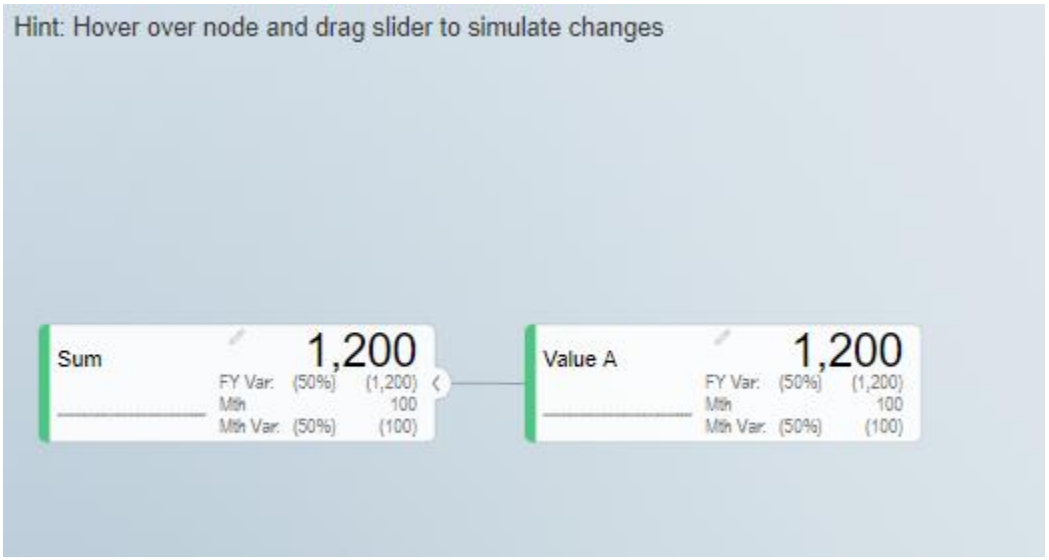


Figure 17.8: Simulated Data at Run time

13. From the above Figure, you can observe that the simulated value for Node “Value A” will be 100 for each month where the total value will be 1200 for 12 months. The Parent Node “Sum” indicates the sum of the Primary values for Node A which is 1200. You will be able to view the Primary value with the comparison value by clicking the Edit icon which would lead you the Table view.

17.3 Single Data Point

The steps below explains on how you will be able to configure the Single Data Point in VDT:

1. As an initial step, proceed with the similar set of steps for creating the Layout as explained in Section 17.1.
2. In this scenario consider that you have two Nodes “Value A” and “Value B” and a Parent Node “Sum” (see Figure below).

