2nd May 2023

TextBOx component (1.0.o)

# Overview

TextBox component is a Iris component that creates a Texbox with copy/paste options disabled and keyboard suggestions removal. Along with that Textbox component will support all the regular features same like Kony Textbox widget.

## A. Use case

Consider a scenario that you want to have a Textbox which must restrict the copy/paste options. Since this is not supporting by the Kony Textbox widget, this component would fulfil that feature.

 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 native methods to prevent the copy/paste options.
2. It supports only Android
3. Uses NFI 3.0

2. Getting Started

## A. Prerequisites:

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

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

 • Volt MX Iris

## Platforms Supported

### Mobile

#### Android

### Tablets

## Importing the app

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

##  **To import the Textbox component, do the following:**

## Open your app project in Volt MX Iris.

 2. 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)

## 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.

# 3. References

## 1. Dynamic Usage:

##  You can also add **TextBox** 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 CuustomTextField component instance \*/

var CustomTextField = new com.CustomTextField ({

 "height": "50%",

 "id": “CustomTextField",

 "isVisible": true,

 "left": "0dp",

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

 "isModalContainer": false,

 "skin": "slFbox",

 "top": "0dp",

 "width": "100%",

 "zIndex": 1,

 "overrides": {

 " CustomTextField": {

 "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 TextBox component to a Form\*/

this.view.add(CustomTextField);

 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

4.General Properties:

## API

 i. setHintAndText

|  |  |
| --- | --- |
| **Description:** | Allows to set the hind and text. |
| **Syntax**: | setHintAndText(“Hint”, “Text”) |
| **Parameters:** | String – Hint, String - Text |
| **Return Value:** |  None |
| **Remarks:**  | NA.  |
|  **Example:** | this.view.CustomTextField.setHintAndText(“Hint”, “Text”); |

###

### ii. getText

|  |  |
| --- | --- |
| **Description:** | Gives text entered in the Texbox. |
| **Syntax**: | getText() |
| **Parameters:** | None |
| **Return Value:** | None |
| **Remarks:**  | NA.  |
|  **Example:** | this.view.CustomTextField.getText(); |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| iii. setTextColor

|  |  |
| --- | --- |
| **Description:** | Changes text color by accepting hex color string. |
| **Syntax**: | setTextColor(“hint color”) |
| **Parameters:** | String – Hint color |
| **Return Value:** | None |
| **Remarks:** | NA.  |
| **Example:** | this.view.CustomTextField.setTextColor (“Hex text color”); |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| iv. setHintTextColor

|  |  |
| --- | --- |
| **Description:** |  Changes hint text color by accepting hex color string. |
| **Syntax**: | setHintextColor(“HEX Color String”)  |
| **Parameters:** | String – hint Color |
| **Return Value:** | None |
| **Remarks:** | NA.  |
| **Example:** | this.view.CustomTextField.setHintTextColor (“Hex text color”); |

 |  |

 |  |

### v. setTextLength

|  |  |
| --- | --- |
| **Description:** |  Restrict text length into method parameter number. |
| **Syntax**: | setTextLength(TextLength) |
| **Parameters:** | Int – text length  |
| **Return Value:** | None |
| **Remarks:** | NA.  |
| **Example:** | this.view.CustomTextField.setTextLength (TextSize); |

### vi. setInputType

|  |  |
| --- | --- |
| **Description:** |  Sets the input type for the textfield |
| **Syntax**: | setInputType(InputTypeConstant) |
| **Parameters:** | Int – input type constant  |
| **Return Value:** | None |
| **Remarks:** | NA.  |
| **Example:** | this.view.CustomTextField.setInputType(InputTypeConstant); |

vii. dissmissKeyboard

|  |  |
| --- | --- |
| **Description:** |  Removes the presented keyboard |
| **Syntax**: | dissmissKeyboard() |
| **Parameters:** | NA |
| **Return Value:** | None |
| **Remarks:** | NA.  |
| **Example:** |  this.view.CustomTextField.dissmissKeyboard(); |

viii. setIMEOptions

|  |  |
| --- | --- |
| **Description:** |  Changes IME options for the Texbox |
| **Syntax**: | setIMEOptions (IME Constant) |
| **Parameters:** | Int – IMEI constants  |
| **Return Value:** | None |
| **Remarks:** | NA.  |
| **Example:** | this.view.CustomTextField.setIMEOptions (IMEConstant); |

ix. setTextAlignment

|  |  |
| --- | --- |
| **Description:** |  Changes the text alignment |
| **Syntax**: | setTextAlignment(Alignment Value) |
| **Parameters:** | Int – Alignment value  |
| **Return Value:** | None |
| **Remarks:** | NA.  |
| **Example:** | this.view.CustomTextField.setTextAlignment (AlignmentValue); |

x. setPadding

|  |  |
| --- | --- |
| **Description:** |  Modifies the padding for the Textfiled |
| **Syntax**: | setPadding(left, top, right, bottom) |
| **Parameters:** | Int – Padding values  |
| **Return Value:** | None |
| **Remarks:** | NA.  |
| **Example:** | this.view.CustomTextField.setPadding(left, top, right, bottom); |

### B. Events

### i.OnDone

|  |  |
| --- | --- |
| **Description:** | Enables a onDone callback.. |
| **Syntax**: | registerOnDoneCallback(this.callback) |
| **Parameters:** | Function - callback |
| **Return Value:** | None |
| **Remarks:**  | NA  |
| **Example:** | onDoneAction() { // Write on done logic }, this.view.CustomTextField.registerOnDoneCallback(this.onDoneAction); |

###  ii. OnTextChangeCallbackcallback

|  |  |
| --- | --- |
| **Description:** | Enables a On Text Change callback. |
|  **Syntax**: | registerOnTextChangeCallback(this.callback) |
| **Parameters:** | Function - callback |
| **Return Value:** | None |
| **Remarks:**  | NA  |
| **Example:** | onTextChange() { // Write on onTextChangelogic }, this.view.CustomTextField. registerOnTextChangeCallback(this.onTextChange); |

 App version 1.0.0:

## Limitations

B. Known Issue