12 -FEB-2024

Advanced hamburger menu Version: 2.1.3

# **Overview**

 The Advanced Hamburger Menu Component adds a menu icon and a sliding menu to your app. When a user clicks on the menu icon, the sliding menu slides on to the screen from either the left or the right side.

You can place the menu icon either on the upper left or the upper right corner of your screen. If the menu icon is on the upper left, the menu slides in from the left side, and if the menu icon is on the upper right, the menu slides in from the right side.

The menu contains a header, menu items, and a footer. Each menu item can also contain submenu items.

## **Use case:**

## Now a days, Advanced Hamburger menu is used in most of the websites we use like amazon, flip cart and many ecommerce websites. For example, if you want to develop a shopping website which has up to 10 categories and some subcategories in those categories, then this component can be used as it reduces screen space and easy to navigate.

## **Percentage of re-use:**

Approximate 80 % of reuse. It sets an expectation of how much can be used out of the box, and how much needs to be customized for a specific app.

## **Features**

* Enriches your app with popular navigation pattern, Hamburger Menu.
* Flexible to display and hide the menu.
* Saves the screen space.
* Menu slides with animation onto the screen.
* Provides controls to customize the UI and to set events.
* Plugs into your app without any code.
* Added Target Container Support.

# **Getting Started**

## **Prerequisites**

Before you start using the Advanced Hamburger Menu component, ensure the following:

• [HCL Foundry](https://manage.hclvoltmx.com/)

• Volt MX Iris

## **Platforms Supported**

### Mobile

#### iOS

#### Android

### Tablets

### PWA

## **Importing the app**

## You can import the Forge components only into the apps that are of the Reference Architecture type.

##  **To import the Advanced Hamburger Menu** **component, do the following:**

## Open your app project in Volt MX Iris.

 2. In the Project Explorer, click the **Templates** tab.

 

3.Right-click **Components**, and then select **Import Component**. The **Import Component** dialog box appears.

 

4.Click **Browse** to navigate to the location of the component, select the component, and then click **Import**. The component and its associated widgets and modules are added to your project.

##

Once you have imported a component to your project, you can easily add the component to a form. For more information, refer [Add a Component to a Form](https://opensource.hcltechsw.com/volt-mx-docs/docs/documentation/Iris/iris_user_guide/Content/C_UsingComponents.html%22%20%5Cl%20%22add-a-component-to-a-form)

## **Building and previewing the app**

After performing all the above steps, you can build your app and run it on your device. For more information, you can refer to the [Building and Viewing an Application](https://opensource.hcltechsw.com/volt-mx-docs/docs/documentation/Iris/iris_user_guide/Content/Cloud_Build_in_VoltMX_Iris.html#cloud) section of the Volt MX User Guide.

 You can then run your app to see the Advanced Hamburger Menu work in real time.

# **References**

## **Dynamic Usage**

 You can also add **Advanced Hamburger Menu** component dynamically. To do so,

1. In the **Project Explorer**, on the **Projects** tab, click **Controllers** section to access the respective **Form Controller**. Create a method and implement the code snippet similar to the sample code mentioned below.

/\* Creating Advanced Hamburger Menu component instance \*/

var advHamMenu= new com.voltmx.advancedhamburgermenu (

 {

 id: "advHamMenu",

 isVisible: true,

 height: "100%"

 top: "0dp",

 left: "0dp",

 width: "100%",

 layoutType: voltmx.flex.FREE\_FORM,

 autogrowMode: voltmx.flex.AUTOGROW\_NONE,

 skin: "slFbox",

 clipBounds:true,

 masterType: constants.MASTER\_TYPE\_USERWIDGET,

 zIndex:12

 }, {}, {});

/\*Setting the component's properties\*/

 advHamMenu.headingText = "Username";

 advHamMenu.footerImageHeight = "28dp";

 advHamMenu.footerImageSrc = "logout\_whitebackground.png";

 advHamMenu.footerImageWidth = "28dp";

 advHamMenu.profileImageSrc = "voltmx\_88.png";

 advHamMenu.profileImageTop = "15%";

 advHamMenu.slidingMenuIcon = "hamburger.png";

 advHamMenu.slidingMenuisVisible = true;

 advHamMenu.subHeadingText = "Email";

 advHamMenu.slidingMenuDirection = "Left";

 advHamMenu.headerAnimation = "Slide In";

 advHamMenu.slidingMenuAnimation = "Push";

 advHamMenu.animationSpeed = "0.25";

 advHamMenu.footerStyle = "Image & Text";

 advHamMenu.profileImageType = "Rounded Corner";

 advHamMenu.showMainMenuImage = true;

 advHamMenu.showSubMenuImage = true;

 advHamMenu.menuOptionsSkin = "voltmxsmsknlblSanFranciscoD8D8D8129";