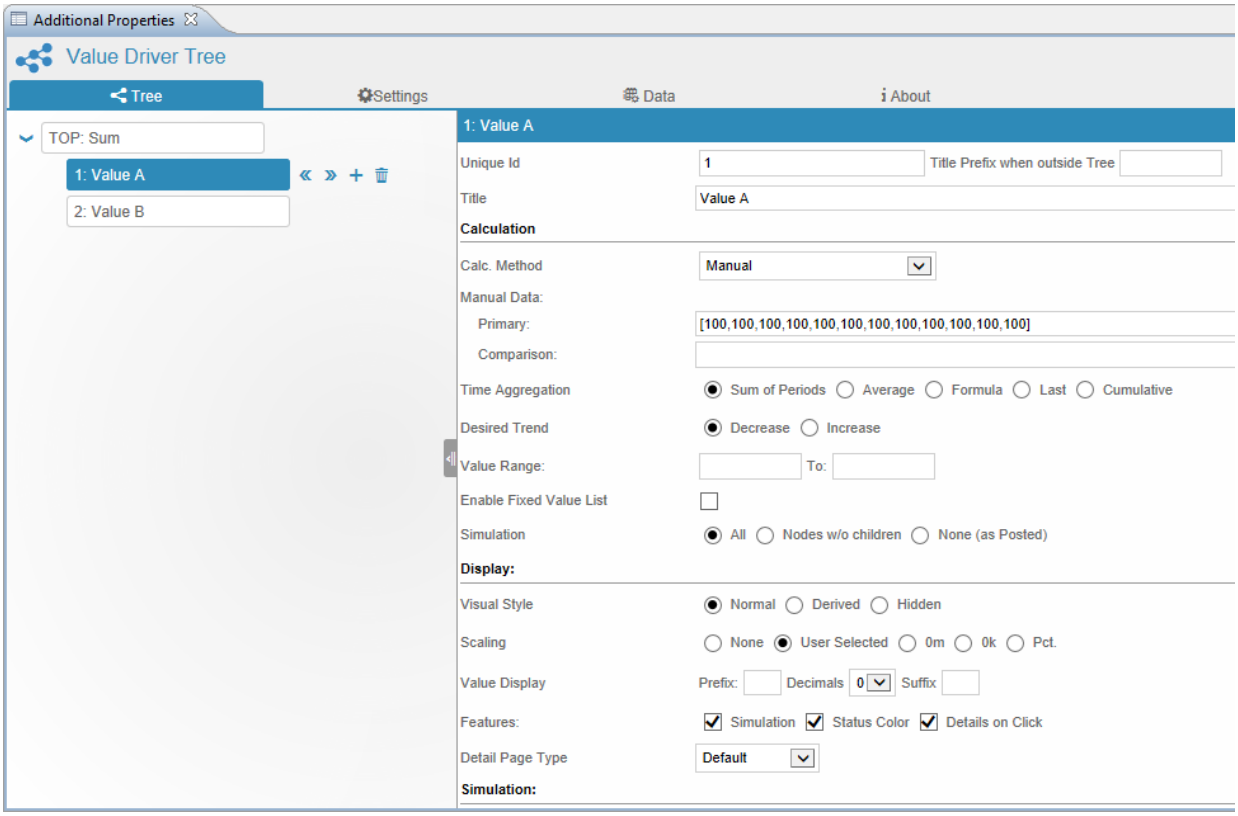


Figure 17.9: Properties Sheet for Node Value A

3. Now navigate to the Properties Sheet for the Node “Value A”. In the area Calculation, set the property Calc. Method to the option Manual.
4. Set the Primary Data to the value [100, 100, 100, 100, 100, 100, 100, 100, 100, 100, 100, 100].
5. Set the property Time Aggregation to the option Sum of Periods.
6. Now navigate to the Properties Sheet for the Node “Value B”. In the area Calculation, set the property Calc. Method to the option Manual (see Figure below).

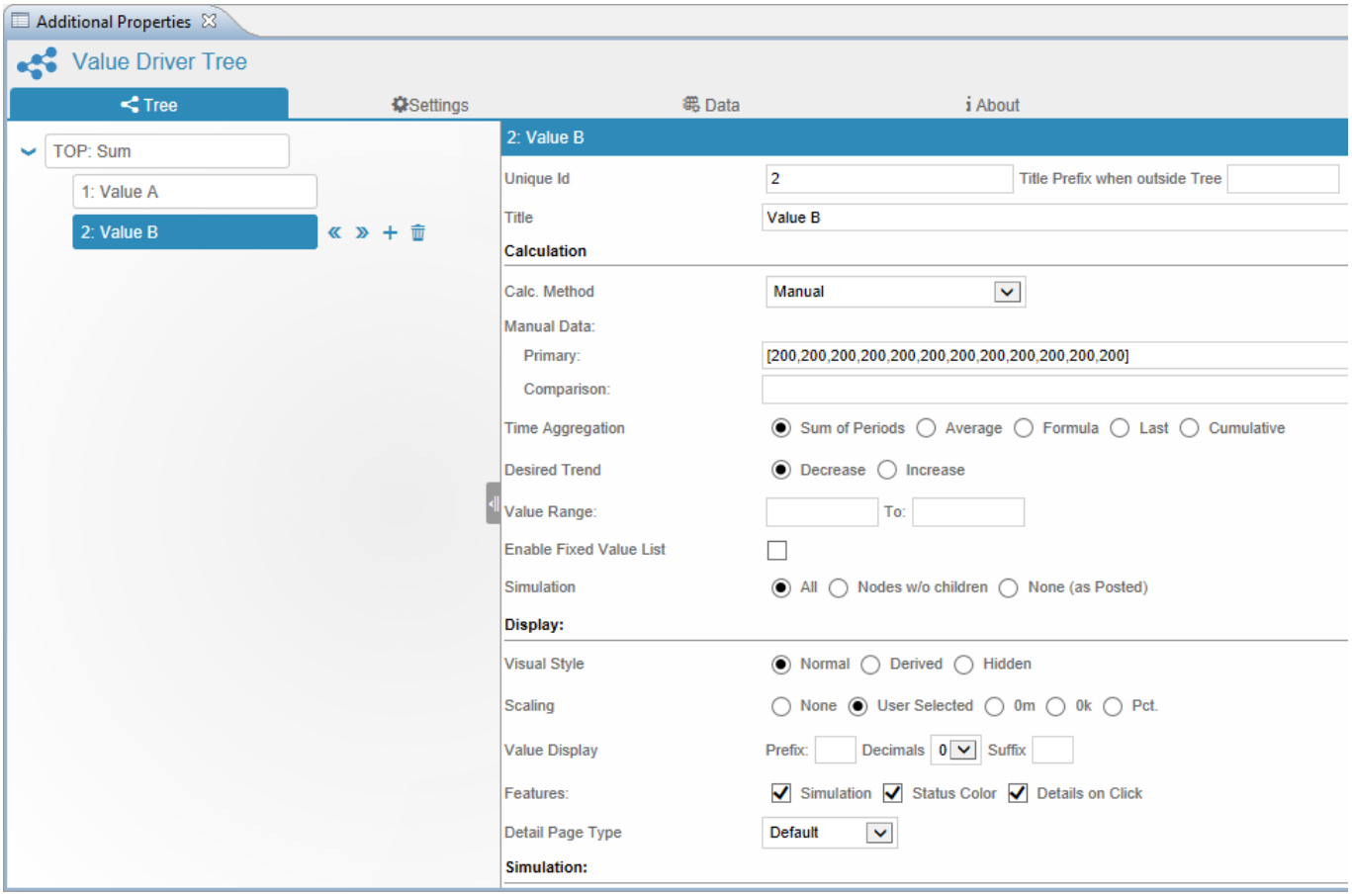


Figure 17.10: Properties Sheet for Node Value B

7. Set the Primary Data to the value [200, 200, 200, 200, 200, 200, 200, 200, 200, 200, 200, 200].
8. Set the property Time Aggregation to the option Sum of Periods.
9. Navigate to the Settings Tab. In the area Initial State, set the property Active Period from value 1 to value 1 as shown in the below Figure.

