# Advanced mode scripting in SAP Analytics Cloud

**Kumail Saifuddin Saif**

kumail.saif@gmail.com
SAP Technical Architect & Projects Delivery Manager, Accenture LLP, USA

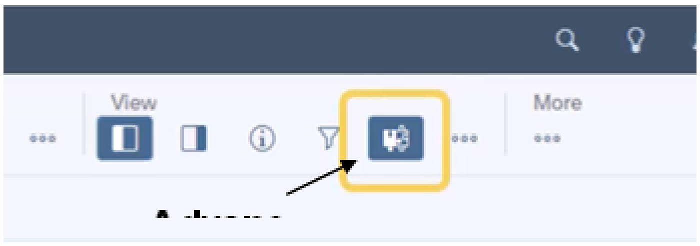
**Abstract: SAP Analytics cloud (SAC) is a strategic tool for reporting offered by SAP and it provides advanced scripting options to enhance the visualizations beyond the standard features. Scripting allows developers to create dynamic interactions between widgets and visualizations in a story. It also enables implementing if-else logic, loops, or more advanced mathematical computations. In this paper we will discuss the details on writing the scripts along with a couple of examples along with the option to debug the scripts.**

**Keywords: SAP Analytics Cloud (SAC), Story, Scripting, SAP Reporting, Advanced mode.**

**1 Introduction:** SAC Stories are most widely used as they provide various options for visual representations using tables, charts, geo maps, text widgets etc. There are lots of great features to build stories, however there could be more complex Business requirements for the presentation and layouts of the Business Data. These complex requirements can be achieved using the Advanced mode scripting option provided by SAP. It helps enhance interactivity, customize functionality, and extend the capabilities of analytics beyond what is possible with standard features. The key areas listed as below:

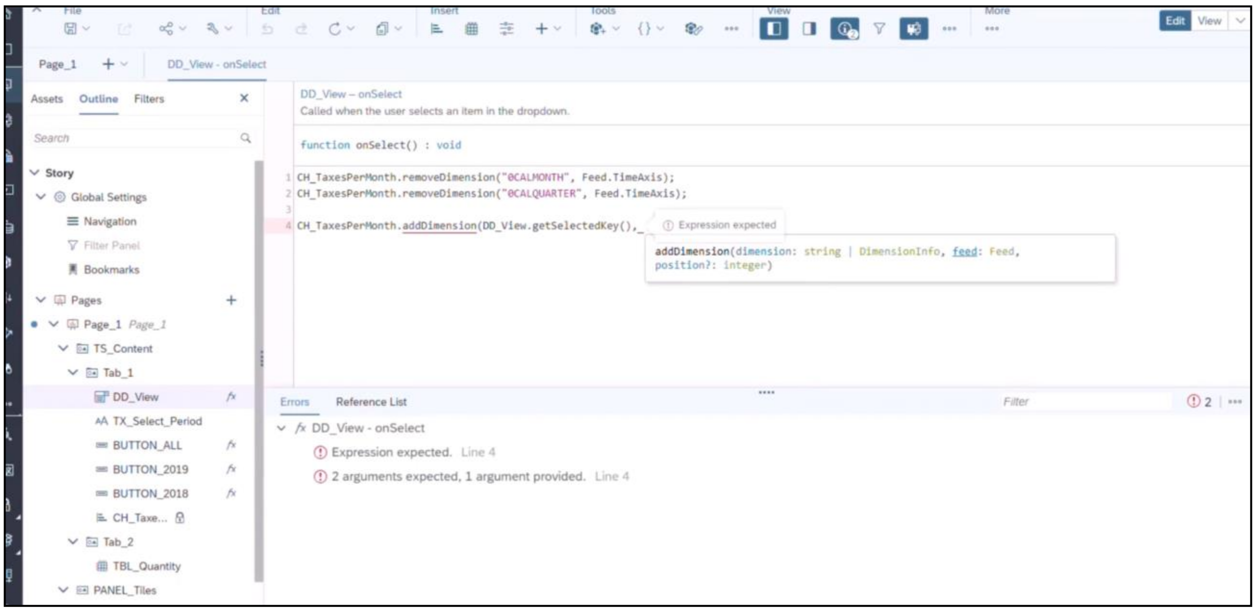
- Interaction
  - Automatically update multiple charts based on user selections or inputs.
  - Enable drill-through actions to navigate between different pages or stories.
  - Respond dynamically to filter changes or trigger actions based on specific events.
- Advanced Calculations
  - Custom aggregations or calculations that depend on multiple variables or conditions.
  - Implementing if-else logic, loops, or more advanced mathematical computations.
- Personalization
  - Display specific data views based on the user's department or region.
  - Customize story elements, such as showing or hiding widgets, based on user input.
  - Highlighting key insights dynamically based on thresholds or trends.
  - Creating custom tooltips, pop-ups, or interactive guides for users.
- Task Automation
  - Resetting filters or inputs to their default state with a single click.
  - Loading default views or data when the story is opened.
  - Performing sequential actions triggered by a single user event.
- Integration
  - Fetching data from external databases or APIs to display alongside SAC data.
  - Writing back data or sending updates to other systems based on user actions.
- Simulation
  - Simulate changes in sales, costs, or other KPIs to project future outcomes.
  - Apply user-defined input variables to modify charts and tables dynamically.

