2nd May 2023

Crop Image And annotate(1.0.o)

# Overview

The CropImageAnnotate component can select and crop the image and then allows to annotate the image with the signature.

## Use case

1. To confirm the Goods delivery from the customer takes the signature in the mobile with the help of this component.

B .Percentage of re-use:

Approximate 90% 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.

C. Features

1. The component utilizes SwiftyJot third party library to take the signature.
2. Uses NFI 3.0

2. Getting Started

## A. Prerequisites

Before you start using the CropImageAnnotate component, ensure the following:

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

• Volt MX Iris

## Platforms Supported

### Mobile

#### iOS

### Tablets

## Importing the app

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

## **To import the CropImageAnnotate 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 CropImageAnnotate work in real time.

# References

## Dynamic Usage

You can also add**CropImageAnnotate** 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.



Graphical user interface, text

Description automatically generated

/\* Creating **CropImageAnnotate** component instance \*/

var **CropImageAnnotate** = new com.technohub.CropImageAnnotate({

"height": "50%",

"id": “CropImageAnnotate",

"isVisible": true,

"left": "0dp",

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

"isModalContainer": false,

"skin": "slFbox",

"top": "0dp",

"width": "100%",

"zIndex": 1,

"overrides": {

"CropImageAnnotate": {

"right": "viz.val\_cleared",

"bottom": "viz.val\_cleared",

"minWidth": "viz.val\_cleared",

"minHeight": "viz.val\_cleared",

"maxWidth": "viz.val\_cleared",

"maxHeight": "viz.val\_cleared",

"centerX": "viz.val\_cleared",

"centerY": "viz.val\_cleared"

}

}

}, {

"overrides": {}

}, {

"overrides": {}

});

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

this.view.add(CropImageAnnotate);

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

Configuring Native Settings (iOS)

To configure the native settings for iOS, follow these steps:

1. From the **Project** explorer, go to **Assets** and expand **Media**.
2. Right-click **Common**, and then select **Resource Location**. Volt MX Iris opens the common resources folder in a file explorer.  
   Graphical user interface, text

   Description automatically generated
3. Open the **infoplist\_configuration.json** file with a text or code editor.
4. At the end of the file, type the following code. You can change the description based on your preference.

"NSCameraUsageDescription" : "Your Description"

Text

Description automatically generated with medium confidence

5. Save the file.

Configuring Deployment Target

1. From the left navigation menu, click **Project Settings**.
2. In the **Project Settings** window, go to **Native** → **iPhone/iPad**.
3. Under **Target Versions**, from the **iOS Version** list, select **11.0** or higher.  
   Graphical user interface, application

   Description automatically generated

General Properties

## API

### i. startGallary

|  |  |
| --- | --- |
| **Description:** | Opens the gallery to pick the image. |
| Syntax: | startGallary() |
| **Parameters:** | None |
| **Return Value:** | None |
| **Remarks:** | Call this API to open the gallery and pick the image, and annotate picked image. |
| **Example:** | this.view.CropImageAnnotate.startGallary (); |

### ii. startCamera

|  |  |
| --- | --- |
| **Description:** | Opens the Camera to pick the image from the Camera |
| Syntax: | startCamera () |
| **Parameters:** | None |
| **Return Value:** | None |
| **Remarks:** | Call this API to open the Camera and pick the image, and annotate picked image. |
| **Example:** | this.view.CropImageAnnotate.stopSession(); |

### C. Events

### 1. resultCallback

|  |  |
| --- | --- |
| **Description:** | Returns the modified image as a base64 image within this callback. |
| Syntax: | resultCallback |
| **Parameters**: | Callback as a parameter |
| **Remarks:** | None |
| **Example:** | postShow(){  kony.runOnMainThread(function(){  this.view.CropImageAnnotate.resultCallback(this.callBackHandler);  }.bind(this), []);  },  callBackHandler: function(data) {    var self = this;  kony.runOnMainThread(function() {  self.view.imgResult.base64 = data;  }, []);  } |

App version 1.0.0:

## Limitations

B. Known Issue