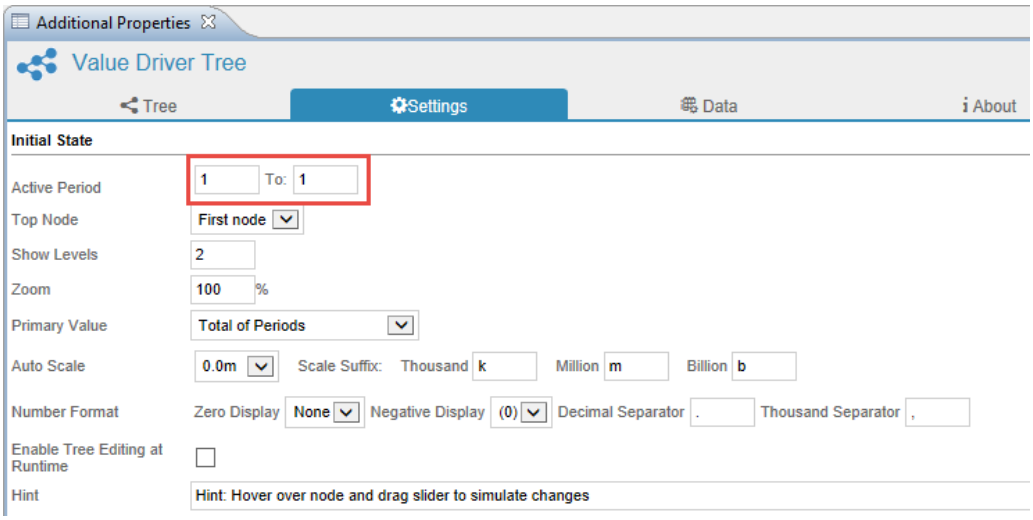


Figure 17.11: Settings Tab

10. Now navigate to the Data Tab. In the area Data Series, set the property Periods per data series to the value 1 and set the property Period Labels to the month Jan (see Figure below).

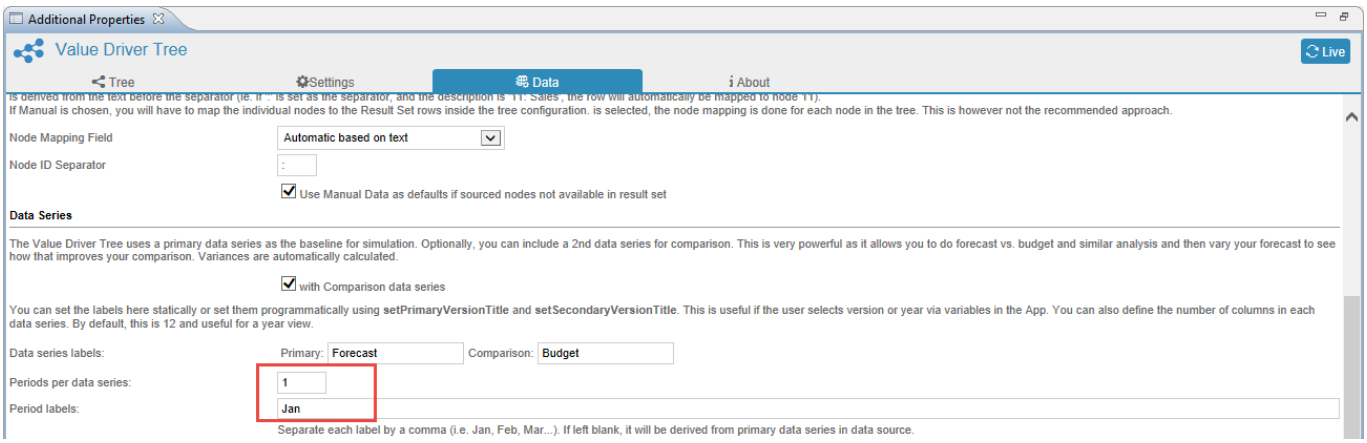


Figure 17.12: Data Tab

11. Based on the above configuration you will be able to view the Value Driver Tree with the simulated values as shown in the below Figure.

