Date :  11-April-2023

SMS - Text Message (Phone) version: 1.0.1

# **Overview**

The SMS - Text Message (Phone) component enables you to quickly integrate native messaging capability into your app. Add 'send SMS' functionality to your app in seconds- just drag and drop the component, configure few properties- that's all! you are ready to go.

## **Features**

* Configure all properties from VoltMX iris properties panel itself.
* Messaging application of the underlying platform is used to send the message.

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

80%

# **Getting Started**

## **Prerequisites**

Before you start using the SMS - Text Message (Phone) component, ensure you have the following:

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

## **Platforms Supported**

### Mobile

#### iOS

#### Android

## **Importing the app**

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

**To import the SMS - Text Message (Phone) component, do the following:**

1. Open your app project in Volt MX Iris.
2. In the Project Explorer, click the Templates tab.

Graphical user interface, text, application

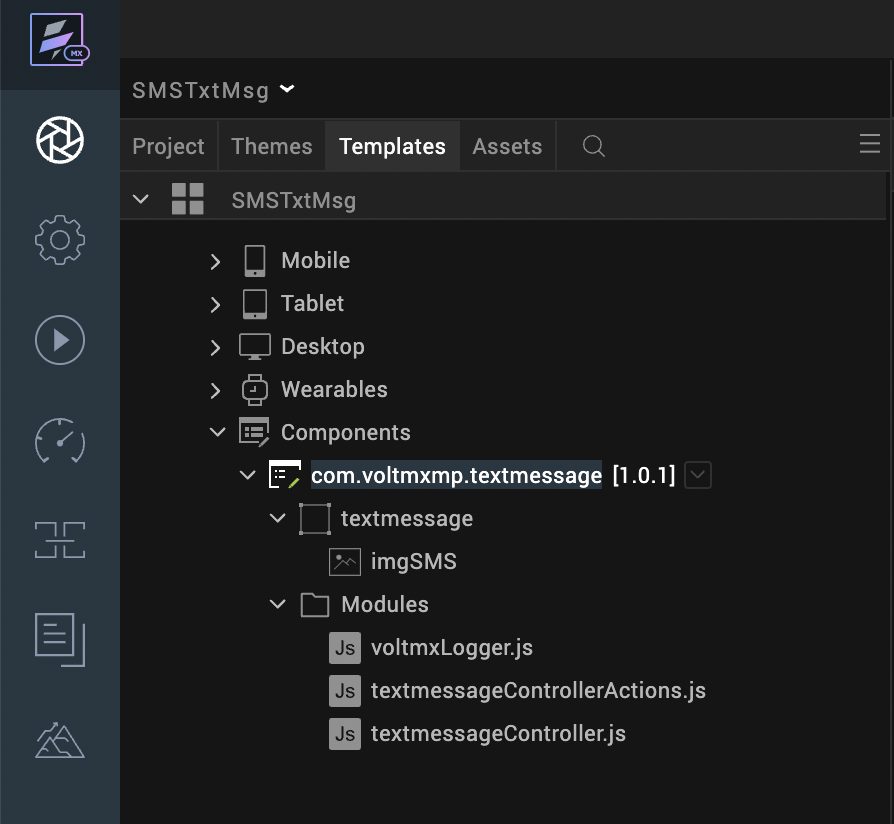
Description automatically generated

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

Graphical user interface, text, application, Teams

Description automatically generated

1. 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).

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 VoltMX 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** properties.

# **References**

## **Dynamic Usage**

You can also add **a SMS - Text Message (Phone)** component dynamically. To do so:

1. In the **Project** **Explorer**, on the **Projects** tab, click **Controllers** section to access the respective **formController**. Create a method and implement the code snippet similar to the sample code mentioned below.

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

var textmessage = new com.voltmxmp.textmessage({

"autogrowMode": voltmx.flex.AUTOGROW\_NONE,

"clipBounds": true,

"height": "50dp",

"id": "textmessage",

"isVisible": true,

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

"left": "0dp",

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

"skin": "slFbox",

"top": "0dp",

"width": "50dp"

}, {}, {});

/\*setting properties\*/

textmessage.message = "Your Text Message";

textmessage.number = "8888888888";

textmessage.smsIconSource = "voltmx\_mp\_textmessage\_smsicon.png";

/\* adding component to form\*/

this.view.add(textmessage);

In the code snippet, you can edit the properties of the component as per your requirement.

1. Save the file.

## **Properties**

The properties provided on the **Component** tab allow you to customize the UI elements in the SMS - Text Message (Phone) component. You can set the properties directly on the **Component** tab or dynamically through code. This section provides information on how to set the properties dynamically through code.

**General**

1. **Phone Number**

|  |  |
| --- | --- |
| **Description:** | Specifies the number to which the message is to be sent. |
| **Syntax:** | number |
| **Type:** | String |
| **Read/Write:** | Read + Write |
| **Example:** | this.view.componentId.number= "8888888888"; |
| **Remarks:** | Only numbers, +, - and space are treated as valid characters. |

1. **Message to be Sent**

|  |  |
| --- | --- |
| **Description:** | Specifies the message to be sent. |
| **Syntax:** | message |
| **Type:** | String |
| **Read/Write:** | Read + Write |
| **Example:** | this.view.componentId.message = "MY MESSAGE"; |

1. **SMS Icon Source**

|  |  |
| --- | --- |
| **Description:** | Specifies the source of message icon. |
| **Syntax:** | smsIconSource |
| **Type:** | String |
| **Read/Write:** | Read + Write |
| **Example:** | this.view.componentId.smsIconSource = "voltmx\_mp\_textmessage\_smsicon.png"; |
| **Remarks:** | Make sure the source file is present in resources. |

## **Events**

-- None of the events are exposed.

## **API’s**

The following API pertains to the SMS - Text Message (Phone) component:

**1. SendSMS**

The API can be invoked to send message.

**Syntax**

sendSMS(number,message)

**Parameters**

number(String),message(String)

**Return Value**

None

**Example**

this.view.componentId.sendSMS ("8888888888", "My Message");

**Remarks**

Only numbers, +, - and space are treated as valid characters for number parameter.

# **Revision History**

App version 1.0.1

## **Known Issues**

## No Known Issues

## **Limitations**

## No Limitations