 /\*Adding the Advanced Hamburger Menu component to a Form\*/

this.view.add(advHamMenu);

/\*Calling APIs form code \*/

var menuData =[

{menuItemName:'opt1', menuItemIcon:'acme.png'},

{menuItemName:'opt2',menuItemIcon:'acme.png'}

];

/\*Array of JSON Objects with submenu items \*/

var submenuData =[

{menuItemName:'opt1', submenuItemName:'subOpt1', submenuItemIcon:"acme.png"},

{menuItemName:'opt1', submenuItemName:'subOpt2', submenuItemIcon:"acme.png"},

];

this.view.advHamMenu.setData(menuData,submenuData);

In the code snippet, you can edit the properties of the component as per your requirement. For more information, see Setting Properties.

2. Save the file

## **Properties**

 The properties provided on the **Component** tab allows you to customize the elements in the **Advanced Hamburger Menu** component. These elements can be UI elements, service parameters, and so on. You can set the properties from the Volt MX Iris Properties panel on the right-hand side. You can also configure these properties using a JavaScript code.

**1. General Properties**

1**. Sliding Menu Visibility (slidingMenuVisibility)**

|  |  |
| --- | --- |
| **Description:** | Enables or disables the visibility of sliding menu. |
| **Syntax:** | slidingMenuIsVisible |
| **Type:** | Boolean |
| **Read/Write:** | Read + Write |
| **Example:** | this. view. advHamMenu. slidingMenuIsVisible = true; |
| **Remarks:** | The default value for the property is true. |

**2. Sliding Menu Icon (slidingMenuIcon)**

|  |  |
| --- | --- |
| **Description:** | Specifies the source of the image that you want to set as the sliding menu icon. |
| **Syntax:** | slidingMenuIcon |
| **Type:** | String |
| **Read/Write:** | Read + Write |
| **Example:** | this. view. advHamMenu. slidingMenuIcon = "image.png"; |
| **Remarks:** | The default value for the property is “image.png”. |

 **3. Menu Options (menuOptions)**

|  |  |
| --- | --- |
| **Description:** |  Specifies the list of menu items that you want to add to the sliding menu  |
| **Syntax:** | menuOptions |
| **Type:** | Data Grid, Array of JSON |
| **Read/Write:** | Read + Write |
| **Example:** | this. view. advHamMenu.menuOptions = [{“menuItemIcon”: “Sports\_Icon.png”, “menuItemName: “Sports”},{“menuItemIcon”: “News\_Icon.png”, “menuItemName: “News”}]; |
| **Remarks:** | This property does not reflect on the preview canvas of Volt MX Iris |

**4. Submenu Options(subMenuOptions)**

|  |  |
| --- | --- |
| **Description:** |  Specifies the list of sub-menu items that you want to add to the sliding menu  |
| **Syntax:** | subMenuOptions |
| **Type:** | Data Grid, Array of JSON |
| **Read/Write:** | Read + Write |
| **Example:** | this. view. advHamMenu.subMenuOptions = [{“subMenuItemName”: “suboption1”, “menuItemName: “Sports”, “subMenuItemIcon”: “sports\_sub.png”},{“subMenuItemName”: “suboption1”, “menuItemName: “News”, “subMenuItemIcon”: “news\_sub.png”}]; |
| **Remarks:** | This property does not reflect on the preview canvas of Volt MX Iris |

**5****. Enable Header(enableHeader)**

|  |  |
| --- | --- |
| **Description:** | Enables or disables the visibility of the header. |
| **Syntax:** | enableHeader |
| **Type:** | Boolean |
| **Read/Write:** | Read + Write |
| **Example:** | this. view. advHamMenu. enableHeader= true; |
| **Remarks:** | The default value for the property is true. |

**6. Placeholder Visibility(placeholderVisibility)**

|  |  |
| --- | --- |
| **Description:** | Enables or disables the visibility of the placeholder content  |
| **Syntax:** | placeholderVisibility |
| **Type:** | Boolean |
| **Read/Write:** | Read + Write |
| **Example:** | this. view. advHamMenu. placeholderVisibility = true; |
| **Remarks:** | This property is only for user reference. It shows where the app content displays at run-time.Make sure that you disable the placeholder before you build your app. |

