28th August 2023

VisionTextRecognition (1.0.o)

# Overview

The VisionTextRecognition component can scan a form or letter and recognise text in it. We call this as OCR. Under the hood, the component contains NFIs that use the Vision and VisionKit API.

## Use case

1. A CamScanner app where app scans a document/file and recognise it.

2. A Text Scanner app where app scans a form and recognises the text in that form.

B .Percentage of re-use:

Approximate 95% 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 Vision and VisionKit APIs to scan and detect text in a form..
2. It is available only in iOS.
3. Uses NFI 3.0

2. Getting Started

## A. Prerequisites

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

• 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 Apple Vision Text Recognition 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 Barcode/QR code Scanner work in real time.

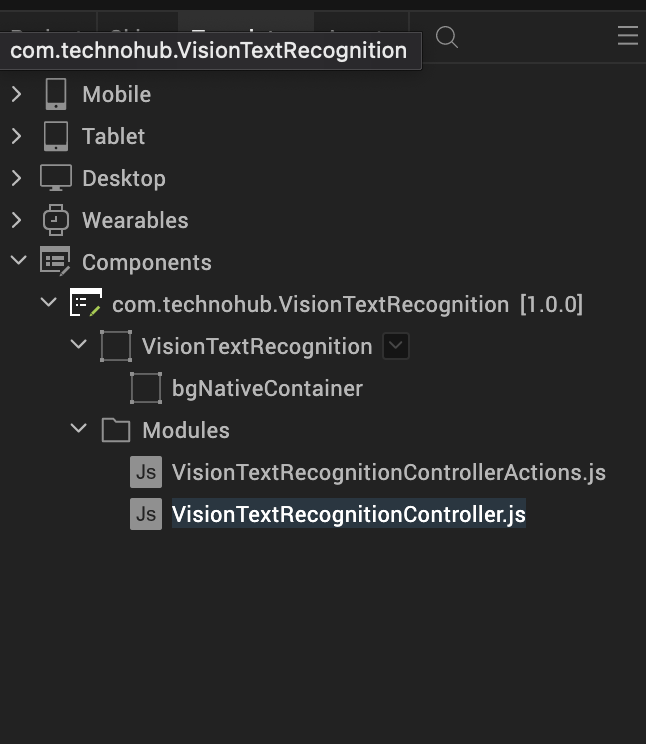
# References

## Dynamic Usage

You can also add**Vision Text Recognition** 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 Vision Text Recognition component instance \*/

var VisionTextRecognition = new com.technohub. VisionTextRecognition ({

"height": "50%",

"id": “VisionTextRecognition",

"isVisible": true,

"left": "0dp",

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

"isModalContainer": false,

"skin": "slFbox",

"top": "0dp",

"width": "100%",

"zIndex": 1,

"overrides": {

" VisionTextRecognition": {

"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 Vision Text Recognition component to a Form\*/

this.view.add(VisionTextRecognition);

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)

