Date : 09-Feb-22

Save image on NFS

version: 1.0.2

# **Overview**

This component can save/capture image using the camera and store it as required. The storage takes place on NFS.

## **Use case:**

### A typical use-case for this component is seen in field service and inspection type applications were capturing the work done by clicking an image is an important part of the field visits/inspections.

## **Features**

### This component can capture image using the camera and store it as required.

### This component can store image in gallery as required.

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

70-80%. (Data can be customizable and skins are not customized but can be changed manually)

# **Getting Started**

## **Prerequisites**

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

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

## **Platforms Supported**

#### Mobile

##### iOS

##### Android

#### Tablets

##### iOS

##### Android

## **Importing the component**

## 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).

## **Configuring Native Settings (Android)**

### Run-time Permissions

On the Android platform, the Save Image on NFS component requires the **WRITE\_EXTERNAL\_STORAGE** permissions.

 To enable the permissions, follow these steps:

1. From the left navigation bar, select **Project Settings**.
2. From the **Project Settings** window, go to **Native** → **Android Mobile/Tablet**.
3. Scroll down to **Manifest Permissions, Tags and Gradle Build Entries**.
4. Set the **WRITE\_EXTERNAL\_STORAGE** permission to **true**.
To set a permission to true, select the permission from the left panel and click **Add>**.


## **Configuring Native Settings (iOS)**

Follow the given steps to enable the permissions.

1. From the **Project** explorer, navigate to the **Assets** tab.
2. Right click **Media** and select **Resource Location** to open the project resources folder.

3. In the browser window that opens, navigate to the **common** folder.
4. Open the **infoplist\_configuration.json** file with a text or code editor.
5. Add "**NSPhotoLibraryUsageDescription**" : "Your Description"，"NSCameraUsageDescription" : " Your Description"



1. **Save** the file.

## **Building and previewing the app**

## Refer to below links:

### [**Build & Generate Native Apps**](https://opensource.hcltechsw.com/volt-mx-docs/docs/documentation/Iris/iris_user_guide/Content/Cloud_Build_in_VoltMX_Iris.html#cloud)

### [**Publish Apps to Enterprise App Store**](https://opensource.hcltechsw.com/volt-mx-docs/docs/documentation/Iris/iris_user_guide/Content/EAS.html#accessing)

# **References**

## **Dynamic Usage**

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.

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

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

 var saveimagenfs = new com.voltmxmp.saveimagenfs(

 {

 "clipBounds": true,

 "height": "50dp",

 "width" : "50dp",

 "id": "saveimagenfs",

 "isVisible": true,

 "centerX": "50%",

 "centerY": "59%",

 "zIndex": 1

 }, {}, {});

 /\* Setting component's properties \*/

 saveimagenfs.imageName = "Image1";

 saveimagenfs.imageFormat = "png";

 saveimagenfs.addImageSrc = "add\_image\_icon.png";

 /\* Adding Component to the Form \*/

 this.view.add(saveimagenfs);

1. **Save** the file.

## **Properties**

### **General**

#### **Add Image Source**

|  |  |
| --- | --- |
| **Description:** | Specifies the source for the image that you want to display on the Add Image button. |
| **Syntax:** | addImageSrc |
| **Type:** | String |
| **Read/Write:** | Read + Write |
| **Default Value:** | "addImage.png" |
| **Example:** | this.view.<componentID>.addImageSrc = "addImage.png"; |

### **Image**

#### **Image Name**

|  |  |
| --- | --- |
| **Description:** | Specifies the file name of the image that you want to save. |
| **Syntax:** | imageName |
| **Type:** | String |
| **Read/Write:** | Write |
| **Default Value:** | "image1" |
| **Example:** | this.view.<componentID>.imageName = "Image1"; |

#### **Image Format**

|  |  |
| --- | --- |
| **Description:** | Specifies the file format of the image that you want to save. |
| **Syntax:** | imageFormat |
| **Type:** | * List Selector
* String
 |
|  **Read/Write:** | Write |
| **Default Value:** | png |
| **Remarks:** | On VoltMX iris, you can select this value from a drop down. |
| **Example:** | this.view.<componentID>.imageFormat = "png"; |



## **Events**

The component invokes events when its corresponding action is performed. You can configure any logic you want the component to perform whenever an event occurs. You can configure the events directly on the Actions tab or by writing a JavaScript.

### **getRawbytes**

|  |  |
| --- | --- |
| **Description:** | Invoked when the user selects an image. |
| **Syntax:** | getRawbytes |
| **Parameters:** | *rawbytes* :The rawBytes data of the image that is selected by the user. |
| **Remarks:** | You can use the rawBytes data of the returned image based on your use cases. |
| **Example:** | this.view.<componentID>.getRawbytes = function(rawbytes){ /\* Assigning rawBytes to an image widget \*/ this.view.imageWidget.rawBytes = rawbytes;}.bind(this); |

### **fileUploadSuccessCallback**

|  |  |
| --- | --- |
| **Description:** | Invoked when the component starts a file upload. |
| **Syntax:** | fileUploadSuccessCallback |
| **Parameters:** | *response [Object]* :The response object from the File System. |
| **Example:** | this.view.<componentID>.fileUploadSuccessCallback = function(response){ alert("File upload success "+ JSON.stringify(response));}.bind(this); |

### **fileUploadFailureCallback**

|  |  |
| --- | --- |
| **Description:** | Invoked when the component is unable to upload the image. |
| **Syntax:** | fileUploadFailureCallback |
| **Parameters:** | *response [Object]* :The response object from the File System. |
| **Example:** | this.view.<componentID>.fileUploadFailureCallback = function(response){ alert("File upload failed "+ JSON.stringify(response));}.bind(this); |

### **imageNotSelected**

|  |  |
| --- | --- |
| **Description:** | Invoked when the user does not select any image. |
| **Syntax:** | imageNotSelected |
| **Example:** | this.view.<componentID>.imageNotSelected = function(){ alert("Please select an image");}.bind(this); |

## **APIs**

The following APIs pertain to the Save Image on NFS component:

### **uploadImage**

|  |  |
| --- | --- |
| **Description:** | Uploads the image to the File System. |
| **Syntax:** | uploadImage() |
| **Parameters:** | None |
| **Return Value:** | None |
| **Example:** | this.view.<componentID>.uploadImage(); |

# **Revision History**

App version 1.0.2:

## **Known Issues**

* N/A

## **Limitations**

* N/A