**7. Enable Seperator(****enableSeperator)**

|  |  |
| --- | --- |
| **Description:** | Enables or disables the separator between the menu items. |
| Syntax: | enableSeperator |
| **Type:** | Boolean |
| **Read/Write:** | Read + Write |
| **Example:** | this. view. advHamMenu. enableSeperator = false; |
| **Remarks:** | The default value for the property is false. |

**8. Enable Menu Item Animation(****enableMenuItemAnimation)**

|  |  |
| --- | --- |
| **Description:** | Enables the animation of the menu items of Sliding Menu. |
| **Syntax:** | enableMenuItemAnimation |
| **Type:** | Boolean |
| **Read/Write:** | Read + Write |
| **Example:** | this. view. advHamMenu. enableMenuItemAnimation = true; |
| **Remarks:** | The default value for the property is true. |

9**. Sliding Menu Animation Speed(animationSpeed)**

|  |  |
| --- | --- |
| **Description:** | Specifies the time that the component takes to complete the sliding animation. |
| **Syntax:** | animationSpeed |
| **Type:** | String |
| **Read/Write:** | Read + Write |
| **Example:** | this. view. advHamMenu. animationSpeed = "0.4"; |
| **Remarks:** | The default value for the property is “0.4”. |

**10.** **Enable Main Menu Item Image(****showMainMenuImage)**

|  |  |
| --- | --- |
| **Description:** | Enables or disables the icon image for the menu items. |
| **Syntax:** | showMainMenuImage |
| **Type:** | Boolean |
| **Read/Write:** | Read + Write |
| **Example:** | this. view. advHamMenu. showMainMenuImage = true |
| **Remarks:** | The default value for the property is true. |

**11. Enable Sub Menu Item Image(s****howSubMenuImage)**

|  |  |
| --- | --- |
| **Description:** | Enables or disables the icon image for the sub-menu items |
| **Syntax:** | showSubMenuImage |
| **Type:** | Boolean |
| **Read/Write:** | Read + Write |
| **Example:** | this. view. advHamMenu. showSubMenuImage = true; |
| **Remarks:** | The default value for the property is true. |

**12. Menu Dropdown Image (imgShowSubMenu)**

|  |  |
| --- | --- |
| **Description:** | Specifies the source of the chevron image for menu items. |
| **Syntax:** | imgShowSubMenu |
| **Type:** | String |
| **Read/Write:** | Read + Write |
| **Example:** | this. view. advHamMenu. imgShowSubMenu= “black\_chevron.png”; |
| **Remarks:** | The default value for the property is “black\_chevron.png”. |

 **13. Content Text (****contentText)**

|  |  |
| --- | --- |
| **Description:** | Specifies the content when any menu or sub-menu item is clicked. |
|  **Syntax:** | contentText |
| **Type:** | String |
| **Read/Write:** | Read + Write |
| **Example:** | this. view. advHamMenu. contentText = “Drag App Content Here”; |
| **Remarks:** | The default value for the property is “Drag App Content Here”. |

 **14. Closing Icon (closeIcon)**

|  |  |
| --- | --- |
| **Description:** | Specifies the image source of close icon. |
|  **Syntax:** | closeIcon |
| **Type:** | String |
| **Read/Write:** | Read + Write |
| **Example:** | this. view. advHamMenu. closeIcon = “black\_cross.png”; |
| **Remarks:** | The default value for the property is “ black\_cross.png”. |

**2.Transitions**

**15. Sliding Menu Direction(****slidingMenuDirection)**

|  |  |
| --- | --- |
| **Description:** | Specifies the direction from which the sliding menu slides on the screen.  |
| **Syntax:** | slidingMenuDirection |
| **Type:** | List Selector, String |
| **Read/Write:** | Read + Write |
| **Example:** | this. view. advHamMenu.slidingMenuDirection= "Right"; |
| **Remarks:** | The default value for the property is “Left”. |
| **Values:**  | Left: The menu slides from left side of the screen.Right: The menu slides from the right side of the screen. |

**16. Header Animation(headerAnimation)**

|  |  |
| --- | --- |
| **Description:** | Specifies the animation that you want to set for the header. |
| **Syntax:** | headerAnimation |
| **Type:** | List Selector, String |
| **Read/Write:** | Read + Write |
| **Example:** | this. view. advHamMenu.headerAnimation = “Slide In”; |
| **Remarks:** | The default value for the property is “No Animation” |