**2 Advanced Mode:**
Advanced Mode is provided by SAP to allow developers to add custom logic to the story with more widgets, additional functionality, and scripting capabilities.



To switch ON advanced mode, go to the View section in the toolbar, choose the Advanced Mode button. Using advanced mode you can add scripts to buttons, checkbox groups or dropdowns, input fields and text areas, and flow layout panels, pagebooks, and tab strips. It is also possible to create custom pop-ups using scripting. SAC also offers the use of APIs, script variables, and script objects, supported by the Script Editor and the Info Panel.

**3 Script Editor:**
Script Editor is used to write scripts for each widget, creating interactive and highly custom-defined stories. It has the code completion and value help functionality to help with the development of scripts. Code completion can be used using the keyboard shortcut Crtl + Space. Using this the editor suggests value help options, or functions to be used for the syntax of the script you are writing depending upon the context. SAC validates the script as you write it, so you can check the syntax of your string using the information panel. If there is any error, it is shown in the information panel at the bottom as seen in the example below.



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

```
function onSelect() : void

CH_TaxesPerMonth.removeDimension("0CALMONTH", Feed.TimeAxis);
CH_TaxesPerMonth.removeDimension("0CALQUARTER", Feed.TimeAxis);

CH_TaxesPerMonth.addDimension(DD_View.getSelectedKey(),_
```

Expression expected

```
addDimension(dimension: string | DimensionInfo, feed: Feed,
position?: integer)
```

DD_View - onSelect
Expression expected. Line 4
2 arguments expected, 1 argument provided. Line 4

**SAC Script Editor**

There are four main categories of the statements that can be written in script editor:
1. Call Statements - Calling API methods for an object like Chart or Button etc.
2. Conditional Execution Statements - IF … ELSE or SWITCH statements

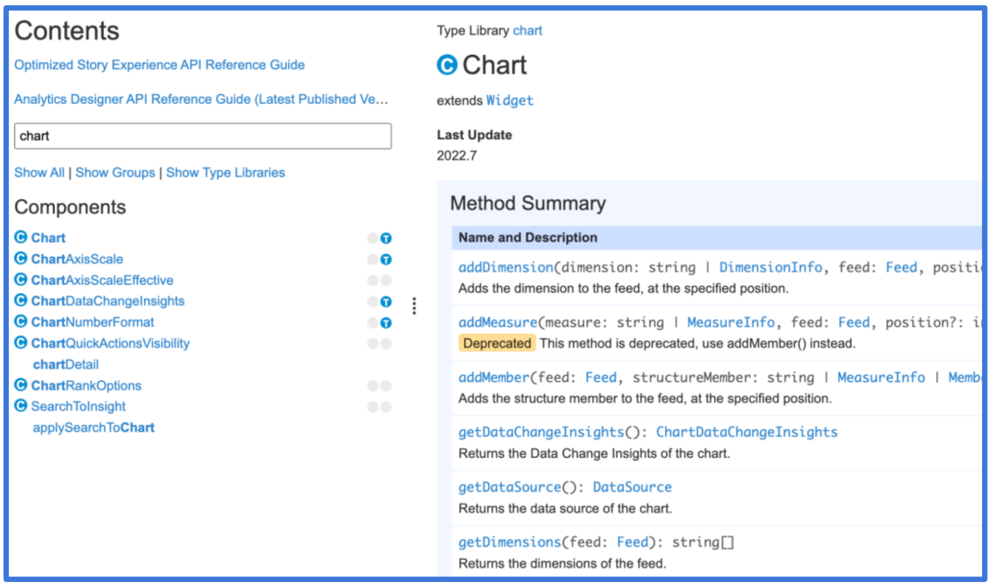
3. Loops - FOR or WHILE loop statements
4. Assignment Statements - Assigning values to variables

