Date : 31-July-24

VOICE INPUT INTERFACE

VERSION: 2.0.3

**1. OVERVIEW**

The Voice Input Interface takes voice input from a user and converts it to text. Under the hood, the component uses the android. Speech and the [Apple Speech](https://developer.apple.com/documentation/speech?language=objc) frameworks. The component supports both Mobile and Web platforms. On Android, 30 seconds is set for idle limit. The voice input will terminate if no voice is captured in 30 seconds.

You can customize the images and scale modes of the images on the component and use any language that is supported by the package. You can use the component in scenarios where you want to convert any voice input to text, for example, in voice searches, or voice assistants.

## **Use case**

### Convert speech to text

### **B. Features**

### Ready to use UI

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

100% (Data can be customizable and skins are not customized but can be changed manually)

# **2.** **GETTING STARTED**

## **Prerequisites**

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

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

## **Platforms Supported**

### Mobile

### *iOS*

#### *Android*

### Tablets

### PWA

## **C.** **Importing the Component**

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

## **To import the Voice Input Interface 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

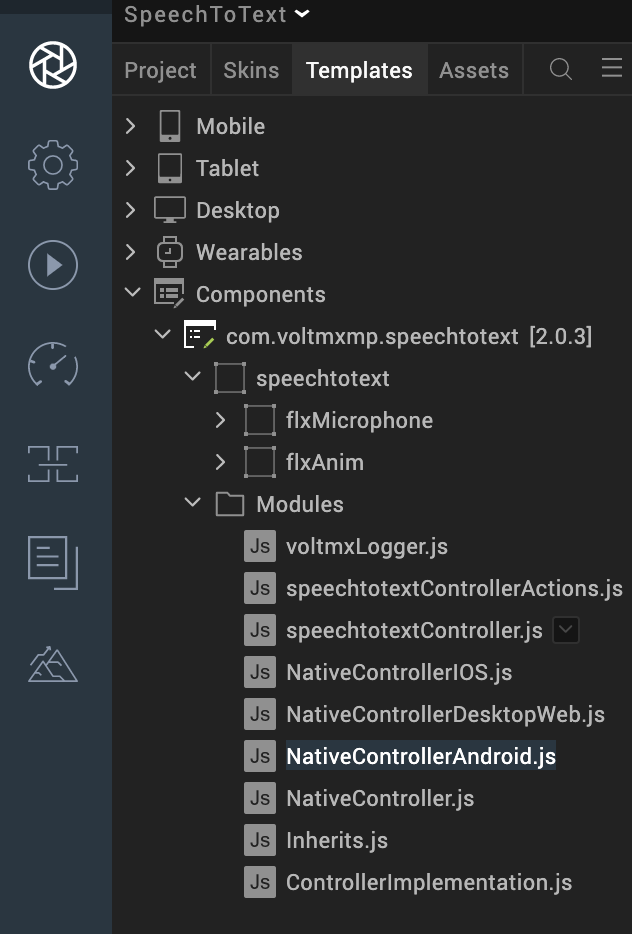
Description automatically generated

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

Graphical user interface, text, application

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#add-a-component-to-a-form).

# **3.References**

## **A. Dynamic Usage**

You can also add a Voice Input Interface 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.

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 \*/

initSpeechToText: function()

{

var speechToText = new com.voltmxmp.speechtotext(

{

"clipBounds": true,

"top": "85%",

"height": "10%",

"width": "12%",

"id": "speechtotext",

"isVisible": true,

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

"bottom": "5%",

"centerX": "50%",

}, {}, {});

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

speechToText.micImage = "voltmxmp\_stt\_microphone.png";

speechToText.cancelImage= "voltmxmp\_stt\_cancel.png";

speechToText.listeningGIF= "voltmxmp\_stt\_imggif.gif";

speechToText.listeningText = "Listening...";

speechToText.micImageScaleMode = "Maintain Aspect Ratio";

speechToText.cancelImageScaleMode = "Maintain Aspect Ratio";

speechToText.GIFImageScaleMode = "Maintain Aspect Ratio";

speechToText.packageName = "com.voltmxmp.speechtotext";

speechToText.setAndroidLanguage = "en-US";

speechToText.setIphoneLanguage = "en-US";

speechToText.setWebLanguage = "en-US";

/\* Adding the component to the form \*/

this.view.add(speechToText);

1. 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.