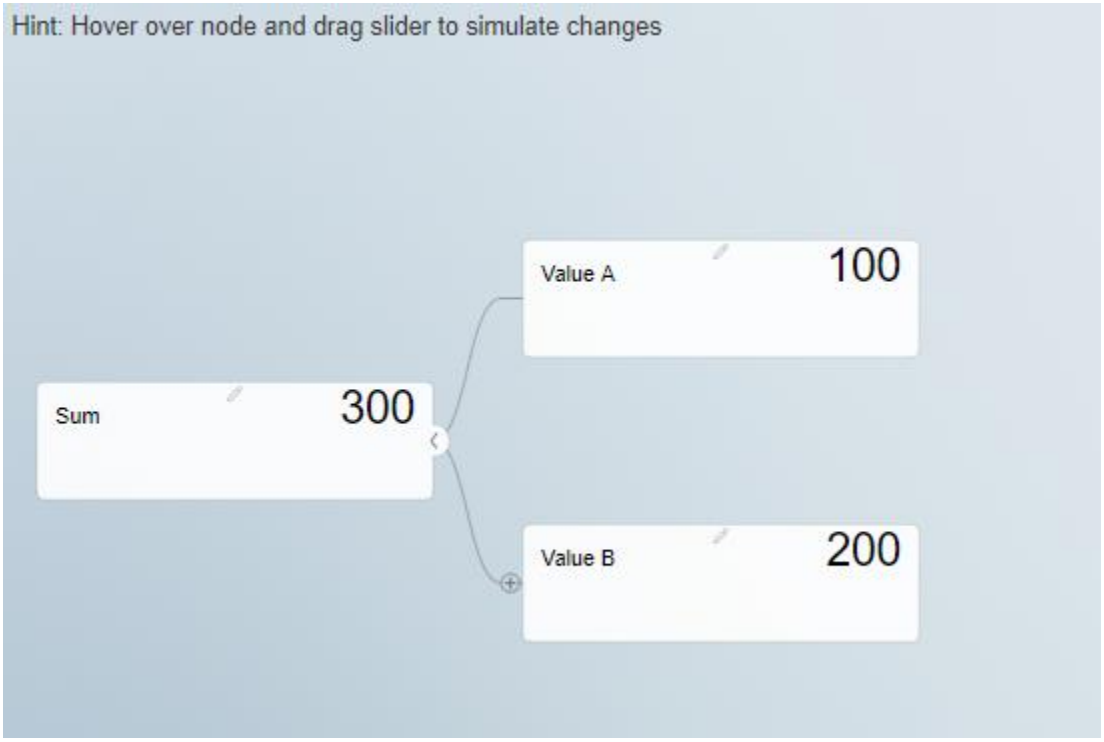


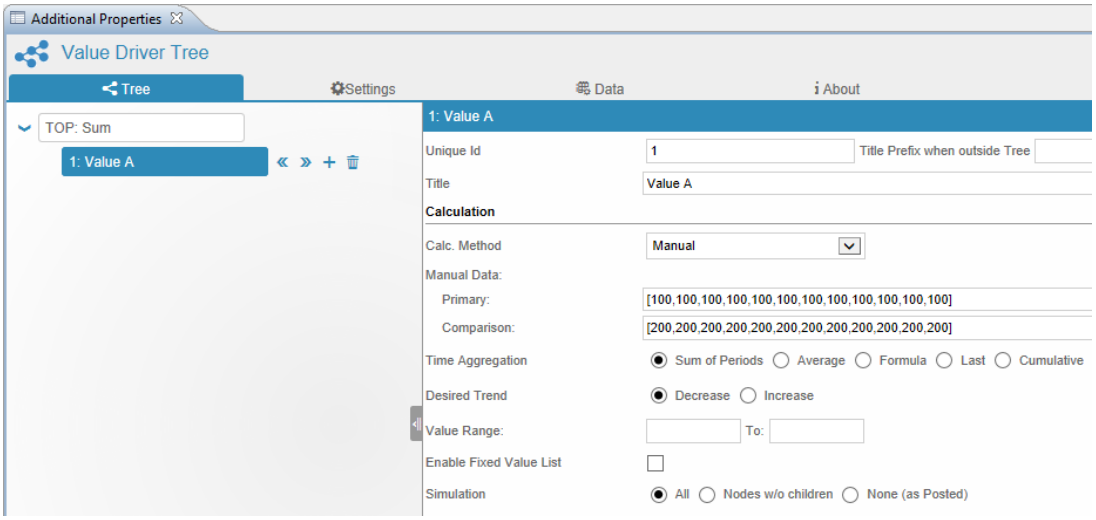
Figure 17.13: Simulated Data at Run time

14. From the above Figure, you can observe that the primary value for Node “Value A” will be 100 for January month and the Primary value for Node “Value B” will be 200 for January month. The total simulated value for the Parent Node “Sum” will be the sum of two nodes Value A and Value B which is 100+200 → 300.

17.4 Single Data Point with Comparison

The steps below explains on how you will be able to configure the Single Data Point with Comparison in VDT:

1. As an initial step, proceed with the similar set of steps for creating the Layout as explained in Section 17.1.
2. In this scenario consider that you have only one Node “Value A” and a Parent Node “Sum” (see Figure below).



3. *Figure 17.14: Properties Sheet for Node Value A*

4. Navigate to the Additional Properties of the Tree Tab and to the area Calculation as shown in the Figure above.
5. For our example, set the property Calc. Method to the option Manual.
6. Set the Primary Data to the value [100, 100, 100, 100, 100, 100, 100, 100, 100, 100, 100, 100].
7. Set the Comparison Data to the value [200, 200, 200, 200, 200, 200, 200, 200, 200, 200, 200, 200].
8. Set the property Time Aggregation to the option Sum of Periods.
9. Navigate to the Settings Tab. In the area Initial State, set the property Active Period from value 1 to value 1 as shown in the below Figure.

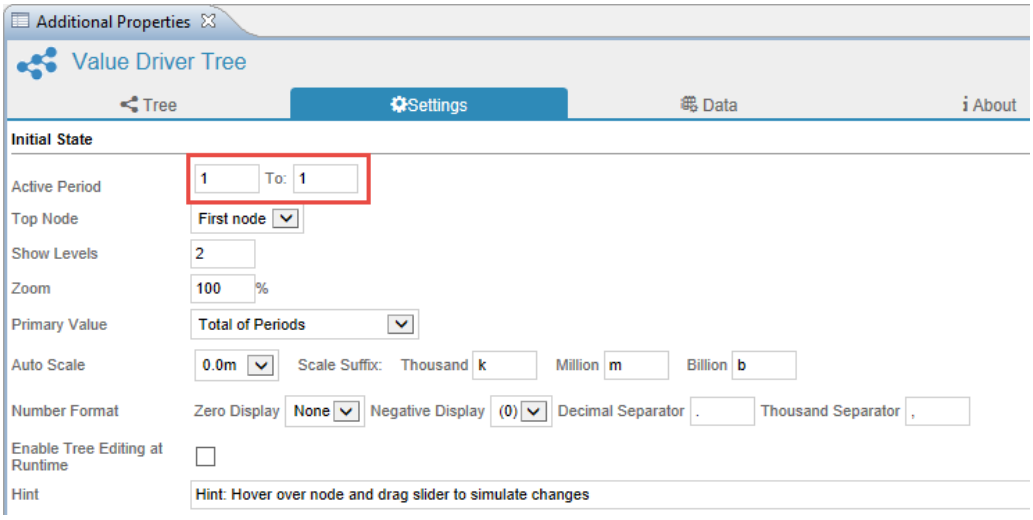


Figure 17.15: Settings Tab

12. Now navigate to the Data Tab. In the area Data Series, set the property Periods per data series to the value 1 and set the property Period Labels to the month Jan (see Figure below).

