



# Kony Visualizer

## User Guide

Release 2.0

### Document Relevance and Accuracy

This document is considered relevant to the Release stated on this title page and the document version stated on the Revision History page. Remember to always view and download the latest document version relevant to the software release you are using.

Copyright © 2014 Kony, Inc.

All rights reserved.

February, 2015

This document contains information proprietary to Kony, Inc., is bound by the Kony license agreements and may not be used except in the context of understanding the use and methods of Kony Inc, software without prior, express, written permission. Kony, Empowering Everywhere, Kony MobileFabric, Kony Modeler, and Kony Visualizer are trademarks of Kony, Inc. Microsoft, the Microsoft logo, Internet Explorer, Windows, and Windows Vista are registered trademarks of Microsoft Corporation. Apple, the Apple logo, iTunes, iPhone, iPad, OS X, Objective-C, Safari, and Xcode are registered trademarks of Apple, Inc. Google, the Google logo, Android, and the Android logo are registered trademarks of Google, Inc. Chrome is a trademark of Google, Inc. BlackBerry, PlayBook, Research in Motion, and RIM are registered trademarks of BlackBerry. All other terms, trademarks, or service marks mentioned in this document have been capitalized and are to be considered the property of their respective owners.

## Revision History

| Date       | Document Version | Description of Modifications/Release   |
|------------|------------------|--|
| 01/05/2014 | 1.0              | Kony Visualizer User Guide   |
| 05/05/2014 | 1.1              | Properties renamed, images refreshed, added section: Importing and Exporting a project, and added iOS 7 specific properties.   |
| 08/19/2014 | 1.2              | Added the following sections: Importing Photoshop Styles, Upgrading Kony Visualizer, and Visualizer Admin Console. Updated various widget PSP.   |
| 12/15/2014 | 2.0              | Added the following sections: Actions, Internationalizing (i18n) Application Content, Windows PSP and Modules.   |
| 02/23/2015 | 2.1              | Updated the document for Visualizer 2.0.2 release with the following features: <a href="#">Form Forking</a> , added new PSP for widgets, <a href="#">Support for legacy widgets</a> , <a href="#">Search Feature</a> , and <a href="#">Autogrowth of a Segment</a> . |

## Table of Contents

---

|                                  |           |
|----------------------------------|-----------|
| <b>1. Preface .....</b>          | <b>12</b> |
| 1.1 Purpose .....                | 12        |
| 1.2 Intended Audience .....      | 13        |
| 1.3 Formatting Conventions ..... | 13        |
| 1.4 Contact Us .....             | 14        |
| <b>2. Introduction .....</b>     | <b>15</b> |
| <b>3. Project Explorer .....</b> | <b>17</b> |
| 3.1 Project Name .....           | 18        |
| 3.2 Project Context Menu .....   | 20        |
| 3.3 Project Tab .....            | 21        |
| 3.4 Skins Tab .....              | 31        |
| 3.5 Templates Tab .....          | 35        |
| 3.6 Assets Tab .....             | 41        |
| 3.7 Search .....                 | 44        |
| 3.8 Filters .....                | 44        |

---

|                                      |            |
|--------------------------------------|------------|
| <b>4. Library Explorer .....</b>     | <b>47</b>  |
| <b>4.1 Library Menu .....</b>        | <b>48</b>  |
| <b>4.2 Widgets .....</b>             | <b>50</b>  |
| <b>4.3 Collections .....</b>         | <b>54</b>  |
| <b>4.4 Library Skins .....</b>       | <b>56</b>  |
| <b>5. Working with Widgets .....</b> | <b>57</b>  |
| <b>5.1 FlexContainer .....</b>       | <b>58</b>  |
| <b>5.2 FlexScrollContainer .....</b> | <b>62</b>  |
| <b>5.3 HBox .....</b>                | <b>72</b>  |
| <b>5.4 ScrollBox .....</b>           | <b>78</b>  |
| <b>5.5 TabPane .....</b>             | <b>88</b>  |
| <b>5.6 VBox .....</b>                | <b>94</b>  |
| <b>5.7 Button .....</b>              | <b>98</b>  |
| <b>5.8 Calendar .....</b>            | <b>104</b> |
| <b>5.9 CheckBoxGroup .....</b>       | <b>120</b> |
| <b>5.10 ComboBox .....</b>           | <b>129</b> |

---

|      |                         |     |
|------|-------------------------|-----|
| 5.11 | DataGridView            | 139 |
| 5.12 | Image2                  | 147 |
| 5.13 | Label                   | 154 |
| 5.14 | Line                    | 160 |
| 5.15 | Link                    | 164 |
| 5.16 | ListBox                 | 169 |
| 5.17 | RadioButtonGroup        | 182 |
| 5.18 | RichText                | 193 |
| 5.19 | Slider                  | 199 |
| 5.20 | TextArea2               | 211 |
| 5.21 | TextBox2                | 223 |
| 5.22 | Browser                 | 243 |
| 5.23 | Camera                  | 251 |
| 5.24 | Horizontal Image Strip2 | 261 |
| 5.25 | ImageGallery2           | 268 |
| 5.26 | Map                     | 274 |

---

|  |            |
|--|------------|
| 5.27 Phone .....                       | 284        |
| 5.28 PickerView .....                  | 289        |
| 5.29 Segment2 .....                    | 294        |
| 5.30 Switch .....                      | 313        |
| <b>6. VBox Form .....</b>              | <b>318</b> |
| 6.1 Form Properties .....              | 318        |
| 6.2 Look .....                         | 318        |
| 6.3 Skin .....                         | 321        |
| 6.4 Platform Specific properties ..... | 322        |
| 6.5 Actions .....                      | 346        |
| 6.6 Notes .....                        | 349        |
| <b>7. Flex Form .....</b>              | <b>352</b> |
| 7.1 Look .....                         | 352        |
| 7.2 Skin .....                         | 353        |
| 7.3 Platform Specific properties ..... | 354        |
| 7.4 Actions .....                      | 382        |

---

|  |            |
|--|------------|
| <b>7.5 Notes .....</b>                             | <b>386</b> |
| <b>8. Flex Container Rules .....</b>               | <b>389</b> |
| <b>9. Popup .....</b>                              | <b>396</b> |
| <b>9.1 Popup Properties .....</b>                  | <b>396</b> |
| <b>9.2 Look .....</b>                              | <b>396</b> |
| <b>9.3 Skin .....</b>                              | <b>397</b> |
| <b>9.4 Platform Specific properties .....</b>      | <b>398</b> |
| <b>9.5 Actions .....</b>                           | <b>413</b> |
| <b>10. Common Properties .....</b>                 | <b>415</b> |
| <b>10.1 Look .....</b>                             | <b>416</b> |
| <b>10.2 Working with Skins .....</b>               | <b>431</b> |
| <b>10.3 Forking .....</b>                          | <b>445</b> |
| <b>11. Action Editor .....</b>                     | <b>452</b> |
| <b>11.1 Important Considerations .....</b>         | <b>452</b> |
| <b>11.2 Accessing Action Editor .....</b>          | <b>452</b> |
| <b>11.3 Adding and configuring an action .....</b> | <b>454</b> |

---

|   |            |
|---|------------|
| 11.4 List of Actions .....                            | 459        |
| <b>12. Importing Photoshop Styles .....</b>           | <b>500</b> |
| 12.1 Prerequisites .....                              | 500        |
| 12.2 Importing Photoshop Styles into Visualizer ..... | 500        |
| 12.3 Naming Conventions .....                         | 503        |
| 12.4 Limitations .....                                | 504        |
| <b>13. App Canvas .....</b>                           | <b>505</b> |
| 13.1 Inline Editing .....                             | 509        |
| <b>14. Top Menu .....</b>                             | <b>512</b> |
| 14.1 Visualizer .....                                 | 513        |
| 14.2 Project .....                                    | 517        |
| 14.3 Run .....  | 521        |
| 14.4 Cloud .....                                      | 522        |
| 14.5 Previewing an App .....                          | 533        |
| <b>15. Importing and Exporting a Project .....</b>    | <b>562</b> |
| 15.1 Exporting a Project to Kony Studio .....         | 562        |

---

|   |            |
|---|------------|
| 15.2 Importing a Project from Kony Studio .....                                   | 568        |
| <b>16. Upgrade Kony Visualizer .....</b>  | <b>572</b> |
| <b>17. Internationalizing (i18n) Application Content .....</b>                    | <b>574</b> |
| 17.1 Configuring i18n Locales and Adding Keys .....                               | 574        |
| 17.2 Assigning an i18n key to a widget .....                                      | 578        |
| 17.3 Custom Locales .....   | 580        |
| 17.4 Functional Preview .....   | 583        |
| <b>18. Visualization Services .....</b>   | <b>584</b> |
| 18.1 Accessing Visualization Services .....                                       | 584        |
| 18.2 Prototypes .....   | 585        |
| 18.3 Visualizer Projects .....  | 586        |
| 18.4 Studio Projects .....  | 587        |
| <b>19. Support for Legacy Widgets and Old Projects .....</b>                      | <b>589</b> |
| 19.1 How Visualizer supports legacy widgets .....                                 | 589        |
| 19.2 How a project created in earlier versions of Visualizer<br>is upgraded ..... | 589        |

---

|  |            |
|--|------------|
| <b>20. Actions .....</b>   | <b>591</b> |
| <b>20.1 Important Considerations .....</b>   | <b>591</b> |
| <b>20.2 Creating an actions from the Action tab .....</b>  | <b>591</b> |
| <b>20.3 Creating an action from the Action tab of a Widget (or<br/>        Pop-up or form) .....</b> | <b>592</b> |
| <b>20.4 Assigning an Action .....</b>  | <b>593</b> |
| <b>21. Keyboard Shortcuts .....</b>  | <b>596</b> |
| <b>22. Flex Layout FAQ .....</b>   | <b>597</b> |

## 1. Preface

Kony Visualizer enables you to create highly interactive designs for mobile and tablet devices. It contains a large number of built-in designing elements that help in creating your application designs specific to a device and its operating system. This WYSIWYG tool provides an instant preview of your design, thereby, helping you make necessary changes to the design quickly.

Manipulating a widget's look and feel, changing its skin, ensuring a property is available for a specific platform or assigning an action event is easily achieved without requiring you to write even a single line of code!

Your team can reuse the user-defined designing elements and skins by importing and exporting them. And not only the designing elements and skins, but the complete project can be shared easily. You can upload the project directly to your cloud and allow your team to download the project. This helps in saving lot of effort in designing duplicates, and in maintaining consistent look and feel across the projects.

One of the unique features of Kony Visualizer is its ability to provide a functional preview of your design. Similar to how an end-user would view the User Interface (UI) of an application, you can generate a functional preview that allows your team to view application UI and execute any action events attached.

Seamless integration with Kony Studio allows you to export (or import) your designs effortlessly.

The product is available on both the Windows and Macintosh platforms. You could easily build your design on one platform, and preview or modify the same design on the other platform.

### 1.1 Purpose

The document helps you familiarize with the Kony Visualizer, and provide procedural information to perform various tasks required to build your application prototype.

The scope of the document is limited to explaining the process involved between creating a design and exporting the design to Kony Studio.

## 1.2 Intended Audience

This document is written for web and application designers who use Kony Visualizer to design and wireframe mobile and tablet application.

## 1.3 Formatting Conventions

The following are the formatting conventions used throughout the document:

| Conventions   | Explanation  |
|---------------|--|
| Monospace     | <ul style="list-style-type: none"><li>■ User input text, system prompts and responses</li><li>■ File Path</li><li>■ Commands</li><li>■ Program Code</li><li>■ File Names</li></ul> |
| <i>Italic</i> | <ul style="list-style-type: none"><li>■ Emphasis</li><li>■ Names of Books and Documents</li><li>■ New Terminology</li></ul>  |
| <b>Bold</b>   | <ul style="list-style-type: none"><li>■ Windows</li><li>■ Menus</li><li>■ Buttons</li><li>■ Icons</li><li>■ Fields</li><li>■ Tabs</li><li>■ Folders</li></ul>                      |

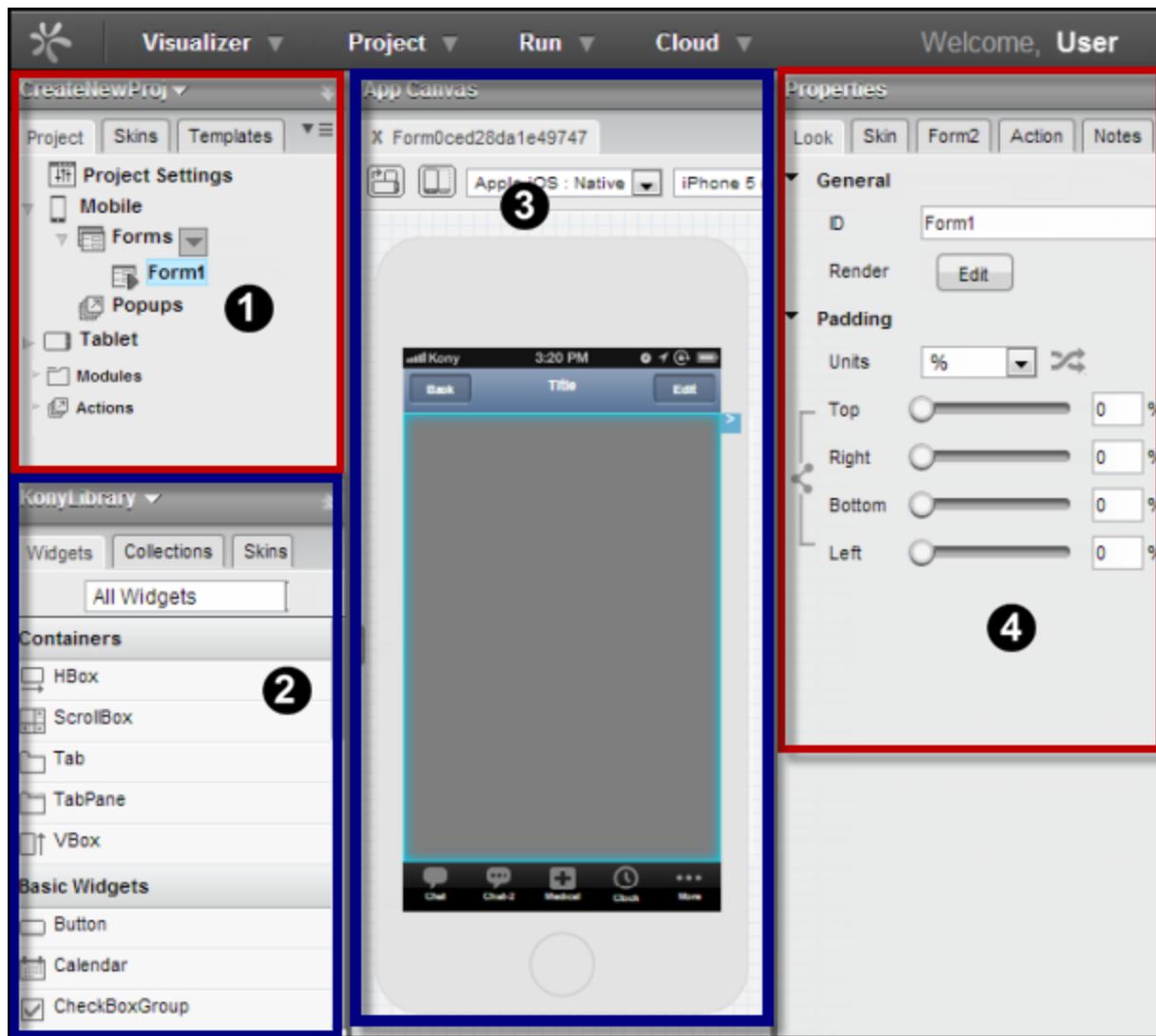
| Conventions      | Explanation   |
|------------------|---|
| <u>URL</u>       | Active link to a URL.   |
| <b>Note</b>      | Provides helpful hints or additional information.                               |
| <b>Important</b> | Highlights actions or information that might cause problems to systems or data. |

## 1.4 Contact Us

We welcome your feedback on our documentation. Write to us at [techpubs@kony.com](mailto:techpubs@kony.com). For technical questions, suggestions, comments, or to report problems on Kony's product line, contact [productsupport@kony.com](mailto:productsupport@kony.com).

## 2. Introduction

Kony Visualizer workspace is a logical grouping of various elements and tasks that enable you to create your designs quickly.



The workspace consists of four major areas and include:

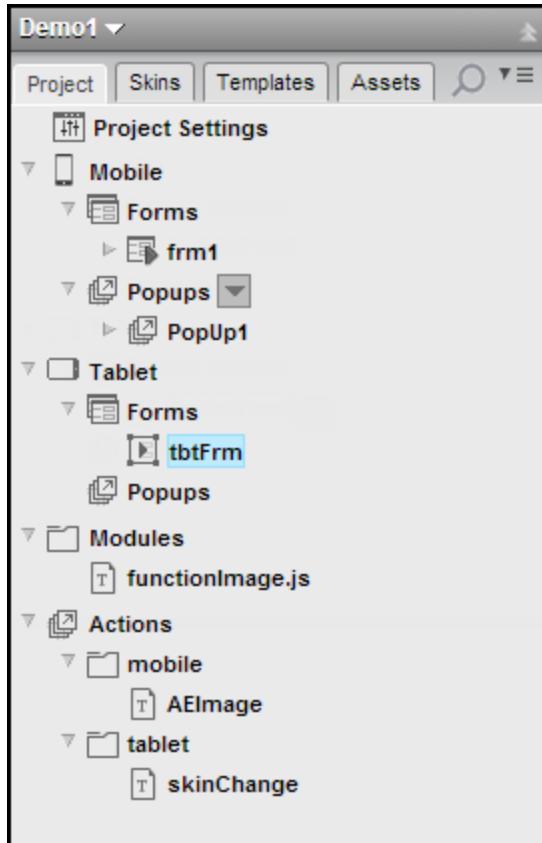
1. **Project Explorer:** You configure project settings, create projects, and add forms for mobile and tablet devices, view the widgets used in a project, edit widgets skins, create templates, and view

the assets available for a project.

2. Library Explorer: It consists of Kony built-in widgets and skins. You can also view the user-defined libraries with collections and themes.
3. App Canvas: It is the central panel of this WYSIWYG environment. Here, you can view the design being created and make the necessary modifications.
4. Properties: Widget properties are displayed in the right-hand side panel of Kony Visualizer workspace. From here, you can:
  - Manipulate widget look and feel.
  - Change, modify, and edit a widget skin.
  - Provide platform specific properties.
  - Apply action events.

## 3. Project Explorer

Project Explorer is located in the top-left panel of Kony Visualizer workspace.



Project Explorer consists of the following sections:

1. [Project Name](#): The name of the project is displayed. You can rename a project by clicking the project name > **Rename** option.
2. [Project tab](#): Allows you to configure project settings, create, and view forms and pop-ups of the mobile and tablet devices.
3. [Skins tab](#) : Allows you to view the skins being used in the design, edit an existing skin and create new skins, and themes (for holding the skins.)

4. [Templates tab](#): Allows you to create templates for a Form (header and footer), Segments, Maps, and Calendar.
5. [Assets tab](#): Allows you to view the images and fonts that are available for a project.
6. [Search](#): Allows you to search for the elements within Project, Skins, Templates , or Assets tab.
7. [Filters](#): Allows you to filter for the elements within Project, Skins, Templates , or Assets tab.

### 3.1 Project Name

Your project name should follow the below guidelines:

- It should not contain more than 13 characters.
- Only alphanumeric characters are allowed.
- Project name should not start with a number.
- Reserved words should not be used as project names.

The following table shows a few examples of valid and invalid project names:

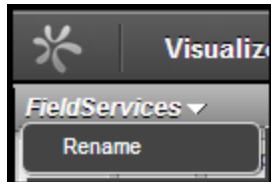
| Valid Project Name | Invalid Project Name | Remarks  |
|--------------------|----------------------|--|
| FirstProject       | 1stProject           | Project name should not start with a number.           |
| myProject          | my Project           | Spaces (and other special characters) are not allowed. |
| bankproject        | continue             | The Project name is a Java keyword.                    |

After Installing Visualizer, when you launch it for the first time, you are required to either create a new project or import an existing project. Until then, the remaining workspace is disabled. You can:

1. Create a new project by clicking **Create New Project > New Project**.
2. Import an existing project from your local machine by clicking **Project > Import Project**.
3. Import an existing project from your cloud account by clicking **Project > Import Cloud Studio Project**.

## 3.2 Project Context Menu

From the project context menu, you can rename the project.



**Rename:** Allows you to rename the project. When you click Rename, the project name changes to an inline editable text field.

## 3.3 Project Tab

The major sections of the Project tab are:

- [Project Settings](#): Provide global settings for your project.
- [Forms](#): It is a visual area (basic application screen) that holds other widgets on it.
- [Popups](#): Similar to a Form, it is a visual component that holds other widgets on it. In general, it is invoked when an action is triggered inside a form.

### 3.3.1 Project Settings

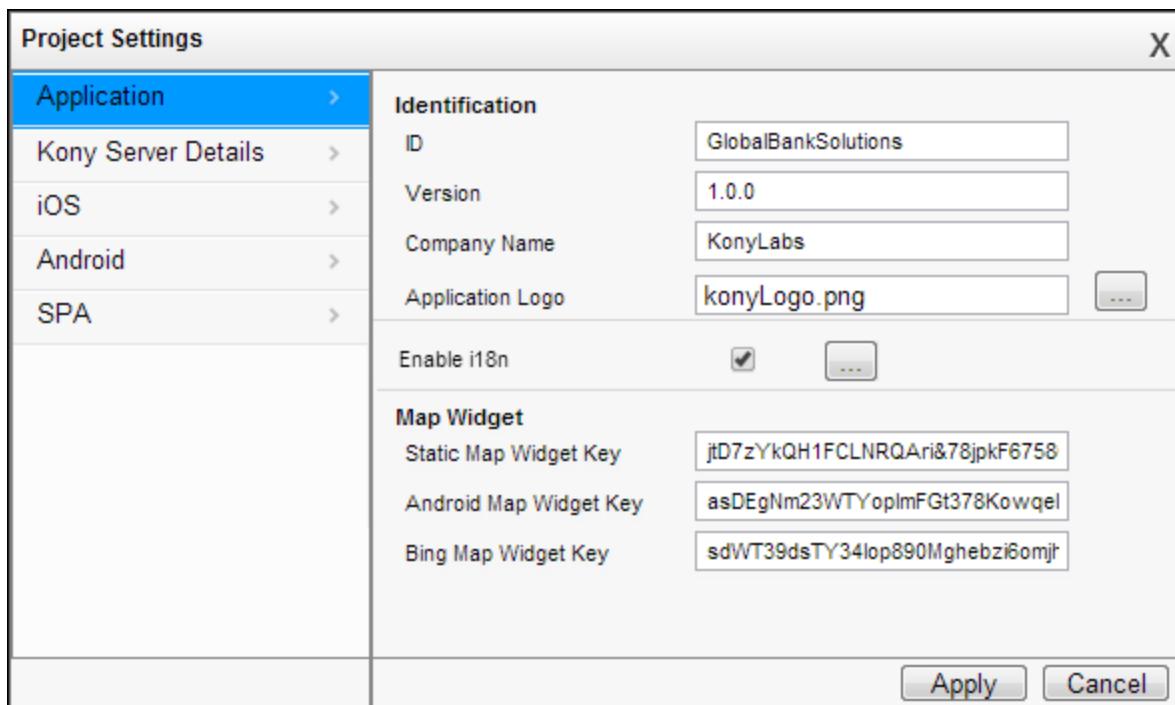
You can provide platform specific settings as well as common settings that are applicable across the platforms.

You can modify the following project settings:

- [Application](#)
- [Kony Server Details](#)
- [iOS](#)
- [Android](#)
- [SPA](#)

#### 3.3.1.1 Application

The following are the list of settings available across the platforms:



1. **ID:** Name of the project.
2. **Version:** Current version of the project.
3. **Company Name:** Specify your company name.
4. **Application Logo:** You may provide a logo for the application by clicking the ellipsis button (...) located next to the **Application Logo** field, and then selecting an image from the available images in the project or you can provide the URL of the image to be used as application logo.

**Note:** Before using an image in a project, navigate to following location:  
<konyworkspace>/<Name of your project>/resources and copy the image in the respective subfolder. For example, if you want an image to be available across all the channels, copy the image in the subfolder **common**. Alternatively, if you wanted an image to be available

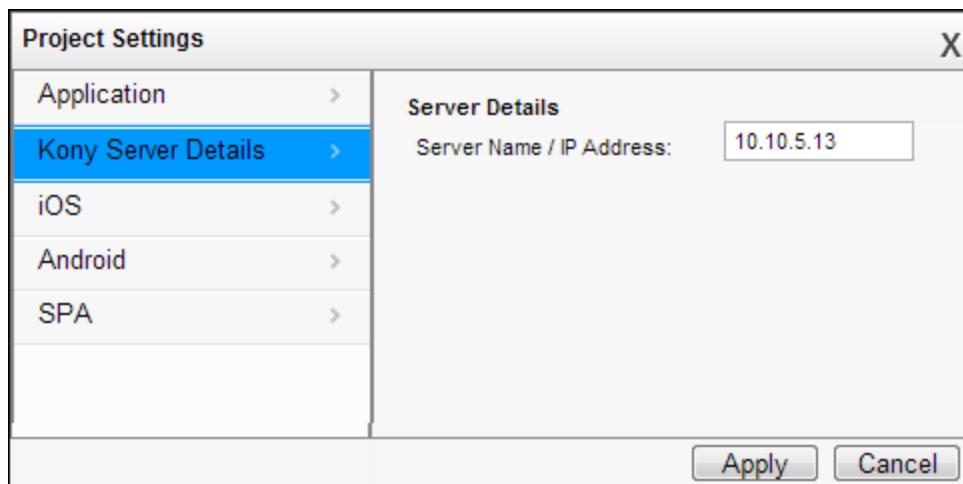
only for a specific platform, for example, mobile channel >native platform, copy the image to mobile/native subfolder. After copying the image, ensure you click the refresh button from the **Project Explorer > Assets** tab (in Kony Visualizer) to bind the newly added image to the project.

**Note:** The above identification elements are required when submitting an application to Android or Google store.

5. **Enable i18N:** Enables you to display the application in multiple languages. For more information, see [Internationalizing \(i18n\) Application Content](#).
6. **Map Widget:** The map widget key needs to be generated based on a domain name. The key generated for a single domain can be used for all sub-domains, URLs on hosts in those domains, and all ports on those hosts. You can provide the following type of Map keys:
  - a. **Static Map Widget Key:** This key is applicable for Android, iOS and SPA. For generating a specific map API key and use it in the properties, see <https://developers.google.com/maps>.
  - b. **Android Map Widget Key:** You need the Google Maps API key for Android to enable Maps in the applications you develop for Android platform. Maps API keys are linked to specific certificate/package pairs, rather than to users or applications. You only need one key for each certificate, no matter how many users you have for an application. Applications that use the same certificate can use the same API key. For generating maps API key for Android, you need to provide the fingerprint of the signed certificate. For more information about signing Android applications, see <http://developer.android.com/guide/developing/eclipse-adt.html> and <https://developers.google.com/maps/documentation/android/start?hl=en>
  - c. **Bing Map Widget Key:** To generate a Bing Map Key for Windows platform, refer <http://msdn.microsoft.com/en-us/library/ff428642.aspx>.

### 3.3.1.2 Kony Server Details

You can provide a server name or an IP address of a machine containing project specific services such as validating user credentials and interacting with the database.

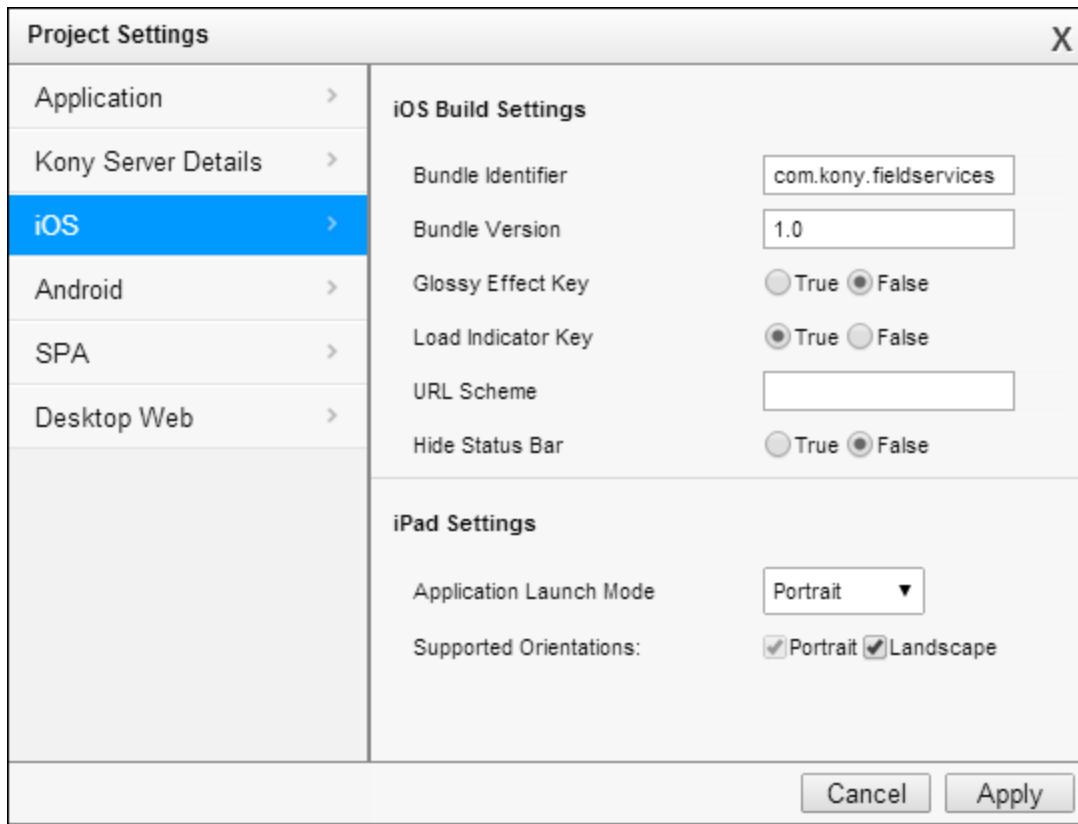


When a Kony Studio project is imported, the server details are automatically updated in Kony Visualizer. These details are useful while performing functional preview of the project (using option Run > Run-IDE), where the project needs to interact with the server to run the required services.

**Note:** You can also type a server name or an IP address manually. This option is useful when the primary sever (whose details were automatically updated in Kony Visualizer when Kony Studio project is imported) is unavailable, and you wanted to provide the details of a backup server.

### 3.3.1.3 iOS

The following are the list of settings available for iOS platform:



- **iOS Build Settings**

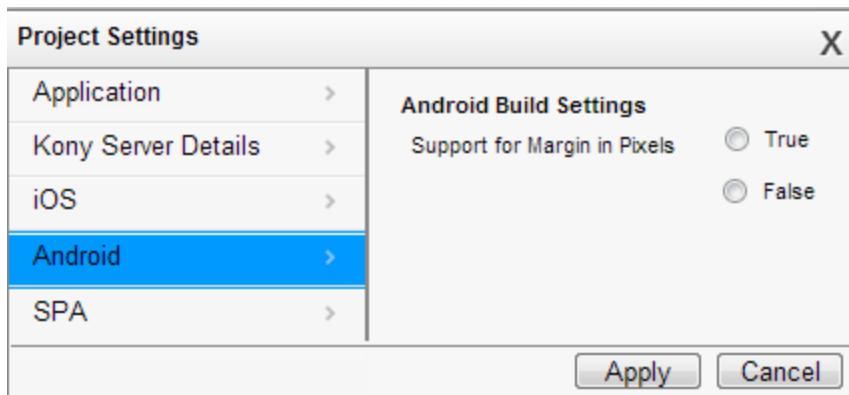
- Bundle Identifier: Specifies a unique name that identifies the application bundle. This is usually in three parts and follows the convention of *com.kony.<appname>*.
- Bundle Version: Specifies the version number of your project.
- Glossy Effect Key: Specifies Glossy effect key.

- **iPad Settings**

- Application Launch Mode: Specifies the default mode of launching the application on an iPad.
- Supported Orientations: Specifies the supported orientations for the iPad. Both orientation modes are selected by default.

#### 3.3.1.4 Android

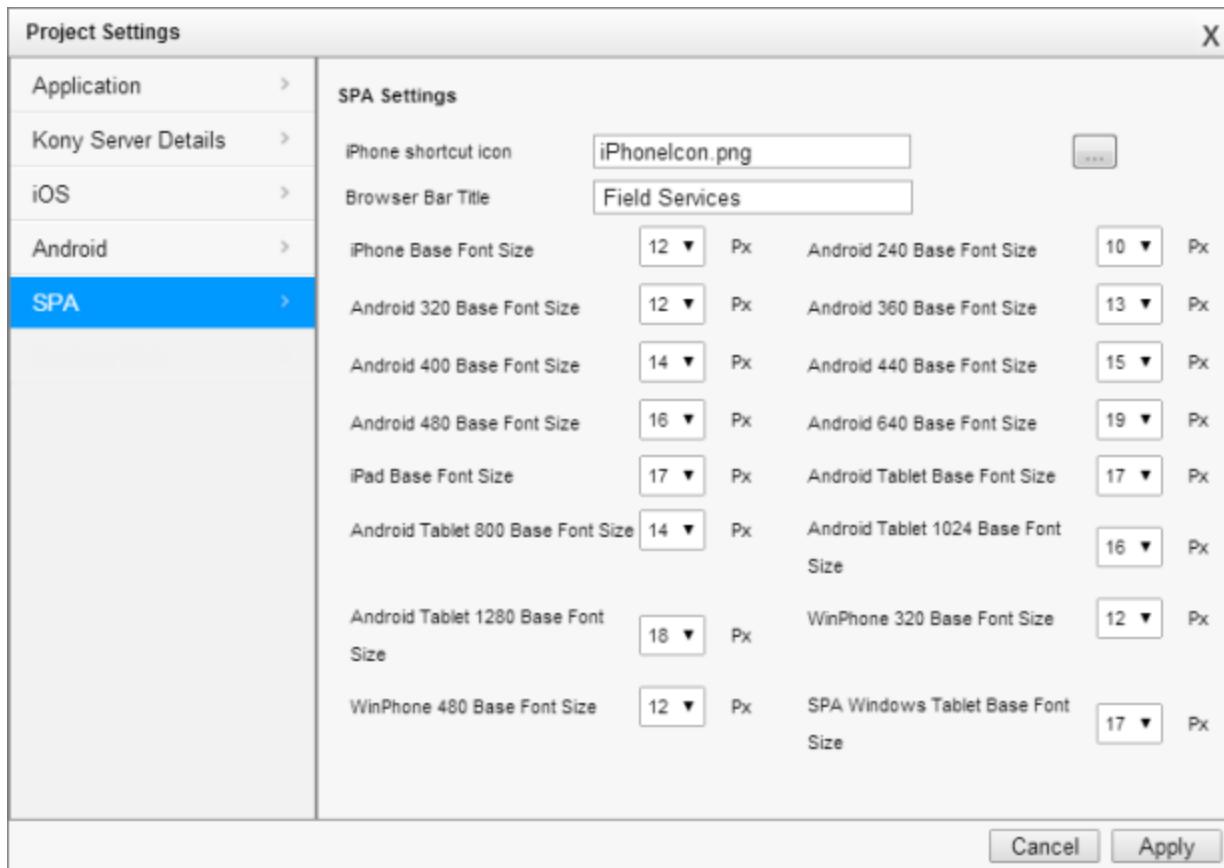
The following are the list of settings available for Android platform.



- Android Build Settings
  - Support for Margin in Pixels: Provides you with an option to choose whether the **Support for the Margin in Pixels** should be allowed.

