Date : 28-Oct-21

Confirmation Popup (1.0.5)

1. Overview

Confirmation popup is used when you need user's explicit confirmation to proceed with an action eg. while deleting a record.

## Use case

### when deleting a record or while cancelling an order placed etc.

## Features

1. Easily configurable- set display text and other properties directly from Volt MX Iris
2. Fully customizable skins allow you to change look and feel of the component

## Percentage of re-use:

90%

# Getting Started

## Prerequisites

Before you start using the Confirmation Pop-up component, ensure the following:

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

## Platforms Supported

### Mobile

#### iOS

#### Android

### Tablets

### PWA

## Importing the Component

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

##  **To import the Confirmation Popup component, do the following:**

## Open your app project in Volt MX Iris.

1. 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#add-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 VoltMX Iris User Guide.

You can then run your app to see the Confirmation Popup work in real time.

# References

## Dynamic Usage

You can also add a **Confirmation popup** 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.

var confirmationpopup = new com.voltmx.confirmationpopup(

 {

 "clipBounds": true,

 "height": "100%",

 "id": "confirmationpopup",

 "isVisible": true,

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

 "left": "0dp",

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

 "skin": "slFbox",

 "top": "0dp",

 "width": "100%",

 "centerX" :"50%",

 "centerY" :"50%"

 }, {}, {});

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

 confirmationpopup.leftButtonText = "No";

 confirmationpopup.rightButtonText = "Yes";

 confirmationpopup.titleText="Confirmation";

 confirmationpopup.textMessage="Do you want to proceed with this <b> operation </b>";

 confirmationpopup.textContent="Lorem ipsum dolor sit et amet.";

 /\*Adding the confirmationpopup component to a Form\*/

 this.view.add(confirmationpopup);

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

1. **Save** the file.

### .

## Properties

The properties provided on the **Component** tab allow you to customize the UI elements in the **Confirmation Popup** component. You can set the properties directly on the **Component** tab or by writing a JavaScript.

 General Properties

**1. LeftBtnText(leftBtnText)**

|  |  |
| --- | --- |
| **Description:** | Specifies the text for left button. |
| Syntax: | leftBtnText |
| **Type:** | String |
| **Read/Write:** | Read + Write |
| **Example:** | this.view.confirmationpopup.leftBtnText= "No"; |
| **Remarks:** | The default value for the property is No. |

**2. RightBtnText(rightBtnText)**

|  |  |
| --- | --- |
| **Description:** | Specifies the text for right button. |
| Syntax: | rightBtnText |
| **Type:** | String |
| **Read/Write:** | Read + Write |
| **Example:** | this.view.confirmationpopup.rightBtnText = "Yes"; |
| **Remarks:** | The default value for the property is "Yes". |

**3. Text Message(textMessage)**

|  |  |
| --- | --- |
| **Description:** | Specifies the text for Display Message. |
| Syntax: | textMessage |
| **Type:** | String |
| **Read/Write:** | Read + Write |
| **Example:** | this.view.confirmationpopup.textMessage = " Do you want to proceed with this operation ?"; |
| **Remarks:** | The default value for the property is " Do you want to proceed with this operation ?" |
|  |  |

**4. Title Text(titleText)**

|  |  |
| --- | --- |
| **Description:** | Specifies the text for Display Title. |
| Syntax: | titleText |
| **Type:** | String |
| **Read/Write:** | Read + Write |
| **Example:** | this.view.confirmationpopup.titleText= "Confirmation"; |
| **Remarks:** | The default value for the property is "Confirmation " |
|  |  |

**5. Text Content(textContent)**

|  |  |
| --- | --- |
| **Description:** | Specifies the text for Display Text Content. |
| Syntax: | textContent |
| **Type:** | Constant |
| **Read/Write:** | Read + Write |
| **Example:** | this.view.componentID.textContent=”Lorem ipsum dolor sit et amet. ”; |
| **Remarks:** | The default value for the property is "Lorem ipsum dolor sit et amet. ” |