**17. Sliding Menu Transisitions(****slidingMenuAnimation)**

|  |  |
| --- | --- |
| **Description:** | Specifies the interaction of the sliding menu with app screen during its transition  |
| **Syntax:** | slidingMenuAnimation |
| **Type:** | List Selector, String |
| **Read/Write:** | Read + Write |
| **Example:** | this. view. advHamMenu. slidingMenuAnimation = “Push”; |
| **Remarks:** | The default value for the property is “Reveal”. |

**3.Headings**

 **18. Heading Height (headingHeight)**

|  |  |
| --- | --- |
| **Description:** | Specifies the height of the heading label. You can specify the height in Dp, Px, or percentage. |
| **Syntax:** | headerHeight |
| **Type:** | String |
| **Read/Write:** | Read + Write |
| **Example:** | this. view. advHamMenu.headingHeight = “5%”; |
| **Remarks:** | The default value for the property is “Preferred” |

 **19.** **Heading Left (****headingLeft)**

|  |  |
| --- | --- |
| **Description:** | Specifies the left edge of the heading label. You can specify the value in Dp, Px, or percentage. |
| **Syntax:** | headingLeft |
| **Type:** | String |
| **Read/Write:** | Read + Write |
| **Example:** | this. view. advHamMenu. headingLeft = ”28.5%”; |
| **Remarks:** | The default value for the property is “28.5%”; |

 **20. Header Top (****headingTop)**

|  |  |
| --- | --- |
| **Description:** | Specifies the top edge of the heading label. You can specify the value in Dp, Px, or percentage |
| **Syntax:** | headingTop |
| **Type:** | String |
| **Read/Write:** | Read + Write |
| **Example:** | this. view. advHamMenu.headingTop = "52%"; |
| **Remarks:** | The default value for the property is “52%”. |

  **21. Heading Width (headingWidth)**

|  |  |
| --- | --- |
| **Description:** | Specifies the width of the heading label. You can specify the value in Dp, Px, or percentage |
| **Syntax:** | headingWidth |
| **Type:** | String |
| **Read/Write:** | Read + Write |
| **Example:** | this. view. advHamMenu. headingWidth = "5%"; |
| **Remarks:** | The default value for the property is “Preferred”. |

 **22.** **Heading Text(****headingText)**

|  |  |
| --- | --- |
| **Description:** | Specifies the text that you want to display on the heading label. |
| **Syntax:** | headingText |
| **Type:** | String |
| **Read/Write:** | Read + Write |
| **Example:** | this. view. advHamMenu. headingText = "Username"; |
| **Remarks:** | The default value for the property is “Username”. |

 **23. Heading Text Visibility(****headingTextIsVisible)**

|  |  |
| --- | --- |
| **Description:** | Enables or disables the visibility of the heading label. |
| **Syntax:** | headingTextIsVisible |
| **Type:** | Boolean |
| **Read/Write:** | Read + Write |
| **Example:** | this. view. advHamMenu. headingTextIsVisible = true; |
| **Remarks:** | The default value for the property is true. |

 **24. Sub Heading Height(****subHeadingHeight)**

|  |  |
| --- | --- |
| **Description:** | Specifies the height of the sub-heading label. You can specify the value in Dp, Px, or percentage |
| **Syntax:** | subHeadingHeight |
| **Type:** | String |
| **Read/Write:** | Read + Write |
| **Example:** | this. view. advHamMenu. subHeadingHeight = "5%"; |
| **Remarks:** | The default value for the property is “Preferred”. |

**25. Sub Heading Left(****subHeadingLeft)**

|  |  |
| --- | --- |
| **Description:** | Specifies the left edge of the sub-heading label. You can specify the value in Dp, Px, or percentage |
| **Syntax:** | subHeadingLeft |
| **Type:** | String |
| **Read/Write:** | Read + Write |
| **Example:** | this. view. advHamMenu.subHeadingLeft = "29%"; |
| **Remarks:** | The default value for the property is “29%”. |

  **26. Sub Heading Text(****subHeadingText)**

|  |  |
| --- | --- |
| **Description:** |  Specifies the text that you want to display on the sub-heading label. |
| **Syntax:** | subHeadingText |
| **Type:** | String |
| **Read/Write:** | Read + Write |
| **Example:** | this. view. advHamMenu.subHeadingText = "Email ID"; |
| **Remarks:** | The default value for the property is “Email ID”. |