#### 3.3.1.5 SPA

The following are the list of settings available for SPA platform.



- iPhone shortcut icon: Specifies the project icon that appears on an iPhone.
- Browser Bar Title: Specifies the project title that appears on browser bar.
- Android Tablet Base Font Size: Specifies the project base font size that appears on an Android tablet.
- iPad Base Font Size: Specifies the project base font size that appears on an iPad.
- WinPhone Base Font Size: Specifies the project base font size that appears on a Windows phone.
- SPA Windows Tablet Base Font Size: Specifies the project base font size that appears on a Windows tablet.

### 3.3.2 Forms

*Form is the starting point for any application.* A form is the top most container. A form can contain any number of widgets but cannot contain another form.

#### Adding a Form for a Mobile

1. From the **Project** tab of **Project Explorer**, expand **Mobile** folder. Right-click **Form** > **New Form**. Select either **VBox Form** or **Flex Form**. This result in creating a new form with a default name assigned to it.
2. You may choose to rename the form by clicking the down-arrow next to the form and then clicking **Rename**. The name of the form is in edit mode allowing you to rename the form.
3. To make a form as a startup form (that is, the first form to be displayed after the application is loaded), click the down-arrow next the form and then click **Mark as Startup**.

**Note:** To add a Form for Tablet devices, follow similar steps as stated above.

#### 3.3.2.1 Form Context Menu

Form context menu allows you to perform various actions such as:

- **Open:** Allows you to open and display a form's design on the App Canvas pane.
- **Rename:** Allows you to rename a form.
- **Duplicate:** Allows you to duplicate a form. Duplicating a form creates an exact copy of the form under the same channel.
- **Copy:** Allows you to copy a form.
- **Paste:** Allows you to paste a copied form. This option gets activated only when you have copied a form. You can paste a form across the channels. That is, a form designed for mobile channel can be copied and pasted to a tablet channel.
- **Delete:** Allows you to delete the form. Before deleting, system displays a warning message.

- **Mark as StartUp:** Allows you to identify a form as the first form to be displayed after an application is loaded. By default, the first form you add to a newly created project is marked as StartUp form.

**Note:** When a startup form is deleted, ensure that you choose another form as a startup form for a project. Else, you may encounter errors while previewing or building the project.

For details on Form properties, refer [Form](#) section.

### 3.3.3 Popups

A popup usually appears in the center of the screen on top of the form from which you have invoked the popup. It does not span the entire screen width. Popups allow you to partition UI design into smaller parts.

Following are the steps for adding a popup for Mobile applications:

1. From the **Project** tab of **Project Explorer**, expand **Mobile** folder. Rest the pointer on **Popups** to display a down-arrow. Click this down-arrow and then click **New Popup**. This result in creating a popup with a default name assigned to it.
2. You may choose to rename the popup by clicking the down-arrow next to the popup and then clicking **Rename**. The name of the form is in edit mode, allowing you to type a new popup name.

**Note:** To add a popup for tablet applications, follow similar steps as stated above.

## 3.4 Skins Tab

A widget skin defines the look and feel of the widget when it is rendered on the form in an application. You can specify different skins for different widgets, or different skins for different platforms. The Kony Visualizer allows you to create skins and specify skins for widgets while setting the properties.

Some of the features of **Project Explorer > Skins** are:

- Creating new themes
- Duplicating themes
- Creating new skins
- Assigning new skins
- Editing skins
- Forking Skins
- Duplicating Skins
- Adding skins to themes

Theme

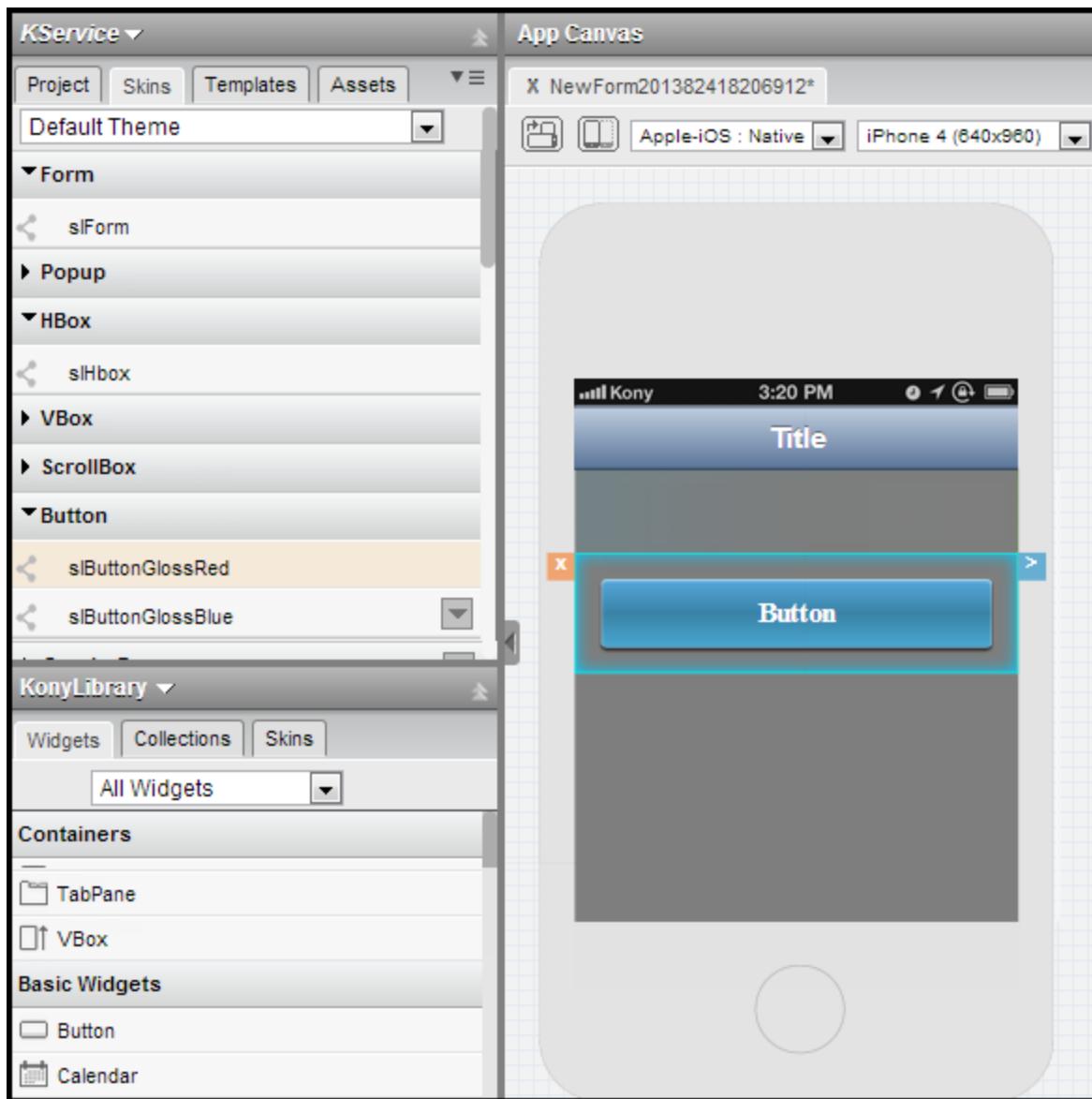
Theme holds the skins that are created

### 3.4.1 Default Theme

The skins that are available in the **Default Theme** are provided by Kony Visualizer.

At the time of creating a new project, **Default Theme** contains empty widget folders. As you start adding widgets to your design, the respective skins for the widgets will be added automatically.

For example,



The above design contains a Form, Button and HBox. Only the skins for these elements are available, while the remaining widget folders remain empty.

The **Default Theme** and the default skins cannot be modified, deleted or renamed. Attempt to modify a default skin results in creating a new skin. These new skins can be used within the same project. If you require these skins for other projects, save them in a user-defined theme, and then export them.

### 3.4.2 Create a New Theme

For storing default as well as modified skins you create a new theme.

From the **Skin** tab of the **Project Explorer**, click the **Default Theme** list and then click **Create New Theme**. A new theme with a default name is created. *All the skins (default and customized) that are used in the project till the time of creating this new theme are added automatically to the newly created theme.*

### 3.4.3 Rename a Theme

You can rename a theme by clicking the Theme drop-down under **Skins** tab and then click **Rename**. From the **Rename Theme** dialog box type a new name for the Theme.

### 3.4.4 Duplicate Current Theme

You can duplicate an existing theme, by selecting the theme from the list below the **Skins** tab and then clicking **Duplicate Current Theme** from the same list.

### 3.4.5 Delete Current Theme

To delete an existing theme, select a theme from the list below the **Skins** tab and then select **Delete Current Theme** from the same list.

### 3.4.6 Skin Context Menu

To access the skin context menu list, right-click on any of the skins, items include:

- Assign To: Allows you to assign a skin to a widget state such as Normal or Focus.
- Edit: Opens the skin in edit mode and allows you to modify its properties.
- Copy: Copies the skins.
- Paste: Pastes a skin into the selected theme. This option is available only when a skin is copied.
- Duplicate: Duplicates a skin.
- Rename: Rename a skin.

- Delete: Deletes a skin.

**Note:** When a skin is deleted, widgets attached to the deleted skin will revert back to default skins.

- Fork: Forks the skin for mobile, tablet and SPA platforms.
- Add to Theme: Adds the skin to an existing library theme.

## 3.5 Templates Tab

Templates provide an easy way to display the same information across the forms, segments and maps.

Following are the templates available for both the mobile and tablet devices:

1. [Headers](#)
2. [Footers](#)
3. [Segments](#)
4. [Maps](#)
5. [Gridcalendars](#)

### 3.5.1 Headers

Header is a section of the form that is docked at the top of the form, and can be reused across the forms.

To create a header template for a mobile applications, follow these steps:

1. Click the **Templates** tab from the **Project Explorer**.
2. Expand **Mobile** and rest the pointer on the **Headers** to display a down-arrow. Click this down-arrow, and then click **New Template**. If required, rename the template.
3. Drag and drop an HBox, a ScrollBox, or a FlexContainer onto a template.

**Note:** You cannot place widgets directly on a template. You need to first add a container widget and then place other widgets inside this container widget. Use either an HBox or a ScrollBox container widget to create a template for a VBox Form. Use FlexContainer container to create a template for a Flex Form.

4. Drag and drop the required widgets onto the HBox or ScrollBox. Set the properties of these widgets and save the header.

### 3.5.2 Footers

Footer is a section of the form that is docked at the bottom of the form and can be reused across the forms.

To create a footer template for mobile applications, follow these steps:

1. Click the **Templates** tab from the **Project Explorer**.
2. Expand **Mobile** and rest the pointer on the **Footers** to display a down-arrow. Click this down-arrow and then click **New Template**. If required, rename the template.
3. Drag and drop an HBox or a ScrollBox onto the template.

**Note:** You cannot place widgets directly on a template. You need to first add a container widget and then place other widgets inside this container widget. Use either an HBox or a ScrollBox container widget to create a template for a VBox Form. Use FlexContainer container to create a template for a Flex Form.

4. Drag and drop the required widgets onto the HBox or ScrollBox. Set the properties of these widgets and save the footer.

### 3.5.3 Inserting Header and Footer to a Form

After creating header and footer templates, you need to insert them in a form, so that they are visible.

To insert a header and a footer, follow these steps:

1. From the **Project** tab of **Project Explorer**, expand either **Mobile** or **Tablet** (depending on whether you have created the header and footer for a mobile or a tablet device).
2. Expand **Forms** and click the form to which you want to insert the header and footer.

3. From the form properties, click **Form2** tab.
4. For inserting header,
  - a. Click the **Edit** button against the **headers** field to open the **Headers** dialog box.
  - b. From the available list, choose a header and then click **OK**.
5. For inserting footer,
  - a. Click the **Edit** button against the **footers** field to open the **Footers** dialog box.
  - b. From the available list, choose a footer and then click **OK**.

#### 3.5.4 Segments

Segment template enables you to define a template for section headers and rows of the segment. This is primarily useful for achieving common look and feel of section headers along with few widgets added as part of section header of a segment.

To create a segment template for mobile, follow these steps:

1. Click the **Templates** tab from the **Project Explorer**.
2. Expand **Mobile** and rest the pointer on the **Segments** to display a down-arrow. Click this down-arrow and then click **New Template**. If required, rename the template.
3. Drag and drop an HBox onto a template.

**Note:** You cannot place widgets directly on a template. You need to first add an HBox widget, and then place other widgets inside this HBox widget.

4. Drag and drop the required widgets onto the HBox or ScrollBox. Set the properties of these widgets and save the Segment template.

### 3.5.5 Inserting a Segment Template to a Segment Widget

To insert a Segment template to a Segment widget, follow these steps:

1. From the **Project** tab of **Project Explorer**, expand either **Mobile** or **Tablet** (depending on whether you have created the segment template for a mobile or tablet device).
2. Expand **Forms** and navigate to the form that contains a segment widget, and to which you wanted to add the segment template.
3. Expand this form and click the segment widget.
4. From the Segment properties, click **Segment** tab.
5. The segment templates are available under **General > Row Template and Section Header Template** list. You may choose to use the template as either Row Template or Section Header Template for the **Segment** widget.

### 3.5.6 Maps

Map Templates are used to set a template for a map callout.

To create a map template for mobile or tablet applications, follow these steps:

1. Click the **Templates** tab from the **Project Explorer**.
2. Expand **Mobile** or **Tablet** (depending on which devices you want to create a map template) and rest the pointer on the **Maps** to display a down-arrow. Click this down-arrow and then click **New Template**. If required, rename the template.
3. Drag and drop an HBox onto a template.

**Note:** You cannot place widgets directly on a template. You need to first add an HBox widget , and then place other widgets inside this HBox widget.

4. Drag and drop the required widgets onto the HBox or ScrollBox. Set the properties of these widgets and save the Map template.

### 3.5.7 Inserting a Map Template to a Map Widget

To insert a Map template to a Map widget, follow these steps:

1. From the **Project** tab of **Project Explorer**, expand either **Mobile** or **Tablet** (depending on whether you have created the map template for a mobile or tablet device).
2. Expand **Forms** and navigate to the form that contains a map widget, and to which you want to add the map template.
3. Expand this form and click the map widget.
4. From the Map properties, click **Map** tab.
5. Click **Edit** button against **Callout Template** field. This results in opening **calloutTemplate** dialog box.
6. From the available list, choose a map and click **OK**.

### 3.5.8 Gridcalendars

Gridcalendar template enables you to define a template for Calendar Day cell. Only one template can be used for each Calendar. The gridcalendar templates are used:

- to define a Calendar Day cell with custom look and feel.
- to achieve the behavior of having widgets such as an Image and a label for a Calendar Day cell.
- to perform an action on the event of an onclick of a Calendar Day cell.

To create a Gridcalendar template for mobile or tablet applications, follow these steps:

1. Click the **Templates** tab from the **Project Explorer**.
2. Expand **Mobile** or **Tablet** (depending on which devices you want to create a Gridcalendar template) and rest the pointer on the **Gridcalendar** to display a down-arrow. Click this down-arrow, and then click **New Template**. If required, rename the template.
3. Drag and drop an HBox onto a template.

**Note:** You cannot place widgets directly on a template. You need to first add an HBox widget , and then place other widgets inside this HBox widget.

4. Drag and drop the required widgets onto the HBox or ScrollBox. Set the properties of these widgets and save the Map template.

## 3.6 Assets Tab

Assets are a repository of Images and Fonts that are copied into a project. Click the **Assets** tab from the **Project Explorer** to view a list of Images and Fonts.

### 3.6.1 Images

Before you can use an image in a project, you need to copy the image to an appropriate project subfolder.

For example, if you wanted an image to be used only for a tablet device and in native mode, you need to copy this image to the following folder:

<Konyworkspace>/<project>/resources/tablet/native.

Following are the subfolders under which you can store the images:

1. **Common:** The images stored here are available across all the channels (mobile and tablet applications) and modes (native and web).
2. **Mobile:** Images stored here are available for all Mobile devices.
  - a. Common: The images stored here are available for both native and web applications.
  - b. Native: The images stored here are available only for building native applications.
  - c. Web: The images stored here are available only for web applications.
3. **Tablet:** Images stored here are available for all tablet devices.
  - a. Common: The images stored here are available for both native and web applications.
  - b. Native: The images stored here are available only for building native applications.
  - c. Web: The images stored here are available only for web applications.

**Note:** Click refresh () to ensure that new images and fonts that are copied into the project folder are loaded into the project.

### 3.6.2 Retina Display on iPhone

iPhone 4.0 has a high resolution screen (960x640) with a density of 326 ppi. When the screen density is more than 300 ppi, images, videos, text, and other objects appear very smooth with high clarity on the screen. This high resolution screen enables a very clear view of all the objects on the screen, which is similar to high quality printed output. Apple has named this technology Retina Display.

All the iPhone applications are developed for 320x480 resolution. The images used in the application are automatically scaled up to suit the high resolution screen. When the images are scaled up for a high resolution screen, the clarity of images is distorted. The Retina Display feature ensures excellent image clarity by preventing automatic scaling.

To support images within an application for Retina Display, follow these steps:

1. Create an image twice the pixel size of the original image. For more information, see the example given below.
2. Suffix the image name with @2x.

For example, if you have a company logo image `logo.png`, 50X50 pixels, which you want to use for a high resolution screen, create the same image with a 100x100 pixel size and name it `logo@2x.png`.

**Note:** Place the images created for the high resolution screen in the Resources folder along with the other images. The application will automatically pick up the high resolution images for devices that support Retina Display.

### 3.6.3 Fonts

Kony Visualizer supports a large number of font families and most of these fonts are available to you by default.

**Note:** To view the available fonts for a project, from the **Properties** pane of any widget, click the **Skin** tab, and under **Fonts**, click the **Edit** button next to the Font Family.

You can import custom fonts into a project by copying the fonts (of type TTF) to an appropriate platform font's subfolder. Ensure that you do not change the name of the custom font while copying the font to a project.

For example, if you wanted a font family to be available only for Android devices, then copy that font family to the following location:

<Konyworkspace>/<project>/resources/fonts/Android.

Below are list of subfolders under which you can store the fonts:

1. Android: Fonts stored here are available for all Android devices.
2. Android Tablet: Fonts stored here are available for all Android tablet devices.
3. IPad: Fonts stored here are available for all iPad devices.
4. IPhone: Fonts stored here are available for all iPhone devices.
5. SPA Android: Fonts stored here are available for all SPA Android devices.
6. SPA Android Tablet: Fonts stored here are available for all SPA Android Tablet devices.
7. SPA IPad: Fonts stored here are available for all SPA iPad devices.
8. SPA iPhone: Fonts stored here are available for all SPA iPhone devices.

**Note:** After copying a font family to a project folder, click refresh () in the **Assets** tab to ensure that the new font family is loaded into the project.

### Fonts Limitations

1. Open Type Fonts (OTF) are not supported by Visualizer. You may, however, convert the OTF to True Type Fonts (TTF) and use them in Visualizer.

2. Faux font weight (normal and bold) is not supported by iOS. You need to use the original TTF (such as Helvetica Bold and Helvetica Light) to produce the desired font weight.
3. Strikethrough of fonts is not supported.
4. Functional Preview does not support custom fonts.

## 3.7 Search

The Search option allows to search for the elements within Project, Skins, or Templates tab.

The following are the searchable elements from each of the tabs:

- From the **Projects** tab, you can search for forms and pop-ups.
- From the **Skins** tab, you can search for default and user-defined skins.
- From the **Templates** tab, you can search for Headers, Footers, Segments, Maps, and Gridcalendars templates.

To initiate search, follow these steps:

1. Click on a tab. For example, **Project**.
2. Click the search icon.
3. In the Search box, as you type, relevant results are displayed.
4. Click on a search result.

## 3.8 Filters

Filters allow you to view elements of similar type thereby hiding the remaining elements. For example, when you filter for Headers in the Templates tab, only header templates of the Mobile and the Tablet channels are displayed. You can apply filters for the following tabs:

- [Project](#)
- [Skins](#)
- [Templates](#)
- [Assets](#)

### 3.8.0.1 Project

Project filter allows you to filter for Forms or pop-ups.

To apply the filters, click the **Project** tab from **Project Explorer**, and then click the down-arrow next to the **Assets** tab. You can select one of the below filters:

- **Show All**: Displays all the project elements.
- **Forms**: Displays only **Forms**.
- **Popups**: Displays only **Pop-ups**.

### 3.8.0.2 Skins

Skins filter allows you to filter for widget specific skins.

To apply a filter, click the **Skins** tab from **Project Explorer**, and then click the down-arrow next to the **Assets** tab. Select a widget to filter for its skins. You can select one of the below filters:

- **All Skins**: Displays all the available skins.
- **Individual Widget Skins**: Click any of the individual widgets (such as HBox, Button or Slider) to display the skins of the selected widget.

### 3.8.0.3 Templates

Template filters allows you to filter for specific type of templates.

To apply a filter, click the **Templates** tab from **Project Explorer**, and then click the down-arrow next to the **Assets** tab. You can select one of the below filters:

- **Show All:** Displays all the templates.
- **Headers:** Displays only **Header** templates.
- **Footers:** Displays only **Footer** templates.
- **Segments:** Displays only **Segment** templates.
- **Maps:** Displays only Map templates.
- **Gridcalendars:** Displays all the grid calendar templates that are available for the project.

#### 3.8.0.4 Assets

Assets filter allows you to filter for either the images or fonts that are copied into a project.

To apply a filter, click the **Assets** tab from **Project Explorer** and then click the down-arrow next to the **Assets** tab. You can apply one of the below filters:

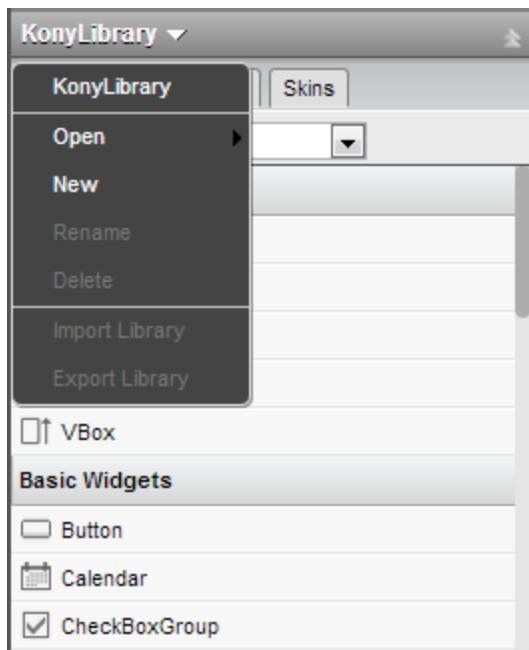
- **Show All:** Displays all the assets.
- **Images:** Displays all the images that are available for the project.
- **Fonts:** Displays the fonts that are available for the project.

## 4. Library Explorer

Library Explorer contains libraries that are repositories of [widgets](#), [collections](#) and [skins](#) and include:

- Kony Library: Contains predefined set of widgets and skins. You may modify the skins of this library and save them in a user-defined library.
- User-defined Libraries: These libraries are created by the user and are stored at the Visualizer application level and are available across all projects. (Example: A library created in project X is available for project Y provided these projects are created in the same workspace.)
- Imported Libraries: These libraries are imported into the project from other workspace. Once a library is imported, it is stored at the Visualizer application level and available for all the projects in the same workspace.

## 4.1 Library Menu



Click the down-arrow next to the **KonyLibrary** in the **Library Explorer** to open the Library Menu Items. Following are Library Menu items:

1. **KonyLibrary**: Opens KonyLibrary. This is the default library and available across all the projects.

**Note:** You cannot delete, rename, import, or export KonyLibrary.
2. **Open**: Hover over this menu item to display the list of available libraries within the same workspace. Click the desired library to open it.
3. **New**: Opens **Create New Library** dialog box. Type a unique name to create a new library.
4. **Rename**: Enables you to type a new name for the library.
5. **Delete**: Deletes the library.
6. **Import Library**: Displays **Open** dialog box. Locate and select the library to be imported.

7. **Export Library:** Displays **Save As** dialog box. Choose the folder for saving the file. A default file name is assigned to the library. You may choose to rename the library before exporting.

**Note:** If the project is not saved, system will prompt you to save the project before you can export a library.

## 4.2 Widgets

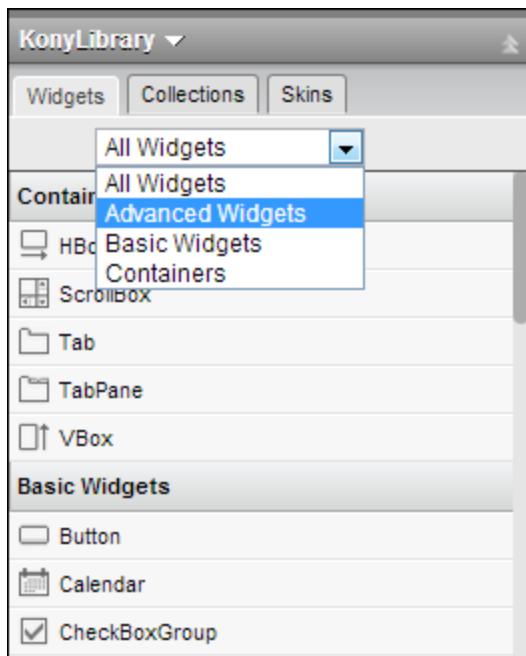
Widgets are small chunks of code that are re-used without additional compilation. Some widgets support interaction with the user (for example: Button, Label, Checkbox, and so on). Others act as containers that group the widgets added to them (for example: Form, Row, Column, and so on).

The Widgets tab lists all the predefined widgets. These widgets are available across all the projects.

The widgets are grouped together based on their common behavior and characteristics. These widgets are categorized as:

1. [Containers](#)
2. [Basic Widgets](#)
3. [Advanced Widgets](#)

You can filter for specific type of widgets from the filter list shown below.



## 4.2.1 Containers

These are a set of widgets that hold other widgets within them. The following is a list of Container Widgets:

Following widgets are classified as Container Widgets:

- [HBox](#)
- [ScrollBox](#)
- [TabPane](#)
- [VBox](#)

## 4.2.2 Basic Widgets

These widgets are components that act independently of the Container Widgets. The following is a list of Basic Widgets:

- [Button](#)
- [Calendar](#)
- [CheckBoxGroup](#)
- [ComboBox](#)
- [DataGrid](#)
- [Image2](#)
- [Label](#)
- [Line](#)
- [Link](#)
- [ListBox](#)
- [RadioButtonGroup](#)
- [RichText](#)
- [Slider](#)
- [TextArea2](#)
- [TextBox2](#)

### 4.2.3 Advanced Widgets

These widgets help build some of the complex actions and include:

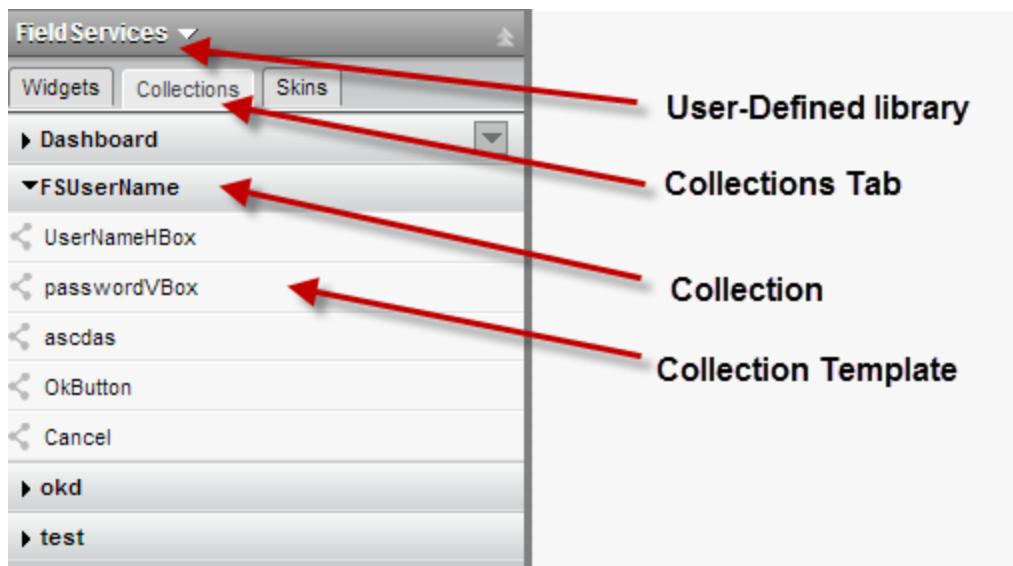
- [Browser](#)
- [Camera](#)
- [Hz Image Strip2](#)
- [Image Gallery2](#)
- [Map](#)
- [Phone](#)
- [PickerView](#)
- [Segment2](#)
- [Switch](#)
- Video

## 4.3 Collections

Collection consists of user-defined widget templates.

The listing in the Collections tab is available only for the user-defined collections that are added to a selected library.

Below image illustrates the Collections tab.



### 4.3.1 Collection

To expand or collapse a collection, click the triangle located at the left the collection.

To delete a collection, hover over the collection to display a triangle to the right of the collection. Click the triangle, and then click **Delete**.

#### 4.3.1.1 Widgets Template

Collection saves a single widget or a group of widgets (that is, widget and its child widgets) as a template. You can perform following actions on templates:

- **Rename:** Click **Rename** to rename a template, from the template contextual menu.
- **Delete:** Click **Delete** to delete a template, from the template contextual menu.

## 4.4 Library Skins

The **Skins** tab of the **Library Explorer** consists of skin themes and can include:

- Predefined skins: By default, these skins are available across all the projects.
- User-defined skins: These skins are created by the user and can be accessed by all projects that are in the same local drive.
- Imported Library Skins: When a library is imported into a project all the skins from the imported library will be available for the project. These libraries are saved and are treated as user-defined skins.

When you select **KonyLibrary** from the **Library Explorer**, the **Skins** tab contains a default theme provided by the Kony Visualizer.

When you select a user-defined library from the **Library Explorer**, the **Skins** tab contains user-defined themes.

Each theme contains widget folders to hold the skins for that particular widget.

While a user-defined skin can be deleted by right-clicking a skin and then selecting **Delete**, the default themes (and consequently skins) cannot be deleted.

Similarly, you can rename a user-defined theme or a skin. But you cannot rename the default **KonyLibrary** themes or its skins.

To apply a skin from the **Skins** tab, select a widget from its project (or from App Canvas) and then from the **Skins** tab of **Library Explorer**, navigate to the widget folder, right-click the skin and then click **Apply**.

**Note:** You can also apply a skin by clicking a skin in the Library Explorer.

## 5. Working with Widgets

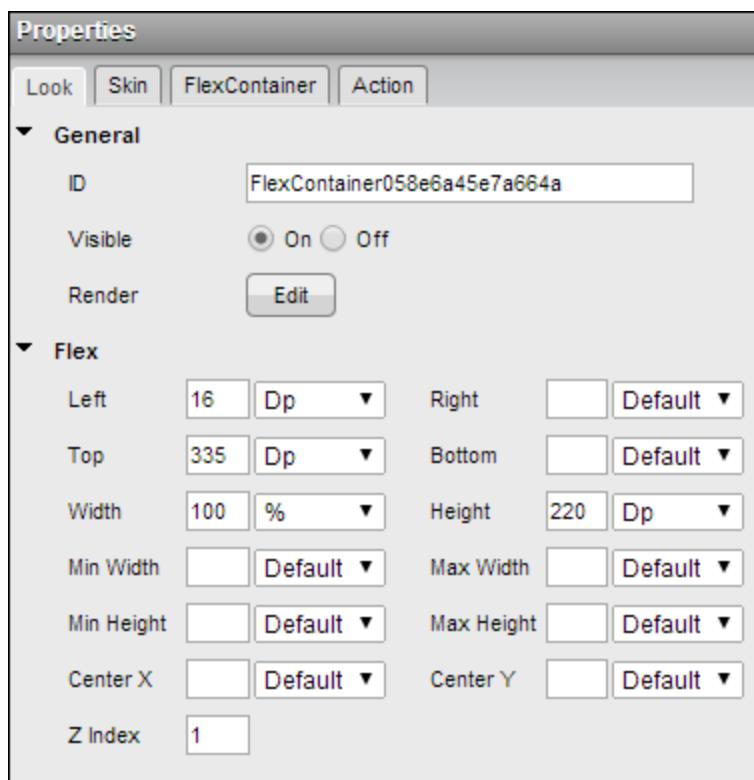
Widgets are small chunks of code that are re-used without additional compilation. Some widgets support interaction with the user such as: Button, Label, and Checkbox. Others act as containers that group the widgets added to them such as HBox, VBox, TabPane and ScrollBox.

## 5.1 FlexContainer

A FlexContainer widget is used to layout widgets in a user specified orientation. It can contain any number of widgets.

### 5.1.1 Look

This section details the **Properties > Look** tab of a FlexContainer widget. Here, you can define various widget properties that are explained in this section.



To specify the look properties of a Flex Container widget, refer section: [Look](#).

### 5.1.2 Skin

Skin defines properties such as background color, borders and shadows for a widget. They also define the widget font (applicable only if text is displayed on a widget.)

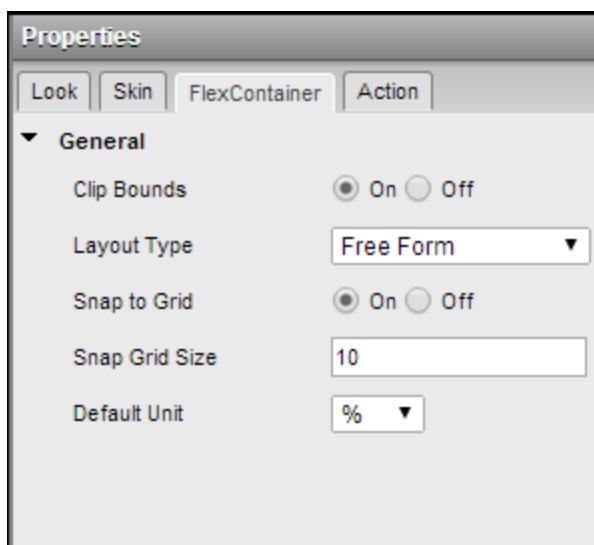
For a FlexContainer widget, you can apply a skin for the following states:

- Normal: Specifies the look and feel of the widget when not in focus.
  - Focus: Specifies the skin that is applied when there is focus on a widget.
  - Blocked UI: Specifies the skin that must be used to block the interface until the action in progress (for example, a service call) is completed.

**Note:** Blocked UI is available only for SPA platforms.

Refer to section [Working with Skins](#) to know more about adding a skin to a widget.

### 5.1.3 Platform Specific Properties



### 5.1.3.1 Clip Bounds

Specifies whether to clip the child widgets when they go out of boundaries or not.

### 5.1.3.2 Layout Type

Specifies if the arrangement of the widgets either in horizontal or vertical direction.

## Default: Vertical

Following are the available options:

- Free Form: The navigation happens in both directions.
- Flow Vertical : The navigation happens in vertical direction.
- Flow Horizontal : The navigation happens in horizontal direction.

#### 5.1.3.3 Snap to Grid

A widget aligns to the nearest intersection of lines in the grid, or other widgets.

- To allow snapping the widgets to a grid, click **On**.
- To disable snapping the widgets to a grid, click **Off**.

#### 5.1.3.4 Snap Grid Size

Specifies the value of the grid size. This option is available only when you enable **Snap to Grid** property.

#### 5.1.3.5 Default Unit

Specifies the default unit to be used for interpretation of numbers with no qualifiers when passed to layout properties.

Following are the available options:

- dp: Specifies the values in terms of device independent pixels.
- px: Specifies the values in terms of device hardware pixels.
- %: Specifies the values in percentage relative to the parent dimensions.

