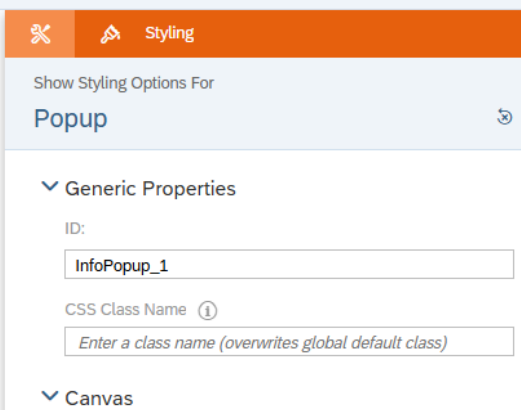
SAC Scripts

Contents

Info - Popup:

1. Create Info Popup and add following function to close button



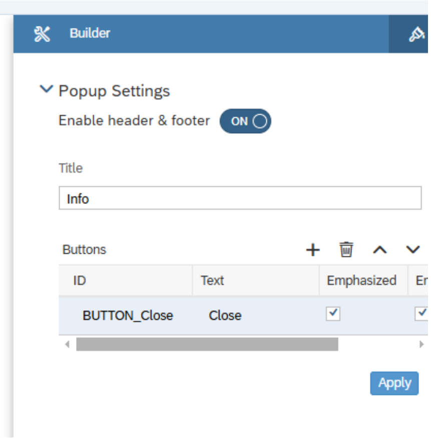
Show Styling Options For
Popup

Generic Properties

ID: InfoPopup_1

CSS Class Name: Enter a class name (overwrites global default class)

Canvas



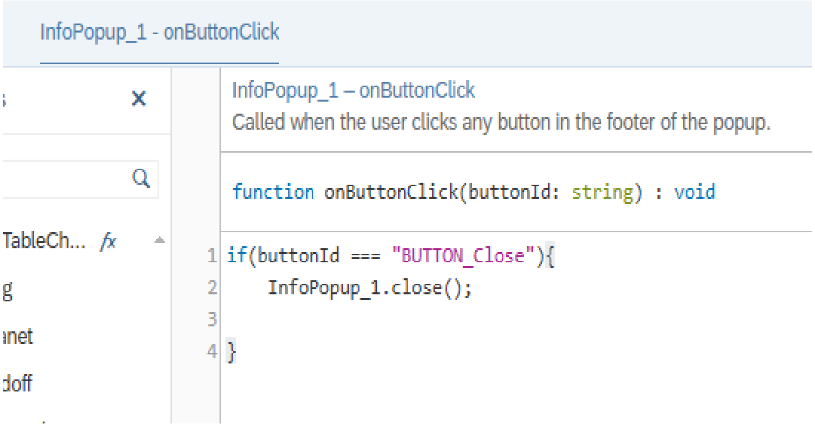
Popup Settings

Enable header & footer: ON

Title: Info

Buttons

ID	Text	Emphasized	Er
BUTTON_Close	Close	☑	☑



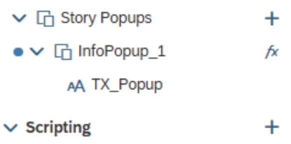
InfoPopup_1 - onButtonClick

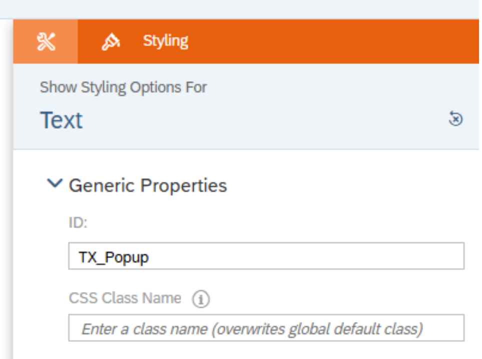
InfoPopup_1 – onButtonClick
Called when the user clicks any button in the footer of the popup.

```
function onButtonClick(buttonId: string) : void
```

```
if(buttonId === "BUTTON_Close"){
    InfoPopup_1.close();

}
```

2. Inside the Info popup create a Text that contains the text that should be shown in the popup





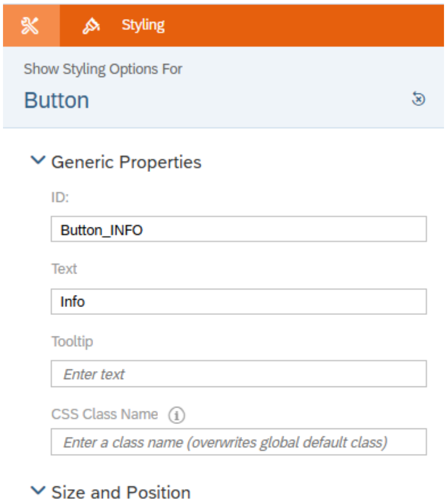
Show Styling Options For
Text

Generic Properties

ID: TX_Popup

CSS Class Name: Enter a class name (overwrites global default class)

3. Create Button and add onClick function to this Button. :



Show Styling Options For

Button

Generic Properties

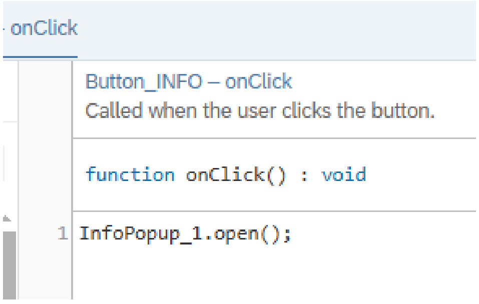
ID: Button_INFO

Text: Info

Tooltip: Enter text

CSS Class Name: Enter a class name (overwrites global default class)

Size and Position

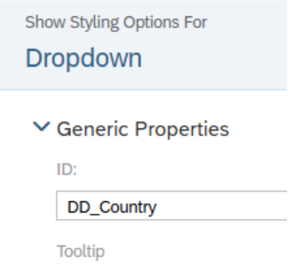


Button_INFO – onClick
Called when the user clicks the button.

```
function onClick() : void

InfoPopup_1.open();
```

Dropdown

1. Create dropdown and use following onInitialization (Page) script (without this script, dropdown remains empty):

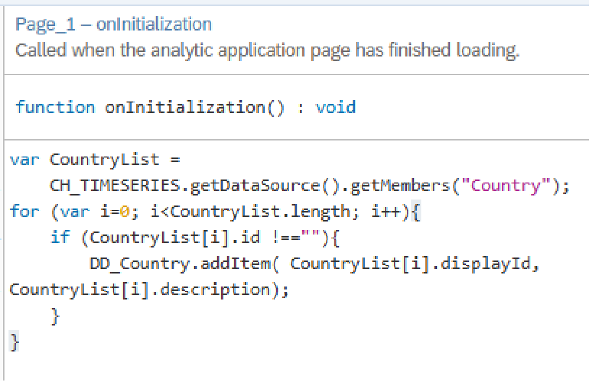

Show Styling Options For

Dropdown

Generic Properties

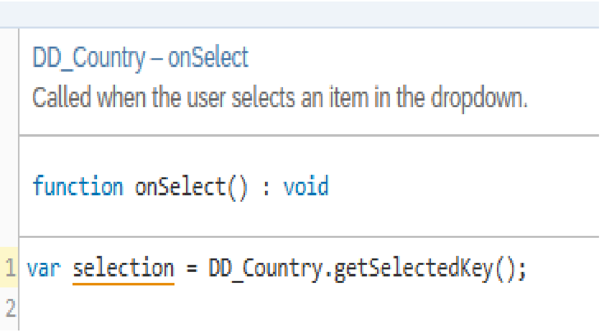
ID: DD_Country

Tooltip


Page_1 – onInitialization
Called when the analytic application page has finished loading.

