Date : 12 Mar 2023

LauNch GooGLE MAPS

Version:1.0.3

**1.OVERVIEW**

The component allows you to display an address/ location on Google Maps/ Apple Maps- just drag and drop the component, customize few properties and you are ready to go!

1. **Use case:**

Provide location to your app users using Google Maps/ Apple Maps

**B. Features:**

1. Configure location, map view type etc. properties directly from Iris property panel.
2. Customize the look and feel as per your company's branding guidelines.
3. The component uses URL schemes provided by Google Maps/ Apple Maps

**C. Percentage of Re-use:**

90%

# **2.Getting Started**

## **A. Prerequisites**

Before you start using the TimeLine component, ensure you have the following:

* [HCL Foundry](https://manage.hclvoltmx.com/)
* Volt MX Iris

## **B. Platforms Supported**

### Mobile

#### iOS

#### Android

### PWA

## **C.** **Importing the Component**

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

## **To import the TimeLine component, do the following:**

## Open your app project in Volt MX Iris.

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

Graphical user interface, text, application

Description automatically generated

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

Graphical user interface, text, application, Teams

Description automatically generated

1. 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.

A screenshot of a computer

Description automatically generated

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" \l "add-a-component-to-a-form).

* 1. **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 Iris User Guide.

You can then run your app to see the Launch Google Maps work in real time.

as per your requirement. For more information, see Setting Properties.

# **3.References**

## **A. Dynamic Usage**

You can also add a Launch Google Maps 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 like the sample code mentioned below.

In the code snippet, you can edit the properties of the component.

/\* Creating a component s object\*/

var launchmap = new com.voltmxmp.launchmap(

{

"autogrowMode": voltmx.flex.AUTOGROW\_NONE,

"clipBounds": true,

"height": "80dp",

"id": "launchmap",

"isVisible": true,

"layoutType": voltmx.flex.FREE\_FORM,

"left": "0dp",

"masterType": constants.MASTER\_TYPE\_USERWIDGET,

"skin": "slFbox",

"top": "0dp",

"width": "80dp"

}, {}, {});

/\* Adding Properties to the component \*/

launchmap.mapTypeApple = "Standard";

launchmap.googlePlaceID = "";

launchmap.iOSMapOptions = "Both Maps";

launchmap.location = "Eiffel Tower";

launchmap.iconScaleMode = constants.IMAGE\_SCALE\_MODE\_MAINTAIN\_ASPECT\_RATIO;

/\* Adding Component to the form \*/

this.view.add(launchmap);

In the code snippet, you can edit the properties of the component as per your requirement. For more information, see [Properties](https://docs.kony.com/marketplace/V8Marketplace/Content/Marketplace/launchmap.htm#Properti).

1. Save the file.

## **B. Setting Properties**

The Properties provided on the Component tab allow you to customize the UI elements and set constraints for fields in the **launch map** component. You can set the properties directly on the Component tab or by writing a JavaScript.

### **i. General Properties**

[**[Open](javascript:void(0);)1.Icon Source**](javascript:void(0);)

|  |  |
| --- | --- |
| **Description:** | Specifies the source of the component's icon. |
| **Syntax:** | iconSrc |
| **Type:** | String |
| **Read/Write:** | Read + Write |
| **Remarks:** | Make sure that the source file is present in the project resources. |
| **Example:** | this.view.componentID.iconSrc = voltmxmplmlaunchmapicon.png ; |

[**[Open](javascript:void(0);)2. Icon Scale Mode**](javascript:void(0);)

|  |  |  |  |
| --- | --- | --- | --- |
| **Description:** | | Specifies the scale mode of the component's icon. For further information,  you can refer to the [imageScaleMode](http://docs.kony.com/konyonpremises/Subsystems/Widget_User_Guide/Content/Image_Layout_Properties.htm" \l "imageScaleMode) property in the Quantum Widget  User Guide. | |
| **Syntax:** | | iconScaleMode | |
| **Type:** | | Constant | |
| **Read/Write:** | | Read + Write | |
| **Values:** | | * constants.IMAGE\_SCALE\_MODE\_CROP * constants.IMAGE\_SCALE\_MODE\_FIT\_TO\_DIMENSIONS * constants.IMAGE\_SCALE\_MODE\_MAINTAIN\_ASPECT\_RATIO | |
| **Example:** | this.view.componentID.iconScaleMode = constants.IMAGE\_SCALE\_MODE\_MAINTAIN\_ASPECT\_RATIO; | |