#### 5.1.4 Action

A Flex Container has the following actions associated with it:

- onTouchStart: An event callback is invoked by the platform when the user touches the touch surface. This event is invoked asynchronously.

- **onTouchMove:** An event callback is invoked by the platform when the touch moves on the touch surface continuously until movement ends. This event is invoked asynchronously.
- **onTouchEnd:** An event callback is invoked by the platform when the user touch is released from the touch surface. This event is invoked asynchronously.

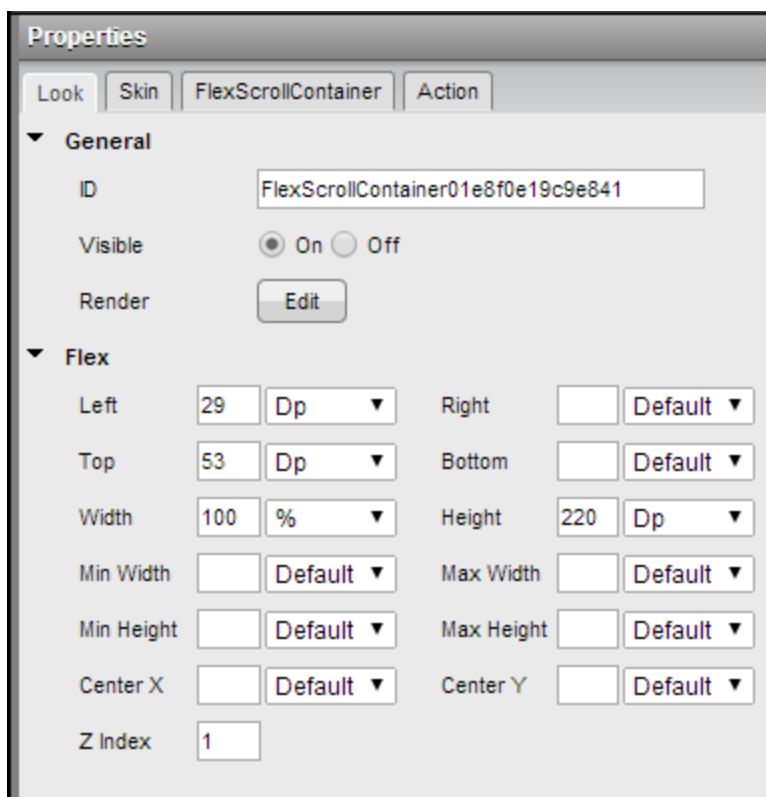
For more information on using the above action, refer to section [Action Editor](#).

## 5.2 FlexScrollContainer

A FlexScrollContainer is a scrollable container which allows you to scroll the content in horizontal or vertical direction on a FlexForm or FlexContainer. It can contain any number of widgets within it.

### 5.2.1 Look

This section details the **Properties > Look** tab of a **FlexScrollContainer** widget. Here, you can define various widget properties that are explained in this section.



To specify the look properties of an FlexScrollContainer widget, refer section: [Look](#).

### 5.2.2 Skin

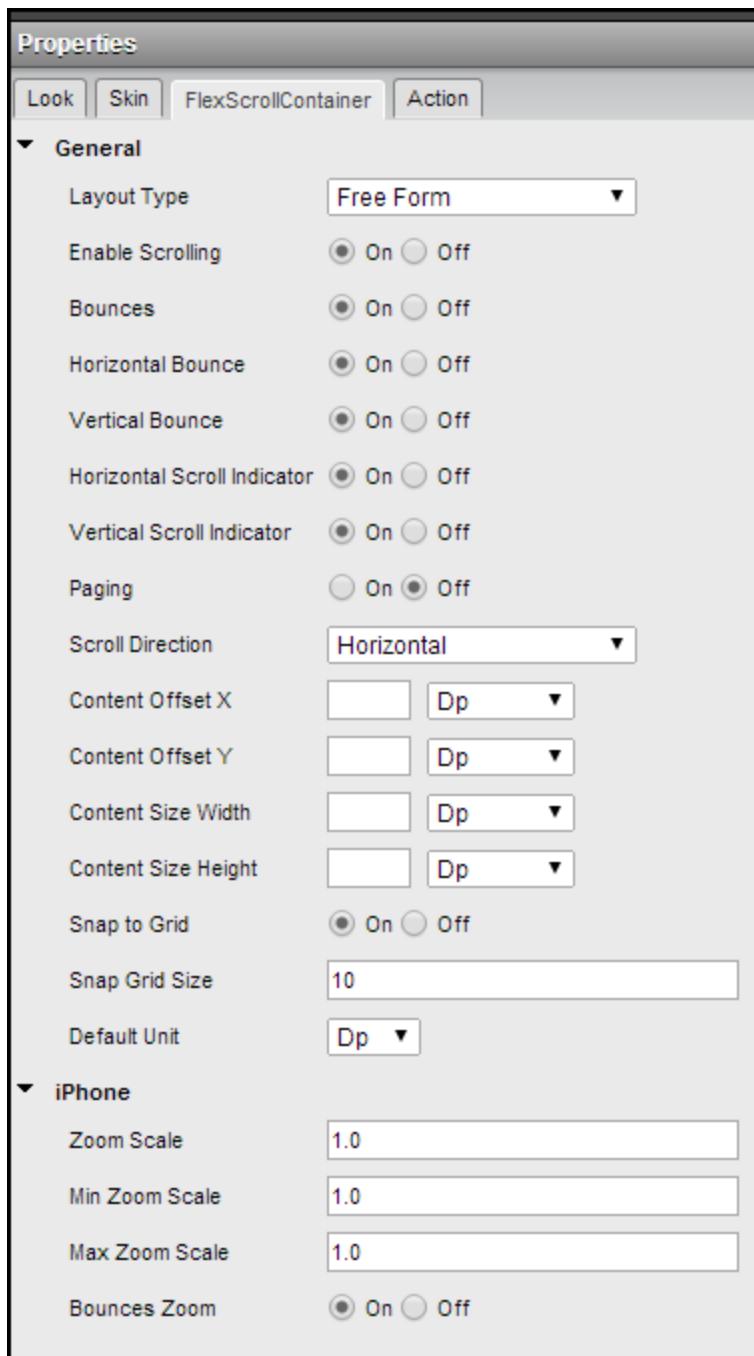
Skin defines properties such as background color, borders and shadows for a widget. They also define the widget font (applicable only if text is displayed on a widget.)

For a FlexScrollContainer, you can apply a skin for the following states:

- Normal: Specifies the look and feel of the widget when not in focus.
- Focus: Specifies the skin that is applied when there is focus on a widget.
- Blocked UI: Specifies the skin that must be used to block the interface until the action in progress (for example, a service call) is completed.

**Note:** Blocked UI is available only on SPA platforms.

### 5.2.3 Platform Specific Properties



### 5.2.3.1 Layout Type

Specifies if the arrangement of the widgets either in horizontal or vertical direction.

Default: Vertical

Following are the available options:

- Free Form: The navigation happens in both directions.
- Flow Vertical : The navigation happens in vertical direction.
- Flow Horizontal : The navigation happens in horizontal direction.

### 5.2.3.2 Enable Scrolling

Specifies whether the scrolling is enabled on the container or not.

- To enable scrolling, click **On**.
- To disable scrolling, click **Off**.

### 5.2.3.3 Bounces

Specifies whether the scroll bounce is enabled or disabled.

- To enable scroll bounce, click **On**.
- To disable scroll bounce, click **Off**.

### 5.2.3.4 Horizontal Bounce

Specifies whether the scroll bounce is enabled or disabled in the horizontal direction.

- To enable horizontal scroll bounce, click **On**.
- To disable horizontal scroll bounce, click **Off**.

### 5.2.3.5 Vertical Bounce

Specifies whether the scroll bounce is enabled or disabled in the vertical direction.

- To enable vertical scroll bounce, click **On**.
- To disable vertical scroll bounce, click **Off**.

#### 5.2.3.6 Horizontal Scroll Indicator

Specifies whether the scroll indicator to be shown or not in the horizontal direction.

**Note:** Scroll Indicators may not be shown permanently. But depending on the platform scroll indicators may appear only during scrolling.

#### 5.2.3.7 Vertical Scroll Indicator

Specifies whether the scroll indicator to be shown or not in the horizontal direction.

**Note:** Scroll Indicators may not be shown permanently. But depending on the platform scroll indicators may appear only during scrolling.

#### 5.2.3.8 Paging

Specifies the whether the paging is enabled for the scroll container. If this property is set to true, the scroll view stops on multiples of the scroll view's bounds when the user scrolls.

#### 5.2.3.9 Scroll Direction

Specifies the direction in which the FlexScrollForm should scroll. This property is supported only when the

Following are the available options:

- Horizontal: Specifies the Flex Scroll Form to scroll in horizontal direction.
- Vertical: Specifies the Flex Scroll Form to scroll in vertical direction.
- Both: Specifies the Flex Scroll Form to scroll in both the horizontal and vertical directions.

#### 5.2.3.10 Content Offset X

Specifies the x coordinates of the top-left of the scrollable region. When the values are set, the scroll container scrolls even if the scrolling is disabled. This will always returns the value that developer has set but never reflects the actual computed offset.

#### 5.2.3.11 Content Offset Y

Specifies the Y coordinates of the top-left of the scrollable region. When the values are set, the scroll container scrolls even if the scrolling is disabled. This will always returns the value that developer has set but never reflects the actual computed offset.

#### 5.2.3.12 Content Size Width

Specifies the width of the container to accommodate all the widgets placed in it. This will returns the values that developer has set, but never reflects the actual computed content size.

#### 5.2.3.13 Content Size Height

Specifies the height of the container to accommodate all the widgets placed in it. This will returns the values that developer has set, but never reflects the actual computed content size.

#### 5.2.3.14 Snap to Grid

A widget aligns to the nearest intersection of lines in the grid, or other widgets.

- To allow snapping the widgets to a grid, click **On**.
- To disable snapping the widgets to a grid, click **Off**.

#### 5.2.3.15 Snap Grid Size

Specifies the value of the grid size. This option is available only when you enable **Snap to Grid** property.

### 5.2.3.16 Default Unit

Specifies the default unit to be used for interpretation of numbers with no qualifiers when passed to layout properties.

Following are the available options:

- dp: Specifies the values in terms of device independent pixels.
- px: Specifies the values in terms of device hardware pixels.
- %: Specifies the values in percentage relative to the parent dimensions.

### 5.2.3.17 Zoom Scale

Specifies the current scale factor applied to the form content.

Default:1

**Note:** This property is specific to the iOS platform.

### 5.2.3.18 Min Zoom Scale

Specifies the minimum zoom scale factor that can be applied to the form.

Default: 1

**Note:** This property is specific to the iOS platform.

### 5.2.3.19 Max Zoom Scale

Specifies the maximum zoom scale factor that can be applied to the form.

Default: 1

**Note:** This property is specific to the iOS platform.

### 5.2.3.20 Bounces Zoom

Specifies whether the scroll view animates the content scaling when the scaling exceeds the maximum or minimum limits. If the value is set to true, and zooming exceeds either the minimum or maximum limits for scaling, the scroll view temporarily animates the content scaling just past these limits before returning to them. If the property is set to false, zooming stops immediately as it reaches scaling limits.

**Note:** This property is specific to the iOS platform.

## 5.2.4 Actions

A FleScrollContainer has the following actions associated with it:

### 5.2.4.1 onScrollStart

An event callback is invoked by the platform when the user starts scrolling the content. This event is invoked asynchronously.

### 5.2.4.2 onScrollTouchReleased

An event callback is invoked by the platform when the user touch is released from the touch surface. This event is invoked asynchronously.

### 5.2.4.3 onScrolling

An event callback is invoked by the platform when the scrolling is in progress. This event is invoked asynchronously.

### 5.2.4.4 onDecelerationStarted

An event callback is invoked by the platform when the user stops scrolling but the content still moves before the content actually stops. This event is invoked asynchronously.

#### 5.2.4.5 onScrollEnd

An event callback is invoked by the platform when the scrolling is ended. This event is invoked asynchronously.

#### 5.2.4.6 onTouchStart

An event callback is invoked by the platform when the user touches the touch surface. This event is invoked asynchronously.

#### 5.2.4.7 onTouchMove

An event callback is invoked by the platform when the touch moves on the touch surface continuously until movement ends. This event is invoked asynchronously.

#### 5.2.4.8 onTouchEnd

An event callback is invoked by the platform when the user touch is released from the touch surface. This event is invoked asynchronously.

#### 5.2.4.9 widgetToZoom

An event callback is invoked by the platform to return one of the child widgets of source to zoom. The returning source itself makes the complete scroll container zoomable. If a null value is returned then the container does not zoom. This event is invoked asynchronously.

**Note:** This action is specific to the iOS platform

#### 5.2.4.10 onZoomStart

An event callback is invoked by the platform when the container is about to zoom. This event is invoked asynchronously.

**Note:** This action is specific to the iOS platform

#### 5.2.4.11 onZooming

An event callback is invoked by the platform when the container is zooming. This event is invoked asynchronously.

**Note:** This action is specific to the iOS platform

#### 5.2.4.12 onZoomEnd

An event callback is invoked by the platform when the zooming has ended. This event is invoked asynchronously.

**Note:** This action is specific to the iOS platform

## 5.3 HBox

An **HBox** is used to layout widgets in a single horizontal orientation. It can contain any number of widgets. By default, **HBox** occupies the entire available width of its parent widget. However, due to form size limitations, it is advisable not to place many widgets in an HBox.

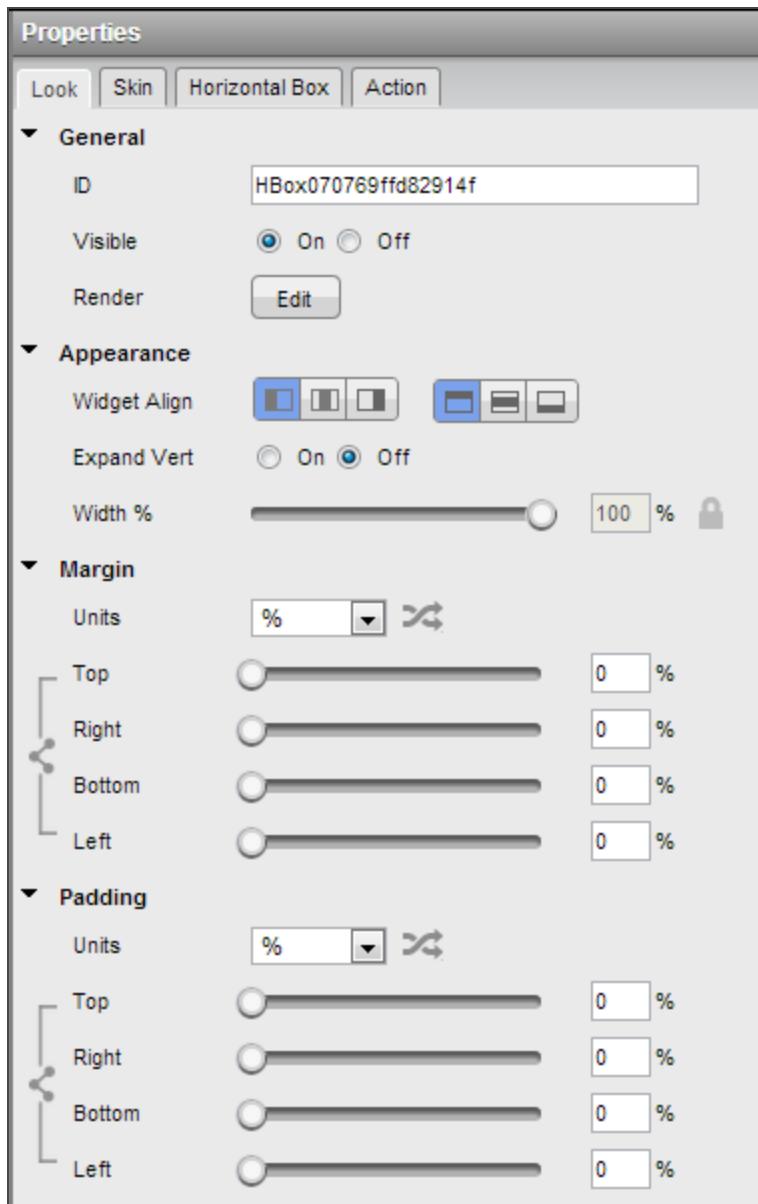
### 5.3.1 Important Considerations

The following are important considerations for an HBox Widget.

- An HBox widget is available only for a VBox Form.
- An HBox can be placed within a Form, VBox, TabPane, a Segment (only if orientation is vertical), or a ScrollBox (only if the orientation is vertical).
- If an **HBox** contains multiple child widgets with varying heights, the height of the child widget with the maximum height is set as the height of the **HBox**.
- If a background image is specified (as part of the normal skin) for the **HBox**, and if the heights of the child widgets are less than the height of the background image, then height of the image is set as the height of the **HBox**.

### 5.3.2 Look

This section details the **Properties > Look** tab of an **HBox** widget.



To specify the look properties of an HBox widget, refer section: [Look](#).

### 5.3.3 Skin

Skin defines properties such as background color, borders and shadows for a widget. They also define the widget font (applicable only if text is displayed on a widget.)

For the HBox widget, you can apply a skin for the following states:

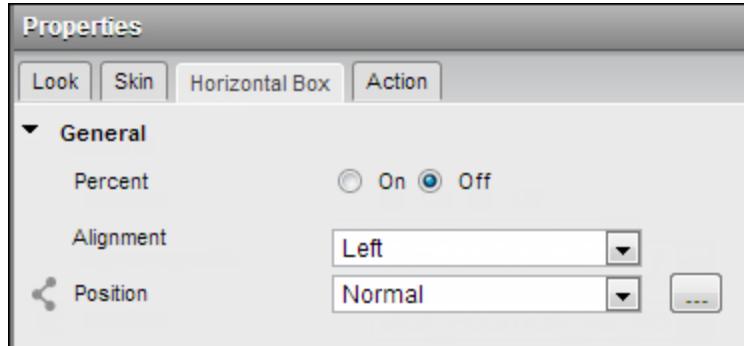
- Normal: Specifies the look and feel of the widget when not in focus.
- Focus: Specifies the skin that is applied when there is focus on a widget.
- Blocked UI: Specifies the skin that must be used to block the interface until the action in progress (for example, a service call) is completed.

**Note:** Blocked UI is available only for SPA platforms.

Refer to section [Working with Skins](#) to know more about adding a skin to a widget.

#### 5.3.4 Platform Specific Properties

**Note:** In this section, the properties that can be forked are identified by an icon ( located to the left of the property. For more details, refer to section [Forking a Widget property](#).



##### 5.3.4.1 Percent

Specifies whether the child widget's weight must be considered during the layout.

**Default:** Percent is enabled.

##### 5.3.4.2 Alignment

Specifies the direction in which the widgets are laid out. This property is displayed only when **Percent** property is disabled.. .

**layoutAlignment** list allows you to set the direction in which the child widgets are laid out within an HBox.

**Default:** Left

Following are the available options:

- Left: The widgets placed inside a box are aligned left.
- Right: The widgets placed inside a box are aligned right.
- Center: The widgets placed inside a box are aligned center.

#### 5.3.4.3 Position

Specifies the position of an HBox as header or footer of the form, popup, segment or map. When you set this property, the HBox is docked along with header or footer.

**Note:** This property is applicable for immediate child widgets placed on a Form.

**Default:** Normal

Following are the options available:

- Normal: The original position of the box is retained.
- Header: The box is fixed at the top of the form.
- Footer: The box is fixed at the bottom of the form.
- Fullscreen Segment Header: This option is useful to fix the position of the box to the top of the form when the box is placed inside a Segment with screenLevelWidget set to true. This property is treated as [normal<sup>1</sup>](#) when a box is not placed in a segment.

---

<sup>1</sup>BOX\_POSITION\_AS\_NORMAL where the original position of the box is retained.

- Fullscreen Segment Footer: This option is useful to fix the position of the box to the bottom of the form when a box is placed inside a Segment with screenLevelWidget set to true. This property is treated as **normal<sup>1</sup>** when a box is not placed in a segment.

The selected option becomes the default **position** value for all the platforms. You can choose to provide a different value for a platform by forking **position** property. Refer to the section [Forking a Widget Property](#) for more details.

**Note:** Position property is not applicable for SPA platform.

### 5.3.5 Actions

HBox widget has the following action associated with it:

- onClick: This action is invoked by the platform when the user performs a click action on the widget.

For more information on using the above action, refer to section [Action Editor](#).

### 5.3.6 Placement inside a Widget

Below table list the widgets inside which an HBox can be directly added.

| Widget     | HBox placement inside a widget                           |
|------------|--|
| Form       | Yes  |
| HBox       | No   |
| VBox       | Yes  |
| Scroll Box | Horizontal Orientation - No<br>Vertical Orientation- Yes |

<sup>1</sup>BOX\_POSITION\_AS\_NORMAL where the original position of the box is retained.

|          |  |
|----------|--|
| Widget   | HBox placement inside a widget                           |
| Tab      | Yes  |
| Segment  | Horizontal Orientation - No<br>Vertical Orientation- Yes |
| Popup    | Yes  |
| Template | Header- Yes<br>Footer- Yes                               |

## 5.4 ScrollBox

A ScrollBox is a scrollable container which allows you to scroll the content within horizontally and vertically. A ScrollBox can contain any widget except a Tab pane. It can contain any number of widgets within it.

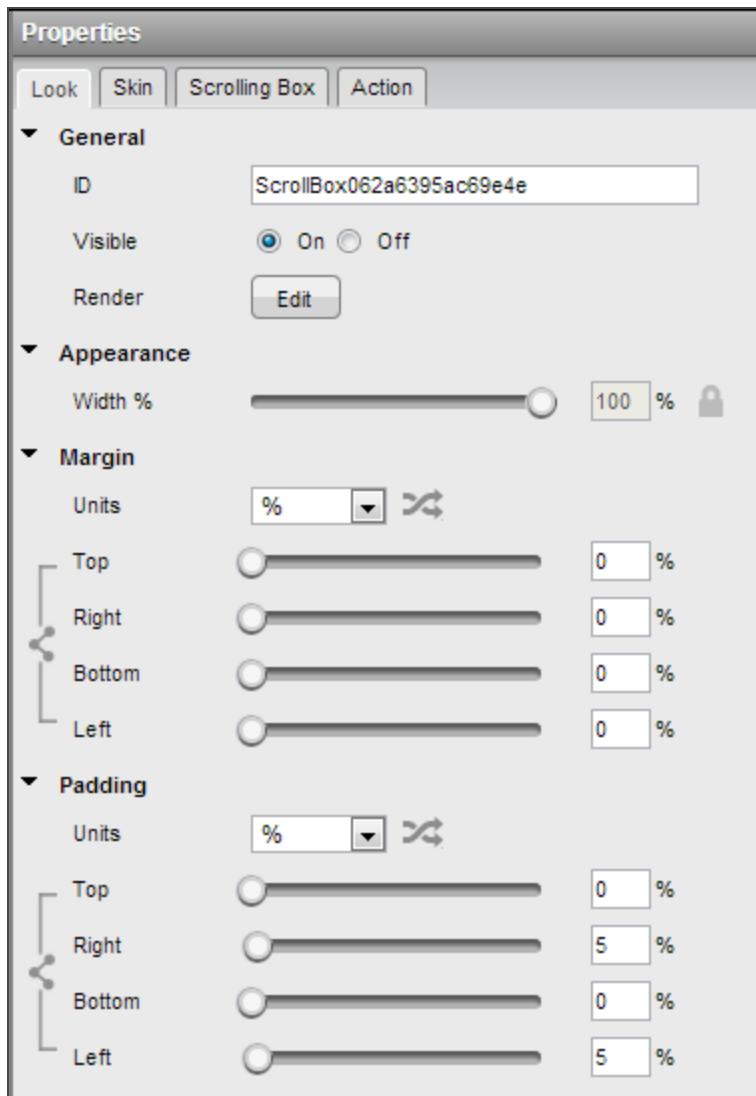
### 5.4.1 Important Considerations

The following is an important consideration of ScrollBox widget:

- A ScrollBox widget is available only for a VBox Form.
- For a good user experience, you must place the ScrollBox using a percentage layout. In a non-percentage layout the width of the ScrollBox varies across platforms.

### 5.4.2 Look

This section details the **Properties > Look** tab of a ScrollBox widget.



To specify the look properties of a ScrollBox widget, refer section: [Look](#).

#### 5.4.3 Skin

Skin defines properties such as background color, borders and shadows for a widget. They also, define the widget font (applicable only if text is displayed on a widget.)

For a ScrollBox widget, you can apply a skin for the following states:

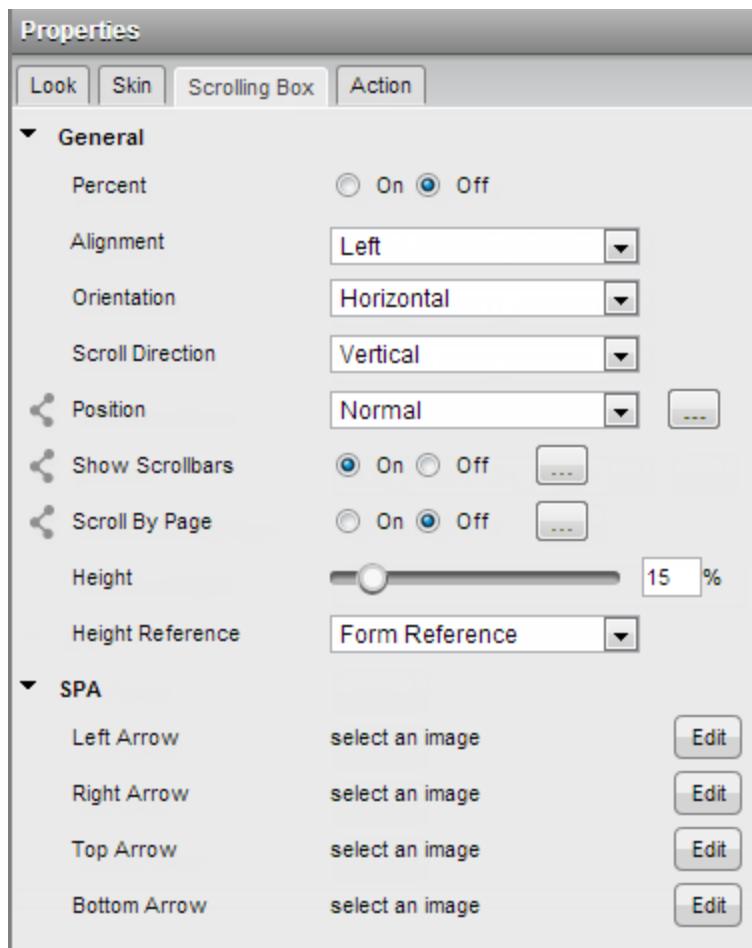
- Normal: Specifies the look and feel of the widget when not in focus.
- Refresh - Pull: Specifies the look and feel of a widget when the scroll bar is pulled.

- Refresh - Push: Specifies the look and feel of a widget when the scroll bar is pushed.

Refer to section [Working with Skins](#) to know more about adding a skin to a widget.

#### 5.4.4 Platform Specific Properties

**Note:** In this section, the properties that can be forked are identified by an icon ( ) located to the left of the property. For more details, refer to section [Forking a Widget property](#).



##### 5.4.4.1 Percent

Specifies whether the child widget's weight must be considered during the ScrollBox layout.

**Default:** percent is enabled

#### 5.4.4.2 Alignment

This property is displayed only when **Percent** is disabled.

**Alignment** allows you to set the direction in which the widgets are laid out.

**Default:** Left

Following are the available options:

- **Left:** The widgets placed inside a box are aligned left.
- **Right:** The widgets placed inside a box are aligned right.
- **Center:** The widgets placed inside a box are aligned center.

#### 5.4.4.3 Orientation

Specifies how you can stack the widgets within the ScrollBox. You can set the orientation of the ScrollBox as horizontal or vertical.

**Note:** ScrollBox with a vertical orientation cannot be placed directly on a form. It has to be placed inside an HBox.

**Default:** Horizontal

Following are the available options:

- Horizontal: Enables you to stack the content within the ScrollBox horizontally.
- Vertical: Enables you to stack the content within the ScrollBox vertically.

#### 5.4.4.4 Scroll Direction

Specifies how you can scroll the content within the ScrollBox.

**Default:** Horizontal

Following are the available options:

- Horizontal: Enables you to scroll the content within the ScrollBox horizontally.
- Vertical: Enables you to scroll the content within the ScrollBox vertically.
- Both: Enables you to scroll the content within the ScrollBox horizontally as well as vertically.
- None: Disables scrolling of the content in the ScrollBox.

**Note:** On SPA platforms, Horizontal scroll direction is not supported when the [Orientation](#) is set as Vertical; Vertical scroll direction is not supported when the [Orientation](#) is set as Horizontal.

#### 5.4.4.5 Position

Specifies the position of a ScrollBox as header or footer of a form.

**Note:** This property is applicable for immediate child widgets (with horizontal orientation) of a Form.

**Default:** Normal

Following are the options available:

- Normal: Retains the original position of the ScrollBox.
- Header: Specifies the position of the ScrollBox is fixed at the top of the Form.
- Footer: Specifies the position of the ScrollBox is fixed at the bottom of the Form.
- Fullscreen Segment Header: This option is useful if the box is placed on a form with a Segment on it. You must set the **screenLevelWidget** property of the particular segment to true. The scrollbox attaches itself to the Segment as a header and scrolls along with the segment. If the **screenLevelWidget** property of the particular segment is not set, then this value is ignored and the box is treated as normal.

- Fullscreen Segment Footer: This option is useful if the box is placed on a form with a Segment on it. You must set the **Full Screen Widget** property of the particular segment to *true*. The scrollbox attaches itself to the Segment as footer and scrolls along with the segment. If the **Full Screen Widget** property of the particular segment is not set, then this value is ignored and the box is treated as normal.

The position of the widget selected from this list becomes the default **position** value for all the platforms. You can choose to provide a different value for a platform by forking **position** property. Refer to the section [Forking a Widget Property](#) for more details.

#### 5.4.4.6 Show Scrollbars

Show Scrollbars specifies the visibility of the Scroll Bars.

**Default:** Show Scrollbars is enabled

- Click **On** to display the Scrollbars.
- Click **Off** to hide the Scrollbars.

The selected option becomes the default for all the platforms. You can choose to provide a different value for a platform by forking **showScrollbars** property. Refer to the section [Forking a Widget Property](#) for more details.

#### 5.4.4.7 Scroll By Page

Enables the scrolling of the content in increments of the width of the scrollbox on swipe gesture.

**Default:** Enabled scroll by page

- Click **On** to enable scroll by page.
- Click **Off** to disallow scroll by page.

The selected option becomes the default for all the platforms. You can choose to provide a different option for a platform by forking **Scroll By Page** property. Refer to the section [Forking a Widget Property](#) for more details.

#### 5.4.4.8 Height

Specifies the available height of the container in terms of percentage. The percentage is with reference to the value of [Height Reference](#) property.

For example, On a Form you have a ScrollBox with 5 labels and 5 buttons in it and a CloseButton below the ScrollBox. If the Height is set as 100 (percentage) and [Height Reference](#) is set as Form Reference, then the ScrollBox occupies the height of the Form excluding the height occupied by the CloseButton.

**Note:** This property is unavailable on Flex Forms.

#### 5.4.4.9 Height Reference

This property is enabled when you set the [Height](#).

**Default:** Form Reference

The container height percentage is calculated based on the following options.

- Form Reference: The scrollbox height is percentage calculated based on the height of the Form excluding headers and footers. This property doesn't have any effect if the scrollbox is placed inside a popup or headers/footers.
- Parent Width: Use this option if the scrollbox is placed inside a Box. The width is calculated based on the width of the Box.

**Note:** This property is unavailable on Flex Forms.

#### 5.4.4.10 Single Page Application (SPA) properties

In SPA, you can provide images for left, right, top and bottom arrows.

### Left Arrow

For inserting a left arrow image, click the **Edit** button against the **Left Arrow** field to open the **leftArrow** dialog box.

You can either:

- Locate the image and click OK.
- Provide image URL and click OK.

### Right Arrow

For inserting a right arrow image, click the **Edit** button against the **Right Arrow** field to open the **rightArrow** dialog box.

You can either:

- Locate the image and click OK.
- Provide image URL and click OK.

### Top Arrow

For inserting a top arrow image, click the **Edit** button against the **Top Arrow** field to open the **TopArrow** dialog box.

You can either:

- Locate the image and click OK.
- Provide image URL and click OK.

### Bottom Arrow

For inserting a bottom arrow image, click the **Edit** button against the **Bottom Arrow** field to open the **bottomArrow** dialog box.

You can either:

- Locate the image and click OK.
- Provide image URL and click OK.

#### 5.4.5 Actions

ScrollView widget has the following actions associated with it:

- onReachingBeginning: Gets called when scrolling reaches the beginning of the ScrollView widget
- onReachingEnd: Gets called when scrolling reaches the end of the ScrollView widget.
- onPull: Gets called when ScrollView is pulled from top.
- onPush: Gets called when ScrollView is pushed from bottom.

For more information on using the above actions, refer to section [Action Editor](#).

#### 5.4.6 Placement Inside a Widget

A ScrollView can be directly added inside the following widgets:

| Widget     | ScrollView placement inside a widget                    |
|------------|---|
| Form       | Yes   |
| HBox       | Yes   |
| VBox       | Yes   |
| ScrollView | Horizontal Orientation - No<br>Vertical Orientation- No |
| Tab        | Yes   |
| Segment    | No  |

|          |                                     |
|----------|-------------------------------------|
| Widget   | ScrollBox placement inside a widget |
| Popup    | Yes                                 |
| Template | Header- Yes<br>Footer- Yes          |

## 5.5 TabPane

TabPane widget allows you to organize multiple tabs within it. Each Tab will in turn hold a collection of widgets within the same area of the Form. You can only view one Tab a time, and every Tab in the TabPane widget consists of a certain type of information and is displayed when the user selects the corresponding Tab.

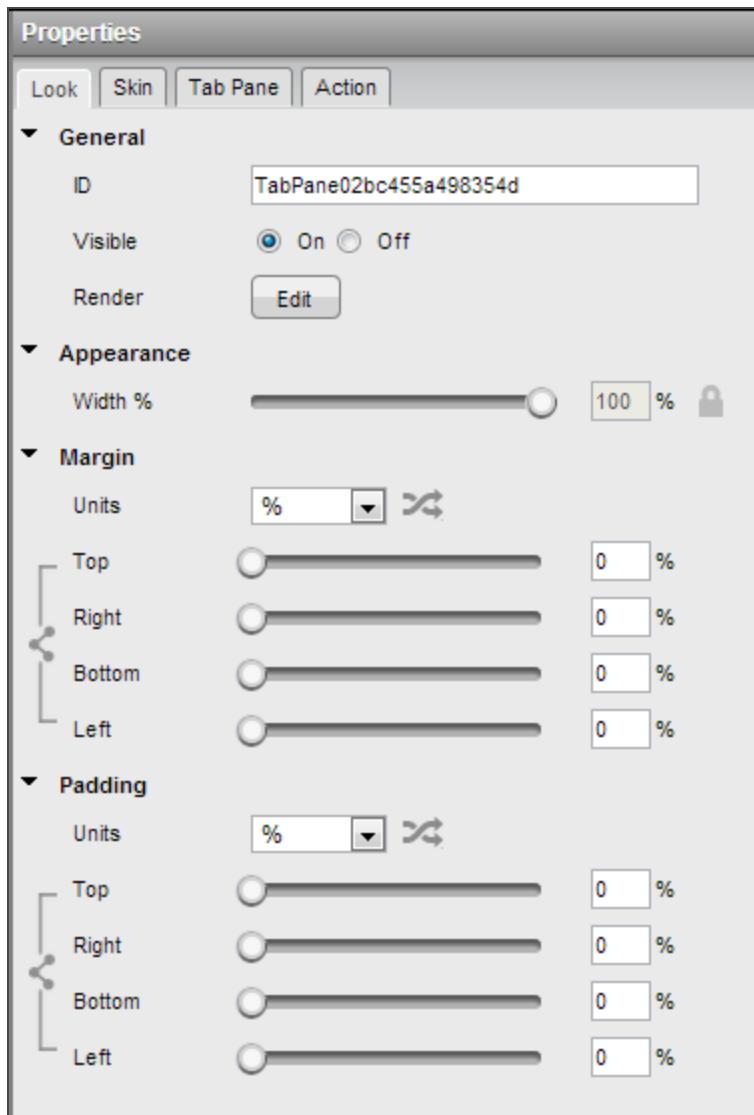
### 5.5.1 Important Considerations

The following are important considerations for a TabPane Widget.