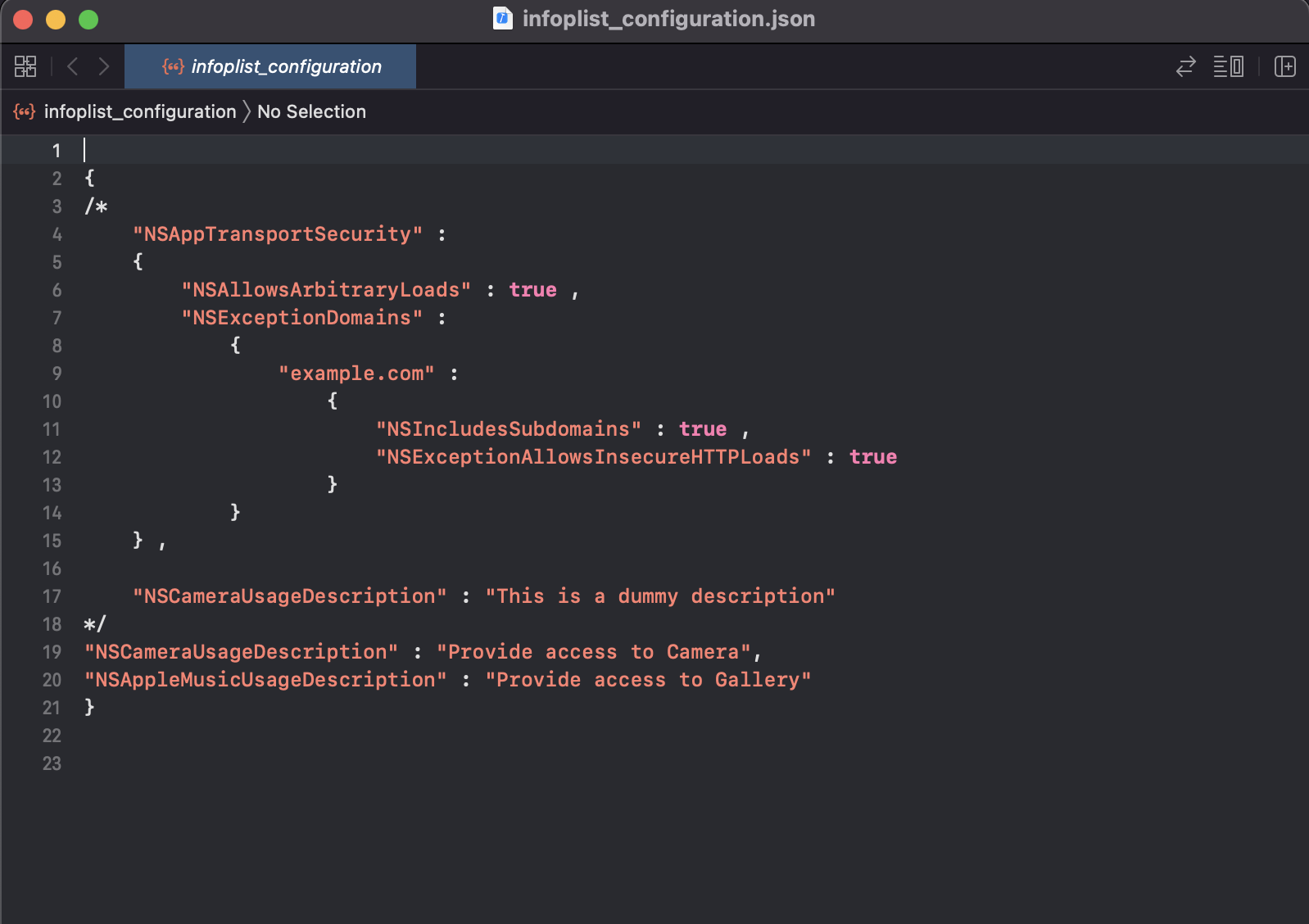
To configure the native settings for iOS, follow these steps:

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"