"NSMicrophoneUsageDescription": "",

"NSSpeechRecognitionUsageDescription": ""

Reference screenshot:



5. Save the file.

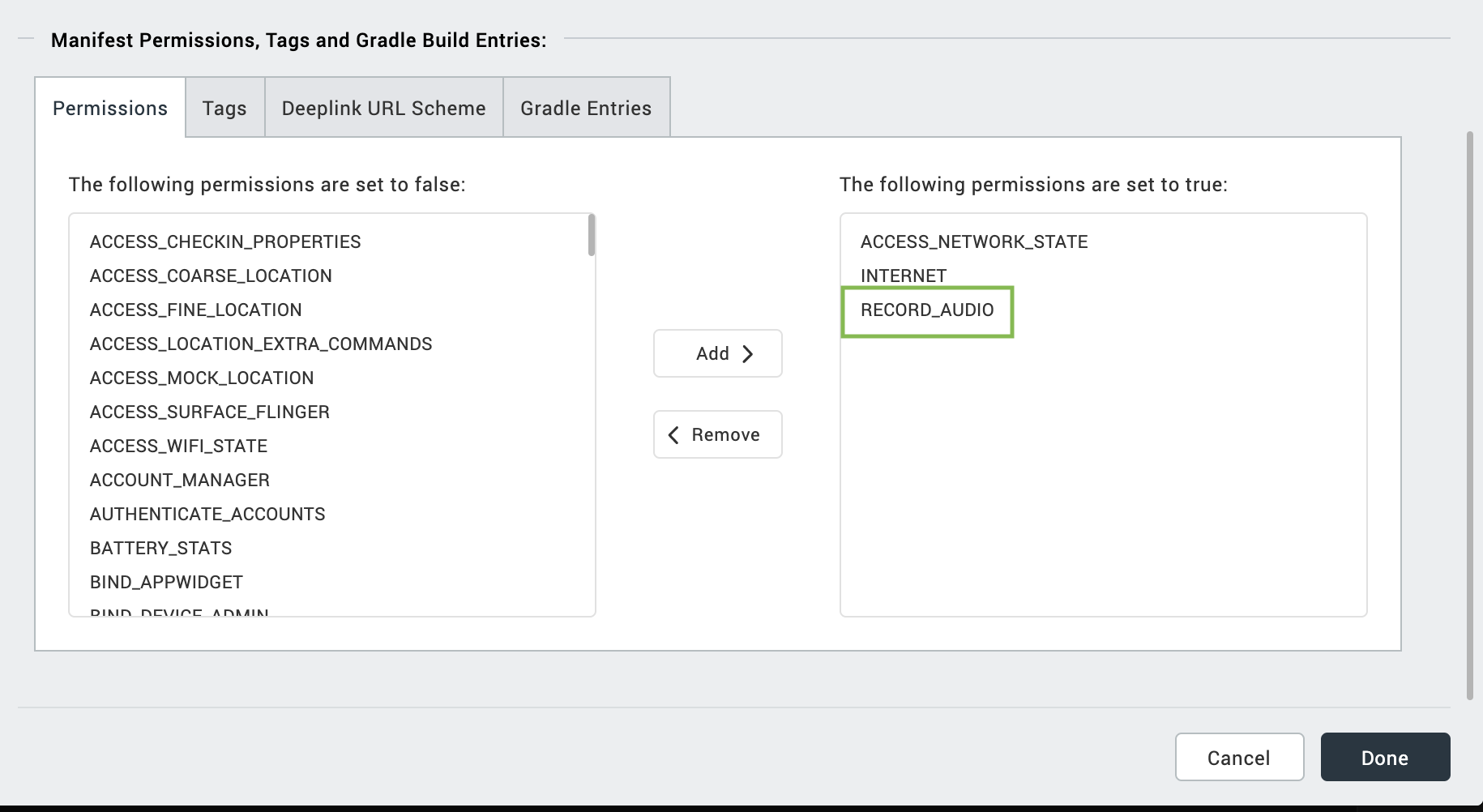
**Configuring Native Settings (Android)**

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

1. From the left navigation menu, click **Project Settings**.

2. In the Project Settings window, go to **Native** → **Android Mobile/Tablet**.

3. Set the **RECORD\_AUDIO** permission to **true**.  
To set a permission to **true**, select the permission from the left panel, and then click **Add**