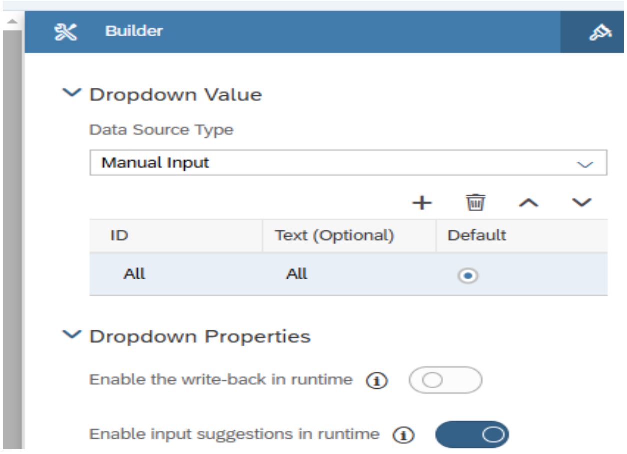
```
function onInitialization() : void

var CountryList =
    CH_TIMESERIES.getDataSource().getMembers("Country");
for (var i=0; i<CountryList.length; i++){
    if (CountryList[i].id !==""){
        DD_Country.addItem( CountryList[i].displayId,
CountryList[i].description);
    }
}
```

An use following on selectScript (DD):


DD_Country – onSelect
Called when the user selects an item in the dropdown.

```
function onSelect() : void

var selection = DD_Country.getSelectedKey();

```

2. onSelect Script shown below, in scripts below 4 charts are affected by the DD menu.



Builder

Dropdown Value

Data Source Type

Manual Input

ID	Text (Optional)	Default
All	All	◉

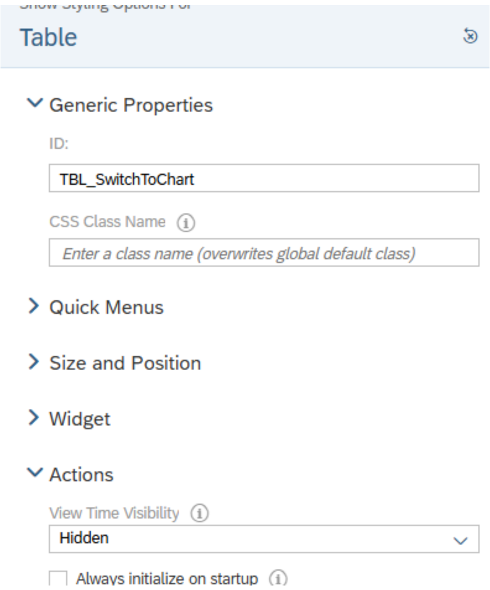
Dropdown Properties

Enable the write-back in runtime

Enable input suggestions in runtime

Switch Charts (Table to Chart)

1. Create a table (here, this one is set to HIDDEN)



Show Styling Options For

Table

Generic Properties

ID:

TBL_SwitchToChart

CSS Class Name

Enter a class name (overwrites global default class)

Quick Menus

Size and Position

Widget

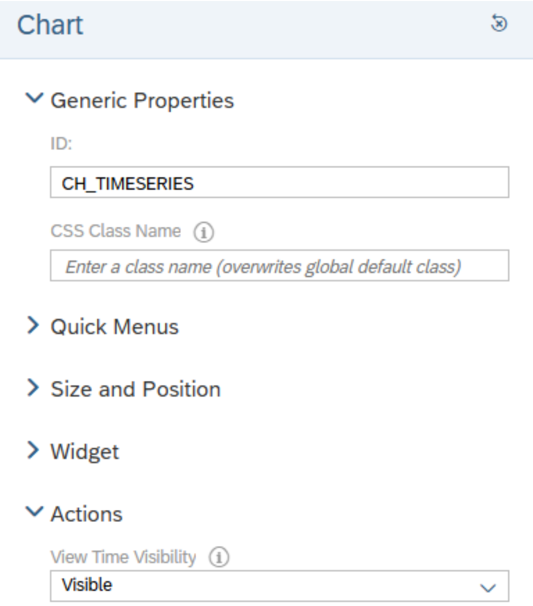
Actions

View Time Visibility

Hidden

□ Always initialize on startup

2. Create a chart (here this one is set to VISIBLE)



Chart

Generic Properties

ID:

CH_TIMESERIES

CSS Class Name

Enter a class name (overwrites global default class)

Quick Menus

Size and Position

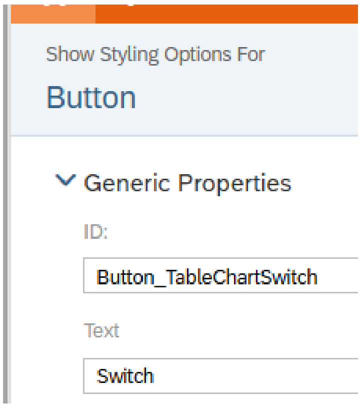
Widget

Actions

View Time Visibility

Visible

3. Create a Button and add onClick script:



Show Styling Options For

Button

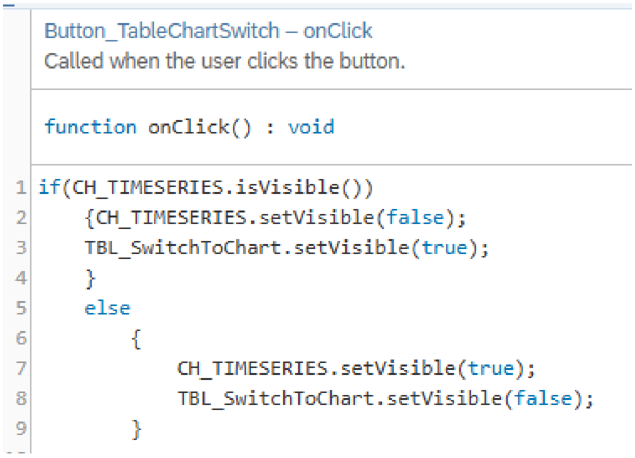
Generic Properties

ID:

Button_TableChartSwitch

Text

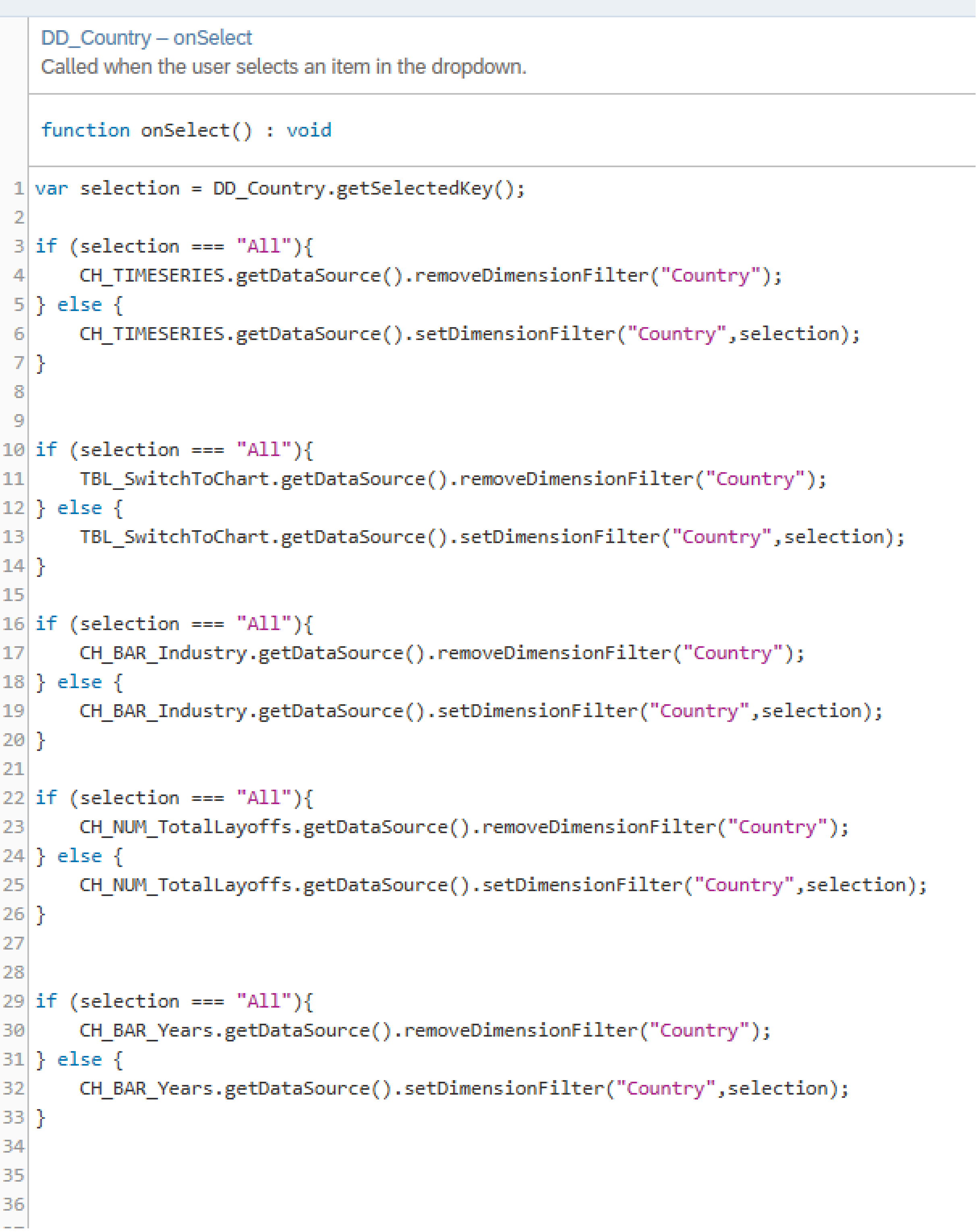
Switch



Button_TableChartSwitch – onClick
Called when the user clicks the button.

```
function onClick() : void
```

```
if(CH_TIMESERIES.isVisible())
    {CH_TIMESERIES.setVisible(false);
    TBL_SwitchToChart.setVisible(true);
    }
    else
        {
            CH_TIMESERIES.setVisible(true);
            TBL_SwitchToChart.setVisible(false);
        }
```



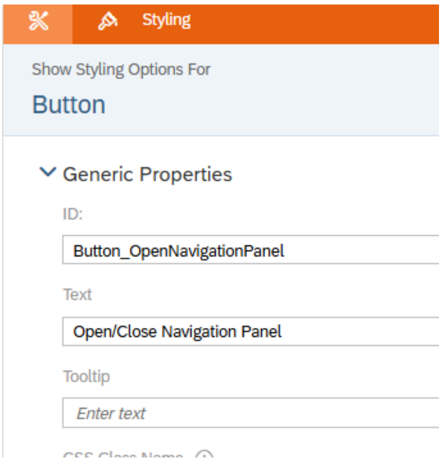
DD_Country – onSelect
Called when the user selects an item in the dropdown.

```
function onSelect() : void
```

```
var selection = DD_Country.getSelectedKey();

if (selection === "All"){
    CH_TIMESERIES.getDataSource().removeDimensionFilter("Country");
} else {
    CH_TIMESERIES.getDataSource().setDimensionFilter("Country",selection);
}


if (selection === "All"){
    TBL_SwitchToChart.getDataSource().removeDimensionFilter("Country");
} else {
    TBL_SwitchToChart.getDataSource().setDimensionFilter("Country",selection);
}

if (selection === "All"){
    CH_BAR_Industry.getDataSource().removeDimensionFilter("Country");
} else {
    CH_BAR_Industry.getDataSource().setDimensionFilter("Country",selection);
}

if (selection === "All"){
    CH_NUM_TotalLayoffs.getDataSource().removeDimensionFilter("Country");
} else {
    CH_NUM_TotalLayoffs.getDataSource().setDimensionFilter("Country",selection);
}


if (selection === "All"){
    CH_BAR_Years.getDataSource().removeDimensionFilter("Country");
} else {
    CH_BAR_Years.getDataSource().setDimensionFilter("Country",selection);
}
```

Button Open/Close Navigation Panel in a Table

1. Create a button and a boolean SV:



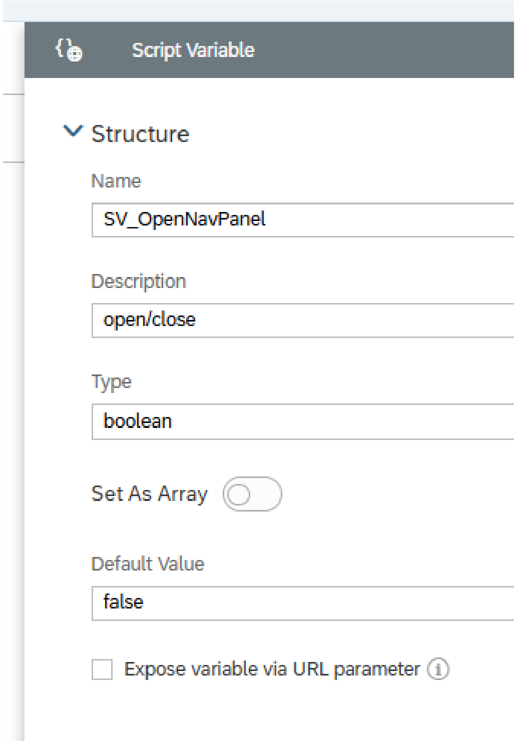
Show Styling Options For
Button

Generic Properties

ID: Button_OpenNavigationPanel

Text: Open/Close Navigation Panel

Tooltip: Enter text



Structure

Name: SV_OpenNavPanel

Description: open/close

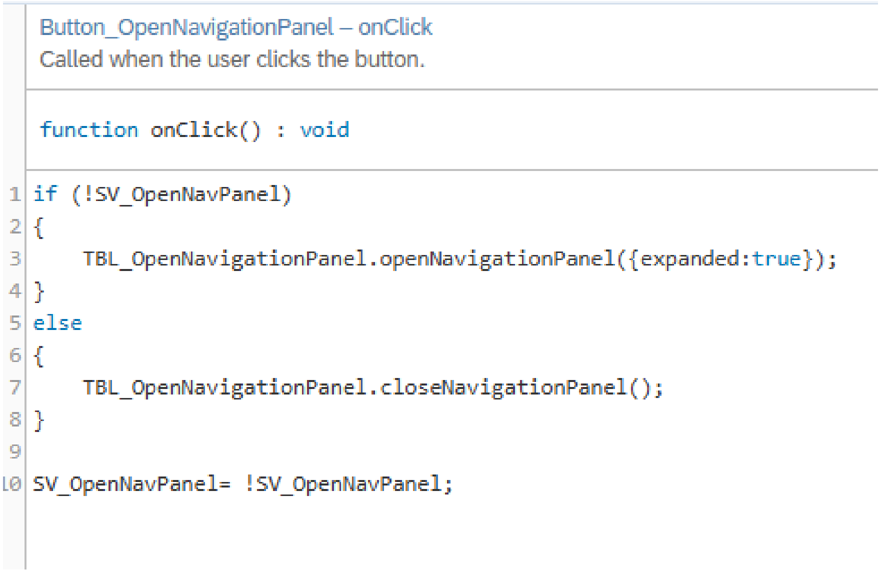
Type: boolean

Set As Array ○

Default Value: false

□ Expose variable via URL parameter

2. Add onClick Script to the button:



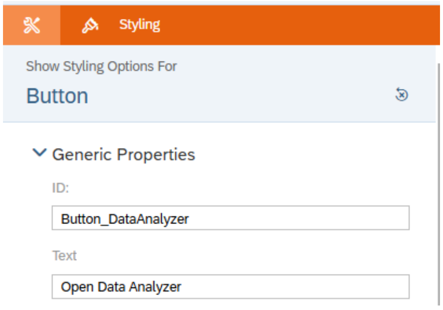
Button_OpenNavigationPanel – onClick
Called when the user clicks the button.

```
function onClick() : void
```

```
if (!SV_OpenNavPanel)
{
    TBL_OpenNavigationPanel.openNavigationPanel({expanded:true});
}
else
{
    TBL_OpenNavigationPanel.closeNavigationPanel();
}

SV_OpenNavPanel= !SV_OpenNavPanel;
```

3.