 **27.** **Sub Heading Text Visibility** **(****subHeadingTextIsVisible)**

|  |  |
| --- | --- |
| **Description:** |  Enables or disables the visibility of the sub-heading label. |
| **Syntax:** | subHeadingTextIsVisible |
| **Type:** | Boolean |
| **Read/Write:** | Read + Write |
| **Example:** | this. view. advHamMenu.subHeadingTextIsVisible = true; |
| **Remarks:** | The default value for the property is true. |

**28. Sub Heading Top(****subHeadingTop)**

|  |  |
| --- | --- |
| **Description:** | Specifies the top edge of the sub-heading label. You can specify the value in Dp, Px, or percentage |
| **Syntax:** | subHeadingTop |
| **Type:** | String |
| **Read/Write:** | Read + Write |
| **Example:** | this. view. advHamMenu.subHeadingTop = "74%"; |
| **Remarks:** | The default value for the property is “74%”. |

**29. Sub Heading Width(subHeadingWidth)**

|  |  |
| --- | --- |
| **Description:** | Specifies the width of the sub-heading label. You can specify the value in Dp, Px, or percentage |
| **Syntax:** | subHeadingWidth |
| **Type:** | String |
| **Read/Write:** | Read + Write |
| **Example:** | this. view. advHamMenu.subHeadingWidth = "30%"; |
| **Remarks:** | The default value for the property is “Preferred”. |

**4. Profile Options**

**30.** **Profile Image Visibility (****profileImageIsVisible)**

|  |  |
| --- | --- |
| **Description:** |  Enables or disables the visibility of the profile image. |
| **Syntax:** | profileImageIsVisible |
| **Type:** | Boolean |
| **Read/Write:** | Read + Write |
| **Example:** | this. view. advHamMenu. profileImageIsVisible = true; |
| **Remarks:** | The default value for the property is true. |

**31.** **Profile Image Source(****profileImageSource)**

|  |  |
| --- | --- |
| **Description:** | Specifies the source of the image that you want to set as the profile image. |
| **Syntax:** | profileImageSrc |
| **Type:** | String |
| **Read/Write:** | Read + Write |
| **Example:** | this. view. advHamMenu. profileImageSrc = "profileimage.png"; |
| **Remarks:** | * The Make sure that the image file exists in the *workspace\resources\common* directory.
* The file name of the image should not contain any upper-case characters.
* Specify the image’s file name along with extension.
* Only PNG format images are supported.
 |

**32.** **Profile Image Type(****profileImageType)**

|  |  |
| --- | --- |
| **Description:** | Specifies the style that you want to set for the profile image. |
| **Syntax:** | profileImageType |
| **Type:** | List Selector, String |
| **Read/Write:** | Read + Write |
| **Example:** | this. view. advHamMenu. profileImageType = "Regular"; |
| **Remarks:** | The default value for the property is “Regular”. |
| **Values:** | * **Regular**: Displays the profile image as provided.
* **Rounded Corner**: Displays the profile image with rounded corners.
* **Circle**: Displays the profile image in a circle.
 |

 **33****. Profile Image Left(****profileImageLeft)**

|  |  |
| --- | --- |
| **Description:** | Specifies the left edge of the profile image. You can specify the value in Dp, Px, or percentage |
| **Syntax:** | profileImageLeft |
| **Type:** | String |
| **Read/Write:** | Read + Write |
| **Example:** | this. view. advHamMenu. profileImageLeft = "30%"; |
| **Remarks:** | The default value for the property is “30%”. |

**34. Profile Image Top (****profileImageTop)**

|  |  |
| --- | --- |
| **Description:** | Specifies the top edge of the profile image. You can specify the value in Dp, Px, or percentage |
| **Syntax:** | profileImageTop |
| **Type:** | String |
| **Read/Write:** | Read + Write |
| **Example:** | this. view. advHamMenu. profileImageTop= "15%"; |
| **Remarks:** | The default value for the property is “15%”. |

**35. Profile Image Height(profileImageHeight)**

|  |  |
| --- | --- |
| **Description:** | Specifies the height of the profile image. You can specify the value in Dp, Px, or percentage |
| **Syntax:** | profileImageHeight |
| **Type:** | String |
| **Read/Write:** | Read + Write |
| **Example:** | this. view. advHamMenu. profileImageHeight= "50dp"; |
| **Remarks:** | The default value for the property is “50dp”. |