**4 API References:**

SAP Provides API Reference Library which is the set of APIs available to help write the scripts to create interactive and highly customised stories. Objects, functions, properties, methods, and events available in writing scripts are listed in the API reference provided by SAP. You can search the contents using a search box. You can either look for events such as OnClick event available for buttons or you can look for widgets like Tables or charts and search for all the available scripting options available.

The basic syntax for call statements is: *<Widget>.<Method>(<Arguments>);*

- *<Widget>* is the name of a data source alias or a component in your story.
- *<Method>* is an operation applied to the object specified on the left of the period.
- *<Arguments>* is a comma-separated list of expressions.



Contents

Optimized Story Experience API Reference Guide

Analytics Designer API Reference Guide (Latest Published Ve...

chart

Show All | Show Groups | Show Type Libraries

Components

- Chart
- ChartAxisScale
- ChartAxisScaleEffective
- ChartDataChangeInsights
- ChartNumberFormat
- ChartQuickActionsVisibility
- chartDetail
- ChartRankOptions
- SearchToInsight
- applySearchToChart

Type Library chart

Chart

extends Widget

Last Update
2022.7

Method Summary

| Name and Description |
|---|
| addDimension(dimension: string \| DimensionInfo, feed: Feed, positi<br>Adds the dimension to the feed, at the specified position. |
| addMeasure(measure: string \| MeasureInfo, feed: Feed, position?: i<br>Deprecated This method is deprecated, use addMember() instead. |
| addMember(feed: Feed, structureMember: string \| MeasureInfo \| Memb<br>Adds the structure member to the feed, at the specified position. |
| getDataChangeInsights(): ChartDataChangeInsights<br>Returns the Data Change Insights of the chart. |
| getDataSource(): DataSource<br>Returns the data source of the chart. |
| getDimensions(feed: Feed): string[]<br>Returns the dimensions of the feed. |

**Example 1 -** Let us take an example of the script as below to be used for a Button:

*CH_TaxesPerMonth.getDataSource().setDimensionFilter("0CALYEAR","2020");*

Here, *<Widget>* is CH_TaxesPerMonth, *<Method>* are getDataSource and setDimensionFilter, and *<Arguments>* are 0CALYEAR and 2020.

This is a simple script which can be assigned to a button using the OnClick event which can be used to filter the records for the year 2020. Similarly there could be multiple buttons designed in a story for different years of data to be filtered easily by the user.

Additionally there could be another button lets say, which is used to clear the filter. The script for the clearing button will look something like this.