- The TabPane widget occupies 100% of the screen width.
  - You can navigate within the TabPane using only the down key.
  - If you press the down key, the focus shifts to the next widget in the TabPane.
  - If you press the down key while you are on the last widget in the TabPane, you are taken to the top most widget in the TabPane.
  - If you press the right or the left arrow keys, you move to next or previous tabs.
  - Tab cycling is supported (that is, if you are on the last tab and you press the right arrow, you move to the first tab).
- On devices which have a navigation key, the following are applicable:
  - Each Tab has a context menu. This menu is displayed in the menu options whenever the Tab is in focus. The Menultems must be placed on other Form specific Menultems.
  - Tab remembers the control on which there was focus. For example, if control 2 of Tab2 is in focus, and you navigate to Tab1, when you navigate back to Tab2, control 2 will be in focus and not the first item in the Tab.

### 5.5.2 Look

This section details the **Properties > Look** tab of a TabPane widget.



To specify the look properties of a TabPane widget, refer section: [Look](#).

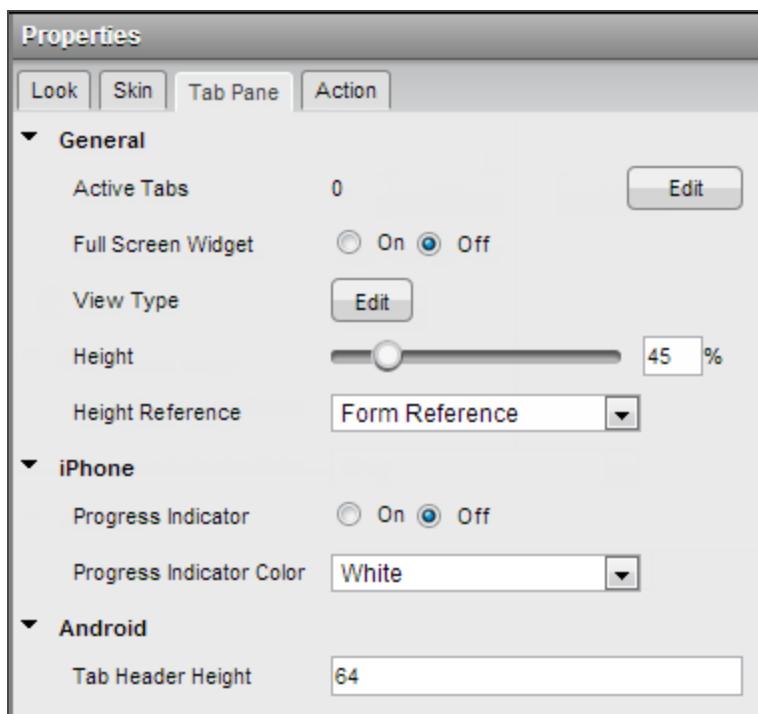
### 5.5.3 Skin

Skin defines properties such as background color, borders and shadows for a widget. They also, define the widget font (applicable only if text is displayed on a widget.)

For a TabPane widget, you can apply a skin for the following states

- Active: Specifies the skin to be applied when a TabPane is active.
- Active Focus: Specifies the skin that is to be applied when a TabPane is active and focused.
- Inactive: Specifies the skin to be applied for all inactive tabs.
- Page: Specifies the skin for page indicator. This property is applicable only when View Type is set as Page and images are selected for Focused Page Icon and Unfocused Page Icon.

#### 5.5.4 Platform Specific Properties



##### 5.5.4.1 Active Tabs

Indicates the selected Tabs indices. Index starts from 0. Specifies the Tab that must be displayed as the default open Tab.

##### 5.5.4.2 Full Screen Widget

Specifies whether the widget should occupy the whole container or not.

**Note:** You cannot place more than one TabPane as a screen level widget inside a form. Also, if you choose to make a TabPane a Screen Level Widget, we recommend that you place only one TabPane in the form and do not place any other widgets in the form.

**Note:** Do not set the screen level widget property to true for more than one widget in the form. If you have multiple widgets with this property set as true, there may be issues with how information is displayed along with some scrolling issues.

- If **On** is selected, the widget occupies the whole container.
- If **Off** is selected, the widget does not occupy the whole container.

#### 5.5.4.3 View Type

Specifies the view type the Tab Pane should display.

To specific view type for various platforms, click the **Edit** button against **View Type** field to open View Type dialog box. Select the desired view type for a choose platform.

Following are the available options:

- Tab
- Collapsible
- Page

#### 5.5.4.4 Height

Specifies the height of the TabPane in terms of percentage. The percentage is with reference to the value of [Height Reference](#) property.

**Note:** This property is unavailable on Flex Forms.

#### 5.5.4.5 Height Reference

The TabPane height percentage is calculated based on the option selected.

- Form Reference: The TabPane height percentage is calculated based on the height of the form excluding headers and footers. This option is not respected if TabPane is placed inside a popup or in templates.
- Parent Width: This option is used if the TabPane is placed inside a popup or in templates. The width is calculated based on the width of the parent container.

**Note:** This property is unavailable on Flex Forms.

#### 5.5.4.6 Progress Indicator

Specifies if the progress indicator must be displayed.

- If Off is selected, the progress indicator is not displayed on the widget.
- If On is selected, the progress indicator is displayed on the widget.

**Note:** This property is specific to the iOS platform.

#### 5.5.4.7 Progress Indicator Color

Specifies the color of the progress indicator. You can either choose Grey color or White color for your Progress Indicator.

**Note:** This property is specific to the iOS platform.

#### 5.5.4.8 Tab Header Height

Specifies the height of a tab header.

**Note:** This property is specific to the Android platform.

### 5.5.5 Actions

TabPane widget has the following action associated with it:

- **onTabClick:** This action is executed when the tab is clicked. This action is available only when the **View Type** is set to **Collapsible View** and is triggered when you expand or collapse any Tab.

For more information on using the above action, refer to section [Action Editor](#).

### 5.5.6 Placement inside a Widget

A TabPane can be directly added inside the following widgets:

| Widget    | TabPane placement inside a widget                      |
|-----------|--|
| Form      | Yes, only 1  |
| HBox      | Yes, only 1  |
| VBox      | Yes, only 1  |
| ScrollBar | Horizontal Orientation -No<br>Vertical Orientation- No |
| Tab       | No   |
| Segment   | Horizontal Orientation -No<br>Vertical Orientation- No |
| Popup     | Yes, only 1  |
| Template  | Header- No<br>Footer- No                               |

## 5.6 VBox

A VBox is used to layout widgets in a single vertical orientation. It can contain any number of widgets. By default, VBox occupies the entire available height of its parent widget.

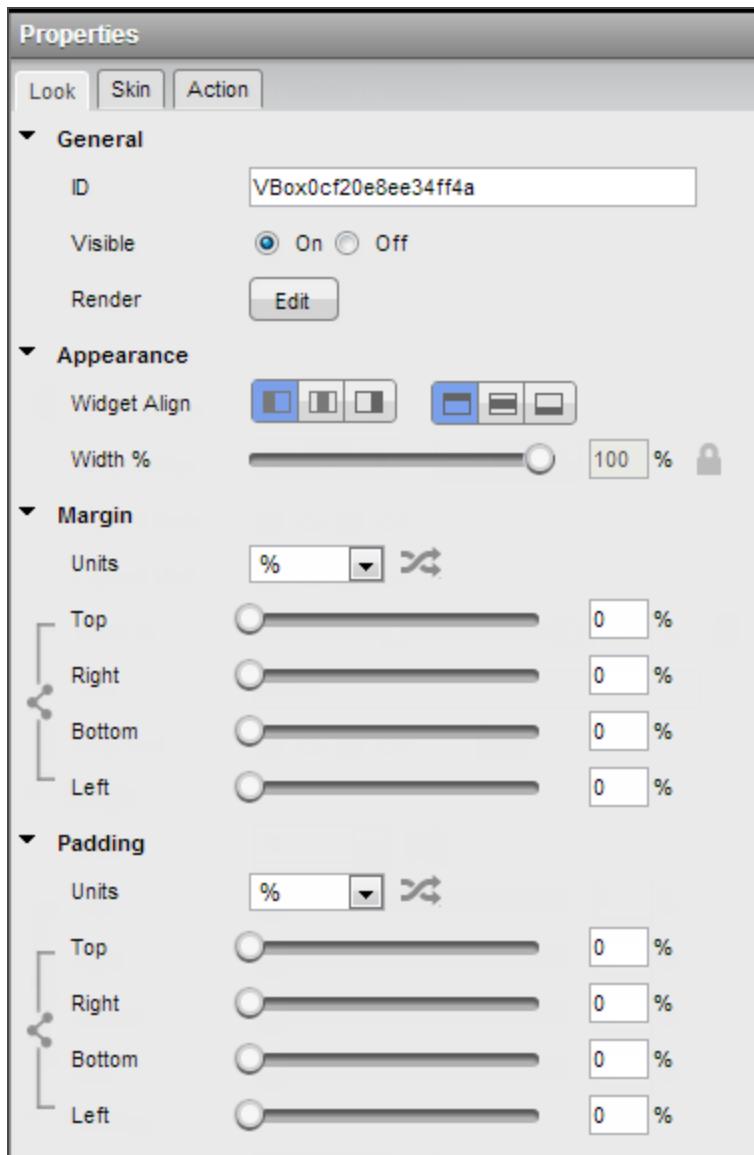
### 5.6.1 Important Considerations

The following are important considerations of a VBox Widget:

- A VBox widget is available only for a VBox Form.
- The height of a VBox is the cumulative height of its child widgets.
- A VBox widget cannot be directly placed inside another VBox.
- A widget becomes invisible if you choose not to render it, and if that widget is placed inside a VBox.

### 5.6.2 Look

This section details the **Properties > Look** tab of a VBox widget.



To specify the look properties of a VBox widget, refer section: [Look](#).

### 5.6.3 Skin

Skin defines properties such as background color, borders and shadows for a widget. They also, define the widget font (applicable only if text is displayed on a widget.)

For a VBox widget, you can apply a skin for the following states:

- Normal: Specifies the look and feel of the widget when not in focus.
- Focus: Specifies the skin that is applied when there is focus on a widget.
- Blocked UI: Specifies the skin that must be used to block the interface until the action in progress (for example, a service call) is completed.
- Hover Skin: Specifies the look and feel of a widget when the cursor hovers on the widget.

**Note:** Hoover Skins is available only on the Windows (native) Tablet platform.

Refer to section [Working with Skins](#) to know more about adding a skin to a widget.

#### 5.6.4 Actions

VBox widget has the following action associated with it:

- onClick: This action is invoked by the platform when the user performs a click action on the widget.

For more information on using the above action, refer to section Action Editor.

#### 5.6.5 Placement Inside a Widget

A VBox can be directly added inside the following widgets:

| Widget    | VBox placement inside a widget                           |
|-----------|--|
| Form      | No   |
| HBox      | Yes  |
| VBox      | No   |
| ScrollBar | Horizontal Orientation - Yes<br>Vertical Orientation- No |

|          |  |
|----------|--|
| Widget   | VBox placement inside a widget                           |
| Tab      | No   |
| Segment  | Horizontal Orientation - Yes<br>Vertical Orientation- No |
| Popup    | No   |
| Template | Header- No<br>Footer- No                                 |

## 5.7 Button

A button is a control that you can use to provide input to an application or trigger an event. For example, navigating to a form, interacting with a dialog box, or confirming an action.

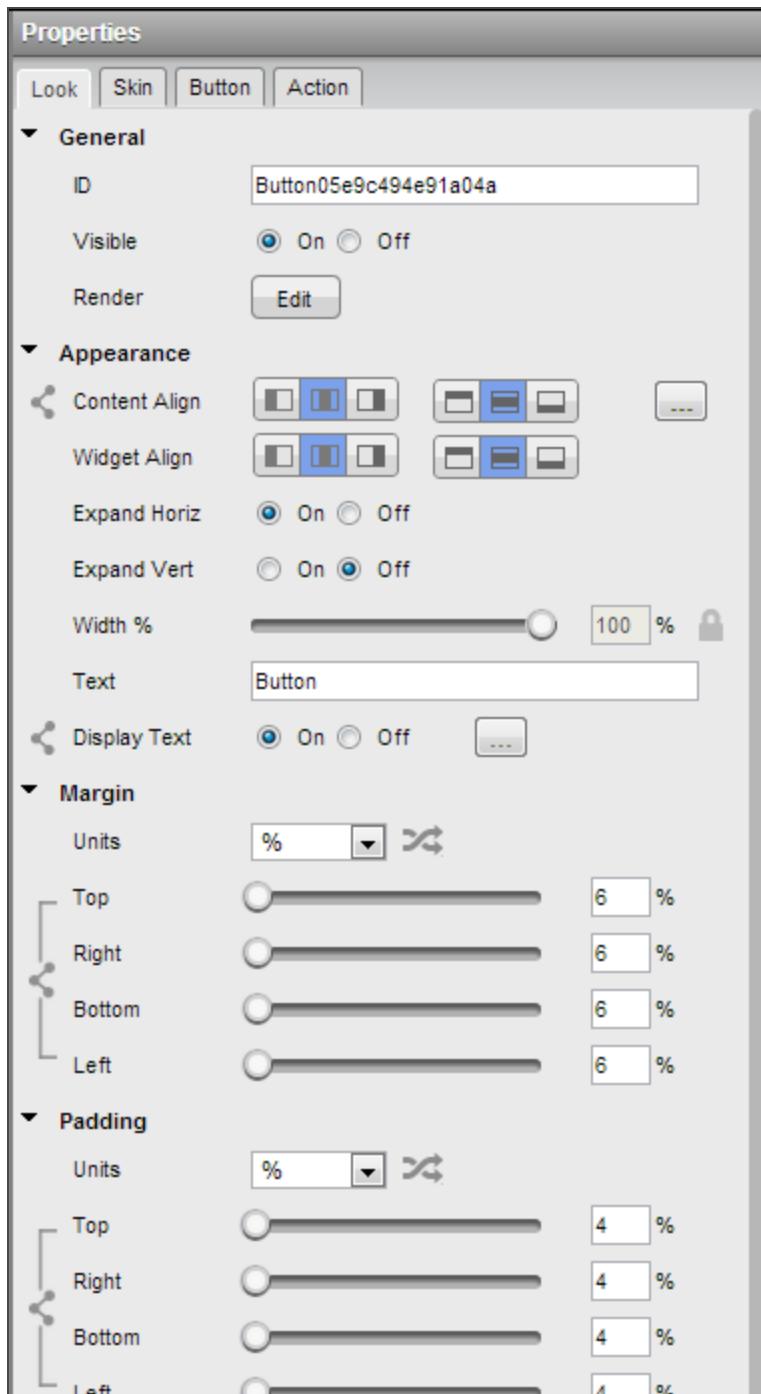
### 5.7.1 Important Considerations

The following are important considerations for a Button Widget.

- You can specify different skins for normal and focus states of a button.
- You can specify a background image for normal and focus states of a button.
- To avoid jumping effect or to avoid overlap of neighboring widgets, you must ensure that the image for normal and focus skins are of the same size.

### 5.7.2 Look

This section details the **Properties > Look** tab of a Button widget.



To specify the look properties of a Button widget, refer section: [Look](#).

### 5.7.3 Skin

Skin defines properties such as background color, borders and shadows for a widget. They also, define the widget font (applicable only if text is displayed on a widget.)

For a Button widget, you can apply a skin for the following states:

- Normal: Specifies the look and feel of the widget when not in focus.
- Focus: Specifies the skin that is applied when there is focus on a widget.
- Blocked UI: Specifies the skin that must be used to block the interface until the action in progress (for example, a service call) is completed.

**Note:** Blocked UI is available only on SPA platforms.

- Pressed: Specifies the skin to indicate that the widget is pressed or clicked.

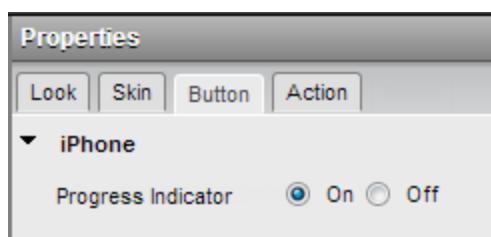
**Note:** Pressed is available only on Android Native platforms.

- Hover Skin: Specifies the look and feel of a widget when the cursor hovers on the widget.

**Note:** Hover Skin is available only on Windows Tablet platforms.

Refer to section [Working with Skins](#) to know more about adding a skin to a widget.

### 5.7.4 Platform Specific Properties



#### 5.7.4.1 Progress Indicator

Specifies if the progress indicator must be displayed when the button is clicked. Typically this property is enabled when the time take to load an application is long.

**Default:** Progress Indicator is enabled.

- Click **On** to enable progress indicator.



- Click **Off** to disable progress indicator.

**Note:** This property is specific to the iOS platform.

#### 5.7.4.2 Tool Tip

Specifies the hint text when the cursor hovers over a widget, without clicking it. The text entered in the tooltip appears as a small box when the cursor hovers over a widget.

**Note:** This property is specific to the Windows Tablet platform.

### 5.7.5 Actions

Button widget has the following action associated with it:

- **onClick:** This action is invoked by the platform when the user performs a click action on the button.

For more information on using the above action, refer to section [Action Editor](#).

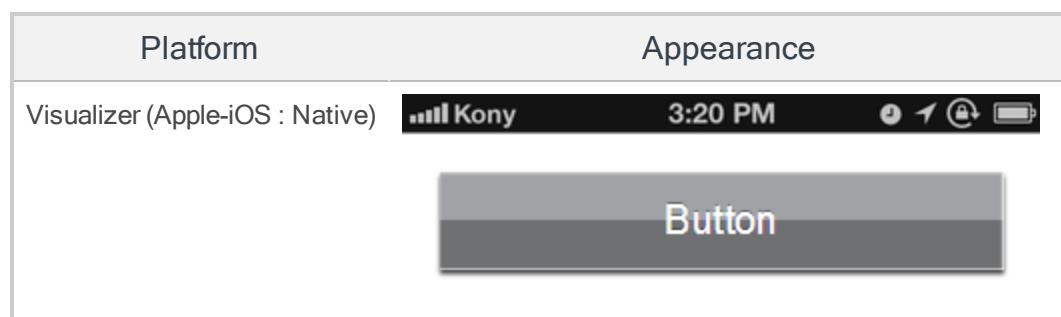
#### 5.7.6 Placement inside a Widget

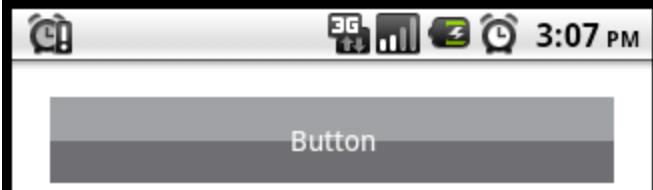
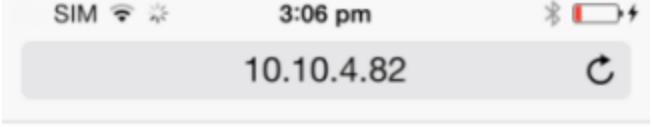
A button can be directly added inside the following widgets:

|           |  |
|-----------|--|
| Widget    | Button placement inside a widget                         |
| Form      | Yes  |
| HBox      | Yes  |
| VBox      | Yes  |
| ScrollBar | Horizontal Orientation -Yes<br>Vertical Orientation -Yes |
| Tab       | Yes  |
| Segment   | Horizontal Orientation -Yes<br>Vertical Orientation -Yes |
| Popup     | Yes  |
| Template  | Header- No<br>Footer- No                                 |

### 5.7.7 Widget Appearance on Platforms

The appearance of the Button widget on various platforms is as follows:



| Platform | Appearance   |
|----------|--|
| Android  |  |
| iOS      |  |
| SPA      |  |

## 5.8 Calendar

Calendar widget allows you to select a date from a graphical calendar. The calendar widget appears as a label with a small calendar icon (icon does not appear on Mobile Web platforms) and displays the date or the date format specified by you. You can interact with the calendar widget by clicking on it.

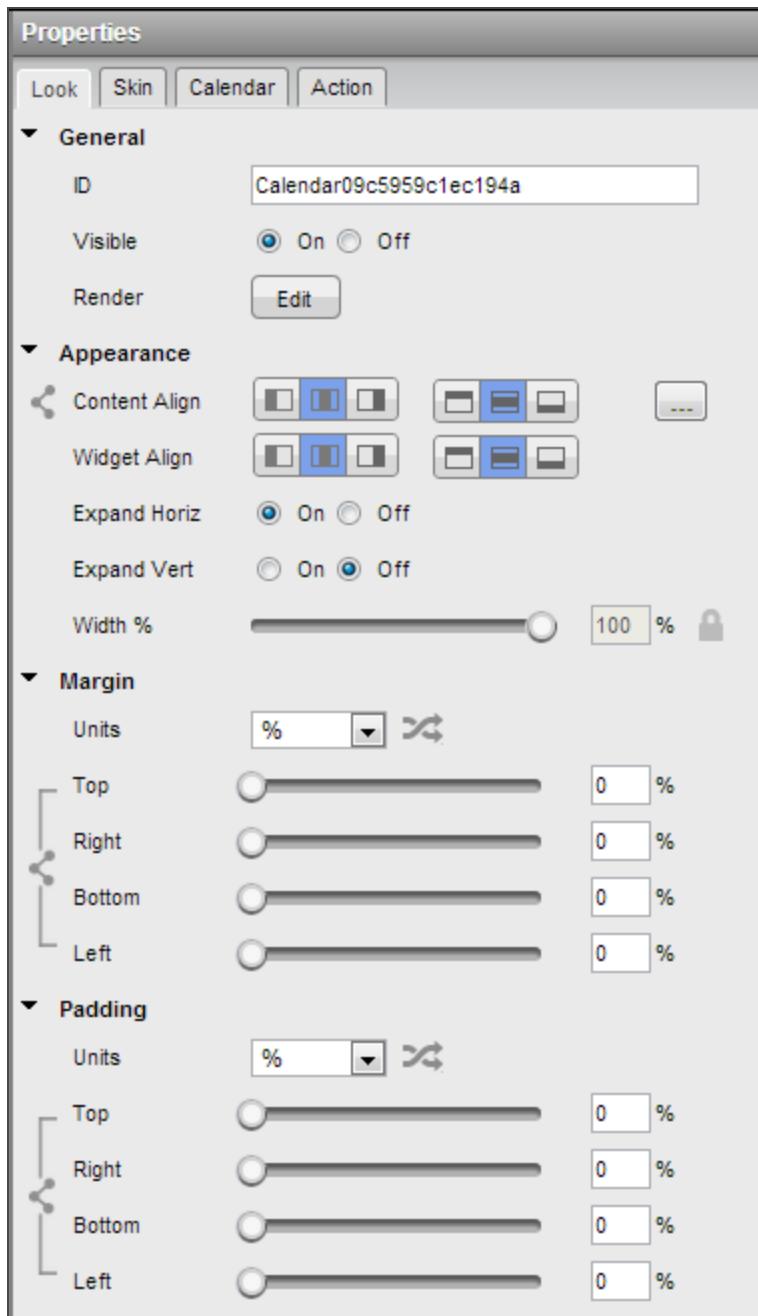
### 5.8.1 Important Considerations

The following are important considerations for a Calendar Widget.

- The default height of calendar widget is 28px.
- If you do not specify an image for popup view then a default image is provided.
- By default, calendar widget occupies the complete width of its parent widget.
- Clicking the Calendar icon, opens the current month calendar. On Android platform, restricting the date selection between validStartDate and validEndDate is not possible with Native Calendar View.

### 5.8.2 Look

This section details the **Properties > Look** tab of a Calendar widget.



To specify the look properties of a Calendar widget, refer section: [Look](#).

### 5.8.3 Skin

Skin defines properties such as background color, borders and shadows for a widget. They also, define the widget font (applicable only if text is displayed on a widget)

For a Calendar widget, you can apply a skin for the following states:

- Normal: Specifies the look and feel of the widget when not in focus.
- Focus: Specifies the skin that is applied when there is focus on a widget.
- Grid: - Specifies the skin for the grid calendar.
- Cell: Specifics the skin for the grid calendar cell.
- Cell - Selected: Specifies the skin that is applied when a grid calendar cell is selected.
- Cell - Focus: Specifies the skin that is applied when there is a focus on the grid calendar cell.
- Cell - Today: Specifies the skin that is applied for the grid calendar today cell.
- Cell - Weekend: Specifies the skin to be applied for the weekend days of a grid calendar.
- Cell - Inactive: Specifies the skin to be applied for non-working inactive days of a grid calendar.
- Done Button: Specifies the skin to be applied for **Done** button that appear on a calendar popup.
- Cancel Button: Specifies the skin to be applied for a **Cancel** button that appear on a calendar popup.
- Day: Specifies the day portion of the currently selected date.
- Month: Specifies the month portion of the currently selected date.
- Hover Skin: Specifies the look and feel of a widget when the cursor hovers over a widget.

**Note:** Hover Skin is available only on the Windows (native) Tablet platform.

Refer to section [Working with Skins](#) to know more about adding a skin to a widget.

## 5.8.4 Platform Specific Properties

**Note:** In this section, the properties that can be forked are identified by an icon ( ) located to the left of the property. For more details, refer to section [Forking a Widget property](#).

**Properties**

Look Skin Calendar Action

**General**

- Start Date
- End Date
- View Type
- Default Date
- Date Format
- Calendar Icon
- Left Nav Image
- Right Nav Image
- Weekend Selectable

**iPhone**

- Mode
- Hide Days Header  On  Off
- Hide Months Header  On  Off
- Day Text Alignment In Cell
- Cell Template
- Data
- Height  %
- Height Reference

**Android**

- Hide Day's Header  On  Off
- Hide Month's Header  On  Off
- Day Textalignment In Cell
- Cell Template
- Data
- Height  %
- Height Reference

**SPA**

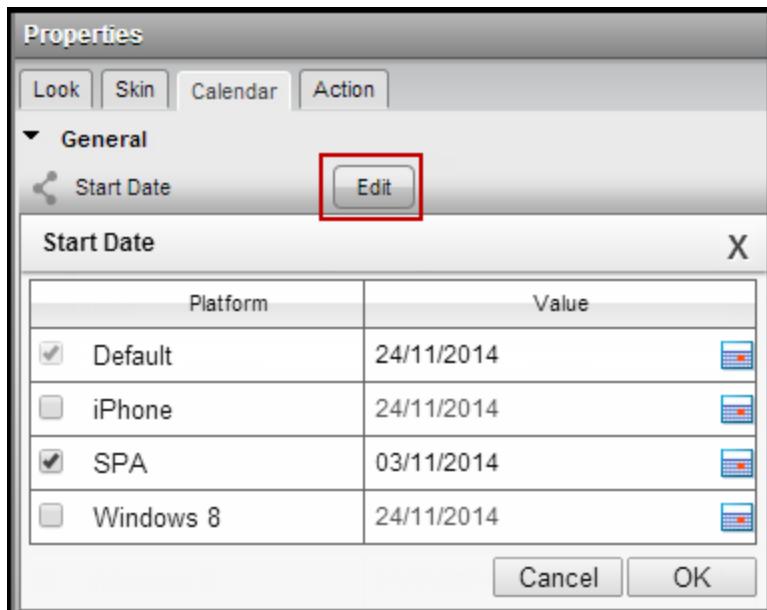
- Popup Title

**Windows 8**

- Placeholder

#### 5.8.4.1 Start Date

Specifies the start date in your choose [Data Format](#). By default, the Start Date is populated with today's date.



To input a different default start date or to provide a platform specific start date, click the **Edit** button against the **Start Date** to open the **Start Date** dialog box.

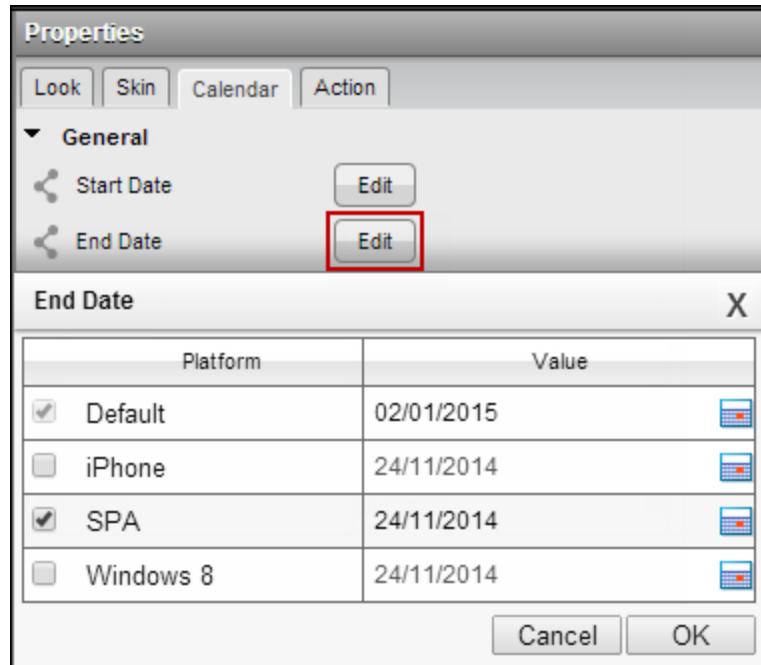
The **Default** platform value is populated with today's date. To change this value, click the calendar icon and from the calendar pop-up select a date.

To provide a platform specific start date,

- a. Click the desired platform check box
- b. In the corresponding **Value** field, click the calendar icon.
- c. from the Calendar pop-up, select a date.

#### 5.8.4.2 End Date

Specifies the end date in your choose [Data Format](#). By default, the valid end date is populated with today's date.



To input a different default end date or to provide a platform specific end date, click **Edit** button against the **End Date**. This results in opening the **Valid EndDate** window.

**Default** platform value is populated with today's date. To change this value, click the calendar icon and from the calendar pop-up select a date.

To provide a platform specific end date,

- click the desired platform check box
- in the **Value** field, click the calendar icon.
- from the Calendar pop-up, select a date.

#### 5.8.4.3 View Type

Specifies the view type of the Calendar.

**Default:** Default

Below are the view types applicable on all platforms:

- Default: Specifies the default native calendar view in respective platforms.
- Onscreen Grid
- Popup Grid ( only SPA platforms supports this option.)

The **View Type** of the widget selected from this list becomes the default for all the platforms. You can choose to provide a different value for a platform by forking **View Type** property. Refer to the section [Forking a Widget Property](#) for more details.

**Note:** On Android platform, when the View Type is set as Default, the user cannot be restricted to select the dates within the validStartDate and validEndDate.

#### 5.8.4.4 Placeholder

Specifies the temporary or substitute text that must be displayed until a date is selected.

For example, you can display the placeholder text on the calendar widget as "Enter your date of travel", until the user clicks the calendar widget and selects a date.

#### 5.8.4.5 Default Date

Specifies a default date to be displayed on a Calendar widget. You can choose a date by clicking the calendar icon, and then selecting a date.

This date replaces any [Placeholder](#) value.

#### 5.8.4.6 Data Format

The date format enables you to select the format in which the date is displayed.

**Default:** DD/MM/YYYY

Following date formats are supported:

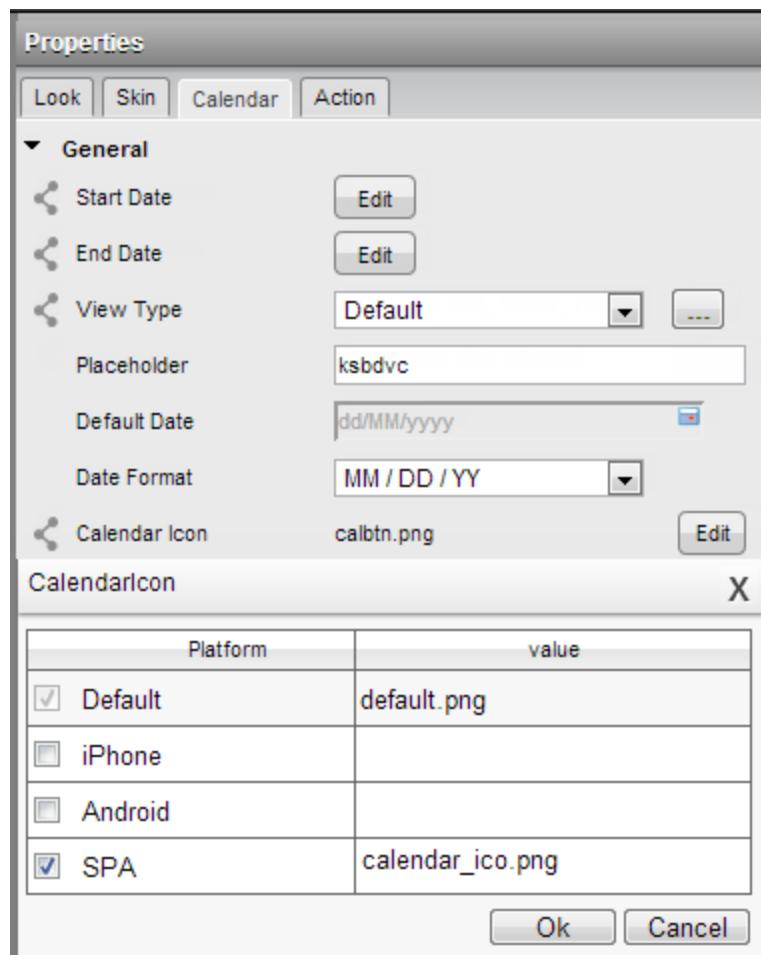
- DD/MM/YYYY
- MM/DD/YYYY
- MM/DD/YY

#### 5.8.4.7 Calendar Icon

Specifies the calendar icon that is displayed next to the placeholder.

System provides a default icon that is displayed across the platforms.

You may choose to replace this default icon or provide a platform specific icon by clicking the **Edit** button against the **Calendar Icon** field to open the **calendar Icon** dialog box.



Select the platform and click inside the corresponding **value** field to open the **Select Image** dialog box.

You can either :

- Select an available image
- Provide image url

#### 5.8.4.8 Left Nav Image

Specifies left navigation image that is displayed on the calendar popup. Click this image to navigate and select a past date from the calendar.

For assigning an image, click the **Edit** button against the **Left Nav Image** to open the **Left Nav Image** search dialog box. You can either:

- Select an image.
- Provide image url.

#### 5.8.4.9 Right Nav Image

Specifies right navigation image that is displayed on the calendar popup. Click this image to navigate and select a future date from the calendar.

For assigning an image, click the **Edit** button against the **Right Nav Image** to open the **Right Nav Image** search dialog box. You may either:

- Select an image.
- Provide image url .

#### 5.8.4.10 Weekend Selectable

Select the check box to allow weekend selection. If unchecked, you will not be able to choose dates that fall on weekends.

#### 5.8.4.11 Mode

Specifies the mode in which the calendar is used.

## Default: Date

Following are the available options:

- Date: Allows you to select only date.
- Date & Time: Allows you to select both date and time.

**Note:** This property is specific to the iOS platform.

### 5.8.4.12 Hide Days Header

This property is available only when View Type is set as Onscreen Grid or Popup Grid. It indicates if the weekdays are hidden on the header of a grid calendar.

