22 Oct 2021

Time picker component (2.0.1)

1. Overview

The Time Picker component uses native widgets to add a clock to your app that lets a user pick a specific time. You can use the Time Picker in scenarios such as: A To-Do app, where the user can set a time for appointments, meetings, doctor visits, and so on.

## Use case

### A To-Do app where the user can set a time for appointments, meetings, doctor visits, and so on.

## Features:

## It is a time input field which expands into the native clock when clicked, where a specific time can be chosen.

## Percentage of re-use:

Approximate 80% of reuse.

# Getting Started

## Prerequisites

Before you start using the Time Picker component, ensure the following:

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

• Volt MX Iris

## Platforms Supported

### Mobile

#### iOS

#### Android

### Tablets

## Importing the Component

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

##  **To import the Info Cards 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.



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%22%20%5Cl%20%22add-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 Info Cards work in real time.

# References

## Dynamic Usage

If you want to use the Time Picker 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.

createComponent : function()

{

/\* Creating the component's object \*/

var TimePicker = new com.voltmxmp.timepicker({

 "id": "timepicker",

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

 "height": "100%",

 "left": "0dp",

 "top": "0dp",

 "width": "100%",

 "isVisible": true,

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

 "skin":"slFbox",

 "zIndex": 1

 },{},{});

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

TimePicker.bgColor = "";

TimePicker.hoursView = false;

TimePicker.time = "";

TimePicker.invokeByDefault = true;

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

this.view.add(TimePicker);

}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

## Properties

The properties provided on the **Component** tab allows you to customize the elements in the **Info Card** component. These elements can be UI elements, service parameters, and so on. You can set the properties from the Volt MX Iris Properties panel on the right-hand side. You can also configure these properties using a JavaScript code.

General Properties

1. Invoke By Default (invokeByDefault )

|  |  |
| --- | --- |
| **Description:** | Specifies whether you want the to display the component by default. |
| **Syntax:** | invokeByDefault |
| **Type:** | Boolean |
| **Read/Write:** | Read + Write |
| **Default Value:** | true |
| **Remarks** | If you set this property to false, you need to call the [show](https://docs.kony.com/marketplace/timepicker/Content/Reference.htm#show) API to initialize the component. |
| **Example:** | this.view.componentID.invokeByDefault = false; |
| **Description:** | Specifies the title for the card. |
| Syntax: | cardTitle |
| **Type:** | String |
| **Read/Write:** | Read + Write |
| **Example:** | this. view. infocard.cardTitle= "#amb557766"; |
| **Remarks:** | The default value for the property is “#amb557766”. |

1. Background Color (bgColor)

|  |  |
| --- | --- |
| **Description:** | Specifies the color that you want to set for the background of the Time Picker. |
| **Syntax:** | bgColor |
| **Type:** | String |
| **Read/Write:** | Write |
| **Remarks:** | * Make sure that you provide the six digit hex code of the color without the **#**.

On Android, make sure that you set the background color such that all the widgets of the component are visible.  |
| **Example:** | this.view.componentID.bgColor = "303030"; |

 3. Set Time(time)

|  |  |
| --- | --- |
| **Description:** | Specifies the time that you want to show on the Time Picker. |
| **Syntax:** | time |
| **Type:** | String |
| **Read/Write:** | Write |
| **Remarks:** | Make sure that you provide the time in a 24-hours format. |
| **Example:** | this.view.componentID.time = "23:45"; |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

ANDROID :

 4.Twenty four hours Format(hoursView)

|  |  |
| --- | --- |
| **Description:** | Toggles the 24-hour format for the Time Picker. |
| **Syntax:** | hoursView |
| **Type:** | Boolean |
| **Read/Write:** | Write |
| **Default Value:** | false |
| **Remarks:** | Make sure that you set this property before you call the [show](https://docs.kony.com/marketplace/timepicker/Content/Reference.htm#show) API

|  |  |
| --- | --- |
| **Example:** | this.view.componentID.hoursView = true; |

 |

**EVENTS ::**

 **5. On Time Changed (**onTimeChanged**)**

|  |  |
| --- | --- |
| **Description:**Invoked when any action changes the time of the component.**Syntax:**onTimeChanged**Parameters:***time [String]* : |  |
| The current time that is selected on the component.**Example:** |  |
| this.view.componentID.onTimeChanged = function(time) |  |
| { |  |
|  | alert("Time Changed to: "+time); |
| }.bind(this); |  |

**APIs:**

**6. Show(**show**)**

|  |  |
| --- | --- |
| **Description:** | Enables the visibility of the Time Picker. |
| **Syntax:** | show() |
| **Parameters:** | None |
| **Return Value:** | None |
| **Example:** | this.view.componentID.show(); |
| **Description:** | Enables the visibility of the Time Picker. |

 **7. Hide (**hide**)**

|  |  |
| --- | --- |
| **Description:** | Disables the visibility of the Time Picker. |
| **Syntax:** | hide() |
| **Parameters:** | None |
| **Return Value:** | None |
| **Example:** | this.view.componentID.hide(); |
| **Description:** | Disables the visibility of the Time Picker. |

**8. Get Time(**getTime**)**

|  |  |
| --- | --- |
| **Description:** | Fetches the time that the user sets on the Time Picker. |
| **Syntax:** | getTime() |
| **Parameters:** | None |
| **Return Value:** | *time [String]* :The time that is set on the Time Picker. |
| **Example:** | var time = this.view.componentID.getTime();alert("The time set on the component is: "+time); |
| **Description:** | Fetches the time that the user sets on the Time Picker. |

 **9. Set Time (**setTime(time)

**)**

|  |  |
| --- | --- |
| **Description:** | Sets the specified time on the Time Picker. |
| **Syntax:** | setTime(time) |
| **Parameters:** | *time [String]* :The time that you want to set on the component. |
| **Return Value:** | None |
| **Remarks:** | Make sure that you provide the time in a 24-hours format. |
| **Example:** | var time = "23:45";this.view.componentID.setTime(time); |
|  |  |

 **10.** **Set Background Color(**setBackgroundColor()**)**

|  |  |
| --- | --- |
| **Description:** | Sets the specified color on the background of the Time Picker. |
| **Syntax:** | setBackgroundColor() |
| **Parameters:** | *color [String]* :The hex code of the color that you want to set on the background of the component. |
| **Return Value:** | None |
| **Remarks:** | Make sure that you give the six digit hex code of the color without the **#**. |
| **Example:** | var color = "ffffff";this.view.componentID.setBackgroundColor(color); |

# Revision History

App version 2.0.1:

## Known Issues

## -

## Limitations

## TimePicker component uses NFI and doesn’t work on PWA/Responsive Web.