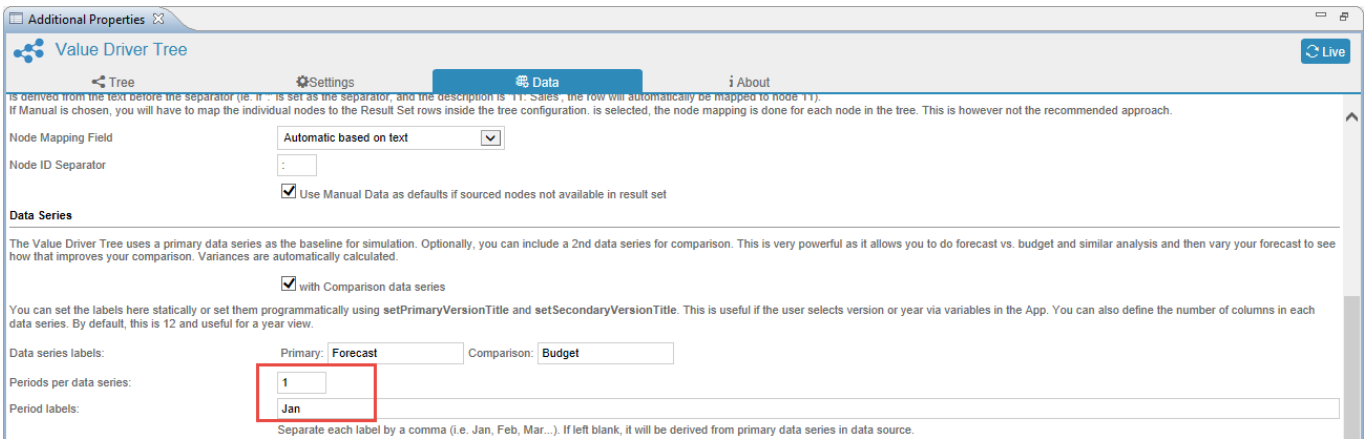


Figure 17.16: Data Tab

13. Based on the above configuration you will be able to view the Value Driver Tree with the simulated values as shown in the below Figure.

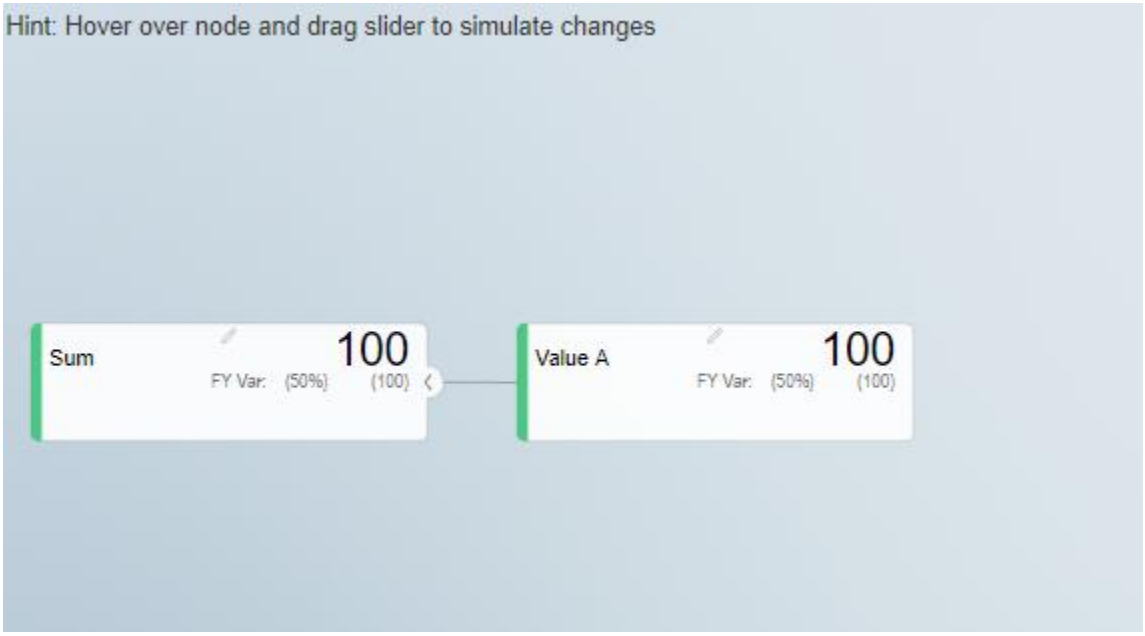


Figure 17.17: Simulated Data at Run time

15. From the above Figure, you can observe that the Primary value for Node “Value A” will be 100 for January month and the Comparison value for Node “Value B” will be 200 for January month. The Parent Node “Sum” indicates the sum of the Primary value for only Node A which is 100 for January month (based on the configuration shown in Figure 17.16). Now you will be able to view the Primary value with the comparison value by clicking the Edit icon which would lead you the Table view.

18 Variance Analysis

As part of VDT Release 3.0, a new option Variance Breakdown has been included in the Run time which shows the complete variance breakdown of the Calculated Components on the Full Period. For our example you can observe the current status of the Node Revenue before applying the Simulation (see Figure below).