Open Data Analyzer-Button

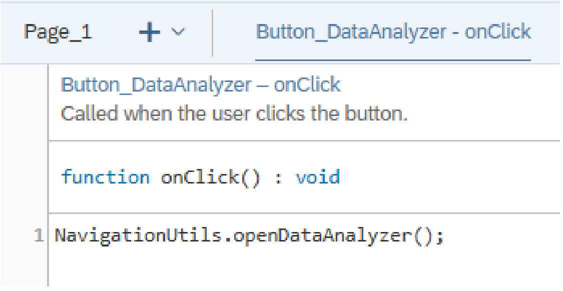


Show Styling Options For
Button

Generic Properties

ID: Button_DataAnalyzer

Text: Open Data Analyzer



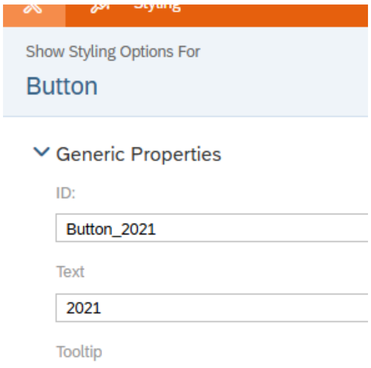
Button_DataAnalyzer – onClick
Called when the user clicks the button.

```
function onClick() : void
```

```
NavigationUtils.openDataAnalyzer();
```

Create Button to Filter on Year and then to remove filter

1. Create button, add script seen below to the button. Table/chart depends the name and also year/dimension depends what to filter on.



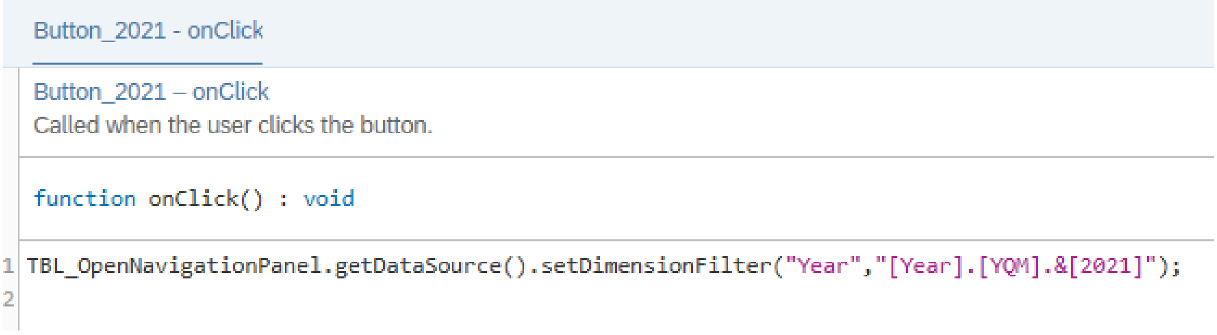
Show Styling Options For
Button

Generic Properties

ID: Button_2021

Text: 2021

Tooltip



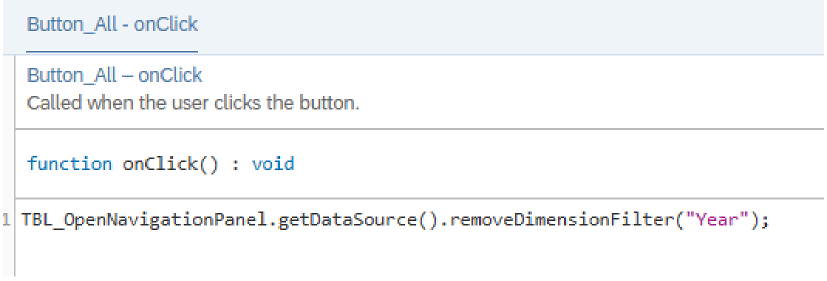
Button_2021 - onClick

Button_2021 – onClick
Called when the user clicks the button.

```
function onClick() : void
```

```
TBL_OpenNavigationPanel.getDataSource().setDimensionFilter("Year","[Year].[YQM].&[2021]");

```

2. Create button ALL and add script seen below:



Button_All - onClick

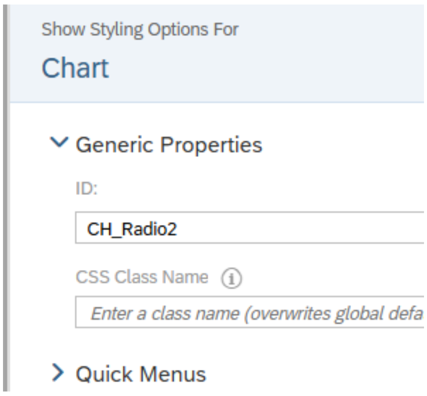
Button_All – onClick
Called when the user clicks the button.

```
function onClick() : void
```

```
TBL_OpenNavigationPanel.getDataSource().removeDimensionFilter("Year");
```

Radio Button Group Switch Charts/Table

1. Create 2 charts (view time visibility of both is on auto)

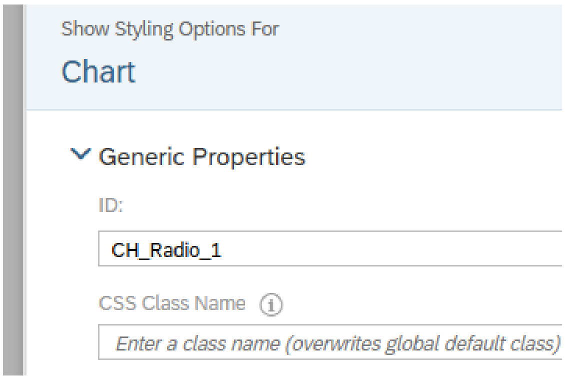


Show Styling Options For
Chart

Generic Properties

ID: CH_Radio2

CSS Class Name: *Enter a class name (overwrites global defa*



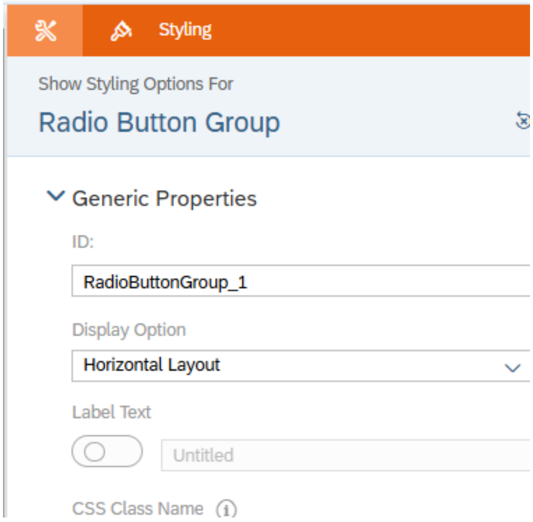
Show Styling Options For
Chart

Generic Properties

ID: CH_Radio_1

CSS Class Name: *Enter a class name (overwrites global default class)*

2. Create RADIO BUTTON and add under the builder the names of the 2 charts. Add the script shown below to the radio button onSelect



Show Styling Options For

Radio Button Group

Generic Properties

ID:

RadioButtonGroup_1

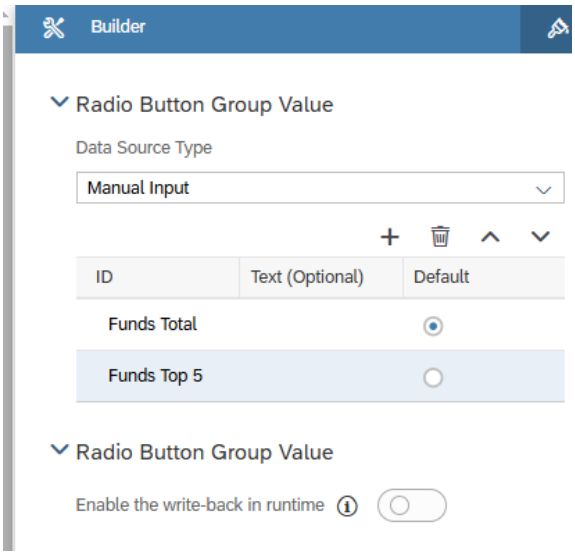
Display Option

Horizontal Layout

Label Text

○ Untitled

CSS Class Name



Radio Button Group Value

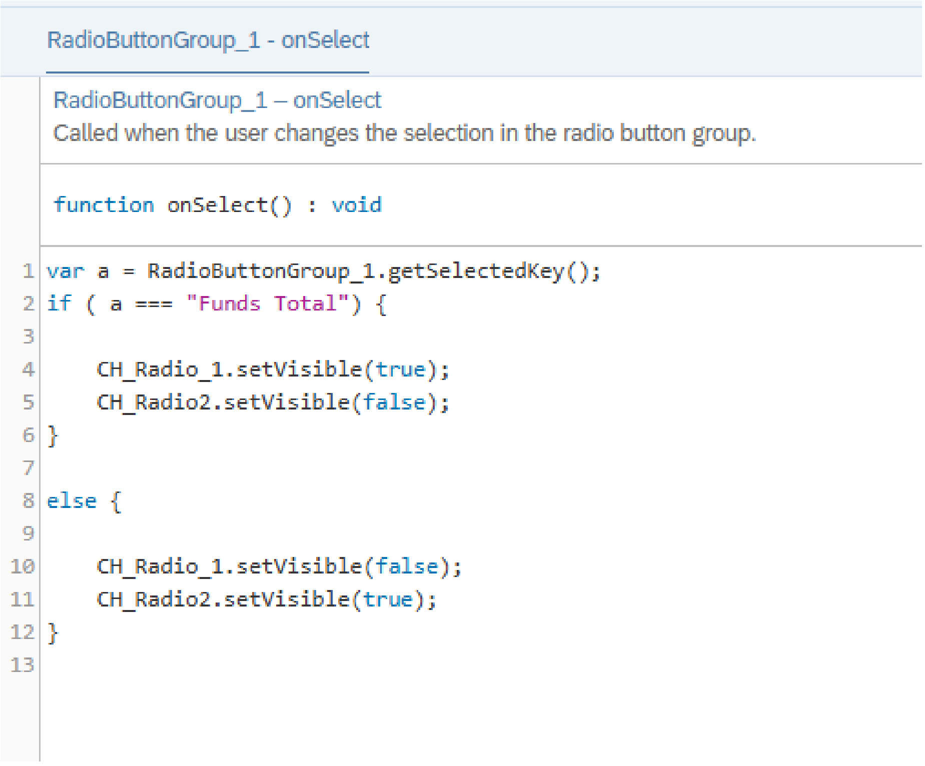
Data Source Type

Manual Input

ID	Text (Optional)	Default
Funds Total		●
Funds Top 5		○

Radio Button Group Value

Enable the write-back in runtime ○



RadioButtonGroup_1 – onSelect
Called when the user changes the selection in the radio button group.

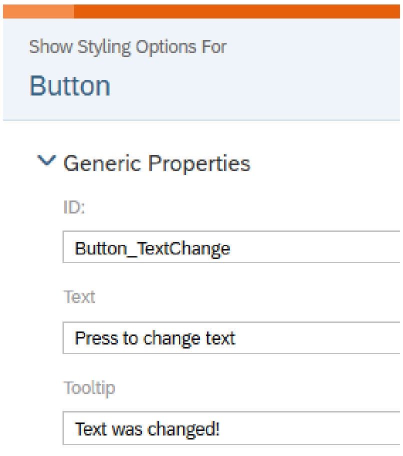
```
function onSelect() : void
```

```
var a = RadioButtonGroup_1.getSelectedKey();
if ( a === "Funds Total") {

    CH_Radio_1.setVisible(true);
    CH_Radio2.setVisible(false);
}

else {

    CH_Radio_1.setVisible(false);
    CH_Radio2.setVisible(true);
}
```

```
var a = RadioButtonGroup_1.getSelectedKey();
if ( a === "Funds Total") {

        CH_Radio_1.setVisible(true);
        CH_Radio2.setVisible(false);
}

else {

        CH_Radio_1.setVisible(false);
        CH_Radio2.setVisible(true);
}
```

Button to change text

1. Create a button and a text



Show Styling Options For

Button

Generic Properties

ID:

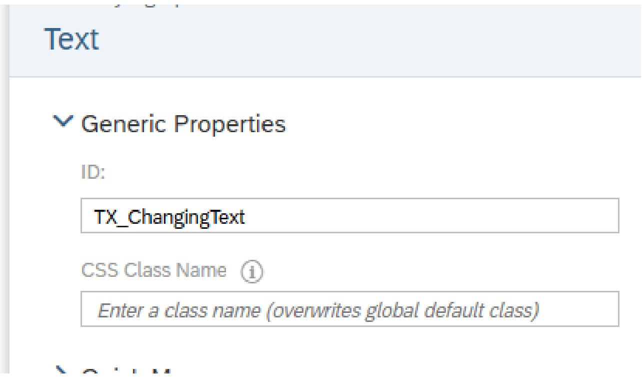
Button_TextChange

Text

Press to change text

Tooltip

Text was changed!



Text

Generic Properties

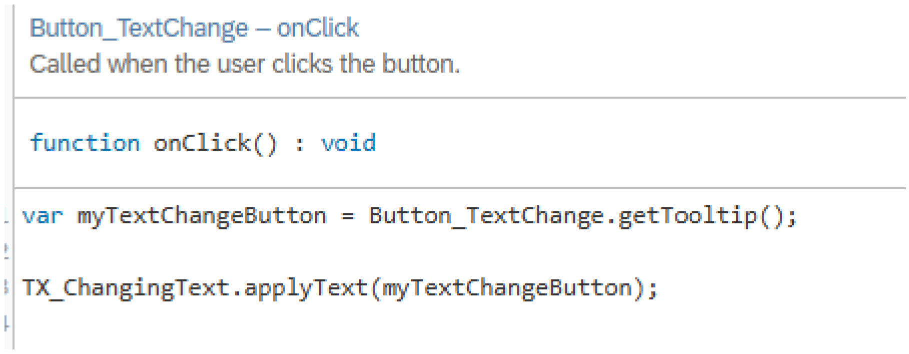
ID:

TX_ChangingText

CSS Class Name

Enter a class name (overwrites global default class)

2. Add following script to button:



Button_TextChange – onClick
Called when the user clicks the button.