4. Switch to the Tags tab.

1. Add below tags at Project Settings/Android Mobile/Tablet/Manifest Permissions, Tags and Gradle Build Entries/Tags/Child tag entries under <manifest> tag/

<queries>

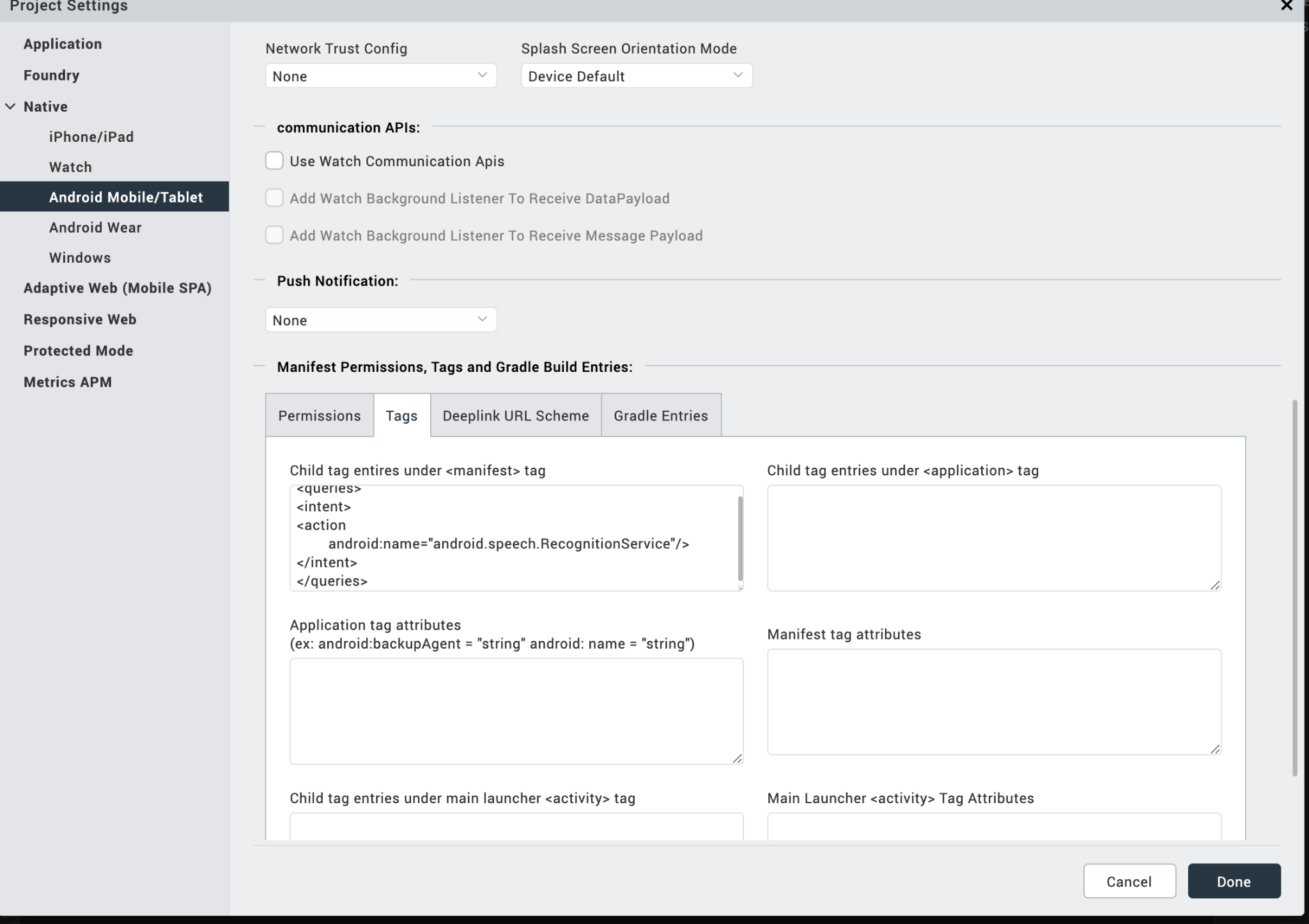
<intent>

<action

android:name="android.speech.RecognitionService" />

</intent>

</queries>



6. Click on Done.

## **B. Properties**

The properties provided on the Component tab allow you to customize the UI elements in the Voice Input Interface component. You can set the properties directly on the Component tab or by writing a JavaScript.