**36. Profile Image Width(profileImageWidth)**

|  |  |
| --- | --- |
| **Description:** | Specifies the width of the profile image. You can specify the value in Dp, Px, or percentage |
| **Syntax:** | profileImageWidth |
| **Type:** | String |
| **Read/Write:** | Read + Write |
| **Example:** | this. view. advHamMenu. profileImageWidth= "50dp"; |
| **Remarks:** | The default value for the property is “50dp”. |

**5. Footer Options**

**37. Footer Style (****footerStyle)**

|  |  |
| --- | --- |
| **Description:** | Specifies the style that you want to set for the footer. |
| **Syntax:** | footerStyle |
| **Type:** | List Selector, String |
| **Read/Write:** | Read + Write |
| **Example:** | this. view. advHamMenu.footerStyle = "Image & Text”; |
| **Remarks:** | The default value for the property is “No Footer”. |
| **Value:** | * **No Footer**: The menu does not display any footer.
* **Only Text**: The menu shows only text in the footer.
* **Only Image**: The menu shows only an image in the footer.
* **Image & Text**: The menu shows both text and an image.
 |

**38. Footer Text (****footerText)**

|  |  |
| --- | --- |
| **Description:** | Specifies the text that you want to display on the footer. |
| **Syntax:** | footerText |
| **Type:** | String |
| **Read/Write:** | Read + Write |
| **Example:** | this. view. advHamMenu.footerText = " Logout”; |
| **Remarks:** | The default value for the property is “Logout”. |

**39. Footer Image Source(f****ooterImageSrc)**

|  |  |
| --- | --- |
| **Description:** | Specifies the source of the image that you want to display on the footer. |
| **Syntax:** | footerImageSrc |
| **Type:** | String |
| **Read/Write:** | Read + Write |
| **Example:** | this. view. advHamMenu. footerImageSrc = " footerimage.png”; |
| **Remarks:** | * The property takes effect only when the value of the *footerStyle* property is IMAGE&TEXT or FOOTER IMAGE.
* Make sure that the image file exists in the *workspace\resources\common* directory.
* The file name of the image should not contain any upper-case characters.
* Specify the image’s file name along with the extension.
* Only PNG format images are supported.
 |

**40. Footer Image Height (footerImageHeight)**

|  |  |
| --- | --- |
| **Description:** | Specifies the height of the image on the footer. You can specify the height in Dp, Px, or percentage. |
| **Syntax:** | footerImageHeight |
| **Type:** | String |
| **Read/Write:** | Read + Write |
| **Example:** | this. view. advHamMenu.footerImageHeight = " 30dp”; |
| **Remarks:** | The default value for the property is “30dp”. |

**41. Footer Image Width (f****ooterImageWidth)**

|  |  |
| --- | --- |
| **Description:** | Specifies the width of the image on the footer. You can specify the width in Dp, Px, or percentage |
| **Syntax:** | footerImageWidth |
| **Type:** | String |
| **Read/Write:** | Read + Write |
| **Example:** | this. view. advHamMenu. footerImageWidth = " 30dp”; |
| **Remarks:** | The default value for the property is “30dp”. |

**Mobile**

**42.** **Target Container Left (%) (****slideMenuAnimMobile)**

|  |  |
| --- | --- |
| **Description:** | Specifies the left of the sliding menu animation for mobile.  |
| **Syntax:** | slideMenuAnimMobile |
| **Type:** | String |
| **Read/Write:** | Read + Write |
| **Example:** | this. view. advHamMenu. slideMenuAnimMobile = " 80%”; |
| **Remarks:** | The default value for the property is “80%”. |

**Tablet**

**43. Target Container Left (%) (****slideMenuAnimTablet)**

|  |  |
| --- | --- |
| **Description:** | Specifies the left of the sliding menu animation for tablet. |
| **Syntax:** | slideMenuAnimTablet |
| **Type:** | String |
| **Read/Write:** | Read + Write |
| **Example:** | this. view. advHamMenu. slideMenuAnimTablet = " 52%”; |
| **Remarks:** | The default value for the property is “52%”. |

**Web**

**44. Target Container Left (%) (****slideMenuAnimWeb)**

|  |  |
| --- | --- |
| **Description:** | Specifies the left of the sliding menu animation for Web. |
| Syntax: | slideMenuAnimWeb |
| **Type:** | String |
| **Read/Write:** | Read + Write |
| **Example:** | this. view. advHamMenu. slideMenuAnimWeb = " 30%”; |
| **Remarks:** | The default value for the property is “30%”. |