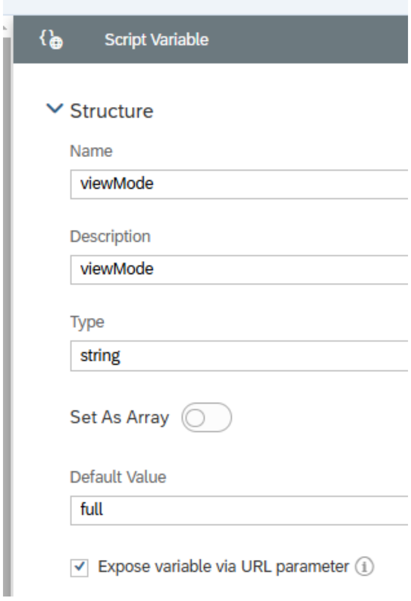
```
function onClick() : void

var myTextChangeButton = Button_TextChange.getTooltip();

TX_ChangingText.applyText(myTextChangeButton);
```

var myTextChangeButton = Button_TextChange.getTooltip();

TX_ChangingText.applyText(myTextChangeButton);

Different view modes with a URL (get 2 URLs)

1. Create a Script variable, string type:



Structure

Name

viewMode

Description

viewMode

Type

string

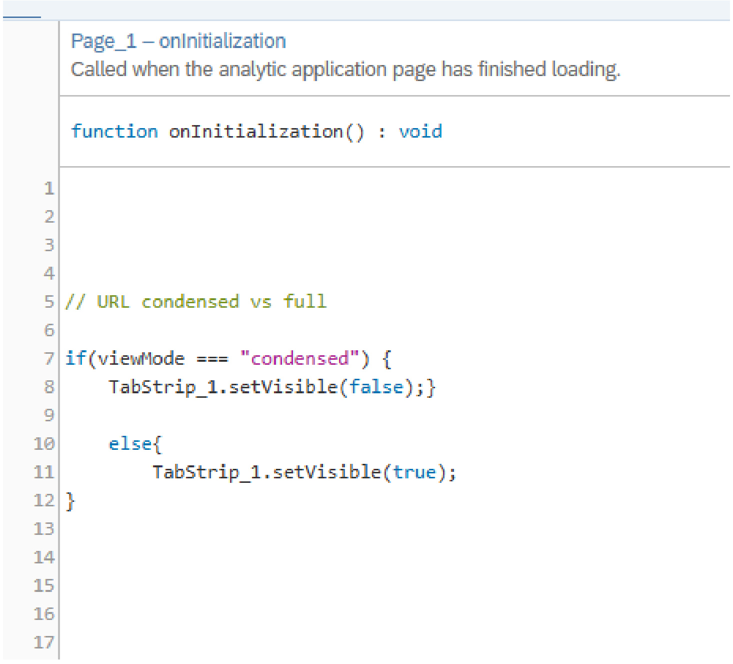
Set As Array ○

Default Value

full

☑ Expose variable via URL parameter

2. To the Page, add following script onInitialization:

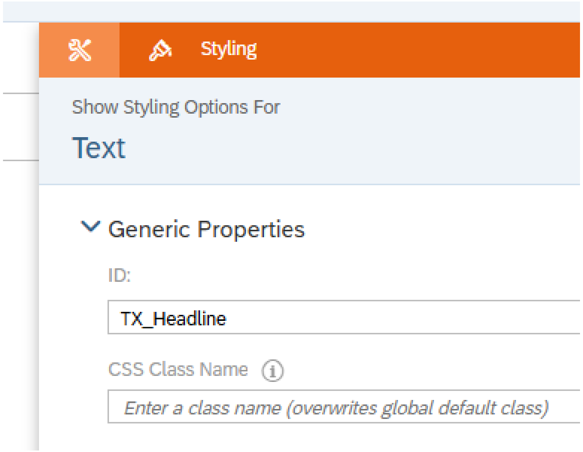


Page_1 – onInitialization
Called when the analytic application page has finished loading.

```
function onInitialization() : void
```

```
// URL condensed vs full

if(viewMode === "condensed") {
    TabStrip_1.setVisible(false);}

    else{
        TabStrip_1.setVisible(true);
}
```

3. In order to have a text or a chart react and change view mode by on clicking: for example in the text headline in a panel you can add the following script:

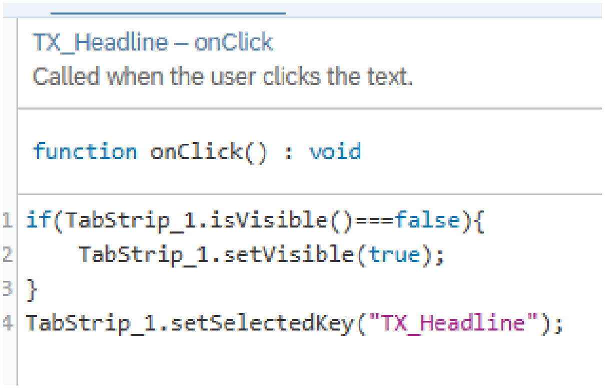


Show Styling Options For
Text

Generic Properties

ID: TX_Headline

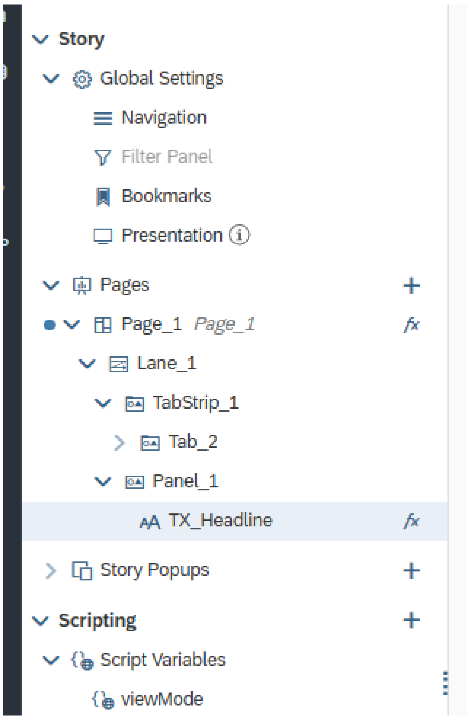
CSS Class Name: *Enter a class name (overwrites global default class)*



TX_Headline – onClick
Called when the user clicks the text.

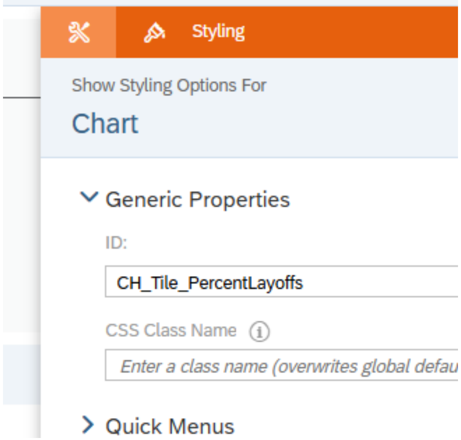
```
function onClick() : void
```

```
if(TabStrip_1.isVisible()===false){
    TabStrip_1.setVisible(true);
}
TabStrip_1.setSelectedKey("TX_Headline");
```



Combine clicking on a (numeric chart) to open for example a specific tab in a tab strip and switching between tabs

1. Create numeric chart/tile, add onClick Script that will open Tab_2 (“Table”) in the tabstrip TS_content



Styling

Show Styling Options For

Chart

Generic Properties

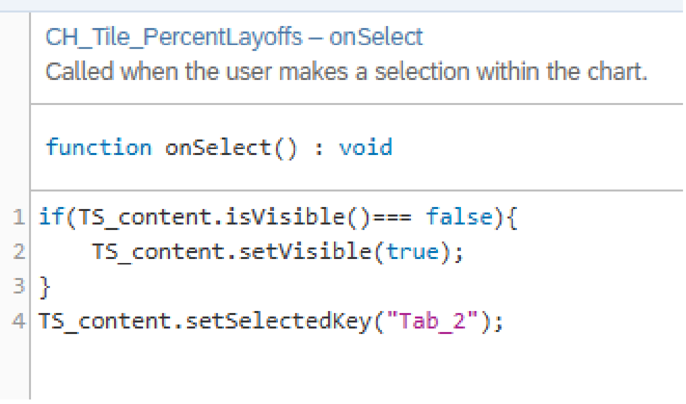
ID:

CH_Tile_PercentLayoffs

CSS Class Name

Enter a class name (overwrites global defau

Quick Menus

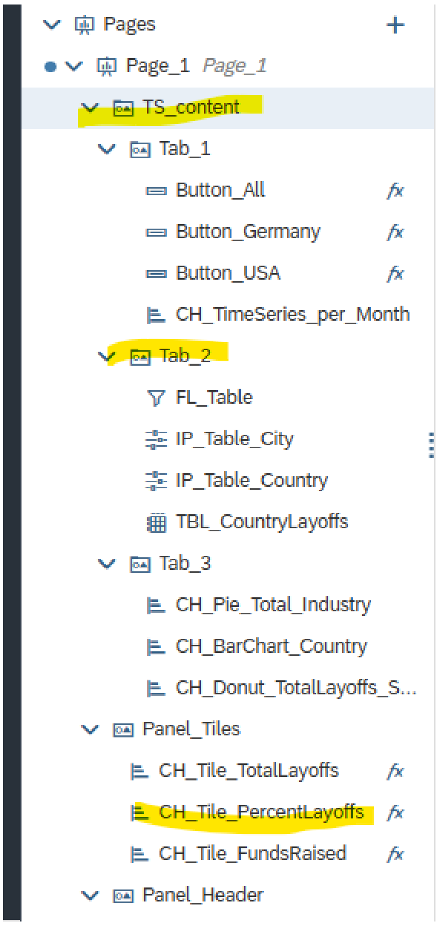


CH_Tile_PercentLayoffs – onSelect

Called when the user makes a selection within the chart.

```
function onSelect() : void
```

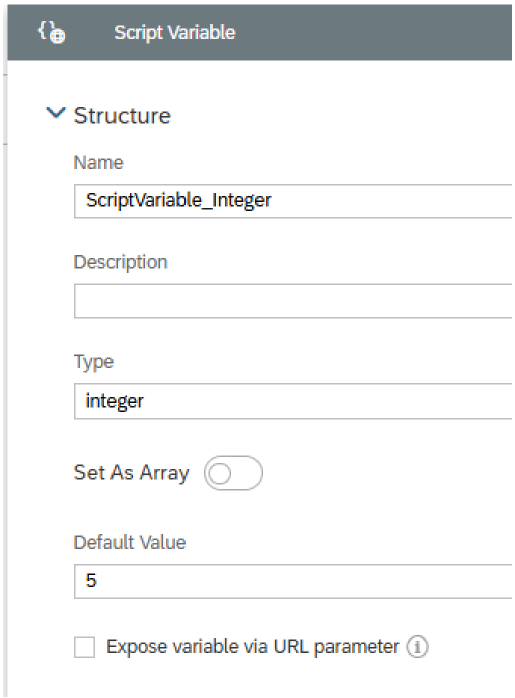
```
if(TS_content.isVisible()=== false){
    TS_content.setVisible(true);
}
TS_content.setSelectedKey("Tab_2");
```



- Pages
  - Page_1 *Page_1*
    - TS_content
      - Tab_1
        - Button_All
        - Button_Germany
        - Button_USA
        - CH_TimeSeries_per_Month
      - Tab_2
        - FL_Table
        - IP_Table_City
        - IP_Table_Country
        - TBL_CountryLayoffs
      - Tab_3
        - CH_Pie_Total_Industry
        - CH_BarChart_Country
        - CH_Donut_TotalLayoffs_S...
    - Panel_Tiles
      - CH_Tile_TotalLayoffs
      - CH_Tile_PercentLayoffs
      - CH_Tile_FundsRaised
    - Panel_Header

Script variable summation / add a number to default value (use case missing)

1. Create an integer script variable (here DV 5) and add a script to + 10 onInitialization
2. You get a 15 on the text area that was designated (here: TextArea_3, needs to be created beforehand)



Name: ScriptVariable_Integer

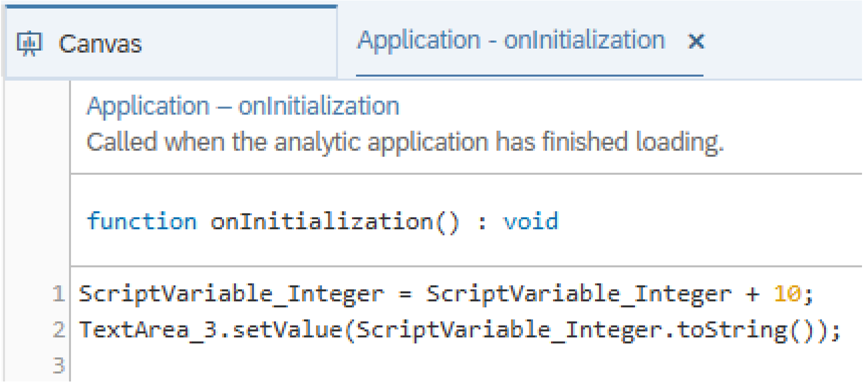
Description:

Type: integer

Set As Array ○

Default Value: 5

□ Expose variable via URL parameter



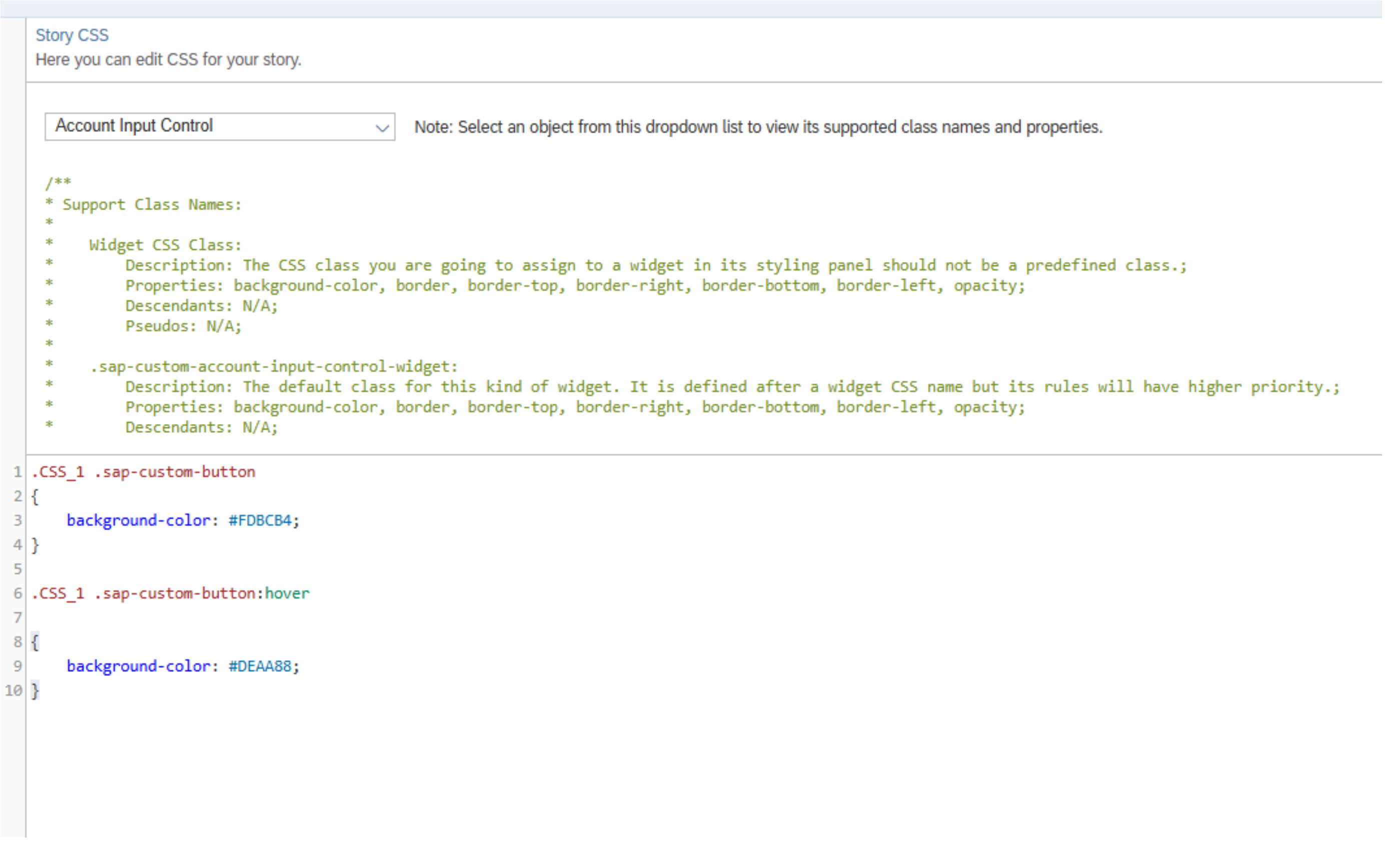
Application – onInitialization
Called when the analytic application has finished loading.

```
function onInitialization() : void
```

```
ScriptVariable_Integer = ScriptVariable_Integer + 10;
TextArea_3.setValue(ScriptVariable_Integer.toString());
```

Set CSS Class of a Button and change when hovering over it

1. Under edit CSS in SAC add a CSS class, and hovering color. For example:



Story CSS
Here you can edit CSS for your story.