**Open1. Mic Image**

| **Category:** | General |
| --- | --- |
| **Description:** | Specifies the source of the image that you want to set as the Microphone icon. |
| **Syntax:** | micImage |
| **Type:** | String |
| **Read/Write:** | Read + Write |
| **Example:** | this.view.componentID.micImage = "microphone.png"; |

**2.OpenCancel Image**

| **Category:** | General |
| --- | --- |
| **Description:** | Specifies the source of the image that you want to set as the Cancel icon. |
| **Syntax:** | cancelImage |
| **Type:** | String |
| **Read/Write:** | Read + Write |
| **Example:** | this.view.componentID.cancelImage = "cancel.png"; |

**3.OpenListening GIF**

| **Category:** | General |
| --- | --- |
| **Description:** | Specifies the source of the image that you want to set as the Listening icon. |
| **Syntax:** | listeningGIF |
| **Type:** | String |
| **Read/Write:** | Read + Write |
| **Example:** | this.view.componentID.listeningGIF = "listening.gif"; |

**4.OpenListening Text**

| **Category:** | General |
| --- | --- |
| **Description:** | Specifies the text that you want to display when the device is in the Listening state. |
| **Syntax:** | ListeningText |
| **Type:** | String |
| **Read/Write:** | Read + Write |
| **Example:** | this.view.componentID.listeningText = "Listening..." |

**5.OpenMic Image Scale Mode**

| **Category:** | General |
| --- | --- |
| **Description:** | Specifies the scale mode of the image that you want to set as the Microphone icon |
| **Syntax:** | micImageScaleMode |
| **Type:** | String |
| **Read/Write:** | Read + Write |
| **Example:** | this.view.componentID.micImageScaleMode = "Maintain Aspect Ratio"; |

**6.OpenCancel Image Scale Mode**

| **Category:** | General |
| --- | --- |
| **Description:** | Specifies the scale mode of the image that you want to set as the Cancel icon. |
| **Syntax:** | cancelImageScaleMode |
| **Type:** | String |
| **Read/Write:** | Read + Write |
| **Example:** | this.view.componentID.cancelImageScaleMode = "Maintain Aspect Ratio"; |

**7.OpenGit Image Scale Mode**

| **Category:** | General |
| --- | --- |
| **Description:** | Specifies the scale mode of the image that you want to set as the Listening icon. |
| **Syntax:** | GifImageScaleMode |
| **Type:** | String |
| **Read/Write:** | Read + Write |
| **Example:** | this.view.componentID.GIFImageScaleMode = "Maintain Aspect Ratio"; |

**8.OpenPackage Name**

| **Category:** | Android |
| --- | --- |
| **Description:** | Specifies the name of the package that the component passes to the speech framework. |
| **Syntax:** | packageName |
| **Type:** | String |
| **Read/Write:** | Read + Write |
| **Example:** | this.view.componentID.packageName = "com.voltmxmp.speechtotext" |

**9.OpenLanguage**

| **Category:** | Android |
| --- | --- |
| **Description:** | Specifies the language that you want to use for the speech to text feature on the Android platform.  The speech to text engine picks the language that matches the specified locale. |
| **Syntax:** | setAndroidLanguage |
| **Type:** | String |
| **Read/Write:** | Read + Write |
| **Example:** | this.view.componentID.setAndroidLanguage = "en-US" |

**10.OpenLanguage**

| **Category:** | IPhone |
| --- | --- |
| **Description:** | Specifies the language that you want to use for the speech to text feature on the iOS platform.  The speech to text engine picks the language that matches the specified locale. |
| **Syntax:** | setIphoneLanguage |
| **Type:** | String |
| **Read/Write:** | Read + Write |
| **Example:** | this.view.componentID.setIphoneLanguage = "en-US" |

**11.OpenLanguage**

| **Category:** | Desktop Web |
| --- | --- |
| **Description:** | Specifies the language that you want to use for the speech to text feature on the Desktop Web platform.  The speech to text engine picks the language that matches the specified locale. |
| **Syntax:** | setWebLanguage |
| **Type:** | String |
| **Read/Write:** | Read + Write |
| **Example:** | this.view.componentID.setWebLanguage = "en-US" |

## **C. Skins**

Skins define the appearance of a widget. You can apply and manipulate skins based on your preference.