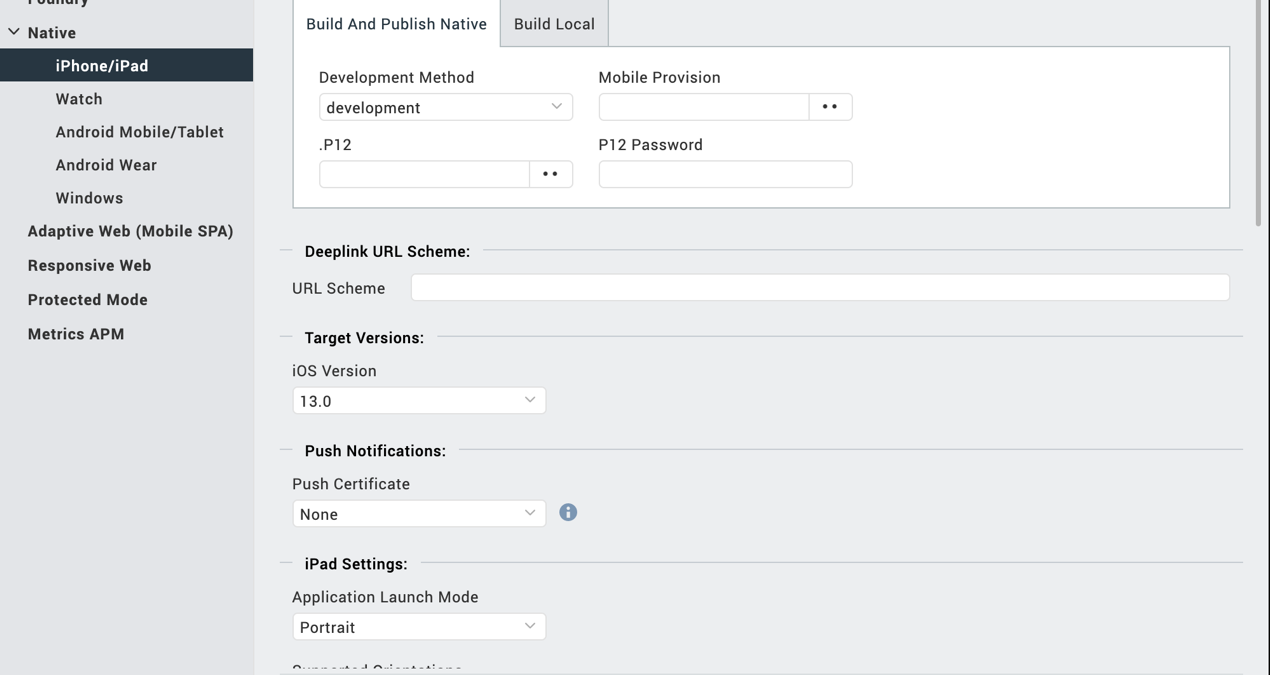
"NSAppleMusicUsageDescription" : "Your Description"



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 **13.0** or higher.



General Properties

## API

### startCamera

|  |  |
| --- | --- |
| **Description:** | Loads camera view to scan a form or any photo which contains text. |
| Syntax: | startCamera () |
| **Parameters:** | None |
| **Return Value:** | None |
| **Remarks:** | Call this API to start the Text Recognition. |
| **Example:** | this.view. VisionTextRecognition.startCamera(); |

### startGallery

|  |  |
| --- | --- |
| **Description:** | Loads Gallery view of device to scan a form or any photo which contains text. |
| Syntax: | startGallery() |
| **Parameters:** | None |
| **Return Value:** | None |
| **Remarks:** | Call this API to start the Text Recognition. |
| **Example:** | this.view. VisionTextRecognition.startGallery(); |

## Events

### resultCallbackForImageAndForText

|  |  |
| --- | --- |
| **Description:** | Invoked after the component scans and detects a text in a form or picture. |
| Syntax: | resultCallbackForImageAndForText |
| **Parameters**: | data[Base64] : The data that is scanned from the Camera and took as photo.  data[String]: The data that is detected in the picture. |
| **Remarks:** | Camera scans form continuously so make sure to save photo before detecting text in that picture. |
| **Example:** | onNavigate: function(context) {  this.view. VisionTextRecognition.resultCallbackForImageAndForText(resCallback, resCallbackText);  },  resCallback: function(data) {    alert(data);  } },  resCallbackText: function(data) {    alert(data);  } }, |

App version 1.0.0:

## Limitations:

#### This component supports portrait mode only.

#### iOS 13, iOS 14 and iOS 15 supports following languages:

###### English US

###### French

###### Italian

###### German

###### Spanish

###### Portuguese

###### Chinese

###### Chinese Traditional

#### iOS 16 and above supports automatic language detection.

B. Known Issue

* Continuously try to scan form/photo unless it is saved.