**3.**[**[Open](javascript:void(0);)Location**](javascript:void(0);)

|  |  |
| --- | --- |
| **Description:** | Specifies the Location to be searched in the visible map region. |
| **Syntax:** | location |
| **Type:** | String |
| **Read/Write:** | Read + Write |
| **Remarks:** | * This property is **mandatory**. The component throws an exception if the **Location** is not provided. * This property refers to:   + The **query** parameter in Google Maps Directions URL scheme.   + The **q** parameter in Apple Maps URL scheme.   For more information, you can refer to the [Google Maps](https://developers.google.com/maps/documentation/urls/guide#directions-action) documentation and [Apple Maps](https://developer.apple.com/library/content/featuredarticles/iPhoneURLScheme_Reference/MapLinks/MapLinks.html) reference.   * While providing the location as (lat,long), make sure there is no space before or after the comma. |
| **Example:** | this.view.componentID.location = paris ;  or  this.view.componentID.location = 48.858371,2.294481 ; |

### **ii. iOS**

**[[Open](javascript:void(0);)1. Map Options](javascript:void(0);)**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Description:** | | This property is used to ask the users which map they want to use. This property invokes an **Action Sheet** pop-up that presents the choice of map to the user - Google Maps, or Apple Maps. This property has the following options:   * Apple Maps * Google Maps * Both Maps | | |
| **Syntax:** | | iOSMapOptions |
| **Type:** | | String |
| **Read/Write:** | | Read + Write |
| **Default Value:** | | "Both Maps" |
| **Remarks:** | | * The component throws an exception if the given value is not a valid string. * The **Action Sheet** will not pop up if you enable only one map. |
| **Example:** | | this.view.componentID.iOSMapOptions = "Both Maps"; |

[**[Open](javascript:void(0);)2. Action Sheet Main Text**](javascript:void(0);)

|  |  |
| --- | --- |
| **Description:** | Specifies the text to be displayed as the main header of the Action Sheet. You can refer to the [Map Options](https://docs.kony.com/marketplace/V8Marketplace/Content/Marketplace/launchmap.htm#Map) property for more information regarding Action Sheet. |
| **Syntax:** | headerMainText |
| **Type:** | String |
| **Read/Write:** | Read + Write |
| **Remarks:** | The Action Sheet will not display the Header if you do not provide any text. |
| **Example:** | this.view.componentID.headerMainText = "Launch Map"; |

**3.**[**[Open](javascript:void(0);)Action Sheet Sub Text**](javascript:void(0);)

|  |  |  |  |
| --- | --- | --- | --- |
| **Description:** | | Specifies the text to be displayed as the sub header of the Action Sheet. You can refer to the [Map Options](https://docs.kony.com/marketplace/V8Marketplace/Content/Marketplace/launchmap.htm#Map) property for more information regarding Action Sheet. | |
| **Syntax:** | | headerSubText | |
| **Type:** | | String | |
| **Read/Write:** | | Read + Write | |
| **Remarks:** | | The Action Sheet will not display the Sub Header if you do not provide any text. | |
| **Example:** | | this.view.componentID.headerSubText = "Which map would you like to use?"; | |

### **Action Sheet Skin Properties**

An **Action Sheet** is a pop-up in iOS that presents the choice of map to the user - Google Maps, or Apple Maps.  
You can change the exposed skins of the Action Sheet component in the **Skins** tab under the **Properties** panel of the Iris.

**1.**[**[Open](javascript:void(0);)Sheet Background Skin**](javascript:void(0);)

|  |  |
| --- | --- |
| **Description:** | Specifies the skin that is applied to the background container of **Header** and **Options** of the Action Sheet. |
| **Syntax:** | sheetBackgroundSkin |
| **Type:** | String |
| **Read/Write:** | Read + Write |
| **Example:** | this.view.componentID.sheetBackgroundSkin = "sheetBGSkin"; |

[**[Open](javascript:void(0);)2. Header Main Text Skin**](javascript:void(0);)

|  |  |
| --- | --- |
| **Description:** | Specifies the skin that is applied to the header's main text in the Action Sheet. |
| **Syntax:** | headerMainTextSkin |
| **Type:** | String |
| **Read/Write:** | Read + Write |
| **Example:** | this.view.componentID.headerMainTextSkin = "headerTextSkin"; |

[**[Open](javascript:void(0);)3. Header Sub Text Skin**](javascript:void(0);)

|  |  |
| --- | --- |
| **Description:** | Specifies the skin that is applied to the header's sub text in the Action Sheet. |
| **Syntax:** | headerSubTextSkin |
| **Type:** | String |
| **Read/Write:** | Read + Write |
| **Example:** | this.view.componentID.headerSubTextSkin = "headerTextSkin2"; |

