Date: 27-Oct-2021

Refresh Animation (Falling Objects & Shopping Cart)

Version: 1.1.2

# **Overview**

The refresh animation (Falling Objects & Shopping Cart) component can be used as a pull-to-refresh (or swipe-to-refresh) pattern that shows up when a user pulls down on a list of data using touch, to retrieve more data. Swipe to refresh works good for lists, grid lists, and card collections that are sorted by recent content (a collection is strictly sorted by descending date).

## **Use case:**

i. Shopping Cart

Considering a case that you want to develop an e-commerce app. In this app, you can add the shopping cart animation to fill the available space when a list (or page) is pulled down to refresh the product list. You can hide the shopping cart animation when the list refresh is complete.

ii. Falling Objects

Consider a case that you want to develop an online hotel booking app that displays a list of hotels based on the selected location and specified price range. In this app, you can add the falling objects animation to fill the available space when the list is pulled down to refresh the accommodation fares. You can hide the falling objects animation when the list refresh is complete.

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

85%

## **Features**

## Provides controls to customize the UI and to set events

# **Getting Started**

## **Prerequisites**

Before you start using the Refresh Animation (Falling Objects & Shopping Cart) component, ensure the following:

• Volt MX Iris

## **Platforms Supported**

### Mobile

#### iOS

#### Android

### Tablets

## **Importing the app**

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

## **To import the Refresh Animation (Falling Objects & Shopping Cart) component, do the following:**

## Open your app project in Volt MX Iris.

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

Graphical user interface, text, application

Description automatically generated

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

Graphical user interface, text, application, Teams

Description automatically generated

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" \l "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 Volt MX User Guide.

You can then run your app to see the Refresh Animation (Falling Objects & Shopping Cart) work in real time.

# **References**

## **Dynamic Usage**

You can also add **Refresh Animation (Falling Objects & Shopping Cart)** 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 Refresh Animation component instance \*/

var refreshAnimations= new com.voltmx.refreshanimations (

{

id: "refreshanimations",

isVisible: true,

height: voltmx.flex.USE\_PREFERRED\_SIZE,

top: "0dp",

left: "0dp",

width: "100%",

layoutType: voltmx.flex.FLOW\_VERTICAL,

skin: "slFbox ",

clipBounds:true,

zIndex:1

}, {}, {});

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

refreshAnimations.themeSelection = “fallingobjects”;

refreshAnimations.cartVisibile = false;

refreshAnimations.fallingObjVisible = false;

/\*Adding the Refresh Animation component to a Form\*/

this.view.add(refreshAnimations);

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 **Refresh Animation (Falling Objects & Shopping Cart)** 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.** **Theme Selection (themeSelection)**

|  |  |
| --- | --- |
| **Description:** | Specifies the list of animations that you want to set when pulling segment. |
| **Syntax:** | themeSelection |
| **Type:** | List Selector , String |
| **Read/Write:** | Read + Write |
| **Example:** | this. view. refreshanimations. themeSelection = “fallingobjects”; |
| **Remarks:** | The default value for the property is “fallingobjects”. |

**2. Cart Visibility (c****artVisible)**

|  |  |
| --- | --- |
| **Description:** | Specifies the visibility of shopping cart animation. |
| **Syntax:** | cartVisible |
| **Type:** | Boolean |
| **Read/Write:** | Read + Write |
| **Example:** | this. view. refreshanimations. cartVisible = false; |
| **Remarks:** | The default value for the property is false. |

**3. Falling Object Visibility (fallingObjVisible)**

|  |  |
| --- | --- |
| **Description:** | Specifies the visibility of falling objects animation. |
| **Syntax:** | fallingObjVisible |
| **Type:** | Boolean |
| **Read/Write:** | Read + Write |
| **Example:** | this. view. refreshanimations. fallingObjVisible = false; |
| **Remarks:** | The default value for the property is false. |

**Specific Animation Properties**

1. **General**

**4.** **Animation Speed (durationOfAnimation)**

|  |  |
| --- | --- |
| **Description:** | Specifies the duration of the cart animation. The duration value is specified in seconds. |
| **Syntax:** | durationOfAnimation |
| **Type:** | String |
| **Read/Write:** | Read + Write |
| **Example:** | this. view. refreshcart.durationOfAnimation = “1”; |
| **Remarks:** | * The property throws an exception if you provide an invalid value or a non-string value. User needs to handle this exception. * The default value for the property is “1”. |

**5. Animation Speed (durationOfAnimation)**

|  |  |
| --- | --- |
| **Description:** | Specifies the duration of the falling objects animation. The duration value is specified in seconds. |
| **Syntax:** | durationOfAnimation |
| **Type:** | String |
| **Read/Write:** | Read + Write |
| **Example:** | this. view. refreshfalling.durationOfAnimation = “1”; |
| **Remarks:** | * The property throws an exception if you provide an invalid value or a non-string value. User needs to handle this exception. * The default value for the property is “1”. |