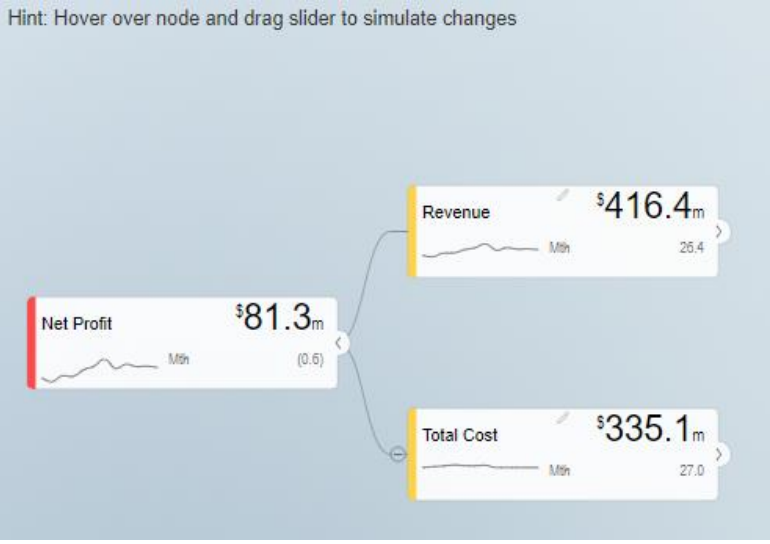


Figure 18.1: Node Revenue in Run time – Before Simulation

In our example, the current value for Node Revenue is 416.4 and by clicking the Edit button you will be able to view the Table showing the Variance Breakdown of calculated components on Full Year. Here you can observe the variance details for the current value 416.4 against the original value 459.4 for the Node Revenue (see Figure below).

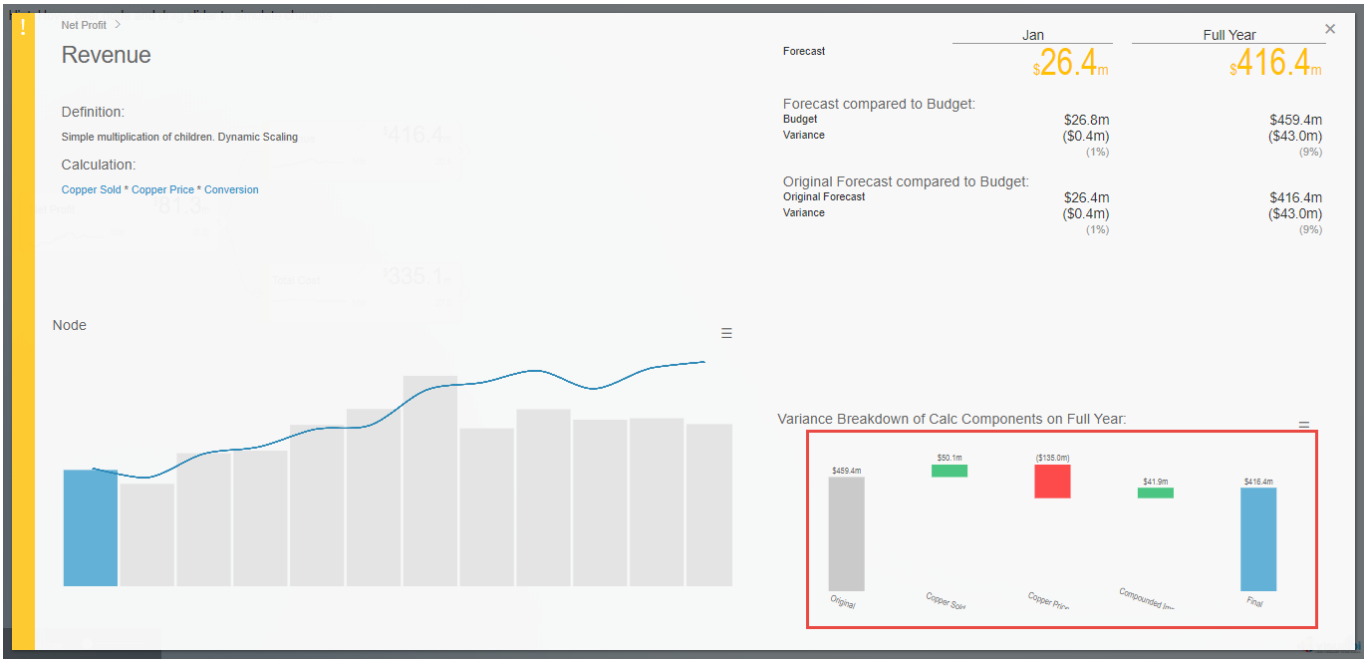


Figure 18.2: Table View on Variance Break before Simulation

After applying the Simulation you can observe that the Node Revenue value gets increased to 505.7 (see Figure below).

