Date: 21-Oct-2021

horizontal parallax (2.0.3)

1. Overview

Parallax scrolling uses multiple objects at different speeds to give the impression of a 3D effect where nearby objects have a larger parallax than more distant objects when observed from different positions. If you're looking for a popular scrolling effect used by native mobile apps and amazing websites, download the horizontal parallax.

## Use case

Consider a case that you are developing an app that provides information about heritage sites. You can use the Horizontal Parallax Animation component in the app and configure the component in such a manner that the app displays the images of a heritage site in the background and related information in the foreground.

## Features:

* Added target container support
* Parallax scrolling effect in a horizontal manner
* Ability to animate background and foreground at different speeds
* Facility to customize the UI elements.

## Percentage of re-use:

90%

# Getting Started

## Prerequisites

Before you start using the Horizontal Parallax component, ensure you have the following:

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

## Platforms Supported

### Mobile

#### iOS

#### Android

### Tablets

### PWA

## Importing the Component

Before you start importing the component to Volt MX Iris, you must download the component from the HCL Volt MX website.

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

**To import the Horizontal Parallax component, do the following:**

1. Open your app project in Volt MX Iris.
2. On the **File** menu, point to **Import**, and click **Component**. The **Open** dialog appears.
3. Navigate to the location where you downloaded the component (zip file) on your computer, select the component, and click **Open**. The **Import Item** dialog appears.
4. In the **Library Name** box, type a name if you want to create a new library. Otherwise, you can select the existing library name.
5. In the **Collection Name** box, type a name if you want to create a new collection. Otherwise, you can select the existing collection name.
6. In the **Component Name** box, the **com.voltmx.horizontalparallax** name is displayed by default.

When you import a component, the component is imported to Iris's workspace, but not directly into your app. Thus, after you import a component, you must add the component to your app.

## **Adding an Horizontal Parallax Component to your App**

You can add the Horizontal Parallax component to the FlexForms, FlexContainer, and FlexScrollContainer widgets.

**To add the Horizontal Parallax component, do the following:**

1. Open your app, and then open the form to which you want to add the component.
2. On the **My Libraries** tab, select the defined Forge library from the drop-down list in which the component exists.
3. From the **Collections** sub-tab, drag the component onto the form on the Iris canvas. The component is added to the form. You can also seen new element, **com.voltmx.horizontalparallax** in the **Components** section on the **Templates** tab.

After adding a component to a form, you can configure the component the way you want it using the **Look**, **Skin**, and **Action** tabs on the **Properties** pane. Configuring the properties on the **Properties** pane is similar to configuring the properties of any widget in Volt MX Iris.

You can also see that a new tab, **Component**, is added on the **Properties** pane. The **Component** tab contains assorted properties relevant to the component that allow you to customize the component as required. The properties on the **Component** tab are categorized as **General** and **Skin** properties.

## 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 Horizontal Parallax work in real time.

# References

## Dynamic Usage

If you want to use the Horizontal Parallax component dynamically, you will need to import the component into your project Templates. Follow the given steps to do so

### Download the component from HCL VoltMX Marketplace as a zip file.

### Go to the Templates tab in your project explorer.

### Right click on Components and select Import Component.

### Navigate to where you downloaded your zip file and import it into Iris.

After you import the component into your project templates, you can add it to your app dynamically. To do so, follow the given steps

### Access the FormController of the form you want to add the component into.

### Create a function called createComponent(); and write the code inside it to create and configure the component.

### You can refer to the given sample code for more information.

/\* creating a component's Object \*/

createComponent : function(){

var horizontalparallax = new com.voltmx.horizontalparallax({

 "clipBounds": true,

 "height": "100%",

 "id": "horizontalparallax",

 "isVisible": true,

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

 "left": "0dp",

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

 "skin": "slFbox",

 "top": "0dp",

 "width": "100%"

 }, {}, {});

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

 horizontalparallax.parallaxSpeed = 5;

 horizontalparallax.isPagingEnabled = true;

 horizontalparallax.isPagingIndicatorEnabled = true;

 horizontalparallax.activeImage = "voltmxmp\_hp\_dot\_selected\_1\_1.png" ;

 horizontalparallax.inactiveImage = "voltmxmp\_hp\_dot\_disable\_1\_1.png" ;

 horizontalparallax.isBgOneVisible = true;

 horizontalparallax.isFgOneVisible = true;

 horizontalparallax.isBgTwoVisible = true;

 horizontalparallax.isFgTwoVisible = true;

 horizontalparallax.isBgThreeVisible = true;

 horizontalparallax.isFgThreeVisible = true;

/\*Adding the Horizontal Parallax component to a Form\*/

 this.view.add(horizontalparallax);

 }

 });;

You can use a component's **Properties** to customize and configure the elements. These elements can be UI elements, service parameters, and so on. You can set the properties from the Iris's Properties panel on the right-hand side. You can also configure these properties using a JavaScript code.