[**[Open](javascript:void(0);)4. Separator Skin**](javascript:void(0);)

|  |  |
| --- | --- |
| **Description:** | Specifies the skin that is applied to the separator lines in the Action Sheet. |
| **Syntax:** | separatorSkin |
| **Type:** | String |
| **Read/Write:** | Read + Write |
| **Example:** | this.view.componentID.separatorSkin = "separatorSkin"; |

[**[Open](javascript:void(0);)5. Options Skin**](javascript:void(0);)

|  |  |
| --- | --- |
| **Description:** | Specifies the skin to be applied to the options in the Action Sheet. |
| **Syntax:** | optionsSkin |
| **Type:** | String |
| **Read/Write:** | Read + Write |
| **Example:** | this.view.componentID.optionsSkin = "optSkin"; |

[**[Open](javascript:void(0);)6. Options Focus Skin**](javascript:void(0);)

|  |  |
| --- | --- |
| **Description:** | Specifies the skin to be applied to the options in the Action Sheet when it is focused. |
| **Syntax:** | optionsFocusSkin |
| **Type:** | String |
| **Read/Write:** | Read + Write |
| **Example:** | this.view.componentID.optionsFocusSkin = "optFocusSkin"; |

**7.**[**[Open](javascript:void(0);)Footer Background Skin**](javascript:void(0);)

|  |  |
| --- | --- |
| **Description:** | Specifies the skin to be applied to the background container of the Footer in the Action Sheet. |
| **Syntax:** | footerBackgroundSkin |
| **Type:** | String |
| **Read/Write:** | Read + Write |
| **Example:** | this.view.componentID.footerBackgroundSkin = "footBGSkin"; |

[**[Open](javascript:void(0);)8. Footer Button Skin**](javascript:void(0);)

|  |  |
| --- | --- |
| **Description:** | Specifies the skin to be applied to the Footer button in the Action Sheet. |
| **Syntax:** | footerButtonSkin |
| **Type:** | String |
| **Read/Write:** | Read + Write |
| **Example:** | this.view.componentID.footerButtonSkin = "footBtnSkin"; |

[**[Open](javascript:void(0);)9. Footer Button Focus Skin**](javascript:void(0);)

|  |  |
| --- | --- |
| **Description:** | Specifies the skin to be applied to the Footer button in the Action Sheet when it is focused. |
| **Syntax:** | footerButtonFocusSkin |
| **Type:** | String |
| **Read/Write:** | Read + Write |
| **Example:** | this.view.componentID.footerButtonFocusSkin = "footBtnFocusSkin"; |

### **iv. Google Maps Properties**

[**[Open](javascript:void(0);)1. Place ID**](javascript:void(0);)

|  |  |
| --- | --- |
| **Description:** | Specifies Google Maps **Place ID**. A **Place ID** is a unique identifier for a place in the Google Places database and Google Maps. |
| **Syntax:** | googlePlaceID |
| **Type:** | String |
| **Read/Write:** | Read + Write |
| **Remarks:** | * This property refers to the **query\_place\_id** parameter of Google Maps Directions URL scheme. For more details you can refer to [Google Maps Documentation](https://developers.google.com/maps/documentation/urls/guide#search-action). * The component throws an exception if the given value is not a **String**. |
| **Example:** | this.view.componentID.googlePlaceID = "frankfurt"; |

### **V. Apple Maps Properties**

[**[Open](javascript:void(0);)1. Map View Type**](javascript:void(0);)

|  |  |
| --- | --- |
| **Description:** | Specifies the map type. The options for Map Type are:   * Standard * Satellite * Hybrid * Transit |
| **Syntax:** | mapTypeApple |
| **Type:** | String |
| **Read/Write:** | Read + Write |
| **Remarks:** | * If you don't provide any value for Map Type, the current map type is used. * This property refers to the **t** parameter of Apple Maps URL scheme. For more details you can refer to [Apple Maps URL Scheme](https://developer.apple.com/library/content/featuredarticles/iPhoneURLScheme_Reference/MapLinks/MapLinks.html) reference. * The component throws an exception if the given value is not a **String**. |
| **Example:** | this.view.componentID.mapTypeApple = "Standard"; |

# **3. Revision History**

App version 1.0.3

* component is malfunctioning in the adaptive web, and the action sheet fails to open when included in the thin client for (iPhone WEB ,Android WEB). So, ThinClient support only Google Maps .