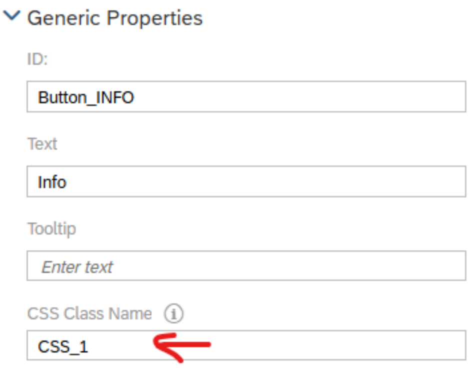
Account Input Control

Note: Select an object from this dropdown list to view its supported class names and properties.

```
/**
* Support Class Names:
*
*    Widget CSS Class:
*        Description: The CSS class you are going to assign to a widget in its styling panel should not be a predefined class.;
*        Properties: background-color, border, border-top, border-right, border-bottom, border-left, opacity;
*        Descendants: N/A;
*        Pseudos: N/A;
*
*    .sap-custom-account-input-control-widget:
*        Description: The default class for this kind of widget. It is defined after a widget CSS name but its rules will have higher priority.;
*        Properties: background-color, border, border-top, border-right, border-bottom, border-left, opacity;
*        Descendants: N/A;
```

```
.CSS_1 .sap-custom-button
{
    background-color: #FDBCB4;
}

.CSS_1 .sap-custom-button:hover

{
    background-color: #DEAA88;
}
```

2. In addition, if a specific button should use this class, add the name of the CSS so it uses it,



Generic Properties

ID:

Button_INFO

Text

Info

Tooltip

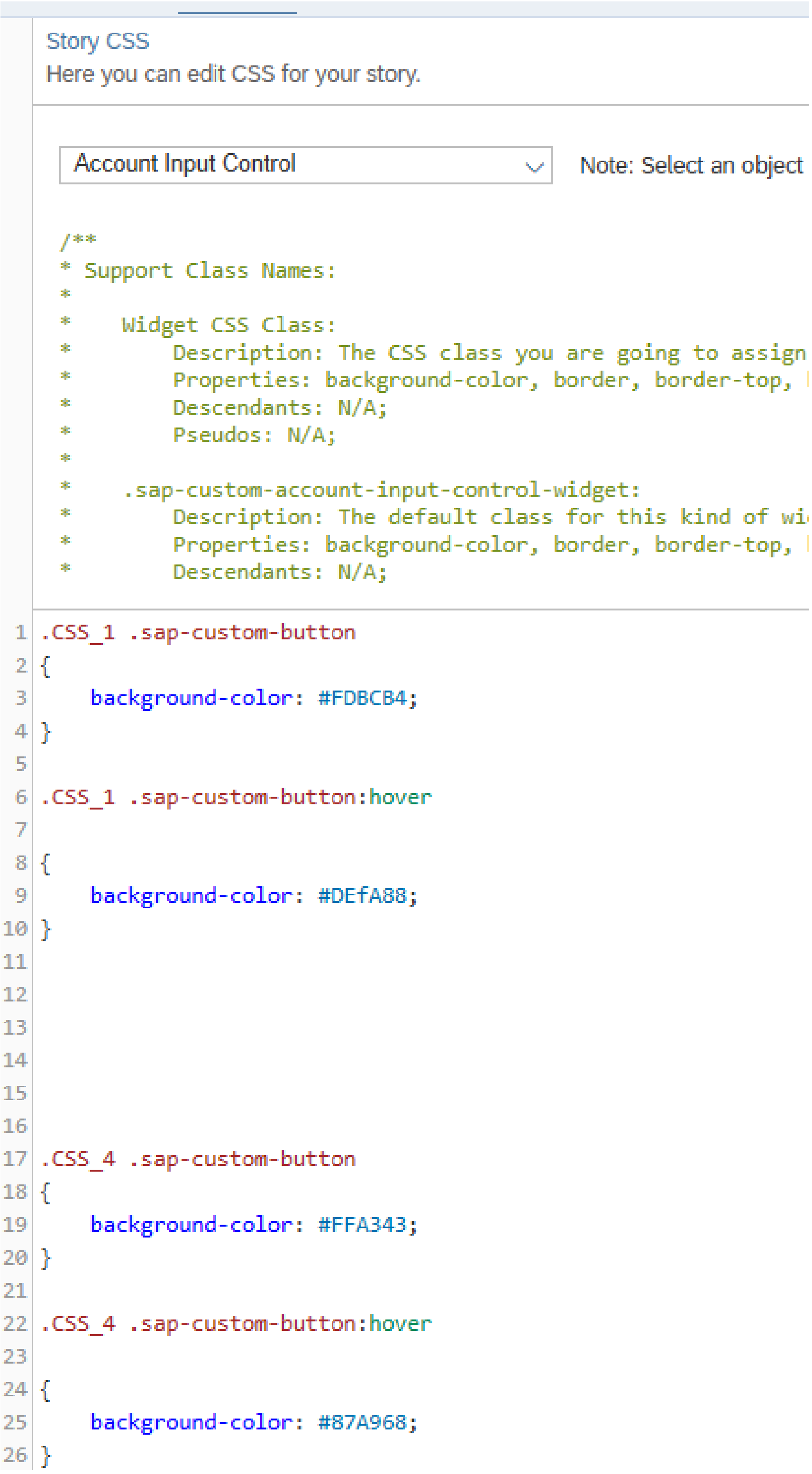
Enter text

CSS Class Name

CSS_1

Change the CSS class of a button (by clicking on another button (Button_2)

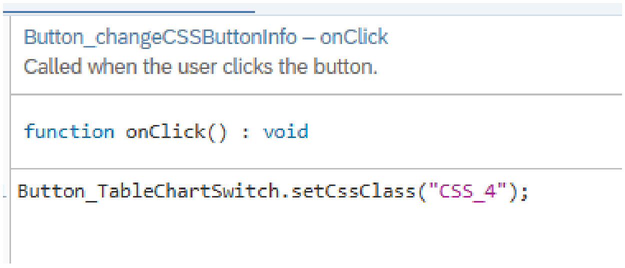
1. Create a CSS class , here CSS_4
2. Create a button that will be used for changing the css class of the other button. Add this script (the button name in the shown script is the button that will be changed):



Story CSS
Here you can edit CSS for your story.

```
/**
* Support Class Names:
*
*    Widget CSS Class:
*        Description: The CSS class you are going to assign
*        Properties: background-color, border, border-top,
*        Descendants: N/A;
*        Pseudos: N/A;
*
*    .sap-custom-account-input-control-widget:
*        Description: The default class for this kind of wi
*        Properties: background-color, border, border-top,
*        Descendants: N/A;
```

```
.CSS_1 .sap-custom-button
{
    background-color: #FDBCB4;
}

.CSS_1 .sap-custom-button:hover

{
    background-color: #DEfA88;
}






.CSS_4 .sap-custom-button
{
    background-color: #FFA343;
}

.CSS_4 .sap-custom-button:hover

{
    background-color: #87A968;
}
```



Button_changeCSSButtonInfo – onClick
Called when the user clicks the button.

```
function onClick() : void

Button_TableChartSwitch.setCssClass("CSS_4");
```