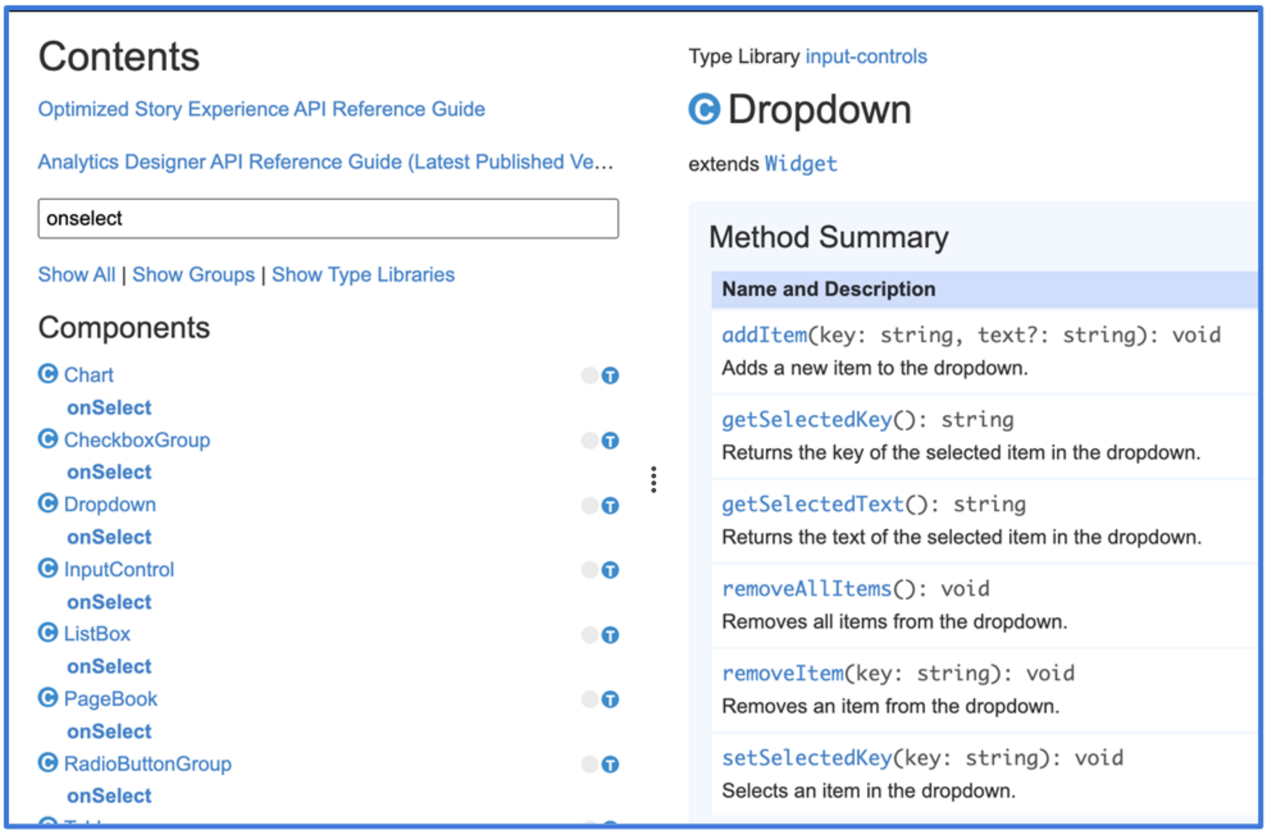
*CH_TaxesPerMonth.getDataSource().removeDimensionFilter("0CALYEAR");*

Above is a simple example where API references are used to design a filtering mechanism using buttons available on the screen. A user can filter the data easily by just a click of a button.

**Example 2 -** Let us take an example of the script as below to be used for a Dropdown:

Similar to the above example, we need the correct event for the dropdown menu option which is OnSelect. You can search for it from the API reference and then click on the DropDown option on the left to get a list

of all methods available. Here you can use the *getSelectedKey()* method to get the key of the selected item in the dropdown and then perform the operation to filter the results based on the key.
*CH_TaxesPerMonth.getDataSource().setDimensionFilter("0CALYEAR",Drop_Down.getSelectedKey());*


Contents

Optimized Story Experience API Reference Guide

Analytics Designer API Reference Guide (Latest Published Ve...

onselect

Show All | Show Groups | Show Type Libraries

Components

- Chart
  - onSelect
- CheckboxGroup
  - onSelect
- Dropdown
  - onSelect
- InputControl
  - onSelect
- ListBox
  - onSelect
- PageBook
  - onSelect
- RadioButtonGroup
  - onSelect