## Properties

The properties provided on the **Component** tab allow you to customize the UI elements in the Horizontal Parallax component. You can set the properties directly on the **Component** tab or by writing a JavaScript. This section provides information about how to set the properties by writing a JavaScript.

General

#### 1.Parallax Speed

|  |  |
| --- | --- |
| **Description:** | Specifies the parallax speed from 1 to 9. |
| **Syntax:** | parallaxSpeed |
| **Type:** | Integer |
| **Read/Write:** | Read + Write |
| **Remarks:** | This property takes values from 1-9. If value for this property is set to more than 9 or less than 1, it will pick default value, which is 5. |

#### 2.Paging

|  |  |
| --- | --- |
| **Description:** | Specifies whether the paging is enabled for the scroll container. If this property is set to true, the scroll view stops on multiples of the scroll view s bounds when the user scrolls. |
| **Syntax:** | isPagingEnabled |
| **Type:** | Boolean |
| **Read/Write:** | Read + Write |
| **Remarks:** | If the property is set to true, the scroll view stops on multiples of the scroll view's bounds when the user scrolls. |

3.Paging Indicator

|  |  |
| --- | --- |
| **Description:** | Specifies visibility for paging indicator. |
| **Syntax:** | isPagingIndicatorEnabled |
| **Type:** | Boolean |
| **Read/Write:** | Read + Write |
| **Remarks:** | If the property is set to true, the scroll view stops on multiples of the scroll view's bounds when the user scrolls. |
| **Example:** | this.view.componentID.isPagingIndicatorEnabled  |

4.Active Page Icon

|  |  |
| --- | --- |
| **Description:** | Specifies image for enable / active paging indicator. |
| **Syntax:** | activeImage |
| **Type:** | String |
| **Read/Write:** | Read + Write |
| **Remarks:** | he default value is "voltmxmp\_hp\_dot\_selected.png". |
| **Example:** | this.view.componentID.activeImage = "voltmx\_hp\_dot\_selected.png"; |

5.Inactive Page Icon

|  |  |
| --- | --- |
| **Description:** | Specifies image for disable / Inactive paging indicator. |
| **Syntax:** | inactiveImage |
| **Type:** | String |
| **Read/Write:** | Read + Write |
| **Remarks:** | * The default value is "voltmxmp\_hp\_dot\_disable.png".
 |
| **Example:** | this.view.componentID.inactiveImage = "voltmx\_hp\_dot\_disable.png"; |

 Placeholder Properties

6..Background One

|  |  |
| --- | --- |
| **Description:** | Specifies the visibility of Background Image Screen One. |
| **Syntax:** | isBgOneVisible |
| **Type:** | Boolean |
| **Read/Write:** | Read + Write |
| **Remarks:** | The default value of the property is true. |
| **Example:** | this.view.componentID.isBgOneVisible = true |

7.Foreground One

|  |  |
| --- | --- |
| **Description:** | Specifies the visibility of Foreground Flex Screen One. |
| **Syntax:** | isFgOneVisible |
| **Type:** | Boolean |
| **Read/Write:** | Read + Write |
| **Remarks:** | The default value of the property is true. |
| **Example:** | this.view.componentID.isFgOneVisible = true; |

8.Background Two

|  |  |
| --- | --- |
| **Description:** | Specifies the visibility of Background Image Screen Two. |
| **Syntax:** | isBgTwoVisible |
| **Type:** | Boolean |
| **Read/Write:** | Read + Write |
| **Remarks:** | The default value of the property is true. |
| **Example:** | this.view.componentID.isBgTwoVisible = true |

9.Foreground Two

|  |  |
| --- | --- |
| **Description:** | Specifies the visibility of Foreground Flex Screen Two. |
| **Syntax:** | isFgTwoVisible |
| **Type:** | Boolean |
| **Read/Write:** | Read + Write |
| **Remarks:** | The default value of the property is true. |
| **Example:** | this.view.componentID.isFgTwoVisible = true; |

10.Background Three

|  |  |
| --- | --- |
| **Description:** | Specifies the visibility of Background Image Screen Three. |
| **Syntax:** | isBgThreeVisible |
| **Type:** | Boolean |
| **Read/Write:** | Read + Write |
| **Remarks:** | The default value of the property is true. |
| **Example:** | this.view.componentID.isBgThreeVisible = true |

11.Foreground Three

|  |  |
| --- | --- |
| **Description:** | Specifies the visibility of Foreground Flex Screen Three. |
| **Syntax:** | isFgThreeVisible |
| **Type:** | Boolean |
| **Read/Write:** | Read + Write |
| **Remarks:** | The default value of the property is true. |
| **Example:** | this.view.componentID.isFgThreeVisible = true; |

## Events

None of the events are exposed

## API’s

### None of the APIs are used

# Revision History

App version 2.0.3:

## Known Issues

* If browser height is minimized then UI got distorted.

## Limitations

* Landscape mode is not supported.