

# Smart Flow Navigations

**Smart Flow Navigations** is a highly customizable Lightning Web Component (LWC) for Salesforce. It allows you to add multiple styled navigation buttons to a Flow screen. You can configure button labels, styles, icons, actions, and more using a simple JSON string.

Developed By: **Anirban Roy**

## Features

- Customizable buttons for **Previous**, **Next**, and **Finish** actions.
- Fully configurable via a simple JSON string, enabling you to customize button labels, styles, icons, actions, and more, including:
  - Button labels
  - Button variants (All supported variants or hex value)
  - Icons and icon alignments
  - Button icon and text color customization
  - Default Button visibility when action unavailable (disabled or hide)
  - Specify Button values explicitly when clicked
- Easy to use within Salesforce Flow screens.
- Supports multiple button configurations at once.
- Ability to display a horizontal line above the buttons for separation.
- Button layout alignments (left, center, right): Align the buttons as required.

To install **Smart Flow Navigations**, follow these steps:

## Installation

1. You can install the **Smart Flow Navigations** component directly into your Salesforce environment using the following links:
  - **Production Environment:** [Install the package in Production](#)
  - **Sandbox Environment:** [Install the package in Sandbox / Scratch Org](#)
2. Add the component to a Flow screen:

- In your Salesforce Flow, Search for the component `Smart Flow Navigations` and add it to a screen where you want the buttons to appear.

## Input Properties

### Buttons List (Required)

- **Type:** String
- **Description:** A JSON array of objects defining the button configurations, including labels, actions, variants, icons, icon colors, and visibility.
- **Example:**

```
[
  {
    "label": "Previous",
    "isNextAction": false,    // For "Next" or "Finish" actions (set to true for "Next")
    "variant": "neutral",
    "outputValue": "Previous",
    "icon": "utility:chevronleft",
    "iconAlignment": "left",
    "hideWhenUnavailable": true // Set to true to hide or disable button when the action is un
  },
  {
    "label": "Next",
    "isNextAction": true,
    "variant": "#1565C0",
    "outputValue": "Next",
    "icon": "utility:chevronright",
    "iconColor": "#fff",
    "labelColor": "#fff",
    "iconAlignment": "right",
    "hideWhenUnavailable": false
  }
]
```

- `label` : **(Required)** The text to be displayed on the button.
- `isNextAction` : **(Optional)** A boolean value indicating whether the button is a "Next" or "Finish" action (set to `true` ), and `false` for the "Previous" action.

- **variant** : **(Required)** The button variant. Supported values include: `neutral` (no color), `blue`, `blue outlined`, `green`, `red`, `red outlined`, or any valid hex color code (e.g., `#FF5733` for a custom color).
- **outputValue** : **(Optional)** The value returned when the button is clicked. This value is used by the Flow when performing decisions. If not explicitly defined, the `label` is returned by default.
  - **Note**: If the `outputValue` is specified as an integer, it will be returned as a string (e.g., `outputValue: 1` will return `"1"` ).
- **icon** : **(Optional)** The LDS icon name to display on the button. E.g. `utility:forward`, `action:delete`, where 'utility' is the category. Refer to this [Lightning Design System icons](#) documentation for all the supported icons.
- **iconColor** : **(Optional)** The color of the icon. This can be a valid hex color code (e.g., `#FF5733` ), or if not specified, the default will be `#fff` (white). If the button has a variant specified (e.g., `blue`, `green` ), and `iconColor` is not explicitly set, the icon color will match the color of the button variant.
- **labelColor** : **(Optional)** The color of the button text. This can be a valid hex color code (e.g., `#FF5733` ), or if not specified, the default will be `#fff` (white). If the button has a variant specified (e.g., `blue`, `green` ), and `labelColor` is not explicitly set, the icon color will match the color of the button variant.
- **iconAlignment** : **(Required when icon is specified)** Specifies where to align the icon with respect to the button label. Expected values are `left` and `right` .
- **hideWhenUnavailable** : **(Optional)** Boolean value to specify whether the button should be hidden or just disabled when the action is unavailable. A button is considered unavailable when its corresponding action is not possible (e.g., the Flow cannot proceed to the next step). The default behavior keeps the button disabled.

## Layout Alignment **(Required)**

- **Type**: String  
**Description**: Specify the alignment of the button layout within the Flow UI. Valid values are:
  - `left` : Align buttons to the left of the screen.
  - `center` : Align buttons in the center of the screen.
  - `right` : Align buttons to the right of the screen (default).
- **Default**: `right`

## Include a Horizontal Line? **(Optional)**

- **Type**: Boolean

- **Description:** Set this to `true` to display a horizontal line above the buttons for better separation from the rest of the screen.
- **Default:** `false`

## Output Properties

### Output Value

- **Type:** String
- **Description:** Returns the value of the button that was clicked. This output can be passed to a Decision block or other Flow elements to manage the flow routing.
- **Default:** Returns the `label` of the button if `outputValue` is not specified in button attributes.

## Example Usage in Flow

You can use the `Smart Flow Navigations` component in your Flow by following the steps below. For a complete demonstration of all features, refer to the example Flow **Smart\_Flow\_Buttons\_Example**, which you can optionally import with the project:

1. Add the **Smart Flow Navigations** component to your Flow screen.
2. Set the `Buttons List` input property with a JSON string of button configurations.
3. Specify the button layout alignment ( `left` , `center` , or `right` ) in the `Layout Alignment` property.
4. Optionally, set the `Include a Horizontal Line?` property to `true` to display a line above the buttons.
5. Use the `outputValue` property to define a button click result value for flow routing (e.g., with a Decision element).

**Tip:** Once the **Smart Flow Navigations** component is added and configured, you can hide the default Flow buttons layout (such as "Next," "Previous," or "Finish") from screen configurations to prevent redundancy and maintain a cleaner interface.

# Error Handling:

- **Description:** If the Buttons List input contains invalid JSON or incorrect attribute values, an error message will be displayed during testing. The component highlights configuration issues, helping you quickly debug and correct them.

## Notes

- Ensure the Buttons List input contains valid JSON syntax..
- This component is designed to be used specifically within Salesforce Flows (Flow Screens).

## Developer Information

- **Author:** Anirban Roy
- **Version:** 1.0
- **API Version:** 62.0