### **Skins Section**

**6. Background Skin (backgroundSkin)**

|  |  |
| --- | --- |
| **Description:** | Specifies the skin to be set as the background of the background flex container. |
| **Syntax:** | backgroundSkin |
| **Type:** | String |
| **Read/Write:** | Read + Write |
| **Example:** | this. view. refreshcart. backgroundSkin= “voltmxrcsknflxBackground20223c”; |
| **Remarks:** | The default value for the property is “voltmxrcsknflxBackground20223c”. |

**7. Background Skin (backgroundSkin)**

|  |  |
| --- | --- |
| **Description:** | Specifies the skin to be set as the background of the background flex container. |
| **Syntax:** | backgroundSkin |
| **Type:** | String |
| **Read/Write:** | Read + Write |
| **Example:** | this. view. refreshfalling. backgroundSkin= “voltmxrfsknFlxBackground181934”; |
| **Remarks:** | The default value for the property is “voltmxrfsknFlxBackground181934”. |

**3. Image Section**

**8. Large Lines Source(largeLinesSource)**

|  |  |
| --- | --- |
| **Description:** | Specifies the file name of the image to be displayed as large lines during the refresh cart animation. |
| Syntax: | largeLinesSource |
| **Type:** | String |
| **Read/Write:** | Read + Write |
| **Example:** | this. view. refreshcart. largeLinesSource = “voltmx\_mp\_rc\_dashlong.png”; |
| **Remarks:** | * Ensure that the image file exists in the *workspace\resources\common* directory. * Do not add any uppercase characters in the file name of the image. * Ensure that the image is in PNG format. * Specify the file name of the image along with the extension. * The default value for the property is “voltmx\_mp\_rc\_dashlong.png”. |

**9. Medium Lines Source(mediumLinesSource)**

|  |  |
| --- | --- |
| **Description:** | Specifies the file name of the image to be displayed as medium lines during the refresh cart animation. |
| **Syntax:** | mediumLinesSource |
| **Type:** | String |
| **Read/Write:** | Read + Write |
| **Example:** | this. view. refreshcart.mediumLinesSource= “voltmx\_mp\_rc\_dashmid.png”; |
| **Remarks:** | * Ensure that the image file exists in the *workspace\resources\common* directory. * Do not add any uppercase characters in the file name of the image. * Ensure that the image is in PNG format. * Specify the file name of the image along with the extension. * The default value for the property is “voltmx\_mp\_rc\_dashmid.png”. |

**10. Small Lines Source(smallLinesSource)**

|  |  |
| --- | --- |
| **Description:** | Specifies the file name of the image to be displayed as small lines during the refresh cart animation. |
| **Syntax:** | smallLinesSource |
| **Type:** | String |
| **Read/Write:** | Read + Write |
| **Example:** | this. view. refreshcart. smallLinesSource = “voltmx\_mp\_rc\_dashsmall.png”; |
| **Remarks:** | * Ensure that the image file exists in the *workspace\resources\common* directory. * Do not add any uppercase characters in the file name of the image. * Ensure that the image is in PNG format. * Specify the file name of the image along with the extension. * The default value for the property is “voltmx\_mp\_rc\_dashsmall.png”. |

**11. Cart Source(cartSource)**

|  |  |
| --- | --- |
| **Description:** | Specifies the file name of the image to be displayed as the cart during the refresh cart animation. |
| **Syntax:** | cartSource |
| **Type:** | String |
| **Read/Write:** | Read + Write |
| **Example:** | this. view. refreshcart.cartSource= “voltmx\_mp\_rc\_shape.png”; |
| **Remarks:** | * Ensure that the image file exists in the *workspace\resources\common* directory. * Do not add any uppercase characters in the file name of the image. * Ensure that the image is in PNG format. * Specify the file name of the image along with the extension. * The default value for the property is “voltmx\_mp\_rc\_shape.png”. |

**12. Wheel Source(wheelSource)**

|  |  |
| --- | --- |
| **Description:** | Specifies the file name of the image to be displayed as the wheel during the refresh cart animation. |
| **Syntax:** | wheelSource |
| **Type:** | String |
| **Read/Write:** | Read + Write |
| **Example:** | this. view. refreshcart.wheelSource = “voltmx\_mp\_rc\_path.png”; |
| **Remarks:** | * Ensure that the image file exists in the *workspace\resources\common* directory. * Do not add any uppercase characters in the file name of the image. * Ensure that the image is in PNG format. * Specify the file name of the image along with the extension. * The default value for the property is “voltmx\_mp\_rc\_path.png”. |

**13. Large Circles Source(largeCirclesSource)**

|  |  |
| --- | --- |
| **Description:** | Specifies the file name of the image to be displayed as large circles during the refresh falling animation. |
| **Syntax:** | largeCirclesSource |
| **Type:** | String |
| **Read/Write:** | Read + Write |
| **Example:** | this. view. refreshfalling. largeCirclesSource = “voltmx\_rf\_circlebig.png”; |
| **Remarks:** | * Ensure that the image file exists in the *workspace\resources\common* directory. * Do not add any uppercase characters in the file name of the image. * Ensure that the image is in PNG format. * Specify the file name of the image along with the extension. * The default value for the property is “voltmx\_rf\_circlebig.png” |