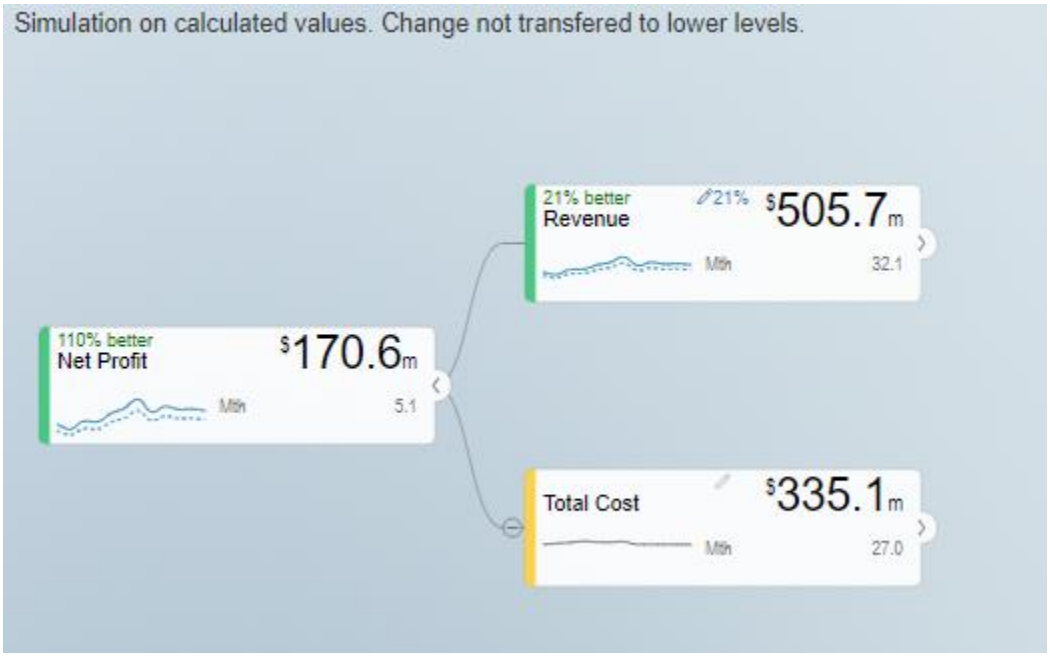


Figure 18.3: Node Revenue in Run time – After Simulation

Now by clicking the Edit button in the Node Revenue, you will be able to view the Table showing the Variance Breakdown of calculated components on Full Year. Here you can observe the variance details for the simulated value 505.7 against the original value 459.4 for the Node Revenue (see Figure below).

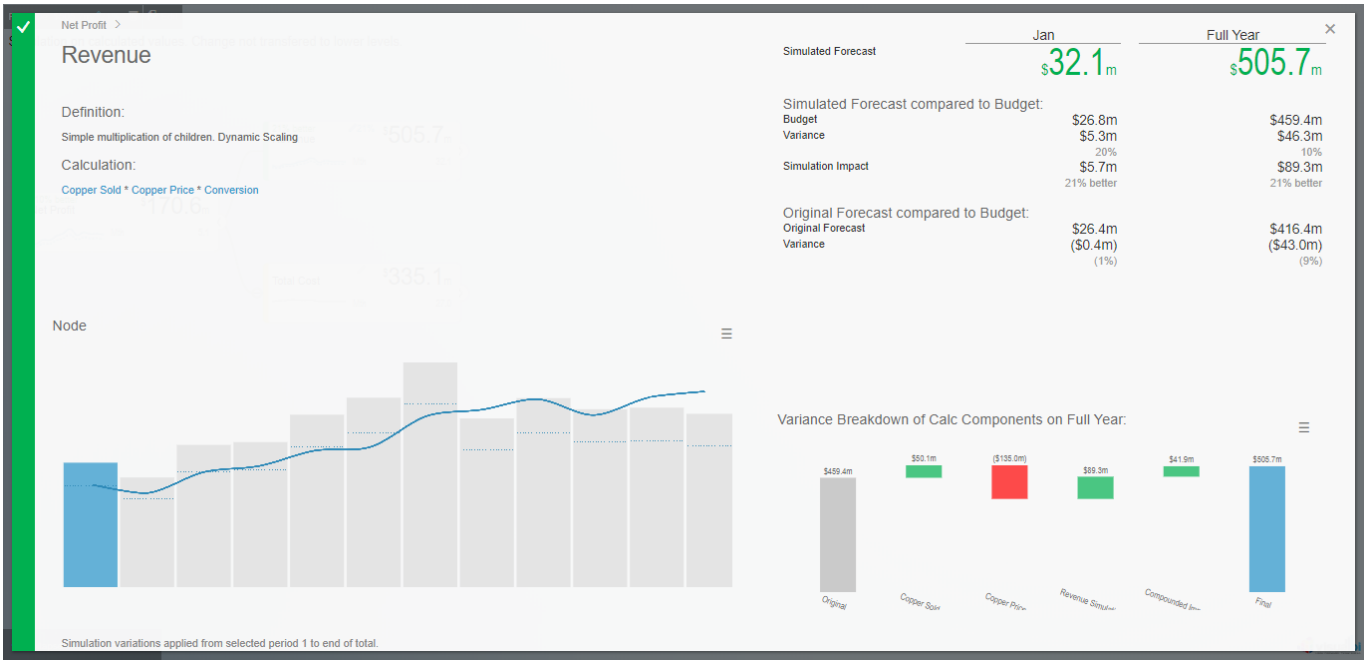


Figure 18.4: Table View on Variance Break after Simulation

19 Dynamic Tree from Data Source

As part of VDT Release 3.0, you will be able to use Hierarchical Data Source for the Value Driver Tree. Based on the Hierarchical Data Source assigned the VDT will show the Parent and the Child Nodes in relevance to the Additional Property settings.

The steps below explains on how you will be able to create the VDT using the Hierarchical Data Source:

1. For our example, create a layout in SAP Lumira Designer by assigning a VDT and a VDT Navigational Panel as shown in the below Figure.

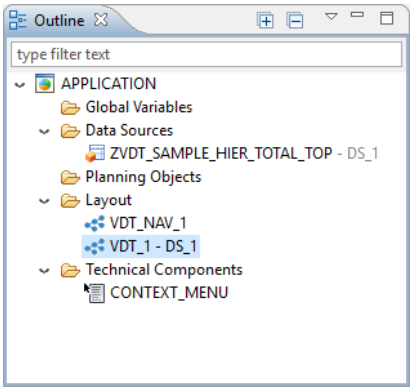


Figure 19.1: Layout

2. For our example, assign the Hierarchical data Source as shown in the above Figure.
3. Now navigate to the Initial View of the Data Source (see Figure below).

