Date : 25-June-24

DATE RANGE PICKER

VERSION: 2.0.4

**1. OVERVIEW**

The Date Range picker component provides an interactive calendar interface for quickly selecting a pair of dates. It is a customized calendar widget that enables a user to select a start date and end date. The component can be customized as required.

## **Use case:**

### App that need user to choose date range. Eg. A travel app that a user needs to book a flight or a hotel room.

### **B. Features:**

### Ready-to-use date range picker component for mobile, tablet apps

### Calendar Interface for user to quickly input a pair of dates

### Configure as per your requirement- specify label names and calendar range

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

90% (Data can be customizable)

# **Getting Started**

## **Prerequisites**

Before you start using the DateRangePicker component, ensure you have the following:

* 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 Date Range Picker component, do the following:**

## Open your app project in Volt MX Iris.

2. In the Project Explorer, click the **Templates** tab.

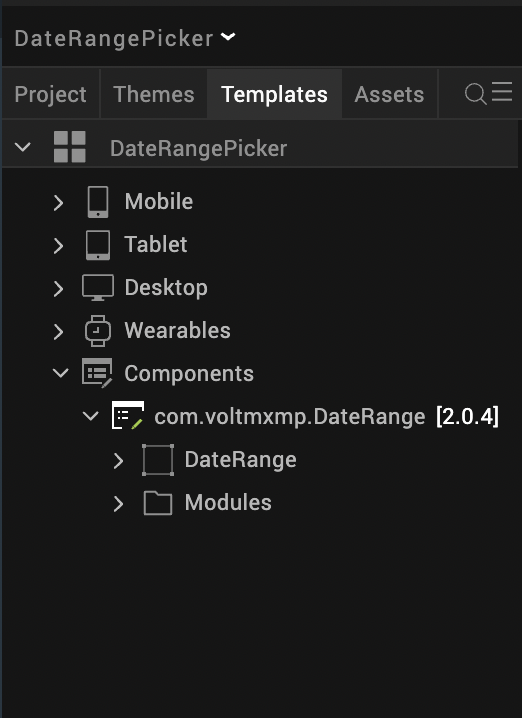
## Graphical user interface, text, application Description automatically generated

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

Graphical user interface, text, application

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 **DateRangePicker** 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 \*/

createComponent: function(data)

{

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

var dateRangeInstance = new com.voltmxmp.DateRange(

{

"clipBounds": true,

"height": "100%",

"id": "DateRange",

"isVisible": true,

"left": "0dp",

"top": "0dp",

"width": "100%",

"zIndex": 1

}, {}, {});

dateRangeInstance.yearLabelSkin = "voltmxmpdrsknflxWhite1";

dateRangeInstance.toDateLabel="To";

dateRangeInstance.fromDateLabel = "From";

dateRangeInstance.calendarStartDate = startDate;

dateRangeInstance.calendarEndDate = endDate;

dateRangeInstance.onDateSelectionDone = function(){

dateRangeInstance.getSelectedFromDate();

dateRangeInstance.getSelectedToDate();

}.bind(this);

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

this.view.add(dateRangeInstance);

}

1. **Save** the file.

## **B. Properties**

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

**Open1. To Date Label**

| **Category:** | General |
| --- | --- |
| **Description:** | Specifies the text that you want to use on the To Date label |
| **Syntax:** | toDateLabel |
| **Type:** | String |
| **Read/Write:** | Write |
| **Example:** | this.view.componentID.toDateLabel = "To"; |

**2. From Date Label**

| **Category:** | General |
| --- | --- |
| **Description:** | Specifies the text that you want to use on the From Date label. |
| **Syntax:** | fromDateLabel |
| **Type:** | String |
| **Read/Write:** | Write |
| **Example:** | this.view.componentID.fromDateLabel = "From"; |

**3. Calendar Start Date**

| **Category:** | General |
| --- | --- |
| **Description:** | Specifies the date from which you want to load the dates on the calendar. |
| **Syntax:** | calendarStartDate |
| **Type:** | String |
| **Read/Write:** | Write |
| **Example:** | this.view.componentID.calendarStartDate="03/08/2022"; |

4. **Calendar End Date**

| **Category:** | General |
| --- | --- |
| **Description:** | Specifies the date till which you want to load the dates on the calendar. |
| **Syntax:** | calendarEndDate |
| **Type:** | String |
| **Read/Write:** | Write |
| **Example:** | this.view.componentID.calendarEndDate="10/01/2022"; |

## **C. Events**

**1. onDateSelectionDone**

Invoked when the user selects the From Date, the To Date, and taps OK.

**Syntax:**

onDateSelectionDone

**Parameters:**

*None*

**Return Value:**

None

**Sample Input Format:**

this.view.componentID.onDateSelectionDone = function()

{

alert("Date range selected!");

}.bind(this);

## **D.APIs**

**1. getSelectedFromDate**

Fetches the From Date that the user selects on the calendar.

**Syntax:**

getSelectedFromDate

**Parameters:**

*None*

**Return Value:**

selectedFromDate[JSON]

**Sample Format:**

var selectedFromDate = this.view.componentID.getSelectedFromDate();

alert("Selected From Date: "+selectedFromDate.dateString);

2**. getSelectedToDate**

Fetches the To Date that the user selects on the calendar.

**Syntax:**

getSelectedToDate

**Parameters:**

*None*

**Return Value:**

selectedToDate[JSON]

**Sample Format**

var selectedToDate = this.view.componentID.getSelectedToDate();

alert("Selected To Date: "+selectedToDate.dateString);

# **4. Revision History**

App version 2.0.4

## **Limitations**

* The calendar won’t be rendered if calendar end date is less than calendar start date.