- Click **On** to hide the weekdays.
- Click **Off** to display weekdays.

**Note:** This property is specific to the iOS and Android platforms.

### 5.8.4.13 Hide Months Header

This property is available only when View Type is set as Onscreen Grid or Popup Grid. It indicates if the months header is hidden for grid calendar including the navigation buttons.

- Click **On** to hide the months header.
- Click **Off** to display the months header.

**Note:** This property is specific to the iOS and Android platforms.

### 5.8.4.14 Day Text Alignment In Cell

This property is available only when the View Type is set as Onscreen Grid or Popup Grid. It specifies the alignment of the text for a Calendar Day cell regarding its boundaries. The alignment values are:

- Top-Left: Specifies the text should align at the top left corner of a Calendar Day cell.
- Top-Center: Specifies the text should align at the top center of a Calendar Day cell.
- Top-Right: Specifies the text should align at the top right of a Calendar Day cell.
- Middle-Left: Specifies the text should align at the middle left of a Calendar Day cell.
- Center: Specifies the text should align at the center of a Calendar Day cell.
- Middle-Right: Specifies the text should align at the middle right of a Calendar Day cell.
- Bottom-Left: Specifies the text should align at the bottom left of a Calendar Day cell.
- Bottom-Center: Specifies the text should align at the bottom center of a Calendar Day cell.
- Bottom-Right: Specifies the text should align at the bottom right of a Calendar Day cell.

**Note:** This property is specific to the iOS and Android platforms.

#### 5.8.4.15 Cell Template

This property is available only when the View Type is set as Onscreen Grid or Popup Grid. It specifies the common template to be used for a Calendar Day cell. A template is used only when the data is present for a Calendar Day cell. If the data is not set to a cell, the cell is displayed with the default look, which has no template.

To assign a template for cell template , follow these steps:

1. Click **Edit**. The **cellTemplate** window appears.
2. From the list of templates, select a template.
3. Click **OK**.

**Note:** This property is specific to the iOS and Android platforms.

#### 5.8.4.16 Data

To modify the **Data** property, you need to assign a grid template to the [Cell Template](#) property.

Data represents the actual data to be rendered in each cell.

To specify the data, follow these steps:

1. Click Edit. The **Master Data for GridCalendar** window appears.
2. From the **Data** column , click Calendar icon.
3. Select a date.
4. From the **Template Data** column, click ellipsis. The **Template Data** appears.
5. Do the following:
  - a. Update the data of a widget, if required.
  - b. Update the skin of a widget, if required.
6. Click **OK**.

**Note:** This property is specific to the iOS and Android platforms.

#### 5.8.4.17 Height

This property is available only when View Type is set as Onscreen Grid or Popup Grid. It specifies the available height of the container in terms of percentage. The percentage is with reference to the value of [Height Reference](#) property.

**Note:** This property is specific to the iOS and Android platforms.

#### 5.8.4.18 Height Reference

This property is available only when View Type is set as Onscreen Grid or Popup Grid and when you set the Height.

The container height percentage is calculated based on the below options.

- Form Reference: The Calendar height is percentage calculated based on the height of the Form excluding header and footer.
- Parent Width: Use this option if the Calendar is placed inside a Box. The width is calculated based on the width of the Box.

**Note:** This property is specific to the iOS and Android platforms.

#### 5.8.4.19 Popup Title

Specifies the title text to be displayed on the calendar popup.

**Note:** This property is specific to the SPA platform.

#### 5.8.4.20 Tool Tip

Specifies the hint text when the cursor hovers over a widget, without clicking it. The text entered in the tooltip appears as a small box when the cursor hovers over a widget.

**Note:** This property is specific to Windows platform.

### 5.8.5 Actions

Calendar widget has the following action associated with it:

- onSelection: This action is triggered when an item is selected.

For more information on using the above action, refer to section [Action Editor](#).

## 5.8.6 Placement Inside a Widget

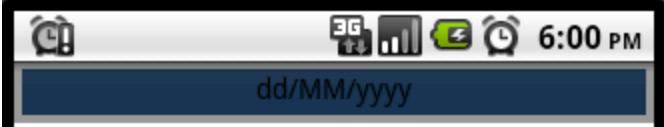
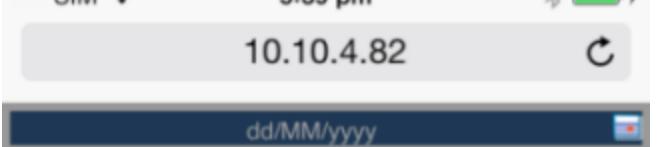
A Calendar can be directly added inside the following widgets:

| Widget    | Calendar placement inside a widget                        |
|-----------|---|
| Form      | Yes   |
| HBox      | Yes   |
| VBox      | Yes   |
| ScrollBox | Horizontal Orientation - Yes<br>Vertical Orientation- Yes |
| Tab       | Yes   |
| Segment   | No  |
| Popup     | Yes   |
| Template  | Header- No<br>Footer- No                                  |

## 5.8.7 Widget Appearance on Platforms

The appearance of the Calendar widget on various platforms is as follows:

| Platform                        | Appearance |
|---------------------------------|------------|
| Visualizer (Apple-iOS : Native) |            |

| Platform | Appearance   |
|----------|--|
| Android  |  |
| iOS      |  |
| SPA      |  |

## 5.9 CheckBoxGroup

The CheckBoxGroup widget allows you to make one or more selections from a group of check boxes. If you select a check box, a check mark appears inside the check box to indicate the selection.

You can use a CheckBoxGroup widget to provide a selection of choices in groups from which the user can select one or more choices.

### 5.9.1 Important Considerations

The following are important considerations for a CheckBoxGroup Widget.

#### 5.9.1.1 All Platforms

- CheckBoxGroup widget is always a group widget.
- Limit the number of choices in the widget. If you need to display several choices (above 15 choices), consider using a [ListBox](#) widget.

#### 5.9.1.2 Android

- If you set the [Orientation](#) to horizontal, we suggest that you do not place more than two items in the group, as there is a platform limitation.

If you place more than two items and the associated text with the items is large, there is a possibility that the additional items will not fit in the screen width and will not be visible on the screen.

#### 5.9.1.3 iPhone

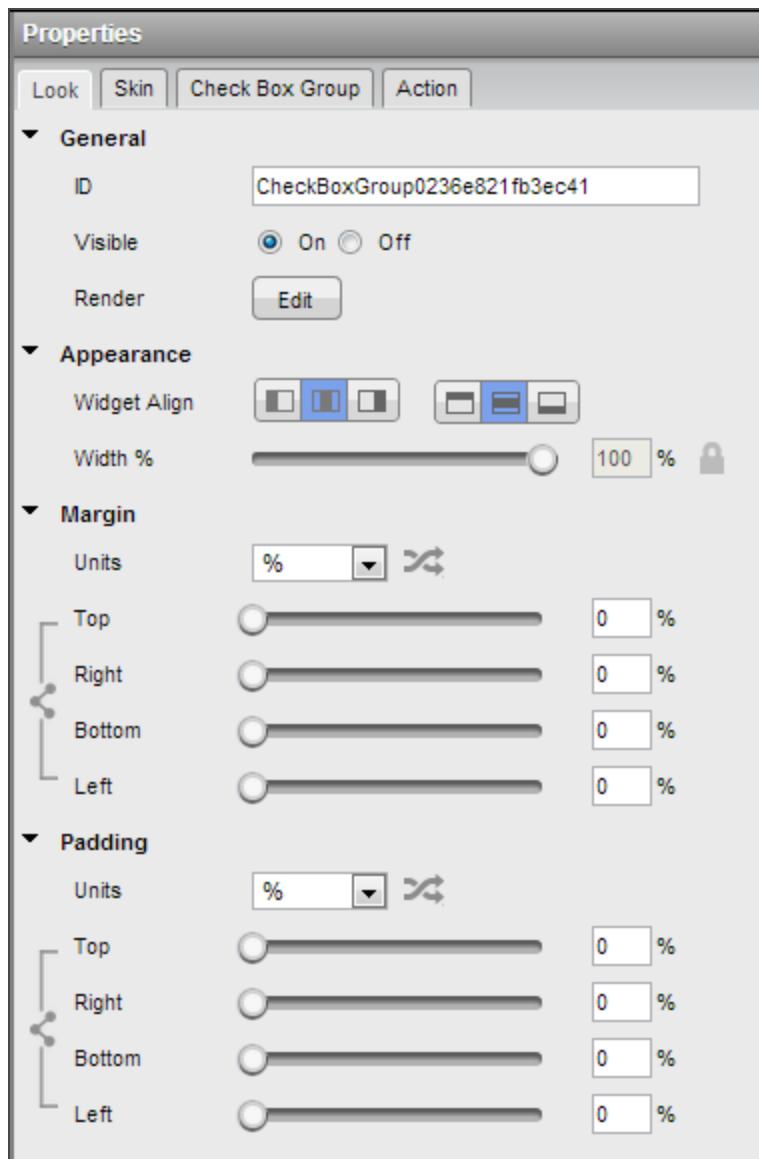
- You cannot apply skins in the on-off switch view.

#### 5.9.1.4 SPA

- Focus skin is not supported.

## 5.9.2 Look

This section details the **Properties > Look** tab of a **CheckBoxGroup** widget.



To specify the look properties of a **CheckBoxGroup** widget, refer section: [Look](#).

### 5.9.3 Skin

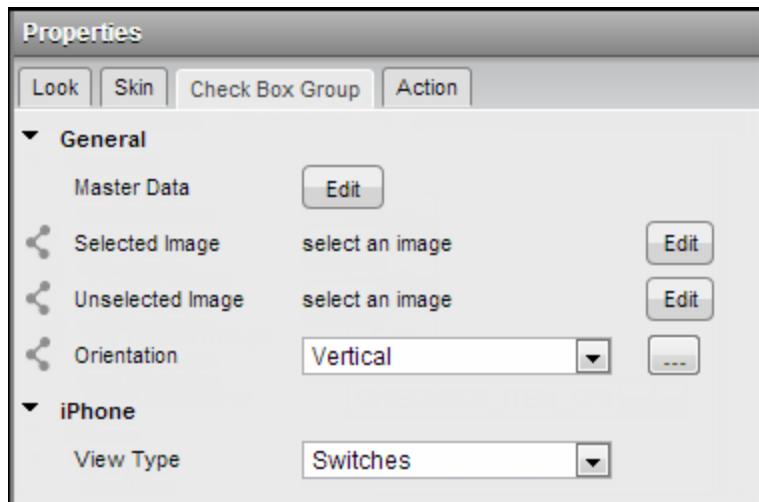
Skin defines properties such as background color, borders and shadows for a widget. They also, define the widget font (applicable only if text is displayed on a widget.)

For a CheckBoxGroup widget, you can apply a skin for the following states:

- Normal: Specifies the look and feel of the widget when not in focus.
- Focus: Specifies the skin that is applied when there is focus on a widget.

Refer to section [Working with Skins](#) to know more about adding a skin to a widget.

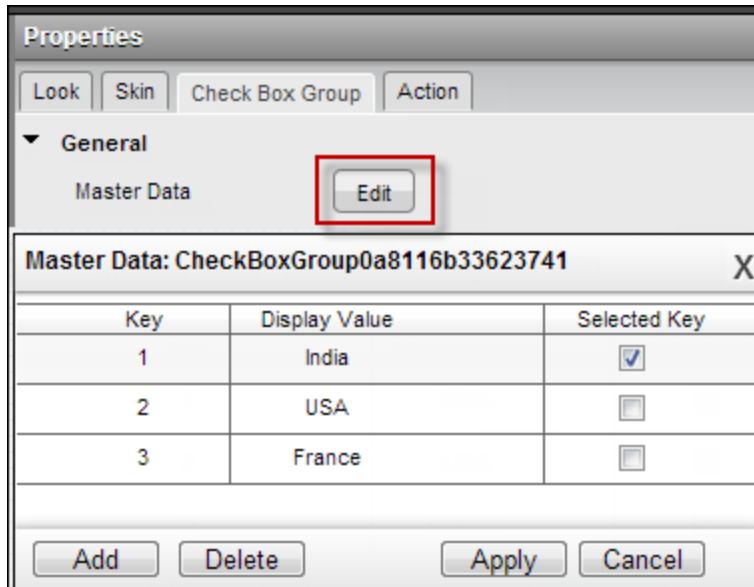
### 5.9.4 Platform Specific Properties



#### 5.9.4.1 Master Data

Specifies the set of values that must be displayed for the user to make a selection from the available choices.

For specifying these set of values, click the **Edit** button against the **masterData** field to open the **MasterData for CheckBox** dialog box.



Master Data contains the following columns:

- Key: It is the unique identifier of each individual Check Box.
- Display Value: This value is displayed in the CheckBoxGroup widget on a form.
- Select Key : Allows you to select individual check boxes from the group. You can select more than one check box.

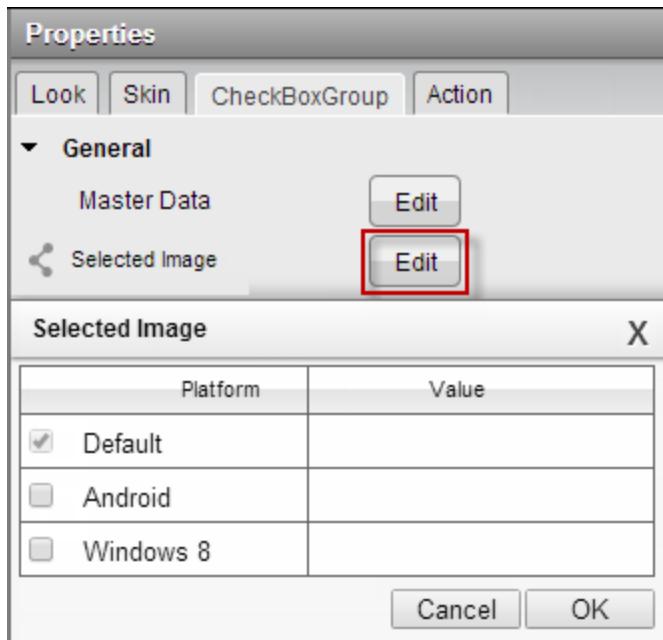
If required, to add more check boxes to the widget, click **Add**. To delete a check box, click inside a check box, and then click **Delete**.

After providing the required details and click **Apply** to create master data.

#### 5.9.4.2 Selected Image

Specifies the image to be displayed when you make a selection.

For providing a default or platform specific image, click the **Edit** button against **Selected Image** field to open the **Selected Image** dialog box.



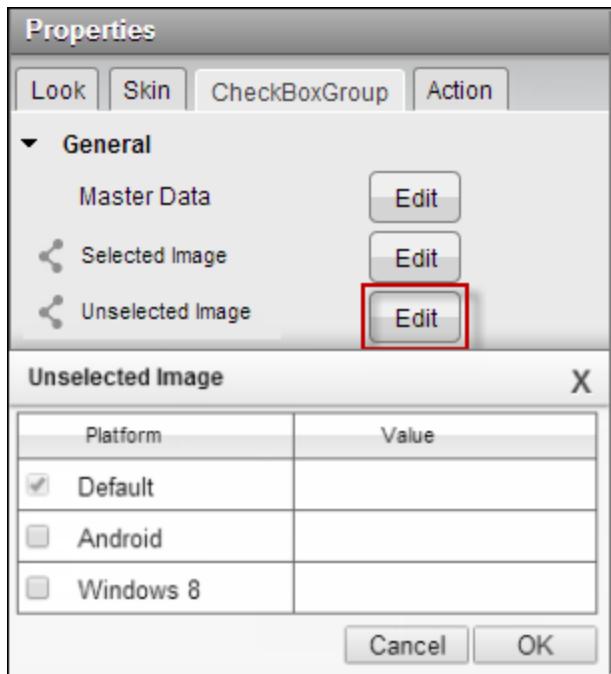
To provide a platform specific image or to replace the default image, select the platform and click inside the corresponding **value** field. This results in opening the **Select Image** dialog box. You can either:

- Select an available image.
- Provide image url.

#### 5.9.4.3 Unselected Image

Specifies the image to be displayed when a selection is cleared.

For providing a default or platform specific image, click the **Edit** button against **unselected Image** field to open the **Unselected Image** dialog box.



To provide a platform specific image or to replace the default image, select the platform and click inside the corresponding **value** field. This results in opening the **Select Image** dialog box. You can either:

- Select an available image.
- Provide image url.

#### 5.9.4.4 Orientation

Specifies the alignment of the check boxes as horizontal or vertical.

**Default:** Vertical

Following orientations are available:

- Vertical
- Horizontal

#### 5.9.4.5 View Type

Specifies the view type of the CheckBox.

## Default:Switches

Following views are available:

- Switches
- Table
- On-screen Wheel

**Note:** This property is specific to the iOS platform.

### 5.9.4.6 Tool Tip

Specifies the hint text when the cursor hovers over a widget, without clicking it. The text entered in the tooltip appears as a small box when the cursor hovers over a widget.

**Note:** This property is specific to the Windows Tablet platform.

### 5.9.5 Actions

CheckBoxGroup widget has the following action associated with it:

- onSelection: This action is invoked by the platform when an item is selected.

For more information on using the above action, refer to section [Action Editor](#).

### 5.9.6 Placement Inside a Widget

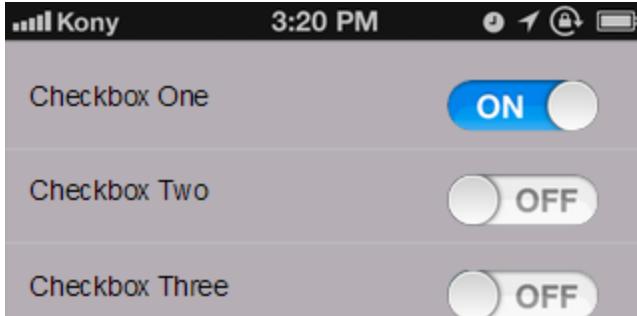
A CheckBoxGroup can be directly added inside the following widgets:

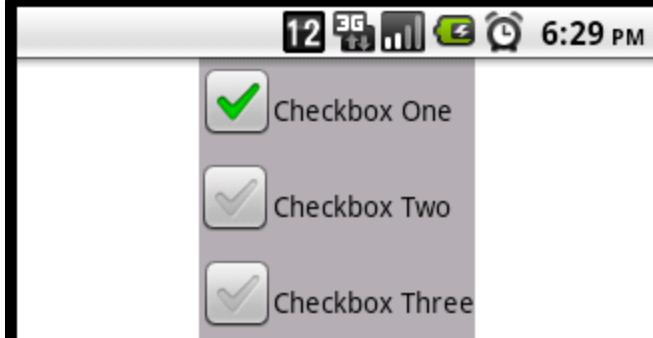
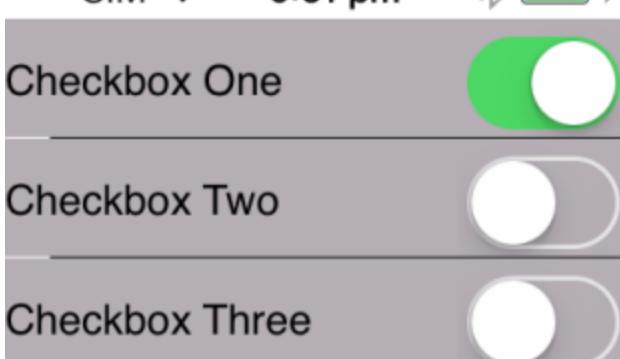
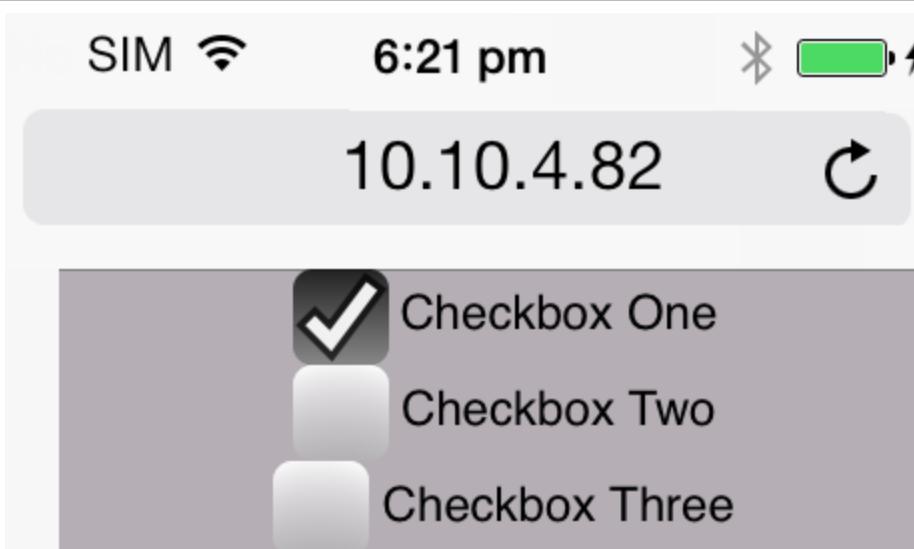
| Widget | CheckBoxGroup placement inside a widget |
|--------|---|
| Form   | Yes                                     |
| HBox   | Yes                                     |

|           |   |
|-----------|---|
| Widget    | CheckBoxGroup placement inside a widget                   |
| VBox      | Yes   |
| ScrollBar | Horizontal Orientation - Yes<br>Vertical Orientation- Yes |
| Tab       | Yes   |
| Segment   | No  |
| Popup     | Yes   |
| Template  | Header- No<br>Footer- No                                  |

### 5.9.7 Widget Appearance on Platforms

The appearance of the CheckBoxGroup widget on various platforms is as follows:

| Platform                        | Appearance   |
|---------------------------------|--|
| Visualizer (Apple-iOS : Native) |  |

| Platform | Appearance   |
|----------|--|
| Android  |    |
| iOS      |   |
| SPA      |  |

## 5.10 ComboBox

A ComboBox is a widget that allows you to select a single item from a list.

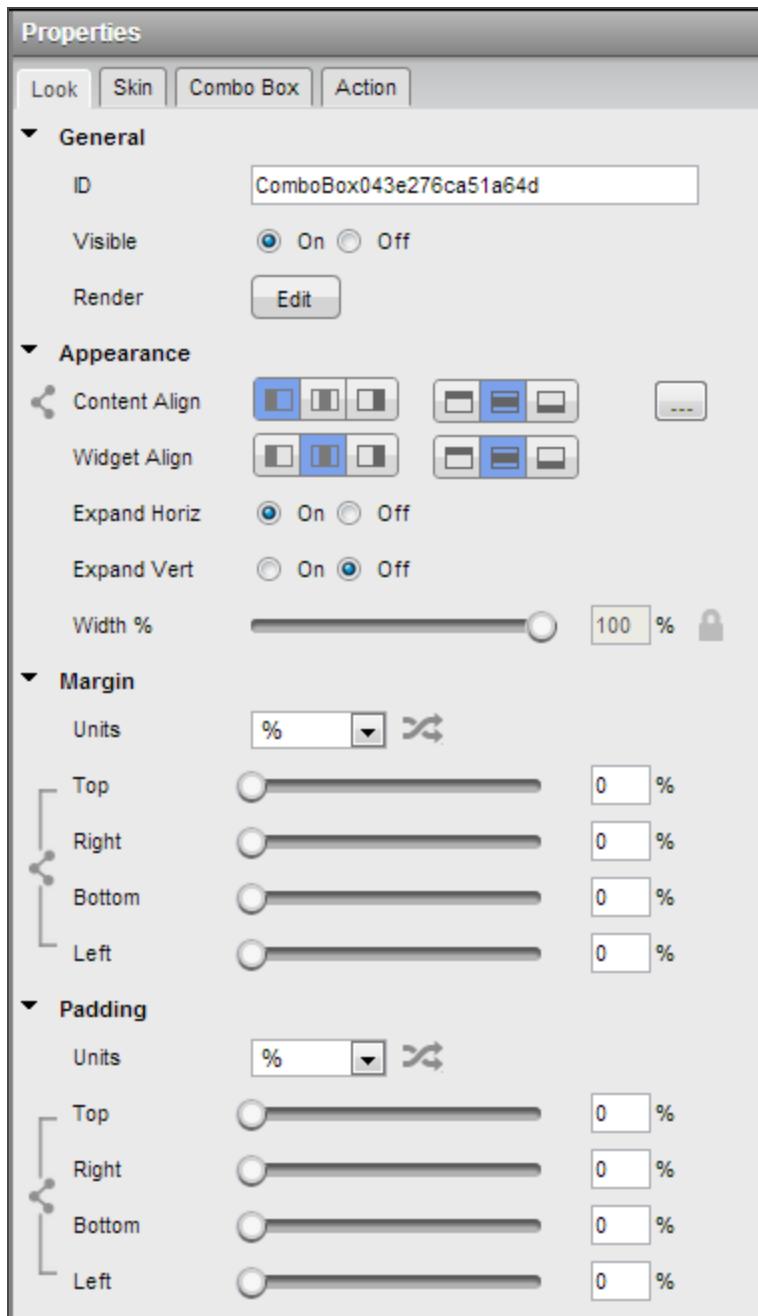
If you select the drop-down arrow on a ComboBox, a list containing a list of items (values) is displayed. When you select an item from the list, the selected item is displayed on the ComboBox.

A ComboBox is similar to a [ListBox](#). However, unlike the ListBox, you can only select a single item at a time.

**Note:** A ComboBox widget is available only for a Flex Form.

### 5.10.1 Look

This section details the **Properties > Look** tab of a **ComboBox** widget.



To specify the look properties of a ComboBox widget, refer section: [Look](#).

### 5.10.2 Skin

Skin defines properties such as background color, borders and shadows for a widget. They also, define the widget font (applicable only if text is displayed on a widget.)

For a ComboBox widget, you can apply a skin for the following states:

- Normal: Specifies the look and feel of the widget when not in focus.
- Focus: Specifies the skin that is applied when there is focus on a widget.
- Blocked UI: Specifies the skin that must be used to block the interface until the action in progress (for example, a service call) is completed.

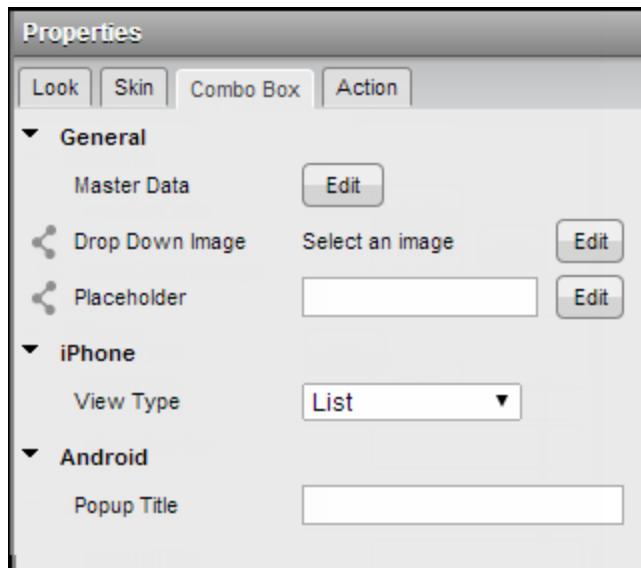
**Note:** Blocked UI is available only on SPA platforms.

- Hover Skin: Specifies the look and feel of a widget when the cursor hovers on the widget.

**Note:** Hover Skin is available only on the Windows Tablet platform.

Refer to section [Working with Skins](#) to know more about adding a skin to a widget.

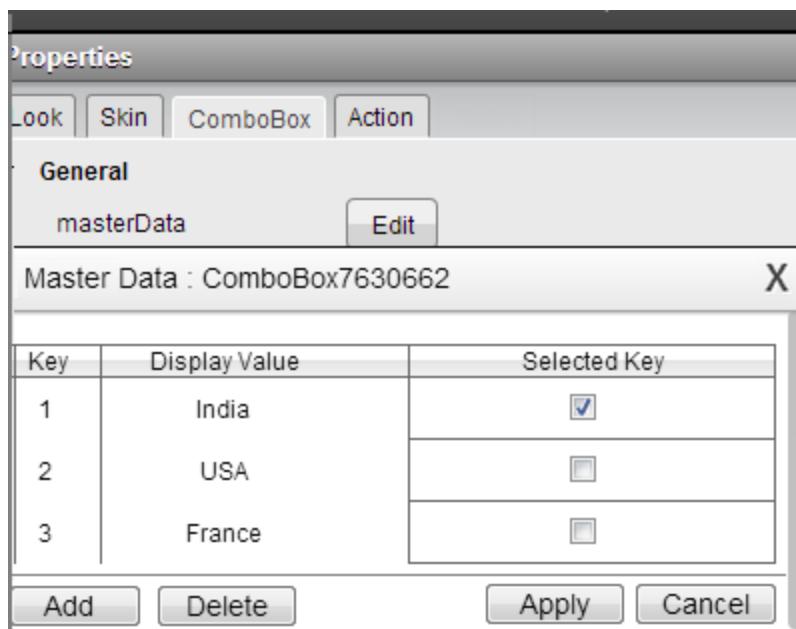
### 5.10.3 Platform Specific Properties



### 5.10.3.1 Master Data

Specifies the set of values that must be displayed for the user to make a selection from the available choices.

For specifying these values, click the **Edit** button against the **Master Data** to open the **Master Data : ComboBox** dialog box.



Master Data contains the following columns:

- Key: It is the unique identifier of each Combo Box value.
- Display Value: This value is displayed in the ComboBoxGroup widget on a form.
- Select Key : Allows you to select an individual Combo Box value from the group.

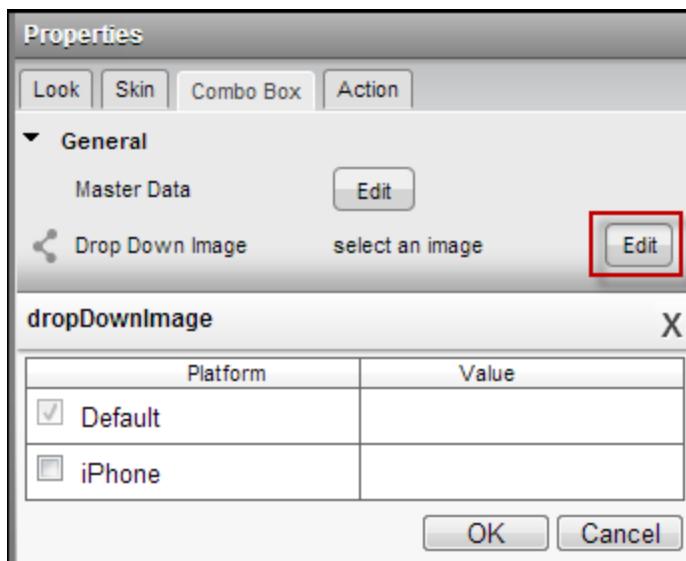
If required, to add more combo box values to a widget, click **Add**. To delete a combo box row, click inside a cell , and then click **Delete**.

After providing the required details and click **Apply** to create master data.

### 5.10.3.2 Drop Down Image

Specifies the image to be used for the drop-down box indicator. The image you specify is used to depict the drop-down box. If you do not specify an image, the drop-down box displays the default image (inverted triangle).

To provide a platform specific image, click the **Edit** button against the **Drop Down Image** to open **dropDownImage** dialog box.



Select the desired platform and click inside corresponding **value** field. From the **Select Image** dialog field, you can either,

- Select an available image.
- Provide image url.

### 5.10.3.3 Placeholder

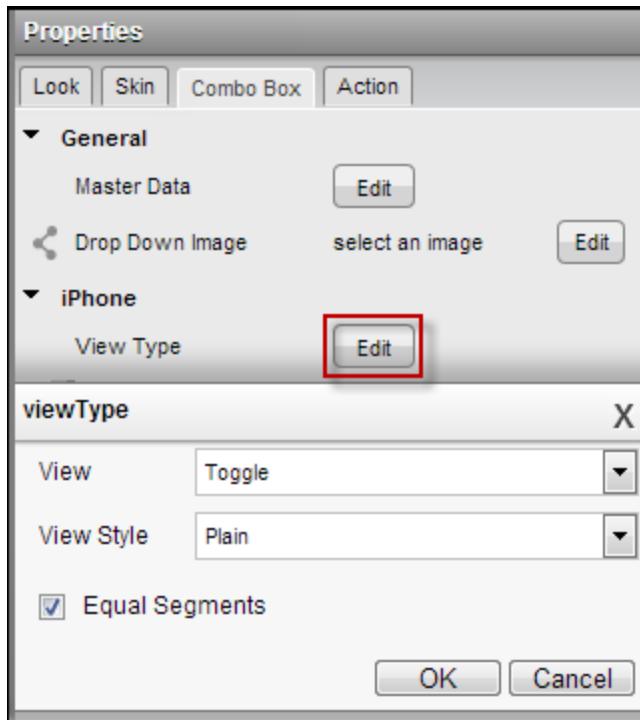
Specifies the temporary or substitute text (a hint provided as a word or phrase) that must be displayed on the ComboBox until the actual selection is made.

**Note:** This property is specific to the iOS and Windows platform.

#### 5.10.3.4 View Type

Specifies the view mode of the ComboBox.

To select a view type, click the **Edit** button against the **View Type** field to open the **View Type** dialog box.



Following are the available options:

- List
- Table
- Toggle
- On-screen Wheel

For Toggle, you can further select one of the following View Styles:

- Plain
- Bordered
- Bar

Apart from the above options, you can also enable and provide **Tint Color** for this widget.

- Enable Tint Color: To provide a tint color when the view type is **Toggle**, select **True** option.
- Tint Color: This option is Enabled when **Tint Color** is set to true. For selecting a tint color, click the color picker to open the color selection dialog box and then Select a desire color.

**Note:** This property is specific to the iOS7 platform.

Click the **Equal Segments** check box to distribute the segments in equal proportions.

**Note:** This property is specific to the iOS platform.

#### 5.10.3.5 Popup Title

Specifies the title text to be displayed for the ComboBox.

**Note:** This property is specific to the Android platform.

#### 5.10.3.6 Tool Tip

Specifies the hint text when the cursor hovers over a widget, without clicking it. The text entered in the tooltip appears as a small box when the cursor hovers over a widget.

**Note:** This property is specific to the Windows Tablet platform.

### 5.10.4 Actions

ComboBox widget has the following action associated with it:

- **onSelection:** This action is triggered when you select or deselect any item in ComboBox.

For more information on using the above action, refer to section [Action Editor](#).