**Skins Section**

**45. Menu Options Text Skin(****menuTextSkin)**

|  |  |
| --- | --- |
| **Description:** | This skin links to the text of the menu items in the Sliding Menu. |
| **Syntax:** | menuTextSkin |
| **Type:** | String |
| **Widget Type:** | Label |
| **Example:** | this. view. advHamMenu. menuTextSkin = " “voltmxsmsknlblSanFranciscoD8D8D8129”; |
| **Remarks:** | The default value for the property is “voltmxsmsknlblSanFranciscoD8D8D8129”. |

**46. Sliding Menu Skin(slidingMenuSkin)**

|  |  |
| --- | --- |
| **Description:** | This skin links to the background of the sliding menu. |
| **Syntax:** | slidingMenuSkin |
| **Type:** | String |
| **Widget Type:** | Flex Container |
| **Example:** | this. view. advHamMenu. slidingMenuSkin = “voltmxsmsknFlxFFFFFF”; |
| **Remarks:** | The default value for the property is “voltmxsmsknFlxFFFFFF”. |

**47. Footer Text Skin(f****ooterTextSkin)**

|  |  |
| --- | --- |
| **Description:** | This skin links to the text on the footer. |
| **Syntax:** | footerTextSkin |
| **Type:** | String |
| **Widget Type:** | Label |
| **Example:** | this. view. advHamMenu. footerTextSkin = “voltmxsmsknlblSanFranciscoD8D8D8129”; |
| **Remarks:** | The This skin is applied only when the value of the *footerStyle* property is IMAGE&TEXT or FOOTER IMAGE.. |

**48. Heading Skin(****headingSkin)**

|  |  |
| --- | --- |
| **Description:** | This skin links text on the heading label. |
| **Syntax:** | headingSkin |
| **Type:** | String |
| **Widget Type:** | Label |
| **Example:** | this. view. advHamMenu. headingSkin = “voltmxsmsknlblSanFranciscoD8D8D8129”; |
| **Remarks:** | This skin is applied only when the value of the Header Text Visibility property is set to **true**. |

**49. Sub Heading Skin(subHeadingSkin)**

|  |  |
| --- | --- |
| **Description:** | This skin links text on the sub-heading label. |
| **Syntax:** | subHeadingSkin |
| **Type:** | String |
| **Widget Type:** | Label |
| **Example:** | this. view. advHamMenu. subHeadingSkin = “voltmxsmsknlblSanFranciscoD8D8D8129”; |
| **Remarks:** | This skin is applied only when the value of the Sub Header Text Visibility property is set to **true**. |

**Dynamic Widgets Skin**

**50. Menu Options (menuOptionsSkin)**

|  |  |
| --- | --- |
| **Description:** | This skin links to dynamically created menu and sub-menu options. |
| **Syntax:** | menuOptionsSkin |
| **Type:** | String |
| **Widget Type:** | Label |
| **Example:** | this. view. advHamMenu. menuOptionsSkin = “voltmxsmsknlblSanFranciscoD8D8D8129”; |
| **Remarks:** | The default value for the property is “voltmxsmsknlblSanFranciscoD8D8D8129” |

**51. Menu Item (menuItemSkin)**

|  |  |
| --- | --- |
| **Description:** | This skin links to dynamically created menu Items |
| **Syntax:** | menuItemSkin |
| **Type:** | String |
| **Widget Type:** | Flex Container  |
| **Example:** | this. view. advHamMenu. menuItemSkin = “voltmxsmsknlblSanFranciscoFFFFFF129”; |
| **Remarks:** | The default value for the property is “voltmxsmsknlblSanFranciscoFFFFFF129” |

**52. Flex Cover(coverSkin)**

|  |  |
| --- | --- |
| **Description:** | This skin links to flex container. |
| **Syntax:** | coverSkin |
| **Type:** | String |
| **Widget Type:** | Flex Container  |
| **Example:** | this. view. advHamMenu. coverSkin= “voltmxsmflxCoverSkin”; |
| **Remarks:** | The default value for the property is “voltmxsmflxCoverSkin” |

**53. Flex Focus (focSkin)**

|  |  |
| --- | --- |
| **Description:** | This skin links to dynamically created flex container. |
| **Syntax:** | focSkin |
| **Type:** | String |
| **Widget Type:** | Flex Container  |
| **Example:** | this. view. advHamMenu. focSkin= “voltmxsmflxfocusSkin”; |
| **Remarks:** | The default value for the property is “voltmxsmflxfocusSkin” |