Skin Section

**6.Left Button Skin(leftButtonSkin)**

|  |  |
| --- | --- |
| **Description:** | This property enables the consumer to change the left button skin of the Confirmation Pop Up component. |
| Syntax: | leftButtonSkin |
| **Widget Type:** | Button |
| **Read/Write:** | Read + Write |
| **Example:** | this.view.confirmationpopup.leftButtonSkin= "leftButtonSkin"; |
| **Remarks:** | Skin assigned to the property is accepted only by same widget type. |

**7. Right Button Skin(rightButtonSkin)**

|  |  |
| --- | --- |
| **Description:** | This property enables the consumer to change the right button skin of the Confirmation Pop Up component. |
| Syntax: | rightButtonSkin |
| **Widget Type:** | Button |
| **Read/Write:** | Read + Write |
| **Example:** | this.view.confirmationpopup.rightButtonSkin= "rightButtonSkin "; |
| **Remarks:** | Skin assigned to the property is accepted only by same widget type. |

**8.** **Flex PopOver Shade Skin (flexPopOverShadeSkin)**

|  |  |
| --- | --- |
| **Description:** | This property enables the consumer to change the Flex PopOver Shade Skinof the Confirmation Pop Up component. |
| Syntax: | flexPopOverShadeSkin |
| **Widget Type:** | Flex Container |
| **Read/Write:** | Read + Write |
| **Example:** | this.view.confirmationpopup.flexPopOverShadeSkin= " flxPopOverShadeSkin"; |
| **Remarks:** | Skin assigned to the property is accepted only by same widget type. |

**9. Flex PopOver Skin (flexPopOverSkin)**

|  |  |
| --- | --- |
| **Description:** | This property enables the consumer to change the Flex PopOver Skin of the Confirmation Pop Up component. |
| Syntax: | flexPopOverSkin |
| **Widget Type:** | Flex Container |
| **Read/Write:** | Read + Write |
| **Example:** | this.view.confirmationpopup.flexPopOverSkin = “flxPopOverSkin”; |
| **Remarks:** | Skin assigned to the property is accepted only by same widget type. |
|  |  |

**10. Flex Center Skin (rightButtonSkin)**

|  |  |
| --- | --- |
| **Description:** | This property enables the consumer to change the Flex Center Skin of the Confirmation Pop Up component. |
| Syntax: | flexCenterSkin |
| **Widget Type:** | Flex Container |
| **Read/Write:** | Read + Write |
| **Example:** | this.view.confirmationpopup.flexCenterSkin = " flxCenterSkin"; |
| **Remarks:** | Skin assigned to the property is accepted only by same widget type. |

## **Events**

**1. btnLeftOnClick**

|  |  |
| --- | --- |
| **Description:** | This event is invoked when the user-touch is released from the left button on the Confirmation Pop Up. |
| Syntax: | btnLeftOnClick |
| **Type:** | String |
| **Read/Write:** | Read + Write |
| **Example:** | this.view.confirmationpopup.btnLeftOnClick= function(){ alert(“Left Button Clicked”); } |
|  |  |

**2. btnRightOnClick**

|  |  |
| --- | --- |
| **Description:** | This event is invoked when the user-touch is released from the right button on the Confirmation Pop Up. |
| Syntax: | btnRightOnClick |
| **Type:** | String |
| **Read/Write:** | Read + Write |
| **Example:** | this.view.confirmationpopup.btnRightOnClick= function(){ alert(“Right Button Click”); } |
|  |  |

 3. btnCloseOnClick

|  |  |
| --- | --- |
| **Description:** | This event is invoked when the user-touch is released from the close button on the Confirmation Pop Up. |
| Syntax: |  btnCloseOnClick |
| **Type:** | String |
| **Read/Write:** | Read + Write |
| **Example:** | this.view.confirmationpopup.btnCloseOnClick= function(){ alert(“Close Button Click”); } |
|  |  |

## APIs

##  --No APIs are exposed.

# Revision History

App version 1.0.5

## Limitations

1.landscape mode is not supported in all channels.