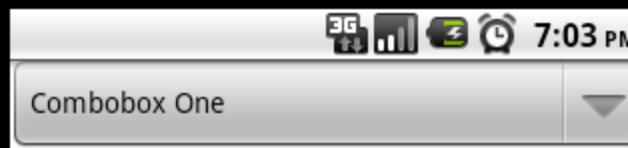
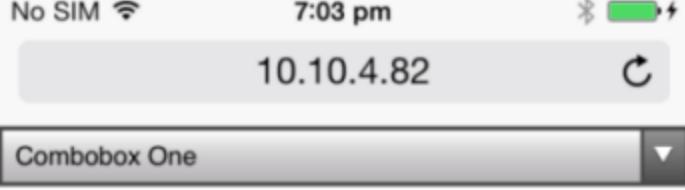
### 5.10.5 Placement Inside a Widget

A ComboBox can be directly added inside the following widgets:

| Widget    | ComboBox placement inside a widget                        |
|-----------|---|
| Form      | Yes   |
| HBox      | Yes   |
| VBox      | Yes   |
| ScrollBar | Horizontal Orientation - yes<br>Vertical Orientation- Yes |
| Tab       | Yes   |
| Segment   | No  |
| Popup     | Yes   |
| Template  | Header- No<br>Footer- No                                  |

### 5.10.6 Widget Appearance on Platforms

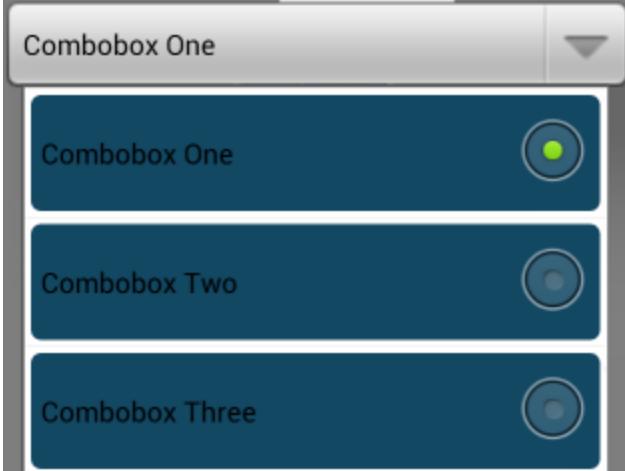
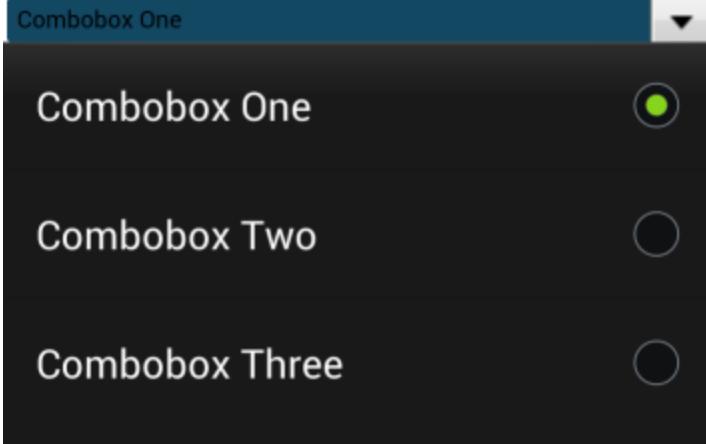
The appearance of the ComboBox with the default properties on various platforms is as follows:

| Platform                      | Appearance  |
|-------------------------------|---|
| Visualizer (Android : Native) |   |
| Androd                        |   |
| iOS                           |   |
| SPA                           |  |

#### 5.10.6.1 UI Behavior

When you click a ComboBox, the UI behavior is not the same on all the platforms.

The UI behavior of a ComboBox on various platforms is as follows:

| Platform | Appearance  |
|----------|---|
| Android  |  A screenshot of an Android combobox. The title bar says "Combobox One". The dropdown menu contains three items: "Combobox One" (selected, indicated by a green dot in a circle), "Combobox Two", and "Combobox Three".  |
| iOS      |  A screenshot of an iOS combobox. The title bar says "Item1". Below it are buttons for "Previous" and "Next", and a "Done" button. The dropdown menu shows three items: "item1" (selected, highlighted in blue), "Item2", and "Item3".  |
| SPA      |  A screenshot of a SPA combobox. The title bar says "Combobox One". The dropdown menu lists three items: "Combobox One" (selected, indicated by a green dot in a circle), "Combobox Two" (unselected, indicated by an empty circle), and "Combobox Three" (unselected, indicated by an empty circle). |

## 5.11 DataGrid

DataGrid widget allows you to present a collection of data in rows and columns (tabular format).

DataGrid also supports common table formatting options, such as alternating the row background color, customizing the gridline, and ability to hide or show headers.

You can use a DataGrid widget to show a read-only view of a small amount of data in a tabular format.

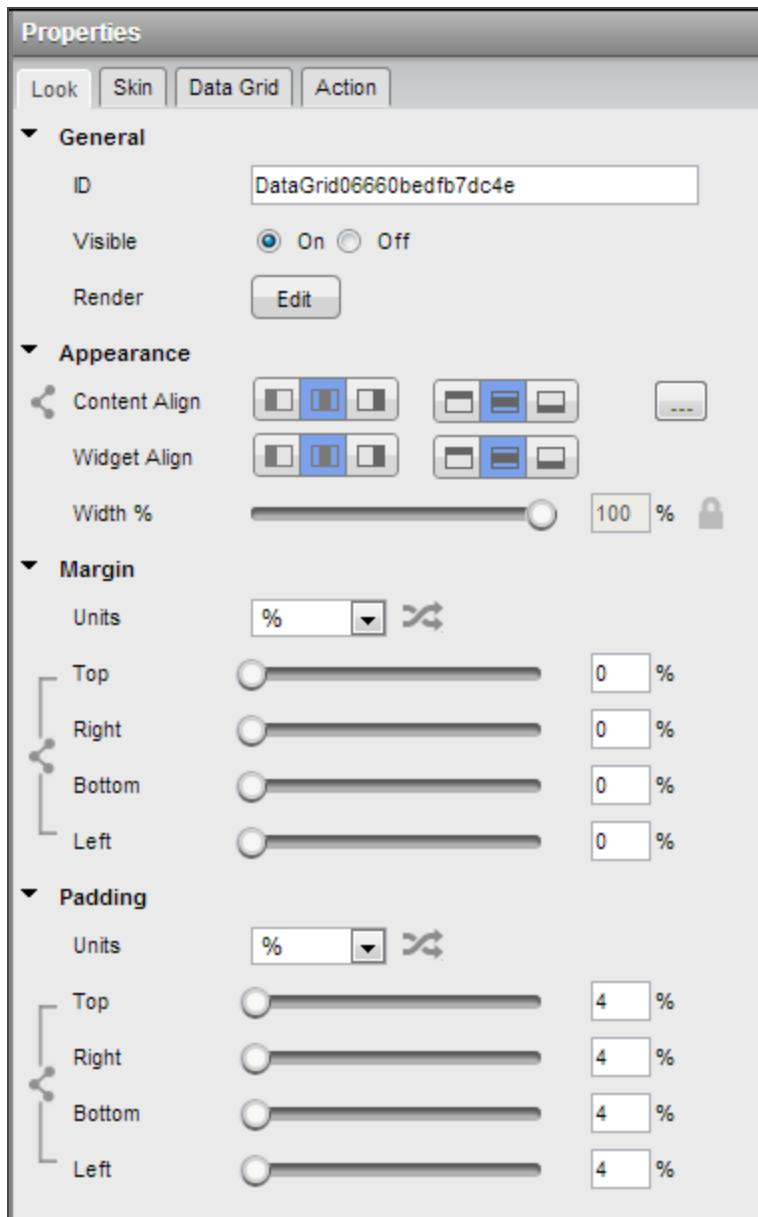
### 5.11.1 Important Considerations

The following are important considerations for a DataGrid Widget.

- To set the data, you must first specify the rows and columns using the [data](#) property.
- If the DataGrid supports the [Multi-Select](#) property , you must ensure that you have specified the [Row - Focus](#) property. Else, you will not be able to distinguish multiple selections.

### 5.11.2 Look

This section details the **Properties > Look** tab of a DataGrid widget.



To specify the look properties of a DataGrid widget, refer section: [Look](#).

### 5.11.3 Skin

Skin defines properties such as background color, borders and shadows for a widget. They also, define the widget font (applicable only if text is displayed on a widget.)

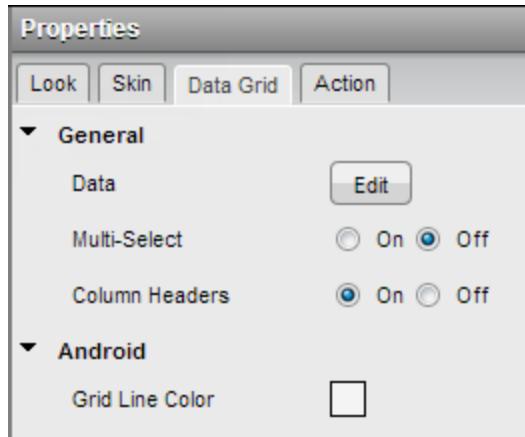
For a DataGrid widget, you can apply a skin for the following states:

- Header: Specifies the skin that is applied to the Header row.
- Row: Specifies the skin that must be applied when the row is not in focus.
- Row - Alternate: Specifies the skin that is applied to the alternative rows.
- Row - Focus: Specifies the skin that must be applied when the row is in focus.
- Hover Skin: Specifies the look and feel of a widget when the cursor hovers on the widget.

**Note:** Hoover Skins is available only on the Windows (native) Tablet platform.

Refer to section [Working with Skins](#) to know more about adding a skin to a widget.

#### 5.11.4 Platform Specific Properties



##### 5.11.4.1 Data

Represents the data displayed in each cell of the data grid.

To input the data, click the **Edit** button against the **Data** field. This results in opening the **Master Data : DataGrid**.

Master Data consist of two tabs:

- Columns

- Rows

### Columns

Master Data: DataGrid079f570b1aead43 X

| Columns |        | Rows        |           |         |           |             |     |
|---------|--------|-------------|-----------|---------|-----------|-------------|-----|
| *ID     | Header | Column Type | *Width(%) | Sort    | Alignment | onClick     |     |
| col1    | Car    | Text ▼      | 33        | false ▼ | Center ▼  | Not Defined | ... |
| col2    | Model  | Text ▼      | 34        | false ▼ | Center ▼  | Not Defined | ... |
| col3    | Image  | Image ▼     | 33        | false ▼ | Center ▼  | Not Defined | ... |

Add Delete Cancel OK

Provide details such as:

- ID: Specifies the unique identifier of a column.
- Header: Specifies the Column header.
- Column Type: Specifies the column to display either as text or image.
  - Text
  - Image
- Width (%): Specify the width of each column in percentage.
- Sort: Specifies if sorting is allowed for the column.
- Alignment: Specifies the alignment of the content in each column. Following alignment options are available:
  - Top-Left
  - Top-Center

- Top-Right
  - Middle-Left
  - Center
  - Middle-Right
  - Bottom-Left
  - Bottom-Center
  - Bottom-Right
- OnClick: Specifies the action that takes place when you click on the header of a column. To specify an OnClick action for a column, click the ellipses button (...).

## Rows

| Car      | Model | Image  |
|----------|-------|--|
| BMW      | Z4    | bmw.png <input type="button" value="..."/> <input type="button" value="X"/>      |
| Ford     | Figo  | ford.png <input type="button" value="..."/> <input type="button" value="X"/>     |
| chrysler | 200   | chrysler.png <input type="button" value="..."/> <input type="button" value="X"/> |
| RC 41    | RC 42 | RC 43  |

All the fields in the **Rows** tab are based on the inputs that you have provided in the **Columns** tab and include:

- Column Headers: The **Header** specified in the **Columns** tab is displayed here.
- Table cells having **Column Type** as **Text**, behave as text fields. You can type the text within these cells.
- Table cells having **Column Type** as **Image**, behave as image fields. You can choose an image to be displayed in these cells. Click the ellipses button (...) in the image field to open **Select Image** dialog box. From the list, select an image or provide the url of the image and click **OK**. The selected image is displayed in the cell.
- Based on the **Alignment**, the text or the image are aligned within the cells.

#### 5.11.4.2 Multi-Select

When this property is enabled, you can choose multiple rows of DataGrid. [Row - Focus](#) skin is applied to those selected rows.

#### 5.11.4.3 Column Headers

This property controls the visibility of the column headers of the DataGrid.

#### 5.11.4.4 Grid Line Color

Specifies the color of the grid line.

**Note:** This property is specific to the Android platform.

#### 5.11.4.5 Tool Tip

Specifies the hint text when the cursor hovers over a widget, without clicking it. The text entered in the tooltip appears as a small box when the cursor hovers over a widget.

**Note:** This property is specific to the Windows Tablet platform.

### 5.11.5 Actions

DataGrid widget has the following action associated with it:

- `onRowSelected`: This action is invoked when a row is selected.

For more information on using the above action, refer to section [Action Editor](#).

### 5.11.6 Placement Inside a Widget

A DataGrid can be directly added inside the following widgets:

| Widget    | DataGrid placement inside a widget                        |
|-----------|---|
| Form      | Yes   |
| HBox      | Yes   |
| VBox      | Yes   |
| ScrollBar | Horizontal Orientation - Yes<br>Vertical Orientation- Yes |
| Tab       | Yes   |
| Segment   | No  |
| Popup     | Yes   |
| Template  | Header- No<br>Footer- No                                  |

### 5.11.7 Widget Appearance on Platforms

The following is the appearance of the DataGrid widget on various platforms with a specified Master Data, Header skin, and row skin:

| Platform | Appearance   |                |         |
|----------|--|----------------|---------|
| Android  | Account Type   | Account Number | Balance |
|          | Checking   | 490            | \$400   |
|          | Checking   | 495            | \$2000  |
|          | Checking   | 496            | \$1000  |
|          | Checking   | 500            | \$2000  |
| iOS      |  |                |         |
| SPA      | Account Type   | Account Number | Balance |
|          | Checking   | 490            | \$400   |
|          | Checking   | 495            | \$2000  |
|          | Checking   | 496            | \$1000  |
|          | Checking   | 500            | \$2000  |

## 5.12 Image2

Image widget is a non-interactive widget that you can use to display a graphic (local or remote) from a PNG file. You can use an Image widget in the following scenarios:

- Display your company's logo.
- Display a snapshot.
- Provide an illustration.

### 5.12.1 Important Considerations

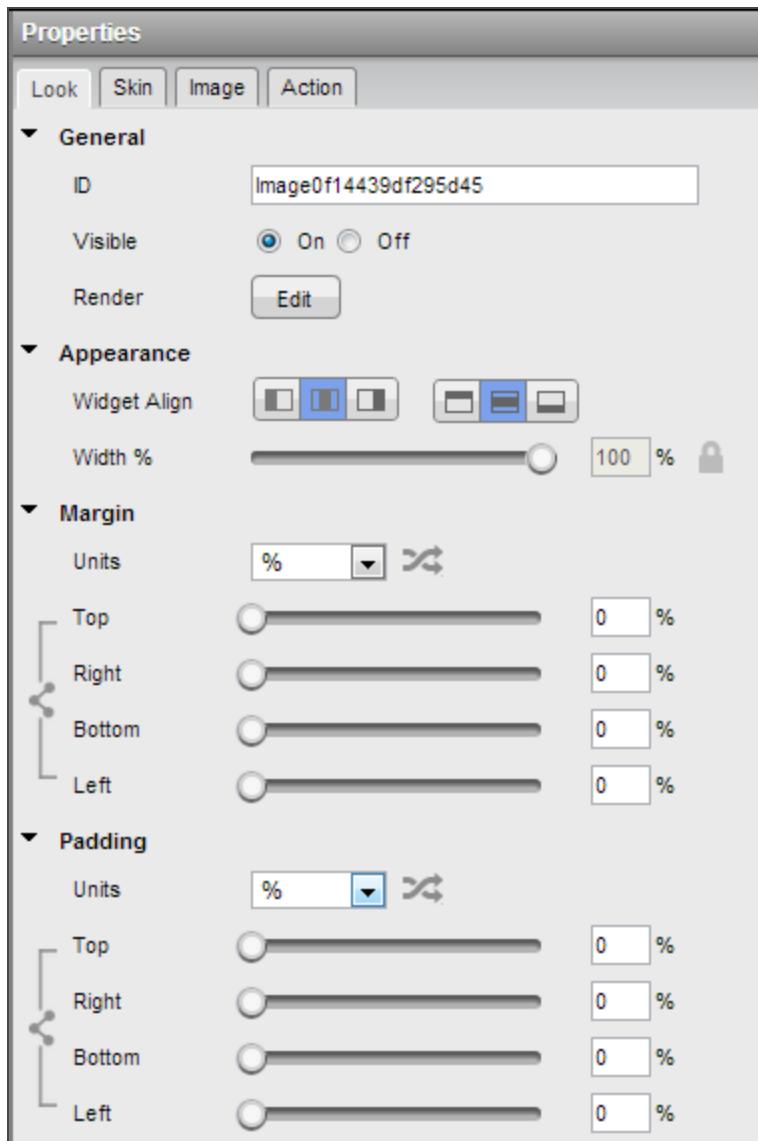
1. Before you can use an image in a project, you need to copy the image to an appropriate project subfolder. For more information, see [Images.htm](#).
2. Following are the important consideration for the Image widget concerning its naming conventions:
  - a. The file name must contain only lowercase characters.
  - b. The file name must start only with an alphabet.
  - c. The file name can contain numbers.
  - d. Do not use any reserved words or keywords ( of JavaScript) as the file names for the images.
  - e. Numerical characters.

The following table shows a few examples of valid and invalid file names for the images:

| Valid File Names    | Invalid File Names | Remarks  |
|---------------------|--------------------|--|
| myicon.png          | Myicon.png         | The file name is invalid because it contains uppercase character.                        |
| icon2.png           | Icon_2.png         | The file name is invalid because it contains uppercase character and an underscore.      |
| acctsummary.png     | aCCNT&summary.png  | The file name is invalid because it contains uppercase characters and special character. |
| accountdetails.png  | Details.png        | The file name is invalid because it contains uppercase character.                        |
| flightstatus123.png | continue.png       | The file name contains a Java keyword.   |

## 5.12.2 Look

This section details the **Properties > Look** tab of an **Image2** widget.



To specify the look properties of an Image2 widget, refer section: [Look](#).

### 5.12.3 Skin

Skin defines properties such as background color, borders and shadows for a widget. They also, define the widget font (applicable only if text is displayed on a widget.)

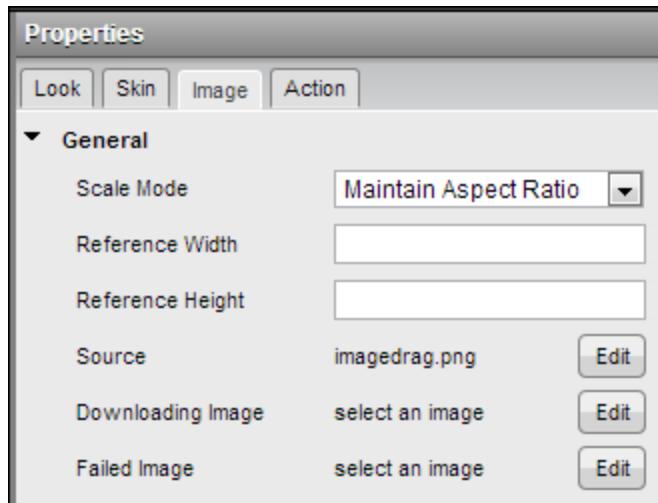
For the Image widget, you can apply a skin for the following states:

- Normal: Specifies the look and feel of the widget when not in focus.
- Hover Skin: Specifies the look and feel of a widget when the cursor hovers on the widget.

**Note:** Hoover Skins is available only on the Windows (native) Tablet platform.

Refer to section [Working with Skins](#) to know more about adding a skin to a widget.

#### 5.12.4 Platform Specific Properties



##### 5.12.4.1 Scale Mode

Specifies how the image's width and height are identified when those of the source image varies from the Image widget itself. Image Widget represents the underlying native widget which renders (and applies alignment to) the source image.

**Default:** Maintain Aspect Ratio

Following are the image scale options available:

- Fit To Dimensions
- Maintain Aspect Ratio

- Crop

#### 5.12.4.2 Reference Width

It is the reference image width in pixels. These are device independent Pixels specified against 163 dpi.

#### 5.12.4.3 Reference Height

It is the reference image height in pixels. These are device independent Pixels specified against 163 dpi.

#### 5.12.4.4 Source

Specifies the source of the image to be displayed. You can specify an image from the resources folder or specify a url of the image.

To specify an image, click **Edit** button against **Source** field to open **Src** dialog box. You can either:

- Select an available image.
- Provide image url.

#### 5.12.4.5 Downloading Image

Specifies the image to be displayed when the remote source is still being downloaded.

To specify an image, click **Edit** button against **Downloading Image** field to open **Downloading Image** dialog box. You can either:

- Select an available image.
- Provide image url.

#### 5.12.4.6 Failed Image

Specifies the image to be displayed when the remote resource is not available.

To specify an image, click **Edit** button against **Failed Image** field to open **Failed Image** dialog box.

You can either:

- Select an available image.
- Provide image url.

#### 5.12.4.7 Tool Tip

Specifies the hint text when the cursor hovers over a widget, without clicking it. The text entered in the tooltip appears as a small box when the cursor hovers over a widget.

**Note:** This property is specific to the Windows Tablet platform.

#### 5.12.5 Actions

Image widget has the following action associated with it:

- **onDownloadComplete:** This action is triggered when the image download from the URL is complete.

For more information on using the above action, refer to section [Action Editor](#).

#### 5.12.6 Placement inside a Widget

Below table list the widgets inside which an image can be directly added:

| Widget     | Image placement inside a widget                           |
|------------|---|
| Form       | Yes   |
| HBox       | Yes   |
| VBox       | Yes   |
| Scroll Box | Horizontal Orientation - Yes<br>Vertical Orientation- Yes |

|          |                                 |
|----------|---------------------------------|
| Widget   | Image placement inside a widget |
| Tab      | Yes                             |
| Segment  | Yes                             |
| Popup    | Yes                             |
| Template | Header- No<br>Footer- No        |

## 5.13 Label

Label widget is used to display non-editable text on the Form and is non-interactive.

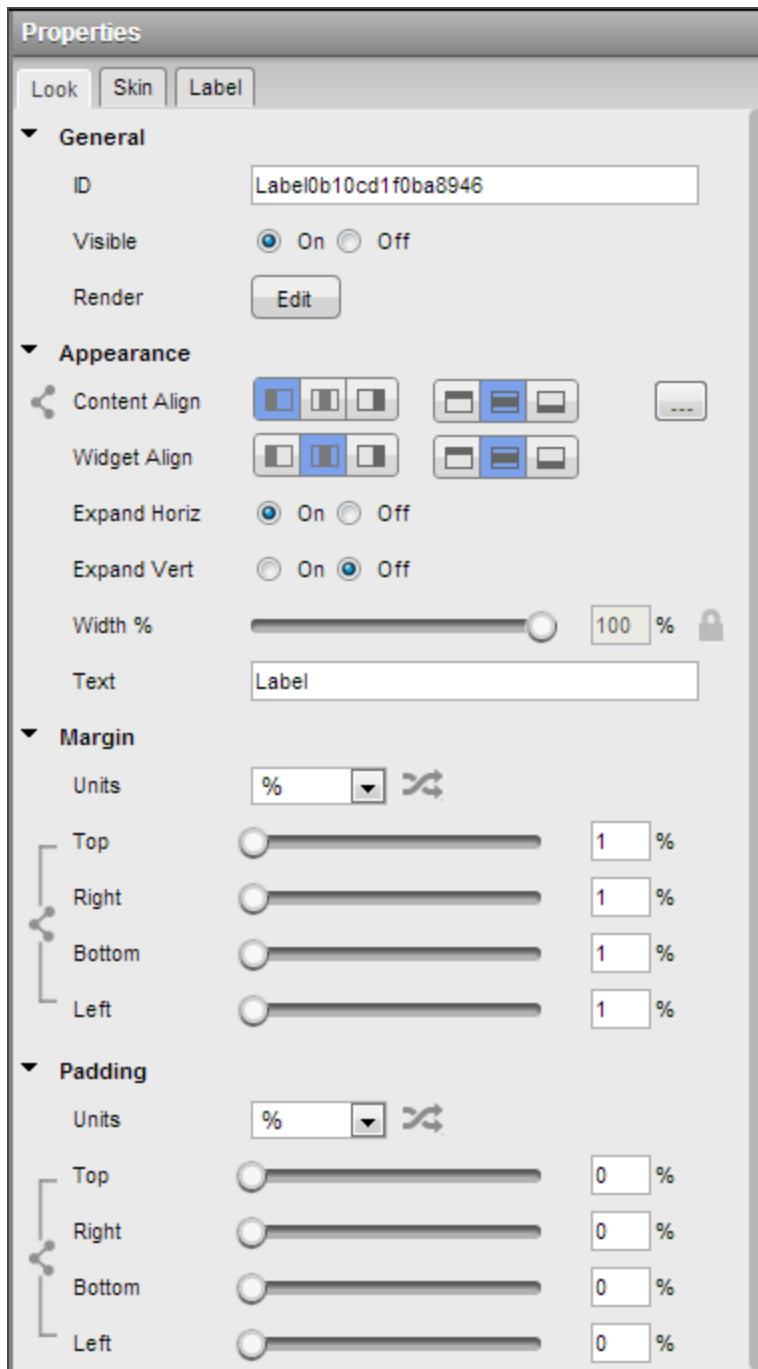
### 5.13.1 Important Considerations

The following is the important considerations for the Label widget.

- If the text in the Label is occupying more space than the allocated height of the Label widget, the Label is stretched vertically to accommodate the full text (infinite wrapping) and does not stretch in the horizontal direction.

### 5.13.2 Look

This section details the **Properties > Look** tab of a Label widget.



To specify the look properties of a Label widget, refer section: [Look](#).

### 5.13.3 Skin

Skin defines properties such as background color, borders and shadows for a widget. They also, define the widget font (applicable only if text is displayed on a widget.)

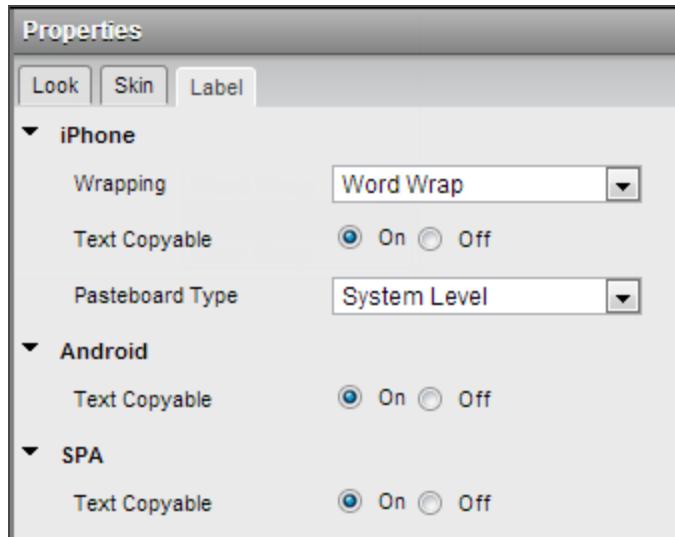
For the Label widget, you can apply a skin for the following states:

- Normal: Specifies the look and feel of the widget when not in focus.
- Hover Skin: Specifies the look and feel of a widget when the cursor hovers on the widget.

**Note:** Hoover Skins is available only on the Windows (native) Tablet devices.

Refer to section [Working with Skins](#) to know more about adding a skin to a widget.

### 5.13.4 Platform Specific Properties



#### 5.13.4.1 Wrapping

When the content of the label reaches the boundaries, it starts wrapping. While wrapping, two strategies can be applied:

- Word Wrapping: Wrap or clip the string only at word boundaries.
- Character Wrapping: Wrap or clip the string at the closest character boundary.

Following are the available options:

- Word Wrap: Specifies if the complete word must be moved to the next line when you reach the right margin. This is the default wrapping property.
- Char Wrap: Specifies if the characters in a word must be moved to the next line when you reach the right margin.

**Note:** This property is specific to the iOS platform.

#### 5.13.4.2 Text Copyable

You can choose whether to allow copying the Label widget's text, and paste this text into other widgets such as TextBox2 and TextArea2.

**Note:** This property is specific to the iOS platform,

#### 5.13.4.3 Pasteboard Type

This property enables an application to share data within the application or between the applications. Another object in the same or a different application then reads that data from the pasteboard and presents it to you in a new location, usually during a paste operation.

**Note:** The destination text box and the source text box must have the same pasteboard type. For example, if you set the pasteboard type as *App Level Persistent*, you can paste the text only to another textbox whose pasteboard type is also set to *App Level Persistent*.

The different pasteboard types are:

- Default: If you choose this option, the value selected in the application properties is applied.
- **System Level:** This is the default selection and if this option is unchanged, the text copied from a Label can be pasted across different applications on the device. Even if you exit the source application, the copied text persists in the memory and can be pasted across applications or within the same application.
- App Level Persistent: If you select this option , the text copied from a Label can be pasted in TextArea or TextBox (with the pasteboard type set as App Level Persistent) within the same application. Even if you close the application, the copied text persists in the memory and can be copied to another TextArea whose pasteboard type is App Level Persistent, when you restart that application.
- App Level Non-Persistent: If you select this option, the text copied from a Label can be pasted in TextArea or TextBox within the same application. This text is not retained in the memory when you close the application.

**Note:** This property is specific to the iOS platform.

#### 5.13.4.4 Text Copyable

You can choose whether to allow copying the Label widget's text, and paste this text into other widgets such as TextBox2 and TextArea2.

#### 5.13.4.5 Tool Tip

Specifies the hint text when the cursor hovers over a widget, without clicking it. The text entered in the tooltip appears as a small box when the cursor hovers over a widget.

**Note:** This property is specific to the Windows Tablet platform.

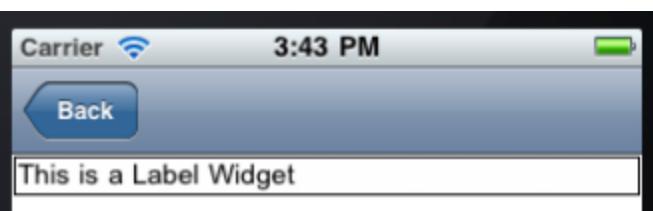
### 5.13.5 Placement Inside a Widget

A label can be directly added inside the following widgets:

|           |   |
|-----------|---|
| Widget    | Label placement inside a widget                           |
| Form      | Yes   |
| HBox      | Yes   |
| VBox      | Yes   |
| ScrollBox | Horizontal Orientation - Yes<br>Vertical Orientation- Yes |
| Tab       | Yes   |
| Segment   | Yes   |
| Popup     | Yes   |
| Template  | Header- No<br>Footer- No                                  |

### 5.13.6 Widget Appearance on Platforms

The appearance of the Label widget on various platforms is as follows:

| Platform | Appearance   |
|----------|--|
| Android  |  |
| iOS      |   |
| SPA      |  |

## 5.14 Line

The Line widget allows you to draw a horizontal or a vertical line on a Form. It is used as a separator between widgets for a better visual experience.

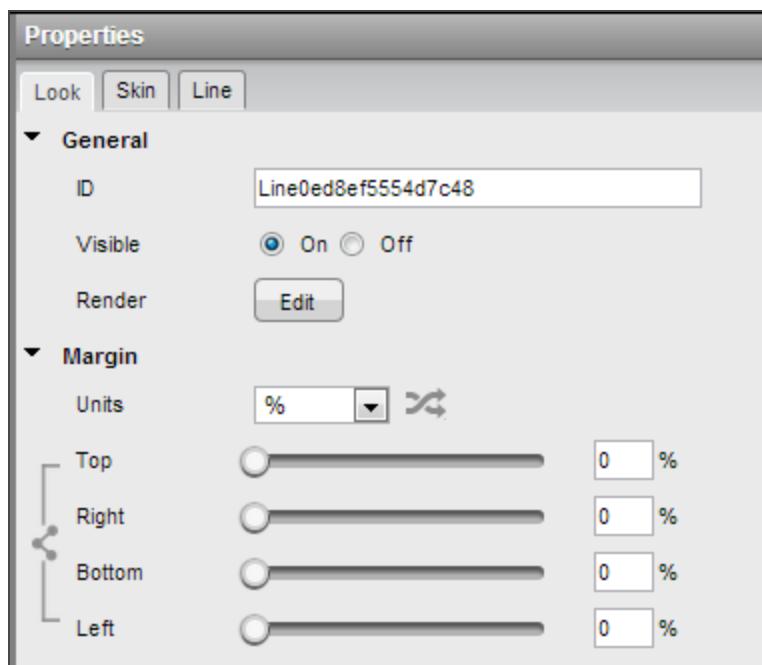
### 5.14.0.1 Important Considerations

The following is an important considerations for an Line Widget.

- A Line widget is available only for a Flex Form.
- Orientation of a line can be horizontal or vertical. The orientation of the line depends on the widget it is placed in. If a Line widget is placed in an HBox, it becomes a VLine and if it placed in a VBox or a Form, it becomes an HLine.

### 5.14.0.2 Look

This section details the **Properties > Look** tab of a Line widget.



To specify the look properties of a Line widget, refer section: [Look](#).

### 5.14.0.3 Skin

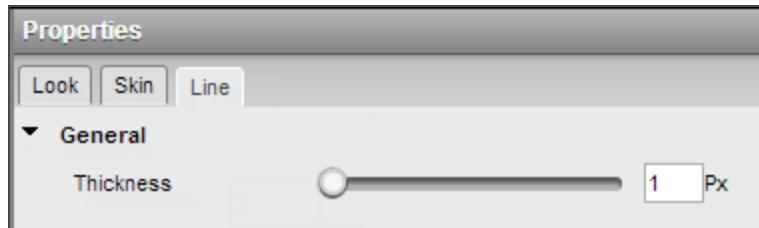
Skin defines properties such as background color, borders and shadows for a widget. They also, define the widget font (applicable only if text is displayed on a widget.)

For the Label widget, you can apply a skin for the following states::

- Normal: Specifies the look and feel of the widget when not in focus.

Refer to section [Working with Skins](#) to know more about adding a skin to a widget.

### 5.14.0.4 Platform Specific Properties



#### Thickness

Specifies the thickness of the widget in pixels. The pixel values are scaled to density specific pixels by the respective platforms.

#### Tool Tip

Specifies the hint text when the cursor hovers over a widget, without clicking it. The text entered in the tooltip appears as a small box when the cursor hovers over a widget.

**Note:** This property is specific to the Windows Tablet platform.

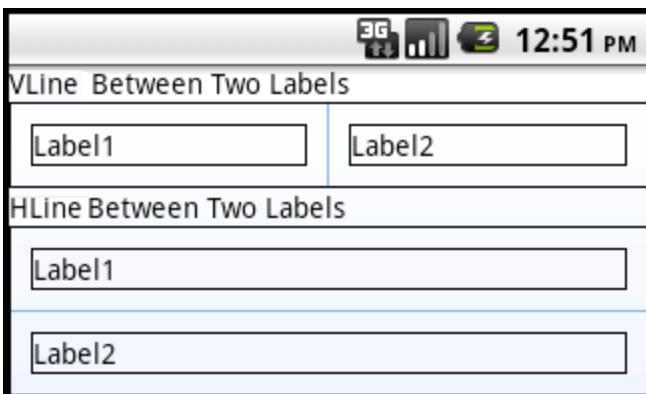
### 5.14.0.5 Placement Inside a Widget

A line can be directly added inside the following widgets:

|           |   |
|-----------|---|
| Widget    | Line placement inside a widget                            |
| Form      | Yes   |
| HBox      | Yes   |
| VBox      | Yes   |
| ScrollBox | Horizontal Orientation - Yes<br>Vertical Orientation- Yes |
| Tab       | Yes   |
| Segment   | Yes   |
| Popup     | Yes   |
| Template  | Header- No<br>Footer- No                                  |

#### 5.14.0.6 Widget Appearance on Platforms

The appearance of the Line widget on various platforms is as follows:

| Platform | Appearance  |
|----------|---|
| Android  |  |

| Platform | Appearance   |
|----------|--|
| iOS      |  <p>The screenshot displays an iOS application interface. At the top, there is a navigation bar with a "Back" button on the left and the time "3:43 PM" in the center. Below the navigation bar, the main content area has a title "VLine Between Two Labels". Under this title, there are two labels: "Label1" on the left and "Label2" on the right, separated by a vertical line. Below this section, another title "HLine Between Two Labels" is present, followed by two labels: "Label1" on the left and "Label2" on the right, separated by a horizontal line.</p> |

## 5.15 Link

Link widget allows you to define a hyperlink that you can interact with (select and click) and navigate to an external location or a location within the application.