**1.OpenListening Flex**

| **Category:** | General |
| --- | --- |
| **Description:** | This skin links to the background of the component in the Listening state |
| **Syntax:** | ListeningFlex |
| **Widget Type:** | FlexContainer |

## 

**2.OpenListening Label**

| **Category:** | General |
| --- | --- |
| **Description:** | This skin links to the Listening text |
| **Syntax:** | ListeningLabel |
| **Widget Type:** | Label |

## **D. 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.

**1.OpenonMicClick**

| **Description:** | Invoked when the user clicks the Microphone icon. |
| --- | --- |
| **Syntax:** | onMicClick |
| **Example:** | this.view.componentID.onMicClick = function()  {  alert("Listening.");  }.bind(this); |

**2.OpenonCancelClick**

| **Description:** | Invoked when the user clicks the Cancel icon. |
| --- | --- |
| **Syntax:** | onCancelClick |
| **Example:** | this.view.componentID.onCancelClick = function()  {  alert("Listening Canceled.");  }.bind(this); |

**3. speechCallBack**

| **Description:** | Invoked when the speech recognition stops, that is, when the component stops listening. |
| --- | --- |
| **Syntax:** | speeckCallBack |
| **Parameters** | speechText [String]  The text that the component converts from the speech |
| **Example:** | this.view.componentID.speechCallBack = function(speechText)  {  alert("Speech converted to Text: "+speechText);  }.bind(this); |

**4. partialSpeechResults**

| **Description:** | Invoked when the speech recognition framework is in the Listening state. |
| --- | --- |
| **Syntax:** | partialSpeechResults |
| **Parameters** | speechText [String]  The text that the component converts from the speech |
| **Example:** | this.view.componentID.partialSpeechResults = function(speechText)  {  alert("Speech partially converted to Text: "+speechText);  }.bind(this); |

**5.Open onBeginningOfSpeech**

| **Description:** | Invoked when the user starts to speak |
| --- | --- |
| **Syntax:** | onBeginningOfSpeech |
| **Parameters** | speechText [String]  The text that the component converts from the speech |
| **Example:** | this.view.componentID.onBeginningOfSpeech = function()  {  alert("Speaking Started.");  }.bind(this); |

**6. onRmsChanged**

| **Description:** | Invoked when there is a change in the sound level of the audio input. |
| --- | --- |
| **Syntax:** | partialSpeechResults |
| **Parameters** | v [Number]  The sound level of the audio input in DB. |
| **Example:** | this.view.componentID.onRmsChanged = function(v)  {  alert("Sound level changed to: "+v);  }.bind(this); |

**7. onError**

| **Description:** | Invoked when any error occurs within the component |
| --- | --- |
| **Syntax:** | onError |
| **Parameters** | code [Number]  The error code from the speech framework. |
| **Example:** | this.view.componentID.onError = function(code)  {  alert("Error Occured! Error Code: "+code);  }.bind(this); |

**8.Open****onBufferReceived**

| **Description:** | Invoked when the component receives more sound. |
| --- | --- |
| **Syntax:** | onBufferReceived |
| **Parameters** | bytes [Array]  The additional sound that the component receives. |
| **Example:** | this.view.componentID.onBufferReceived = function(bytes)  {  alert("Additional Sound Received: "+bytes);  }.bind(this); |

**9.OpenonEvent**

| **Description:** | Reserved for adding future events. |
| --- | --- |
| **Syntax:** | onEvent |
| **Parameters** | i [Number]  Specifies the type of the event that occurred.  Bundle  Contains the passed parameters. |

**10.Open onReadyForSpeech**

| **Description:** | Invoked when the speech recognition framework is ready. |
| --- | --- |
| **Syntax:** | onReadyForSpeech |
| **Parameters** | bundle  Parameters set by the recognition service. |
| **Example:** | this.view.componentID.onReadyForSpeech = function(bundle)  {  alert("Ready to Listen: "+bundle);  }.bind(this); |

## **E. APIs**

-- None of the methods are exposed

# **4. Revision History**

App version 2.0.3

## **A. Limitations**

* On Desktop Web, the component only supports Google Chrome, Edge. If you use the component on any other web browser (such as Firefox or Safari), then the component throws an exception.
* Voice Input Interface does not include punctuation marks in the results from speech recognition.
* App preview is not available because of NFI dependency

**B. Known Issues**

-- NA