**14. Medium Circles Source(mediumCirclesSource)**

|  |  |
| --- | --- |
| **Description:** | Specifies the file name of the image to be displayed as medium circles during the refresh falling animation. |
| **Syntax:** | mediumCirclesSource |
| **Type:** | String |
| **Read/Write:** | Read + Write |
| **Example:** | this. view. refreshfalling. mediumCirclesSource = “voltmx\_rf\_circlemed.png”; |
| **Remarks:** | * Ensure that the image file exists in the *workspace\resources\common* directory. * Do not add any uppercase characters in the file name of the image. * Ensure that the image is in PNG format. * Specify the file name of the image along with the extension. * The default value for the property is “voltmx\_rf\_circlebmed.png” |

**15. Small Circles Source(smallCirclesSource)**

|  |  |
| --- | --- |
| **Description:** | Specifies the file name of the image to be displayed as small circles during the refresh falling animation. |
| **Syntax:** | smallCirclesSource |
| **Type:** | String |
| **Read/Write:** | Read + Write |
| **Example:** | this. view. refreshfalling. smallCirclesSource = “voltmx\_rf\_circlesmall.png”; |
| **Remarks:** | * Ensure that the image file exists in the *workspace\resources\common* directory. * Do not add any uppercase characters in the file name of the image. * Ensure that the image is in PNG format. * Specify the file name of the image along with the extension. * The default value for the property is “voltmx\_rf\_circlesmall.png” |

## **API’s**

### **i. displaySelAnim**

|  |  |
| --- | --- |
| **Description:** | Displays the selected animation. |
| **Syntax:** | displaySelAnim () |
| **Parameter:** | None |
| **ReturnValue:** | None |
| **Example:** | this. view. refreshanimations.displaySelAnim (); |

### **ii. cartShow**

|  |  |
| --- | --- |
| **Description:** | Displays the shopping cart animation. |
| **Syntax:** | cartShow () |
| **Parameter:** | None |
| **ReturnValue:** | None |
| **Example:** | this. view. refreshanimations.cartShow (); |

### **iii. cartHide**

|  |  |
| --- | --- |
| **Description:** | Hides the shopping cart animation. |
| **Syntax:** | cartHide () |
| **Parameter:** | None |
| **ReturnValue:** | None |
| **Example:** | this. view. refreshanimations.cartHide (); |

### 

### **iv. fallingObjShow**

|  |  |
| --- | --- |
| **Description:** | Displays the falling objects animation. |
| **Syntax:** | fallingObjShow () |
| **Parameter:** | None |
| **ReturnValue:** | None |
| **Example:** | this. view. refreshanimations. fallingObjShow (); |

### 

### **v. fallingObjHide**

|  |  |
| --- | --- |
| **Description:** | Hides the falling objects animation. |
| **Syntax:** | fallingObjHide () |
| **Parameter:** | None |
| **ReturnValue:** | None |
| **Example:** | this. view. refreshanimations. fallingObjHide (); |

### **Specific Animation API’s**

### **vi. show**

|  |  |
| --- | --- |
| **Description:** | Displays the refresh cart animation. |
| **Syntax:** | show () |
| **Parameter:** | None |
| **ReturnValue:** | None |
| **Example:** | this. view. refreshcart. show (); |

### 

### **vii. hide**

|  |  |
| --- | --- |
| **Description:** | Hides the refresh cart animation. |
| **Syntax:** | hide () |
| **Parameter:** | None |
| **ReturnValue:** | None |
| **Example:** | this. view. refreshcart. hide (); |

### **viii. show**

|  |  |
| --- | --- |
| **Description:** | Displays the refresh falling animation. |
| **Syntax:** | show () |
| **Parameter:** | None |
| **ReturnValue:** | None |
| **Example:** | this. view. refreshfalling. show (); |

### **ix. hide**

|  |  |
| --- | --- |
| **Description:** | Hides the refresh falling animation. |
| **Syntax:** | show () |
| **Parameter:** | None |
| **ReturnValue:** | None |
| **Example:** | this. view. refreshfalling. hide (); |

## **Sample Usage of the Component**

### **Component with the Segment Widget**

Consider a use case where user pulls down a list to update the list data. You can use the Refresh Animation component to fill the space when the list (segment widget) is pulled down to refresh the data. You can achieve the use case by following the below procedure.

**To use the component in the Segment Widget, do the following:**

1. Import the Refresh Animation component.
2. Add a Flex Container Widget to the form.
3. Set the **Layout Type** of the FlexContainer Widget to **Flow Vertical**.
4. Drag the Refresh Animation component and the Segment Widget onto the FlexContainer Widget.
5. Set the left and top margin values of the Refresh Animation component to Zero percent.
6. Set the left and top margin values of the Segment Widget to Zero percent.

7. Define the onPull event of the Segment Widget as shown in the following code snippet:

8. Copy and paste the following code snippet in the Action editor:

this.view.componentInstance.show();

The above code displays the Refresh animation component on triggering the onPull event of the Segment Widget.

1. Save the file.

# **4. Revision History**

App version 1.1.2

## **A. Limitations**

1. Landscape mode is not supported.