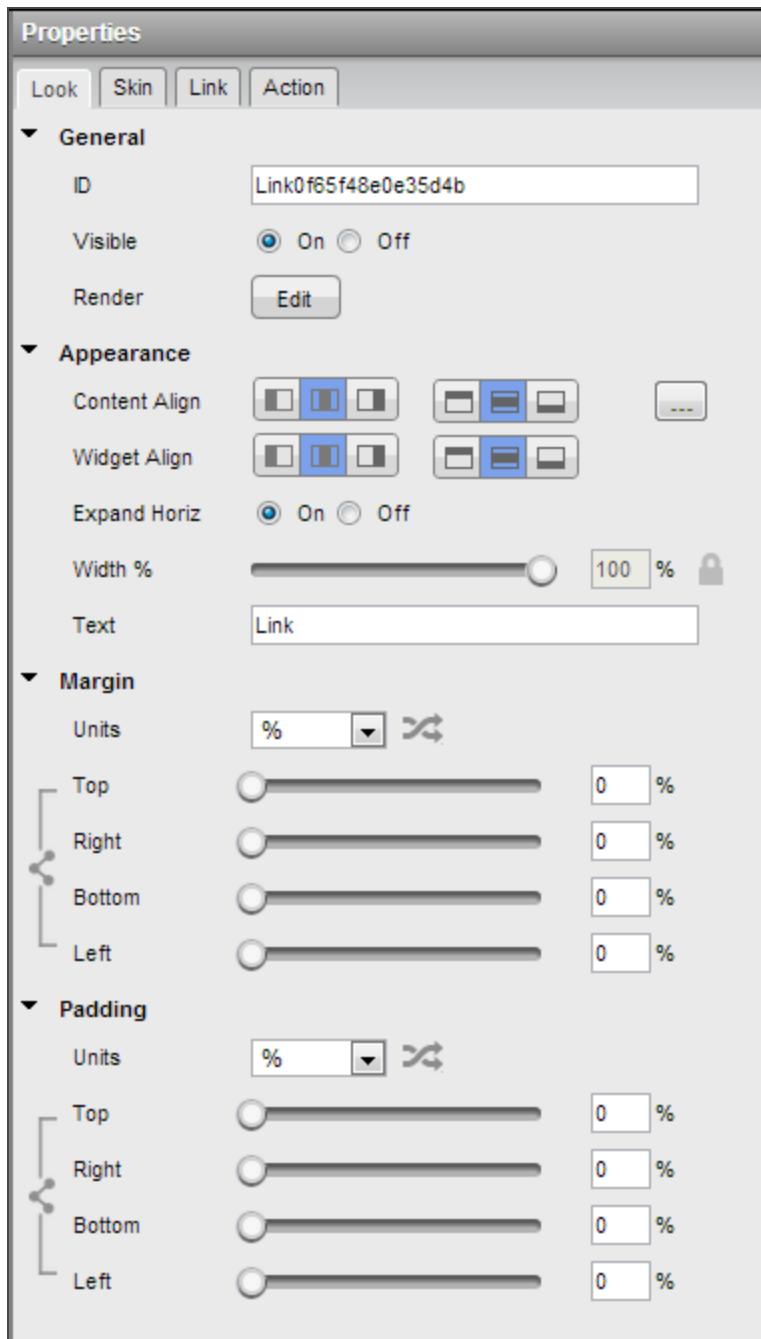
### 5.15.1 Important Considerations

The following are important considerations for an Link Widget.

- A Link widget is available only for a Flex Form.
- If you do not specify a skin, the default skin is applied to the link (link appears in blue font and is underlined).
- If you do not specify a Focus, the default focus skin is applied to the link (link appears in black font and is underlined).
- If you specify a skin or Focus without an underline, when rendered, the link will appear without an underline on the platform.
- For Server side Mobile Web (basic), font styles are not supported.

### 5.15.2 Look

This section details the **Properties > Look** tab of a Link widget.



To specify the look properties of a Link widget, refer section: [Look](#).

### 5.15.3 Skin

Skin defines properties such as background color, borders and shadows for a widget. They also, define the widget font (applicable only if text is displayed on a widget.)

For the Link widget, you can apply a skin for the following states:

- Normal: Specifies the look and feel of the widget when not in focus.
- Focus: Specifies the skin that is applied when there is focus on a widget.
- Blocked UI: Specifies the skin that must be used to block the interface until the action in progress (for example, a service call) is completed.

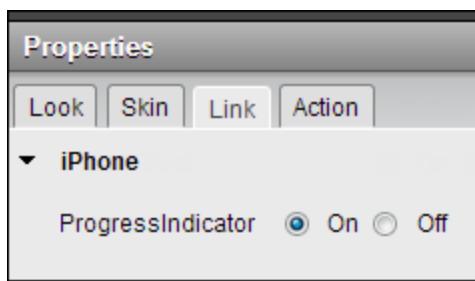
**Note:** Blocked UI is available only for SPA platforms.

- Hover Skin: Specifies the look and feel of a widget when the cursor hovers on the widget.

**Note:** Hoover Skins is available only on the Windows (native) Tablet platform.

Refer to section [Working with Skins](#) to know more about adding a skin to a widget.

### 5.15.4 Platform Specific Properties



#### 5.15.4.1 Progress Indicator

Specifies if the progress indicator must be displayed when the link is clicked.

- Click **On** to enable progress indicator.



- Click **Off** to disable progress indicator. (default)

**Note:** This property is specific to the iOS platform.

#### 5.15.4.2 Tool Tip

Specifies the hint text when the cursor hovers over a widget, without clicking it. The text entered in the tooltip appears as a small box when the cursor hovers over a widget.

**Note:** This property is specific to the Windows Tablet platform.

#### 5.15.5 Actions

Link widget has the following action associated with it:

- **onClick:** This action is invoked by the platform when the user performs a click action on the link.

For more information on using the above action, refer to section [Action Editor](#).

#### 5.15.6 Placement Inside a Widget

A link can be directly added inside the following widgets:

| Widget | Link placement inside a widget |
|--------|--------------------------------|
| Form   | Yes                            |
| HBox   | Yes                            |
| VBox   | Yes                            |

|           |   |
|-----------|---|
| Widget    | Link placement inside a widget                            |
| ScrollBar | Horizontal Orientation - Yes<br>Vertical Orientation- Yes |
| Tab       | Yes   |
| Segment   | Yes   |
| Popup     | Yes   |
| Template  | Header- No<br>Footer- No                                  |

### 5.15.7 Widget Appearance on Platforms

The following is the appearance of a Link widget on various platforms:

| Platform      | Appearance                            |
|---------------|---------------------------------------|
| Android       | <a href="#">This is a Link widget</a> |
| iPhone        | <a href="#">This is a Link widget</a> |
| Windows Phone | <a href="#">This is a Link widget</a> |
| SPA           | <a href="#">This is a Link widget</a> |

## 5.16 ListBox

List Box displays a list of items as a drop-down box and allows you to select a single item at a time.

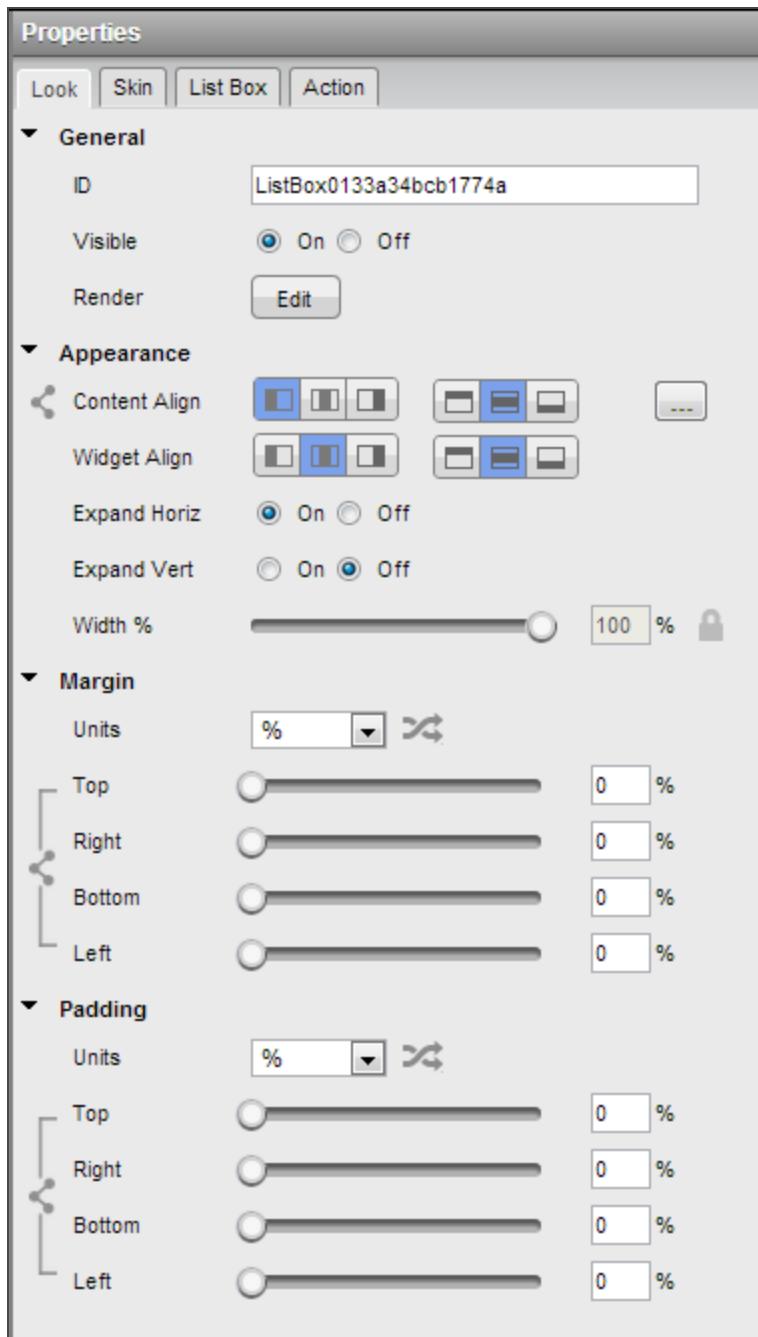
### 5.16.1 Important Considerations

The following are important considerations for a ListBox Widget.

- Shows dynamic set of data in a fixed space.
- The master data of choices should be limited and fetched in a separate service call.

### 5.16.2 Look

This section details the **Properties > Look** tab of a ListBox widget.



To specify the look properties of a ListBox widget, refer section: [Look](#).

### 5.16.3 Skin

Skin defines properties such as background color, borders and shadows for a widget. They also, define the widget font (applicable only if text is displayed on a widget.)

For the ListBox widget, you can apply a skin for the following states:

- Normal: Specifies the look and feel of the widget when not in focus.
- Focus: Specifies the skin that is applied when there is focus on a widget.
- Native Field: Specifies the skin that is applied to each item in the native popup that appears when you click on the ListBox.
- Native Field Focus: Specifies the skin that is applied to a focused item in the native popup that appears when you click on the ListBox.
- Blocked UI: Specifies the skin that must be used to block the interface until the action in progress (for example, a service call) is completed.

**Note:** Blocked UI is available only for SPA platforms.

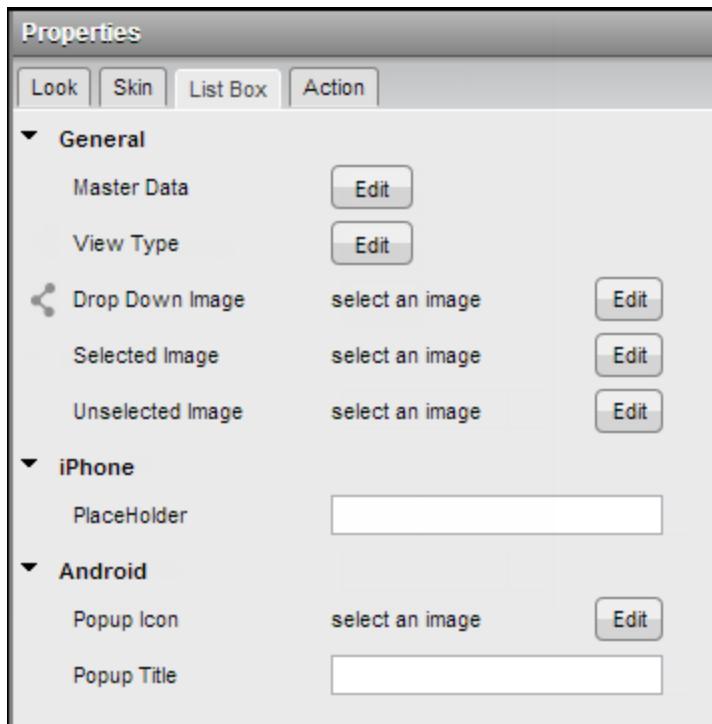
- Placeholder: This property reads the font color set in the skin and ignores the other attributes. Android does not support setting a background color for a placeholder.
- Hover Skin: Specifies the look and feel of a widget when the cursor hovers on the widget.

**Note:** Hover Skins is available only on the Windows (native) Tablet platform.

Refer to section [Working with Skins](#) to know more about adding a skin to a widget.

### 5.16.4 Platform Specific Properties

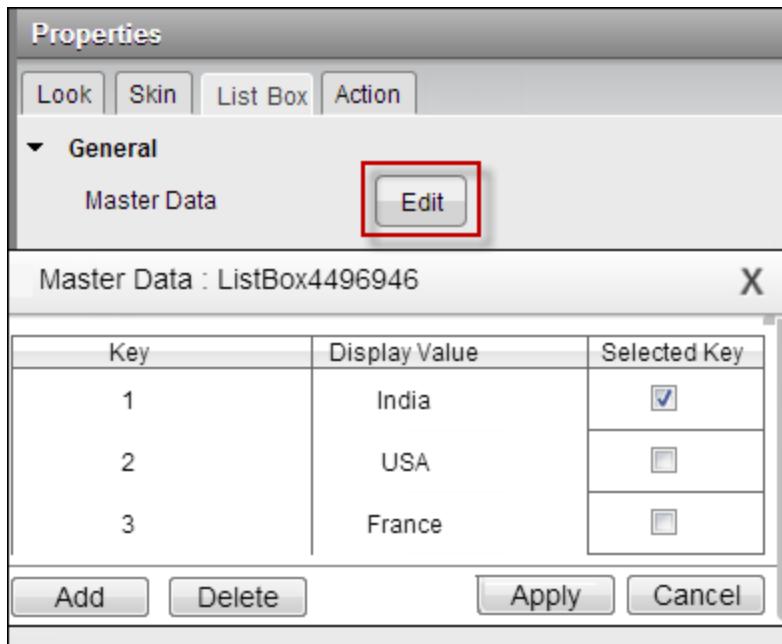
**Note:** In this section, the properties that can be forked are identified by an icon ( located to the left of the property. For more details, refer to section [Forking a Widget property](#).



#### 5.16.4.1 Master Data

Specifies the set of values that must be displayed for the user to make a selection from the available choices.

For specifying these set of values, click the **Edit** button against the **Master Data** to open the **Master Data : ListBox** dialog box.



Master Data contains the following columns:

- Key: It is the unique identifier of each List Box value.
- Display Value: This value is displayed in the List Box widget on a form.
- Select Key : Allows you to select an individual List Box value from the group.

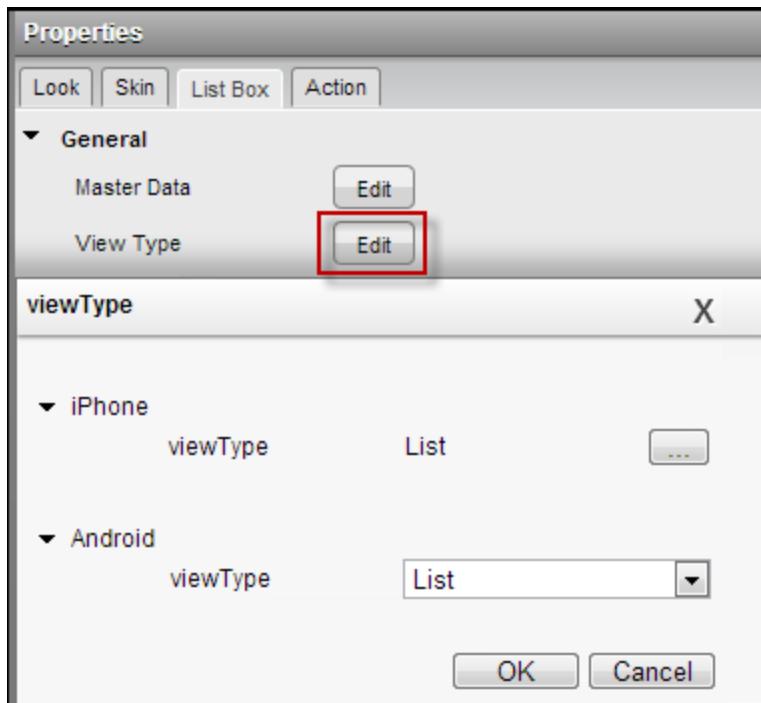
If required, to add more List box values to a widget, click **Add**. To delete a List box row, click inside a cell, and then click **Delete**.

After providing the required details and click **Apply** to create master data.

#### 5.16.4.2 View Type

Specifies the view mode of the List Box.

To select a view type, click the **Edit** button against the **View Type** field to open the **View Type** dialog box.



### **View Type - iOS**

Following are the options available for iOS platform:

- List
- Table
- Toggle
- On-screen Wheel

For Toggle, you can further select one of the following View Styles:

- Plain
- Bordered
- Bar

Apart from the above options, you can also enable and provide **Tint Color** for this widget.

- Enable Tint Color: To provide a tint color when the view type is Toggle, select **True** option.
- Tint Color: This option is Enabled when **Tint Color** is set to true. For selecting a tint color, click the color picker to open the color selection dialog box and then Select a desire color.

**Note:** This property is specific to the iOS7 platform.

Click the **Equal Segments** check box to distribute the segments in equal proportions.

#### **View Type - Android**

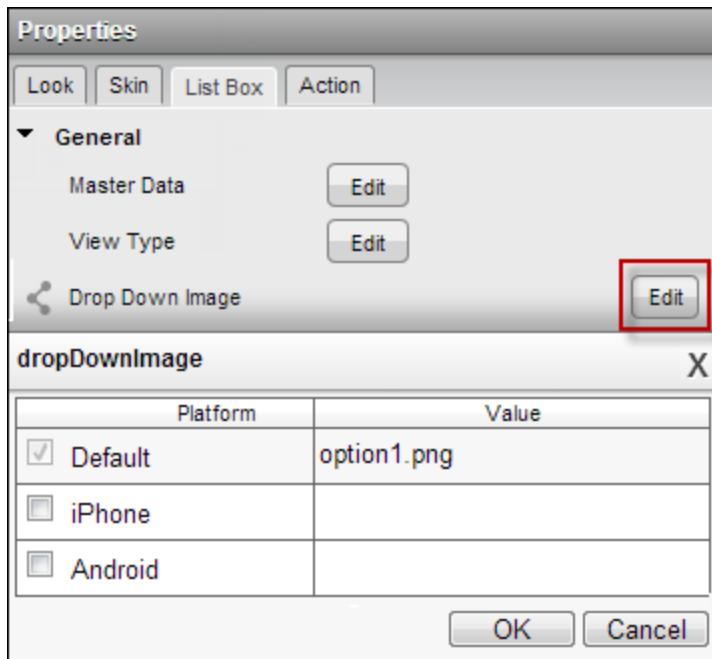
Following are the options available for Android platform:

- List
- Spinner

#### **5.16.4.3 Drop Down Image**

Specifies the image to be used for the drop-down box indicator (inverted triangle by default). The image you specify is used to depict the drop-down box. If you do not specify an image, the drop-down box displays the default image (inverted triangle).

For specifying a default image or a platform specific image, click the **Edit** button against the **Drop Down Image** to open **dropDownImage** dialog box.



Select the desired platform and click inside corresponding **value** field. From the **Select Image** dialog field, you can either,

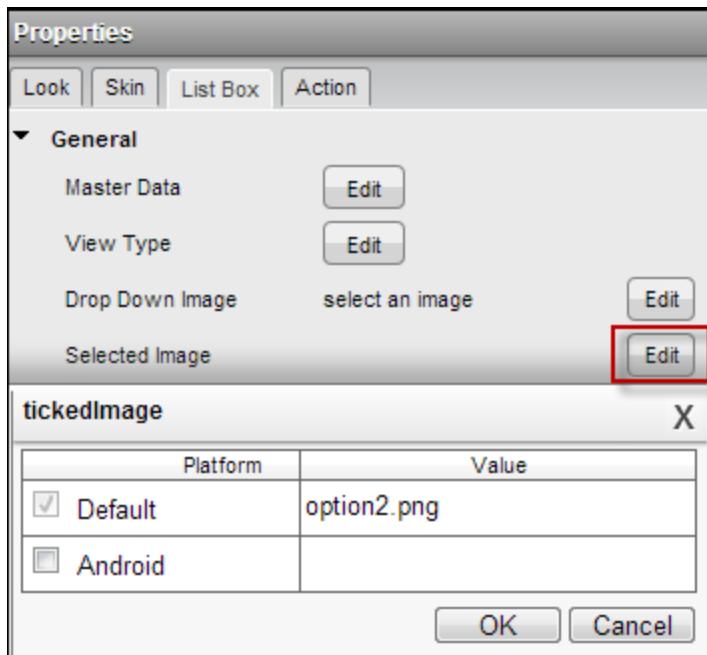
- Select an available image.
- Provide image url.

#### 5.16.4.4 Selected Image

Specifies the image to be displayed when you make a selection.

**Note:** If you specify a **Selected Image**, ensure that you also specify an **Unselected Image**. If not specified, the behavior will be undefined

For specifying a default image or a platform specific image, click the **Edit** button against the **Selected Image** to open **tickedImage** dialog box.



Select the desired platform and click inside corresponding **value** field. From the **Select Image** dialog field, you can either,

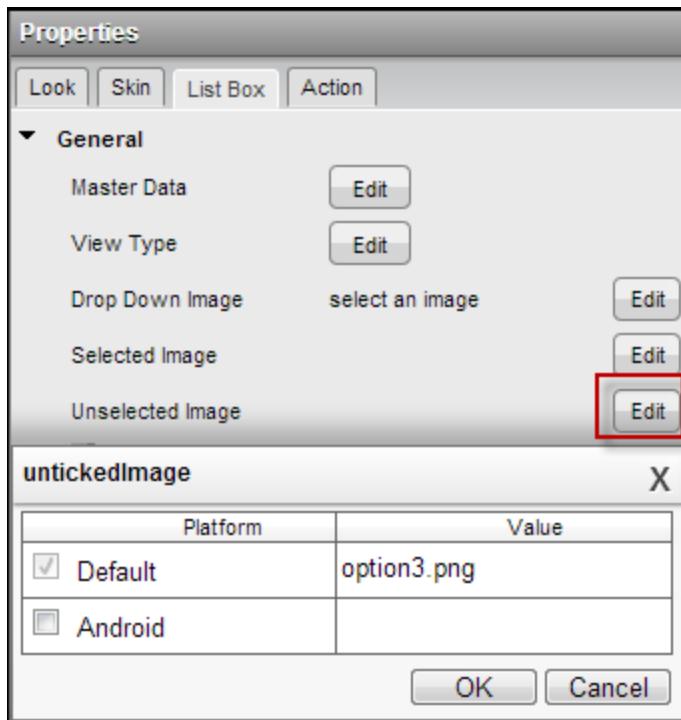
- Select an available image.
- Provide image url.

#### 5.16.4.5 unselected Image

Specifies the image to be displayed when a selection is cleared.

**Note:** If you specify a **Selected Image**, ensure that you also specify an **Unselected Image**. If not specified, the behavior will be undefined

For specifying a default image or a platform specific image, click the **Edit** button against the **unselected Image** to open **tickedImage** dialog box.



Select the desired platform and click inside corresponding **value** field. From the **Select Image** dialog field, you can either,

- Select an available image.
- Provide image url.

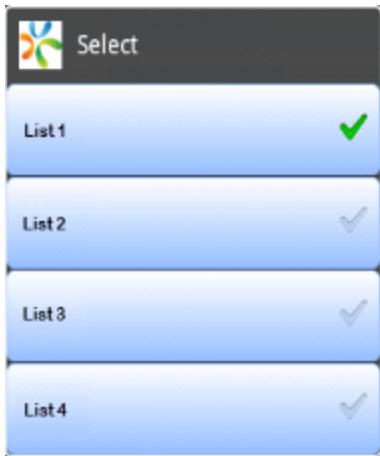
#### 5.16.4.6 Place Holder

Specifies the temporary or substitute text (a hint provided as a word or phrase) that must be displayed on the ComboBox until the actual selection is made.

**Note:** This property is specific to the iOS platform.

#### 5.16.4.7 Popup Icon

Specifies the icon that appears in the title area of the popup (on the top left side of the popup).



For the popup icon to appear, click the **Edit** button against the **Popup Icon** field to open the **popupIcon** dialog box.

Select an image to be displayed as the popup icon and click **OK**.

**Note:** This property is specific to the Android platform.

#### 5.16.4.8 Popup Title

Specifies the title text to be displayed for the ListBox.

**Note:** This property is specific to the Android platform.

#### 5.16.4.9 Tool Tip

Specifies the hint text when the cursor hovers over a widget, without clicking it. The text entered in the tooltip appears as a small box when the cursor hovers over a widget.

**Note:** This property is specific to the Windows Tablet platform.

### 5.16.5 Actions

ListBox widget has the following ListBox action associated with it:

- **onSelection:** This action is invoked by the platform when an item is selected.

For more information on using the above action, refer to section [Action Editor](#).

### 5.16.6 Placement Inside a Widget

A ListBox can be directly added inside the following widgets:

| Widget    | ListBox placement inside a widget                         |
|-----------|---|
| Form      | Yes   |
| HBox      | Yes   |
| VBox      | Yes   |
| ScrollBar | Horizontal Orientation - Yes<br>Vertical Orientation- Yes |
| Tab       | Yes   |
| Segment   | No  |
| Popup     | Yes   |
| Template  | Header- No<br>Footer- No                                  |

### 5.16.7 Widget Appearance on Platforms

The appearance of the widget with the default properties (with and without skin) on various platforms is as follows:

| Platform | Without Skin   | With Skin  |
|----------|--|--|
| Android  | Item1<br> | Item2<br> |
| iOS      | Item1<br> | Item1<br> |
| SPA      | Item1<br> | Item1<br> |

## 5.17 RadioButtonGroup

**RadioButtonGroup** is a widget that allows you to define a set of radio buttons and the user can choose one of it as an option.

### 5.17.1 Important Considerations

The following are important considerations for a RadioButtonGroup Widget.

#### 5.17.1.1 All Platforms

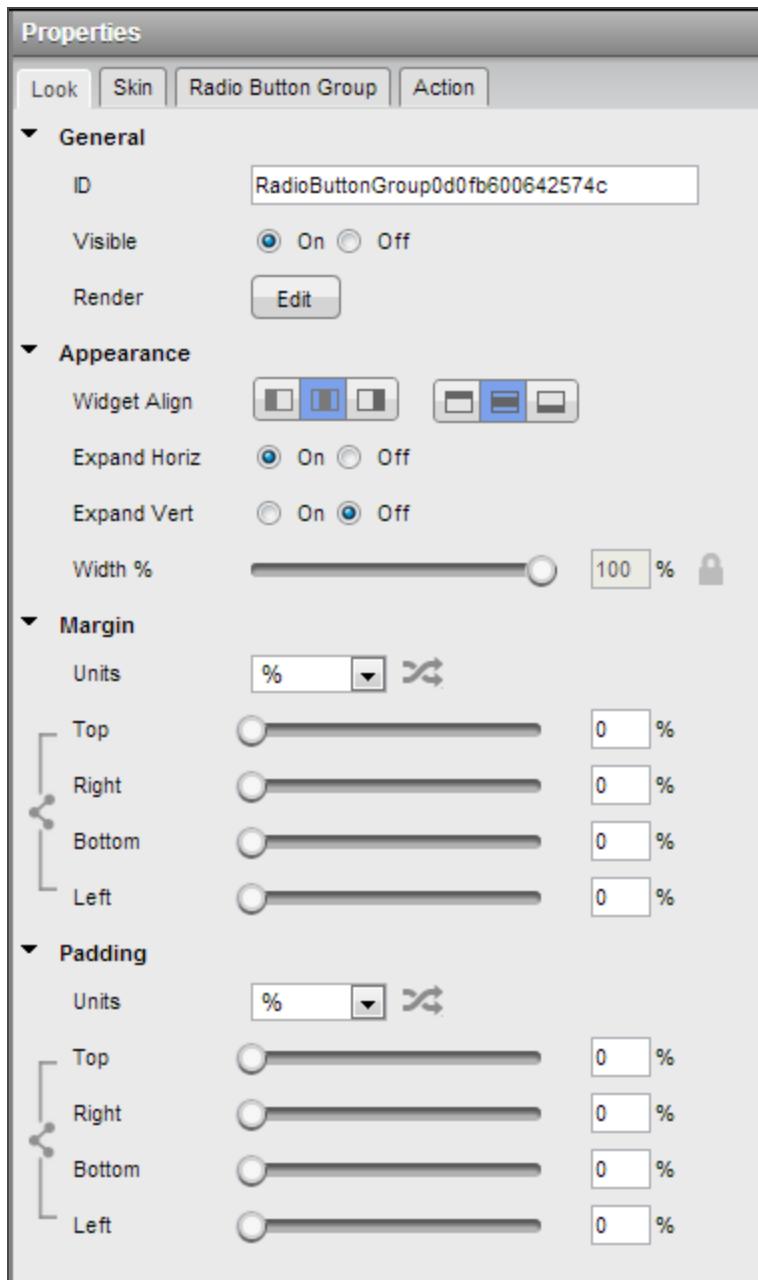
- RadioButtonGroup widget is always a group widget.
- Limit the number of choices in the widget. If you need to display several choices, consider using a [ComboBox](#) widget.

#### 5.17.1.2 Android

- If you set the [Orientation](#) to horizontal, we suggest that you do not place more than two items in the group, as there is a platform limitation.
- If you place more than two items and the associated text with the items is large, there is a possibility that the additional items will not fit in the screen width and will not be visible on the screen.

### 5.17.2 Look

This section details the **Properties > Look** tab of a RadioButtonGroup widget.



To specify the look properties of a RadioButtonGroup widget, refer section: [Look](#).

### 5.17.3 Skin

Skin defines properties such as background color, borders and shadows for a widget. They also, define the widget font (applicable only if text is displayed on a widget.)

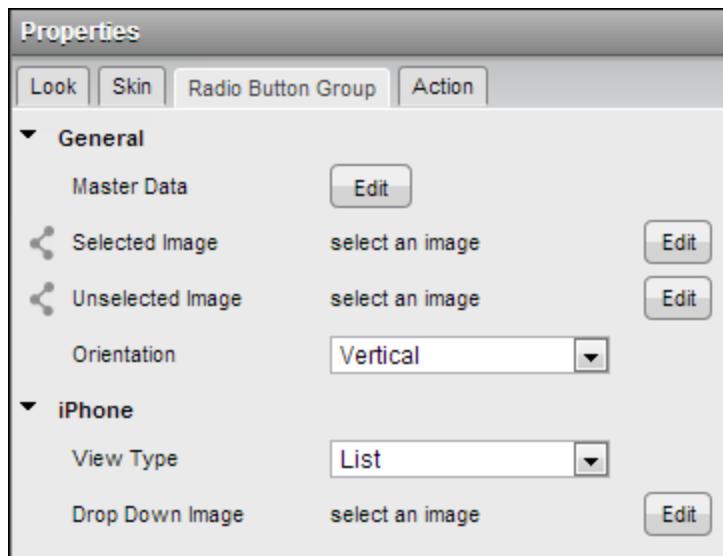
For the RadioButtonGroup widget, you can apply a skin for the following states:

- Normal: Specifies the look and feel of the widget when not in focus.
- Focus: Specifies the skin that is applied when there is focus on a widget.
- Hover Skin: Specifies the look and feel of a widget when the cursor hovers on the widget.

**Note:** Hoover Skins is available only on the Windows (native) Tablet devices.

Refer to section [Working with Skins](#) to know more about adding a skin to a widget.

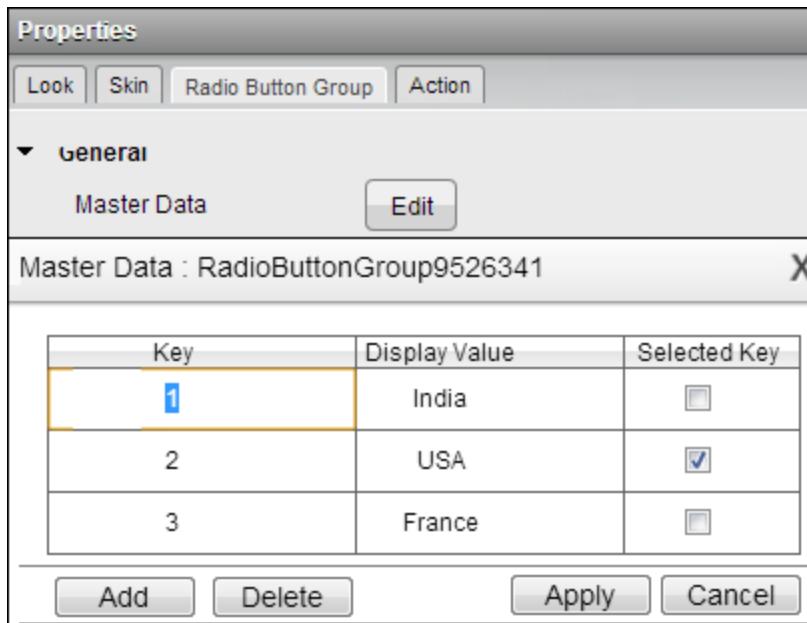
#### 5.17.4 Platform Specific Properties



##### 5.17.4.1 Master Data

Specifies the set of values that must be displayed for the user to make a selection from the available choices.

For specifying the values, click the **Edit** button against the **Master Data** field to open the **RadioButtonGroup** dialog box.



Master Data contains the following columns:

- Key: It is the unique identifier of each RadioButton value.
- Display Value: This value is displayed in the RadioButton group widget on a form.
- Select Key : Allows you to select an individual RadioButton value from the group.

If required, to add more RadioButton values to a widget, click **Add**. To delete a RadioButton row, click inside a cell, and then click **Delete**.

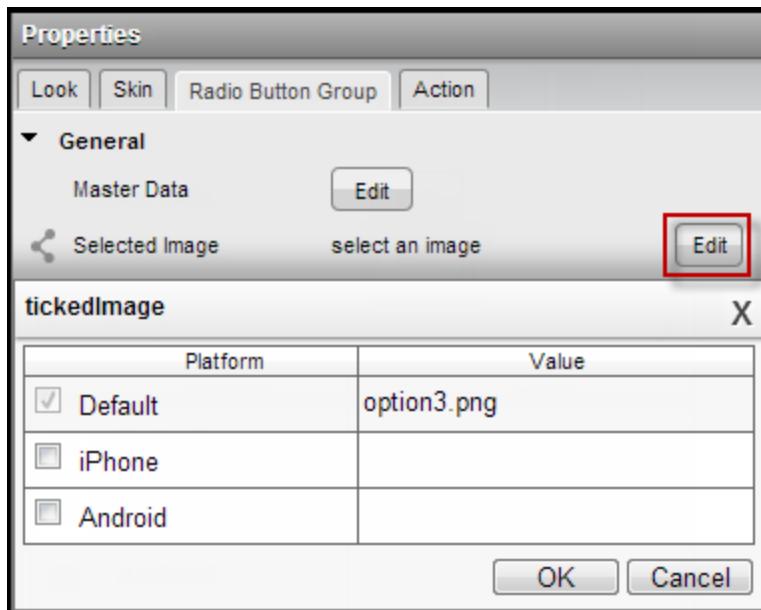
After providing the required details and click **Apply** to create master data.

#### 5.17.4.2 Selected Image

Specifies the image to be displayed when you make a selection.