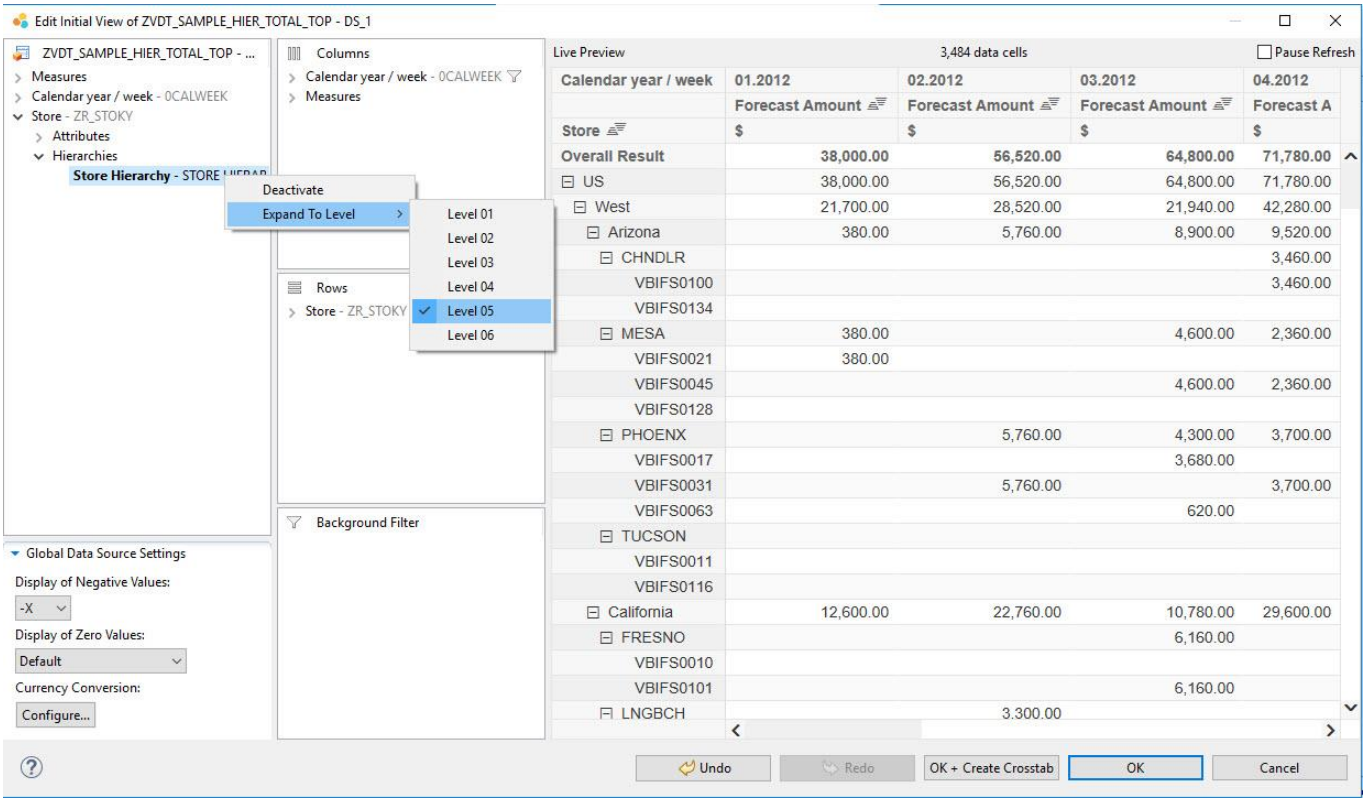


Figure 19.2: Initial View of the Data Source

4. Now activate the 5 Levels of Hierarchy for the Dimension State as shown in the Figure above.
5. Based on the activation of the 5 Levels of Hierarchy, you will be able to view the Tree with 5 levels of Hierarchy having “US” as the Parent Node (see Figure below).

In the above Figure, you can observe that “1” represents the First Level of Hierarchy (US – Parent Node), “2” represents the Second Level of Hierarchy (West, South, North East and Mid West with West as Parent Node), “3” represents the Third Level of Hierarchy (Arizona, California, New Mexico, Washington, Georgia and Orgeon with Arizona as Parent Node), “4” represents the Fourth Level of Hierarchy (CHNDLR, MESA, PHOENX and TUCSON with CHNDLR as Parent Node) and “5” represents the Fifth Level of Hierarchy (VBIFS0100 and VBIFS0134).

20 Known Issues and Limitations

Below are the known issues as of the writing of the document. This list will be updated in the online version to reflect the current state at any point in time.

Pinch zoom is not supported in IE on Windows touch enabled devices	The Pinch zoom doesn't work within Windows touch interfaces. A zoom slider has been added to ensure users has access to zoom even when
Value Driver Trees tested up to 1500 nodes in single tree	As your tree becomes bigger, the most likely challenge becomes your data source performance. As the data source is not refreshed during navigation, this major performance impact is however only material when you open the App. It is a little like opening an Excel workbook. As they become larger, they take longer to load. Value Driver Trees are tested up to 1500 nodes and calculation, visualization and simulation is almost instant on all tested devices. We can however not guarantee performance beyond our tested level. If you have a requirement for more than 1500 nodes in a single tree, please contact us.

www.visualbi.com

solutions@visualbi.com

Phone: +1 888-227-2794

Fax: +1 888-227-7192

Plano, TX ■ Dallas, TX ■ Chennai, India