**54. Rounded Corner Image (circleRoundImage)**

|  |  |
| --- | --- |
| **Description:** | This skin links to profile type flex container |
| **Syntax:** | circleRoundImage |
| **Type:** | String |
| **Widget Type:** | Flex Container |
| **Example:** | this. view. advHamMenu.circleRoundImage = “voltmxsmsknSlidingMenuRoundedCornerImage”; |
| **Remarks:** | The default value for the property is “voltmxsmsknSlidingMenuRoundedCornerImage” |

**55. Rounded Image (circleImage)**

|  |  |
| --- | --- |
| **Description:** | This skin links to profile type flex container. |
| **Syntax:** | circleImage |
| **Type:** | String |
| **Widget Type:** | Flex Container |
| **Example:** | this. view. advHamMenu. circleImage = “voltmxsmSknSlidingMenuCircleImage”; |
| **Remarks:** | The default value for the property is “voltmxsmSknSlidingMenuCircleImage” |

## **API’s**

###  **i. setData**

|  |  |
| --- | --- |
| **Description:** | Adds menu and sub-menu items to the Sliding Menu. |
| **Syntax:** | setData (menuItems , submenuItems) |
| **Parameter (Optional):** | menuItems [Array of JSON]:This object contains details of all the menu items that you want to display on the Sliding Menu.submenuItems [Array of JSON]:This object contains details of all the submenu items that you want to display on the Sliding Menu. |
| **Return Value:** | None |
| **Example:** |  var menuData = [ {menuItemName:'Sports', menuItemIcon:'acme.png'}, {menuItemName:'News',menuItemIcon:'acme.png'} ]; /\*Array of JSON Objects with submenu items \*/ var submenuData =[ {menuItemName: ‘News', submenuItemName:'subOpt1', submenuItemIcon:"acme.png"}, {menuItemName:'Sports', submenuItemName:'subOpt1', submenuItemIcon:"acme.png"}, ];this. view. advHamMenu. setData(menuData, submenuData); |

###  **ii. showSlidingMenu**

|  |  |
| --- | --- |
|  **Description:** | Displays the Sliding Menu. |
|  **Syntax:** | showSlidingMenu () |
|  **Parameters:** | None |
|  **Return Value:** | None |
|  **Example:** | this. view. advHamMenu. showSlidingMenu (); |

###  **iii. hideSlidingMenu**

|  |  |
| --- | --- |
| **Description:** | Hides the Sliding Menu |
| **Syntax:** | hideSlidingMenu () |
| **Parameter:** | None |
| **Return Value:** | None |
| **Example:** | this. view. advHamMenu. showSlidingMenu (); |

###

###  **iv** **removeAllMenuItem**

|  |  |
| --- | --- |
| **Description:** | Removes all the items or options from the Sliding Menu. |
| **Syntax:** | removeAllMenuItem () |
| **Parameter:** | None |
| **Return Value:** | None |
| **Example:** | this. view. advHamMenu.removeAllMenuItem (); |

### **D. Events**

### 1.**onMenuItemClick**

|  |  |
| --- | --- |
| **Description:** | Invoked when the user clicks or taps any of the menu items or sub-menu items. |
| **Syntax:** | onMenuItemClick |
| **Parameters:** | menuindex [Array]: Contains the index of the menu or submenu item clicked. |
| **Example:** | this. view. advHamMenu. onMenuItemClick= function (index) {alert (“Clicked Item”);}; |

### **2.** **o****nFooterClick**

|  |  |
| --- | --- |
| **Description:** | Invoked when the user clicks or taps the footer. |
| Syntax: | onFooterClick () |
| **Example:** | this. view. advHamMenu. onFooterClick = function () {alert (“Clicked Footer”);}; |

### **3.** **onProfileClick**

|  |  |
| --- | --- |
| **Description:** | The event is invoked when a user clicks option 3. |
| Syntax: | onProfileClick () |
| **Example:** | this. view. advHamMenu. onProfileClick = function () {alert (“Clicked Profile”);}; |

#  **4. Revision History**

 App version 2.1.3

##  **A. Limitations**

 1. Landscape mode is not supported.

 2. If browser height is minimized UI gets distorted.

##  **B. Known Issues**

The UI of the component gets distorted if you call the showSlidingMenu or hideSlidingMenu APIs immediately after the dynamic creation of component.