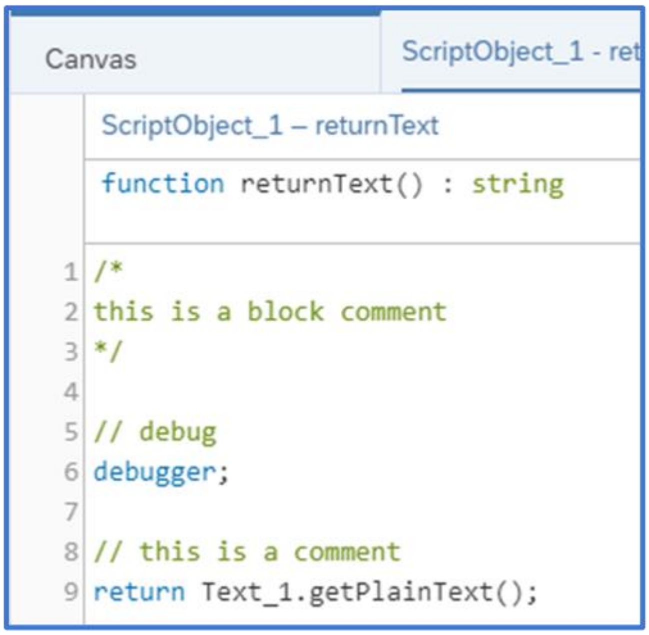
Type Library input-controls

Dropdown

extends Widget

Method Summary

| Name and Description |
|---|
| `addItem(key: string, text?: string): void` Adds a new item to the dropdown. |
| `getSelectedKey(): string` Returns the key of the selected item in the dropdown. |
| `getSelectedText(): string` Returns the text of the selected item in the dropdown. |
| `removeAllItems(): void` Removes all items from the dropdown. |
| `removeItem(key: string): void` Removes an item from the dropdown. |
| `setSelectedKey(key: string): void` Selects an item in the dropdown. |



ScriptObject_1 – returnText

```
function returnText() : string
```

```
/*
this is a block comment
*/

// debug
debugger;

// this is a comment
return Text_1.getPlainText();
```

**5 Debugging:**
SAP Analytics Cloud also provides the option to Debug the script. This is achieved using the *debugger;* statement in the script. It helps to create a breakpoint, so that the script execution is paused automatically wherever this statement is encountered. After setting the debugger command in the script you can launch the debug mode by adding the parameter *debug=true* to the URL.

In debug mode, you will see that your script is a transformed Javascript, however the comments that were written by you in the script are still made available at their locations, so that you can recognize your scripts in the web browser's developer tools.

**Conclusion:**
Scripting in SAP Analytics Cloud (SAC) stories empowers users to go beyond basic analytics, offering flexibility, interactivity, and the ability to tailor dashboards to specific business needs, ultimately leading to better insights and decision-making. API Reference library provides the details on the available methods for different objects and their events to be used to capture user interactions. The easy search option makes it very easy for a developer to find the correct methods or events for different object types like charts, tables, graphs etc. Debugging option provided helps to debug the script with simple options to check the values contained in the variables during the execution by setting a breakpoint using script command as well as runtime during the debugging.

**References:**
1 - SAP Analytics Cloud Help [Online]. Available at:
https://help.sap.com/docs/SAP_ANALYTICS_CLOUD/00f68c2e08b941f081002fd3691d86a7/1fb1f4ce92f44fc983debc25ac1f2cc9.html
2 - Building User Intuitive Stories in SAP Analytics Cloud (Tips & Tricks) [Online]. Available at:
https://community.sap.com/t5/technology-blogs-by-sap/building-user-intuitive-stories-in-sap-analytics-cloud-tips-tricks/ba-p/13486944
3 - API Reference Guide [Online]. Available at:
https://help.sap.com/doc/1639cb9ccaa54b2592224df577abe822/2023.8/en-US/index.html
4 - JavaScript Weak Typing [Online]. Available at:
https://code-basics.com/languages/javascript/lessons/data-types-weak-typing
5 - JavaScript Strict Mode [Online]. Available at:
https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Strict_mode
6 - How to Create Dynamic Images in SAP Analytics Cloud Story [Online]. Available at:
https://community.sap.com/t5/technology-blogs-by-members/part-1-how-to-create-dynamic-images-in-sap-analytics-cloud-story/ba-p/13429202
7 - SAC Product Documentation [Online]. Available at:
https://help.sap.com/docs/SAP_ANALYTICS_CLOUD?locale=en-US
8 - Apply Best Practices for Performance Optimization in Story Design [Online]. Available at:
https://help.sap.com/docs/SAP_ANALYTICS_CLOUD/00f68c2e08b941f081002fd3691d86a7/fbe339efda1241b5a3f46cf17f54cdff.html?locale=en-US
9 - Exploring SAP Analytics Cloud [Online]. Available at:
https://learning.sap.com/learning-journeys/exploring-sap-analytics-cloud
10 - Optimize the Story Builder Performance [Online]. Available at:
https://help.sap.com/docs/SAP_ANALYTICS_CLOUD/00f68c2e08b941f081002fd3691d86a7/bf77843031e04ce6b6e721e67ccadeaf.html?locale=en-US