**Note:** If you specify a **Selected Image**, ensure that you also specify an **Unselected Image**. If not specified, the behavior will be undefined

For specifying a default image or a platform specific image, click the **Edit** button against the **Selected Image** to open **tickedImage** dialog box.

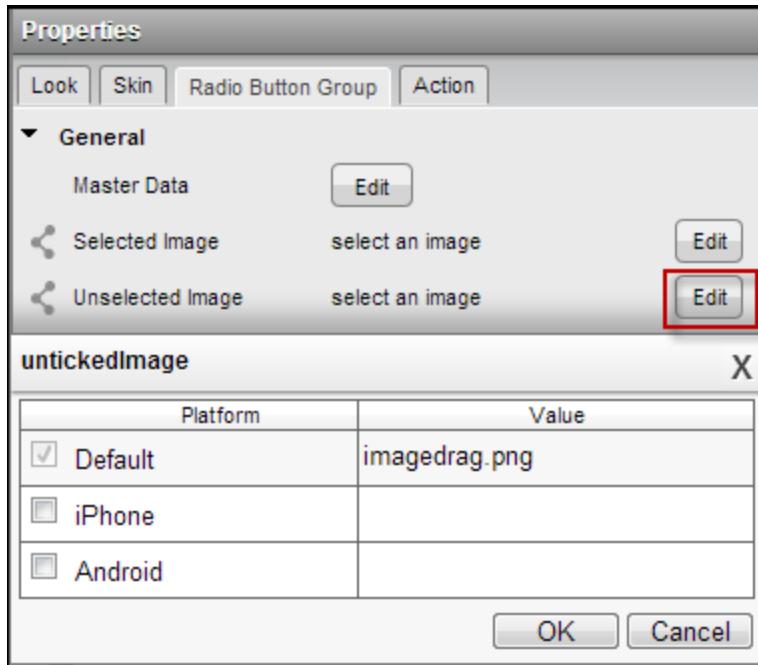


To provide a platform specific drop-down image, select the desired platform and click inside corresponding **value** field. From the **Select Image** dialog field, you can either,

- Select an available image.
- Provide image url.

#### 5.17.4.3 Unselected Image

Specifies the image to be displayed when a selection is cleared.



Select the desired platform and click inside corresponding **value** field. From the **Select Image** dialog field, you can either,

- Select an available image.
- Provide image url.

#### 5.17.4.4 Orientation

Specifies the type of alignment of the items within the widget.

**Default:** Vertical

Following orientations are available:

- Vertical
- Horizontal

#### 5.17.4.5 View Type

Specifies the view type of the RadioButtonGroup.

**Default:**List

Following are the available options:

- List
- Table
- Toggle
- On-Screen Wheel

**Note:** When you select the On-Screen Wheel view type, you cannot view the on screen wheel on App canvas. However, the On-Screen Wheel option works as expected on a device.

For Toggle, you can further select one of the following View Styles:

- Plain
- Bordered
- Bar

Apart from the above options, you can also enable and provide **Tint Color** for this widget.

- Enable Tint Color: To provide a tint color when the view type is Toggle, select **True** option.
- Tint Color: This option is Enabled when **Tint Color** is set to true. For selecting a tint color, click the color picker to open the color selection dialog box and then Select a desire color.

Click the **Equal Segments** check box to distribute the segments in equal proportions.

**Note:** This property is specific to the iOS platform.

#### Drop Down Image

Specifies the image to be used for the drop-down box indicator. The image you specify is used to depict the drop-down box. If you do not specify an image, the drop-down box displays the default image (inverted triangle).

This option is available only when View Type selected in **List**.

To provide a platform specific image, click the **Edit** button against the **Drop Down Image** to open **Drop Down Image** dialog box.

From the **Drop Down Image** dialog box, select an image and click **OK**.

#### 5.17.4.6 Group Cells

Specifies if the group cells style must be applied. This property is available only when the View Type selected is **Table**. When applied, the items in the check box are grouped and they are indicated with a round corner rectangle box.

This option is available only when View Type selected in **Table**.

Default: Cells are not groped.

- If selected, the group cells style is applied.
- If unselected, the group cells style is not applied.

#### 5.17.4.7 View Style

This property is available only when View Type is **Toggle**.

Following are the available options:

- Plain
- Bordered
- Bar

#### 5.17.4.8 Equal Segments

This option is only available when [View Type](#) is Toggle.

- Click **On** to distribute the segments in equal proportions.
- Click **Off** if you do not want to distribute the segments in equal proportions.

#### 5.17.4.9 Tint Color

This option is only available when [View Type](#) is Toggle.

- Click **On** to provide a tint color.
- Click **Off** if you do not want to provide a tint color.

**Note:** This property is specific to the iOS7 platform.

#### 5.17.4.10 Tint Color

This option is enabled when **Tint Color** is set to true. For selecting a tint color, click the color picker to open the color selection dialog box and then Select a desire color.

**Note:** This property is specific to the iOS7 platform.

#### 5.17.4.11 Tool Tip

Specifies the hint text when the cursor hovers over a widget, without clicking it. The text entered in the tooltip appears as a small box when the cursor hovers over a widget.

**Note:** This property is specific to the Windows Tablet platform.

### 5.17.5 Actions

RadioButtonGroup widget has the following action associated with it:

- **onSelection:** This action is invoked by the platform when an item is selected.

For more information on using the above action, refer to section [Action Editor](#).

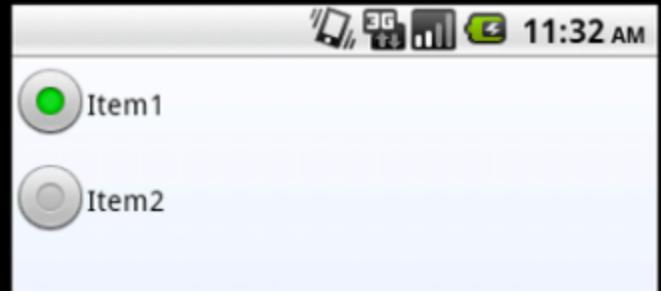
#### 5.17.6 Placement inside a Widget

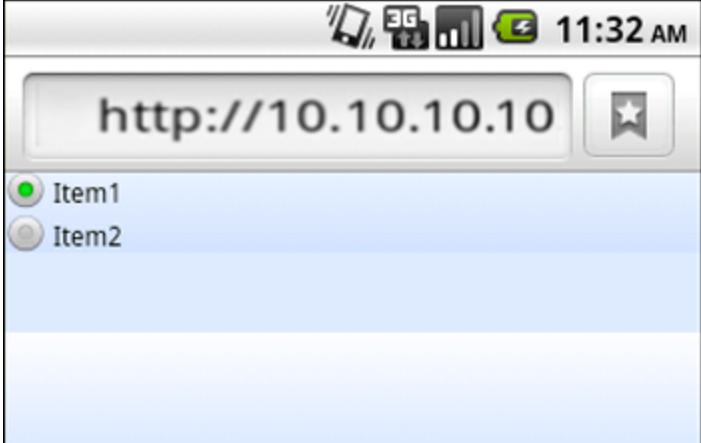
A RadioButtonGroup can be directly added inside the following widgets:

|           |   |
|-----------|---|
| Widget    | RadioButtonGroup placement inside a widget                |
| Form      | Yes   |
| HBox      | Yes   |
| VBox      | Yes   |
| ScrollBox | Horizontal Orientation - Yes<br>Vertical Orientation- Yes |
| Tab       | Yes   |
| Segment   | No  |
| Popup     | Yes   |
| Template  | Header- No<br>Footer- No                                  |

### 5.17.7 Widget Appearance on Platforms

The appearance of the RadioButtonGroup widget on various platforms is as follows:

| Platform | Appearance  |
|----------|---|
| Android  |  |

| Platform | Appearance   |
|----------|--|
| iOS      |   |
| SPA      |  |

## 5.18 RichText

RichText widget is used to display non-editable and well formatted text on the Form. HTML formatting tags are used in RichText widget to display text with styles (bold, underlined etc.), links, and images.

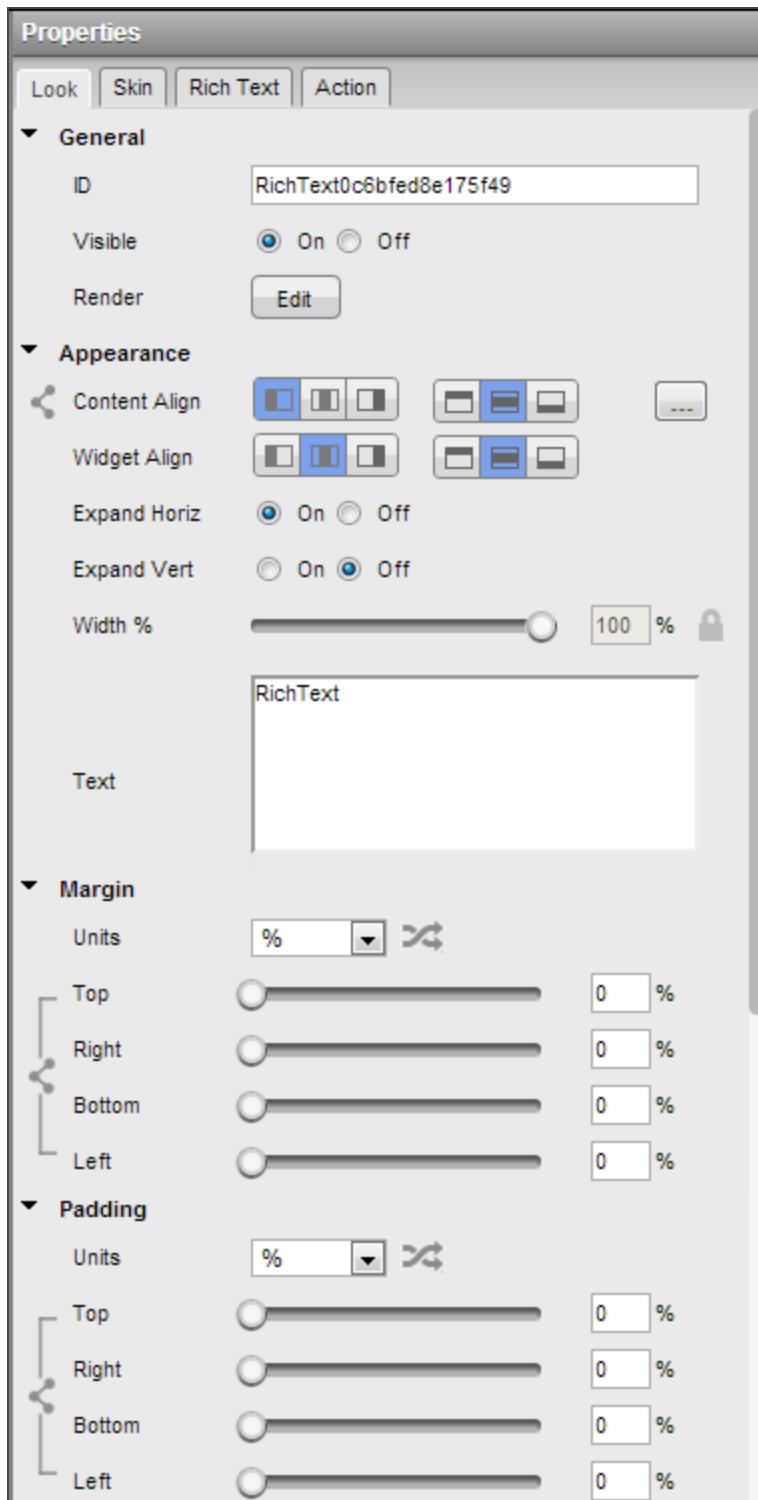
### 5.18.1 Important Considerations

The following is the important considerations for the RichText widget.

- If you specify a skin for the RichText widget, all the font level settings (color style, or size etc.) is applied to the complete content in RichText widget. You can use the label style HTML formatting tag to override the text color specified at the skin level.

### 5.18.2 Look

This section details the **Properties > Look** tab of a RichText widget.



To specify the look properties of a RichText widget, refer section: [Look](#).

### 5.18.3 Skin

Skin defines properties such as background color, borders and shadows for a widget. They also, define the widget font (applicable only if text is displayed on a widget.)

For the RichText widget, you can apply a skin for the following states:

- Normal: Specifies the look and feel of the widget when not in focus.
- Link: Specifies the skin is applied to the link in the RichText widget.
- Link Focus: Specifies the skin is applied to the link when focused.
- Telephone Link: Specifies the skin to be applied to the telephone links in the RichText widget.

**Note:** Telephone Link is available only on the Windows platform.

- Hover: Specifies the look and feel of a widget when the cursor hovers on the widget.

**Note:** Hoover Skins is available only on the Windows (native) Tablet platform.

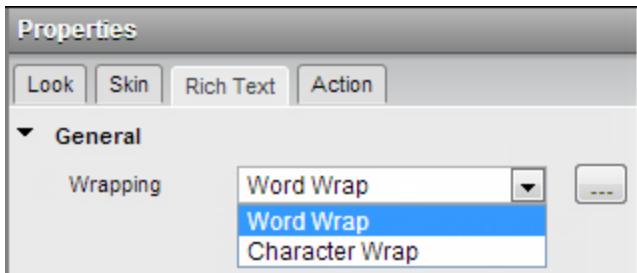
- Super Script: Specifies the skin to be applied to superscripts in the RichText widget.

**Note:** Super Script Skins is available only on the Windows platform.

Refer to section [Working with Skins](#) to know more about adding a skin to a widget.

## 5.18.4 Platform Specific Properties

### 5.18.4.1 Rich Text Wrapping



When the content of the RichText reaches the boundaries, it starts wrapping. There are two wrapping options available:

- Word Wrap: Specifies if the complete word must be moved to the next line when you reach the right margin. This is the default wrapping property.
- Character Wrap: Specifies if the characters in a word must be moved to the next line when you reach the right margin.

**Note:** This property is specific to the iOS platform.

### 5.18.4.2 Tool Tip

Specifies the hint text when the cursor hovers over a widget, without clicking it. The text entered in the tooltip appears as a small box when the cursor hovers over a widget.

**Note:** This property is specific to the Windows Tablet platform.

## 5.18.5 Actions

Rich Text widget has the following action associated with it:

- **onClick:** This action is invoked by the platform when the user performs a click action on the RichText.

For more information on using the above action, refer to section [Action Editor](#).

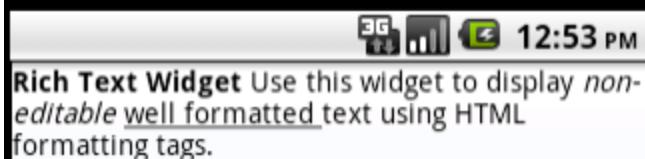
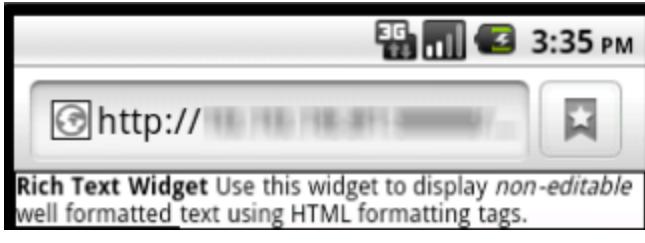
### 5.18.6 Placement Inside a Widget

A RichText can be directly added inside the following widgets:

| Widget    | RichText placement inside a widget                        |
|-----------|---|
| Form      | Yes   |
| HBox      | Yes   |
| VBox      | Yes   |
| ScrollBox | Horizontal Orientation - Yes<br>Vertical Orientation- Yes |
| Tab       | Yes   |
| Segment   | Yes   |
| Popup     | Yes   |
| Template  | Header- No<br>Footer- No                                  |

### 5.18.7 Widget Appearance on Platforms

The appearance of the Rich Text widget on various platforms is as follows:

| Platform | Appearance  |
|----------|---|
| Android  |   |
| iOS      |   |
| SPA      |  |

## 5.19 Slider

Slider widget allows you to select a value from a defined range of values by moving the thumb (an indicator) in a horizontal direction. The Slider widget consists of a seekbar (or track) and a thumb (a control that you can slide). You can optionally choose to display a minimum value and a maximum value. When you drag the thumb along the slider, the value or process is updated continuously and is displayed on the track (you must define an event to achieve this behavior).

**Note:** The Slider widget is not supported in SPA platforms.

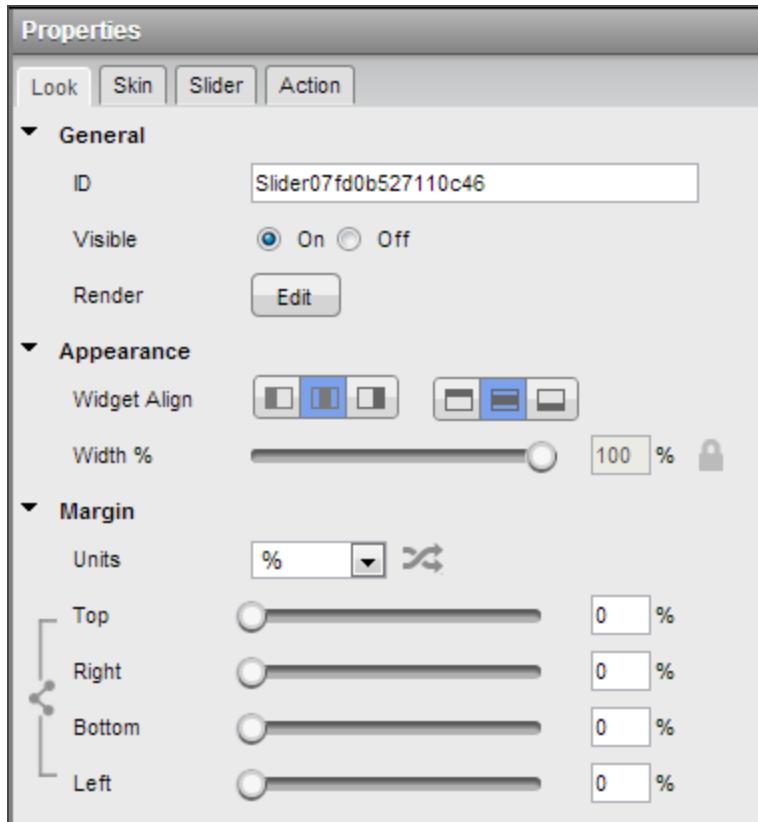
### 5.19.1 Important Considerations

The following are important considerations for an Slider Widget.

- All platforms (except iPhone): The slider widget does not display the minimum and maximum values unless the [Min Label](#) and [Max Label](#) are specified.
- iPhone: You cannot display the minimum and maximum values. You can use the [Min Value Image](#) and the [Max Value Image](#) to indicate the values.

### 5.19.2 Look

This section details the **Properties > Look** tab of a Slider widget.



To specify the look properties of an Slider widget, refer section: [Look](#).

### 5.19.3 Skin

Skin defines properties such as background color, borders and shadows for a widget. They also, define the widget font (applicable only if text is displayed on a widget.)

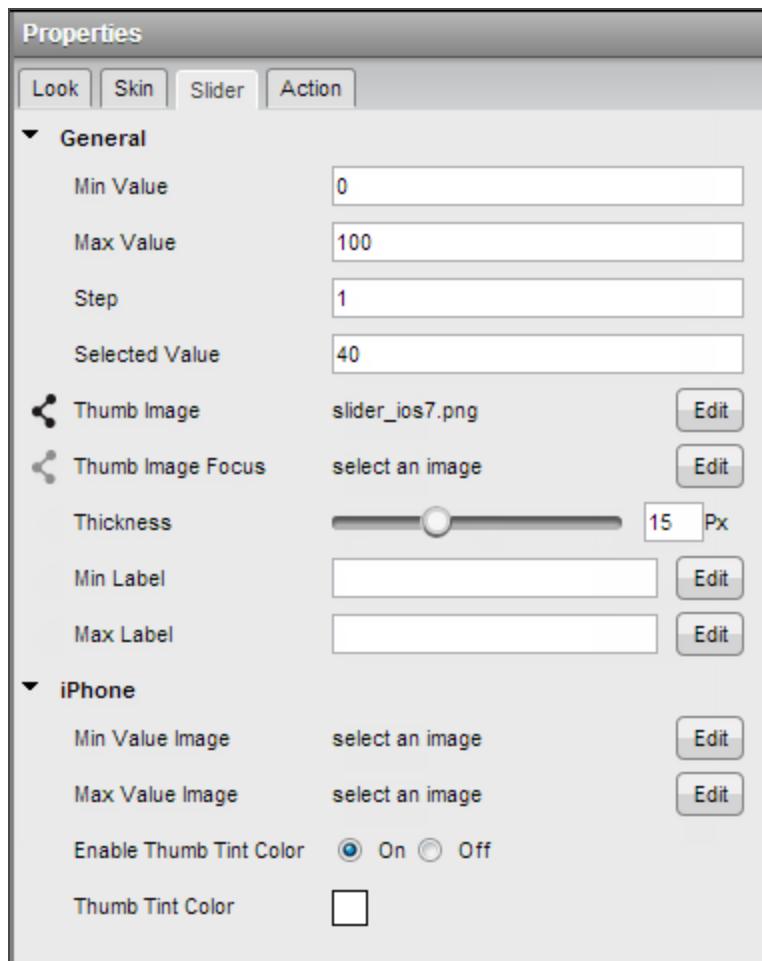
For the Slider widget, you can apply a skin for the following states:

- Left: Specifies the skin that is applied to the background of the slider on left side of the thumb image..
- Right: Specifies the skin that is applied to the background of the slider on right side of the thumb image..
- Min Label: Specifies the skin that is applied to the min property of the slider.
- Max Label: Specifies the skin that is applied to the max property of the slider.

Refer to section [Working with Skins](#) to know more about adding a skin to a widget.

#### 5.19.4 Platform Specific Properties

**Note:** In this section, the properties that can be forked are identified by an icon ( ) located to the left of the property. For more details, refer to section [Forking a Widget property](#).



##### 5.19.4.1 Min Value

Specifies the minimum value on the Slider that you can select.

**Default:** 0

You can set any number (the number can be upto 4 bytes (positive or negative)) as the minimum value.

#### 5.19.4.2 Max Value

Specifies the maximum value on the slider that you can select.

**Default:** 100

You can set any number (the number can be upto 4 bytes (positive or negative)) as the maximum value.

#### 5.19.4.3 Step

Specifies the value by which the slider value is increased or decreased between the allowed slider values.

**Default:** 1

you can specify a value which is in between the difference of the maximum and minimum values of the slider.

For example, if you set the minimum value to 40 and the maximum value to 45, the step value can be a number between 1 and 5.

#### 5.19.4.4 Selected Value

Specifies the value that must be displayed as selected when rendered. If you specify the default selected value, the thumb shows the specified value as the selected value when rendered.

**Default:** 40

You must be aware of the following scenarios for the slider:

**Scenario1:** Below is the calculation, if default value is not specified:

Default value = minimum value + (maximum value - minimum value)/2

The default value is the minimum value plus half the difference between the minimum and the maximum value.

For example, if the minimum value is 0 and the maximum value is 100, the default value is 50.

**Scenario2:** Below is the calculation, if default value is specified:

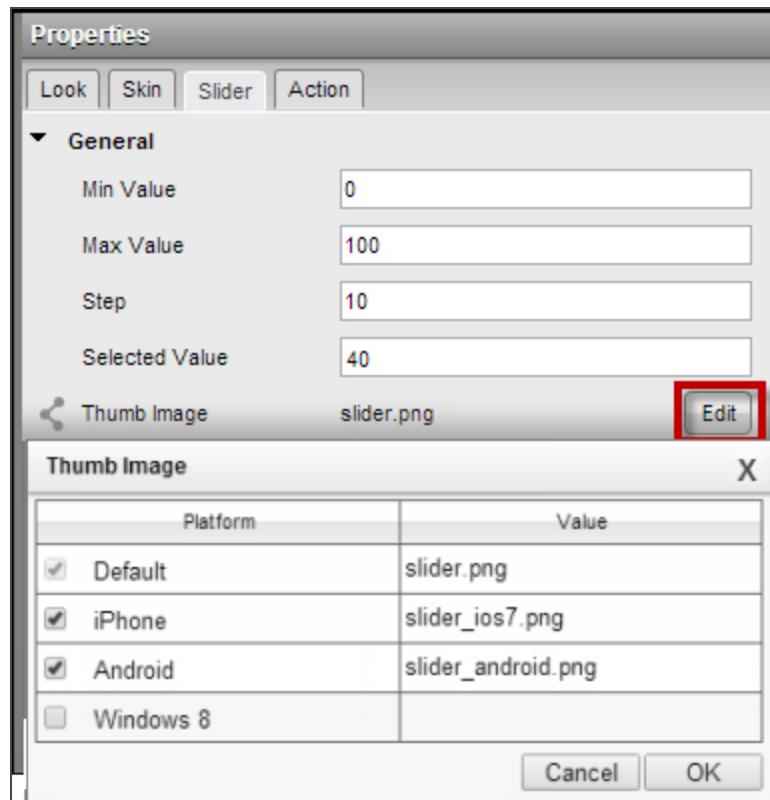
If you specify a default value which is lesser than the minimum value, the minimum value is taken as the default value.

If you specify a default value which is greater than the maximum value, the maximum value is taken as the default value

#### 5.19.4.5 Thumb Image

Specifies the image to indicate the thumb.

To provide a default or platform specific thumb image, click the **Edit** button against the **Thumb Image** to open **thumbImage** dialog box.



System provides a default image for each of the platforms.

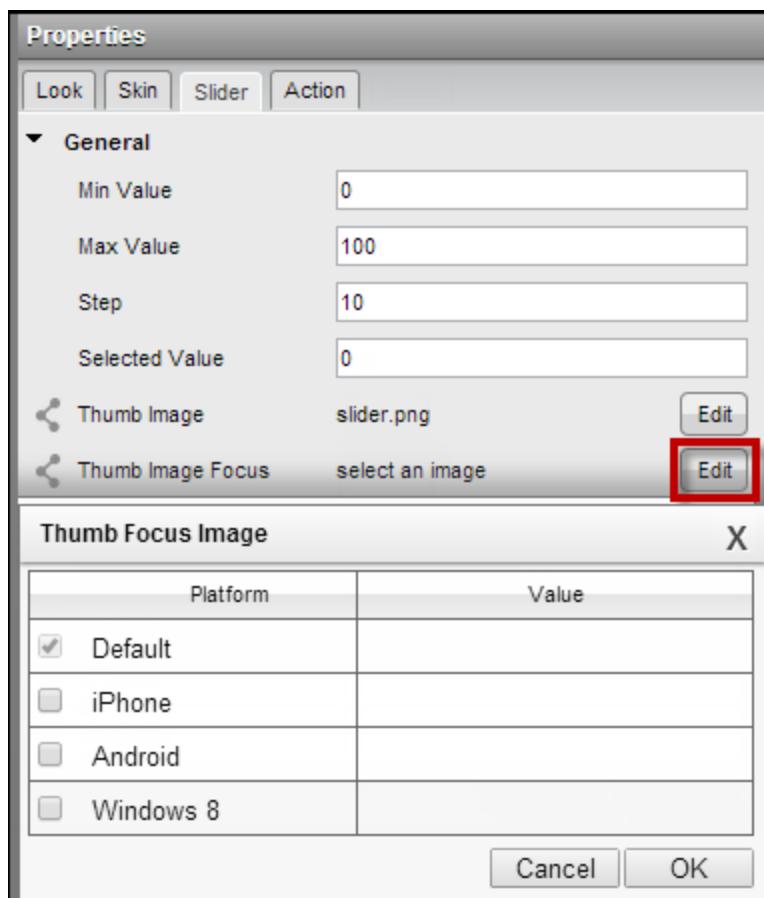
To provide a platform specific image or to replace the default image, select the desired platform and click inside the corresponding **value** field. This results in opening the **Select Image** dialog box. From here, you can either:

- Select a available image.
- Provide image url.

#### 5.19.4.6 Thumb Image Focus

Specifies the image to indicate that there is focus on the thumb.

To provide a default or a platform specific thumb image, click the **Edit** button against the **Focus Thumb Image** to open **focusThumbImage** dialog box.



To provide a platform specific image or to replace the default image, select the desired platform and click inside the corresponding **value** field. This results in opening the **Select Image** dialog box. You can either:

- Select an available image.
- Provide image url.

#### 5.19.4.7 Thickness

Specifies the thickness of the seekbar.

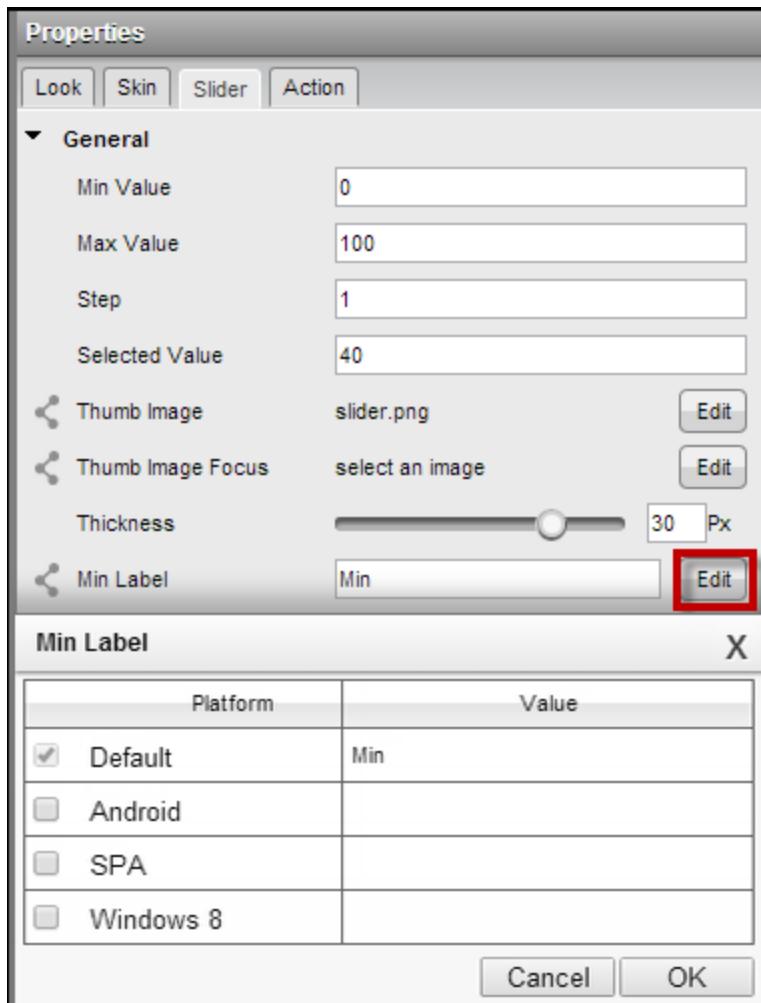
**Default:** 15

#### 5.19.4.8 Min Label

Specifies the text or number to be displayed for the minimum value of the slider. This text is displayed just below the slider on the left.

In the **Min Label** text field, type the name to be used as minimum Label. This text become default for all the platforms.

For providing platform specific label, click the **Edit** button against the **Min Label** to open the **minLabel** dialog box.



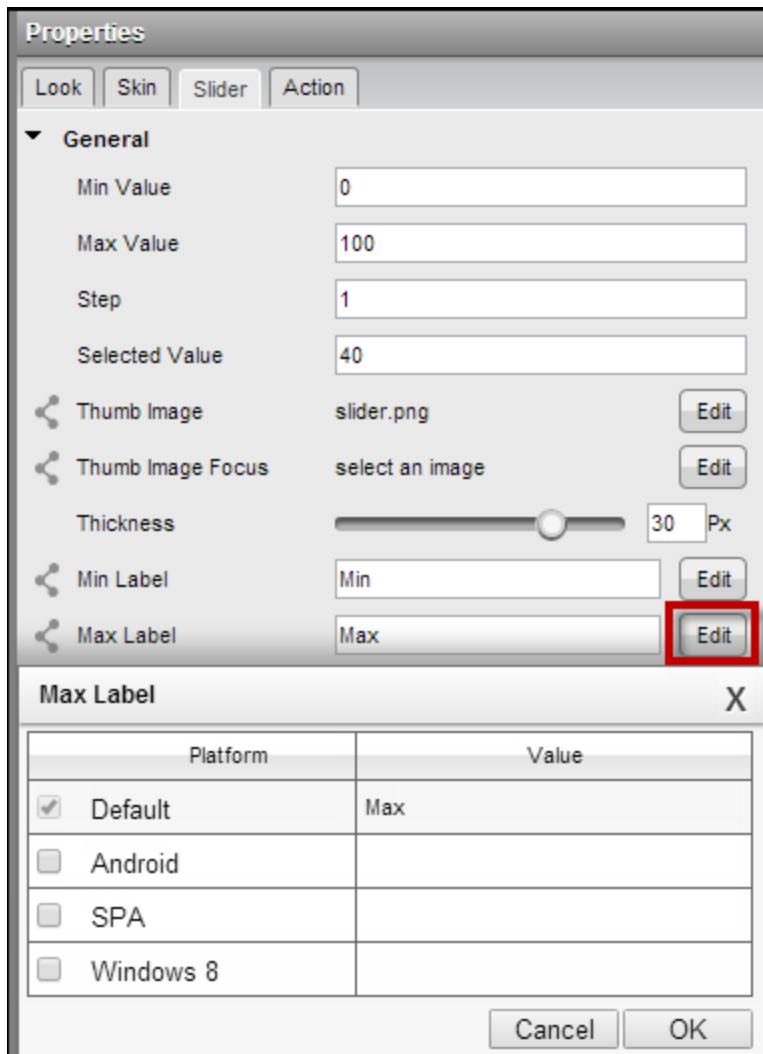
Select a platform and then type a desired name inside the corresponding **value** field. This becomes the **Min Label** for that particular platform.

#### 5.19.4.9 Max Label

Specifies the text or number to be displayed for the maximum value of the slider. This text is displayed just below the slider on the right.

In the **Max Label** text field, type the name to be used as maximum label. This text become default for all the platforms.

For providing platform specific label, click the **Edit** button against the **Max Label** to open the **maxLabel** dialog box.



Select a platform and then type a desired name inside the corresponding **value** field. This becomes the **Max Label** for that particular platform.

#### 5.19.4.10 Min Value Image

Specifies the image for the minimum value of the slider.

Click the **Edit** button against the **Min Value Image** to open the **minValueImage** dialog box. You can either:

- Select a available image.
- Provide image url.

**Note:** This property is specific to the iOS platform.

#### 5.19.4.11 Max Value Image

Specifies the image for the maximum value of the slider.

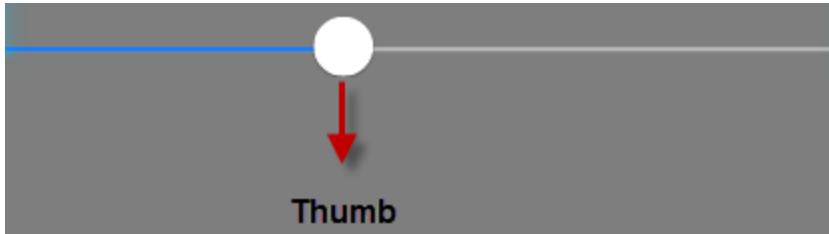
Click the **Edit** button against the **Max Value Image** to open the **maxValueImage** dialog box. You can either:

- Select a available image.
- Provide image url.

**Note:** This property is specific to the iOS platform.

#### 5.19.4.12 Enable Thumb Tint Color

Specifies whether a color should be specified for the Slider widget thumb.



**Note:** This property is specific to the iOS7 platform.

#### 5.19.4.13 Thumb Tint Color

This option is Enabled when **Enable Thumb Tint Color** is set to true. For selecting a tint color, click the color picker to open the color selection dialog box and then Select a desire color.

**Note:** This property is specific to the iOS7 platform.

## 5.19.5 Actions

Slider widget has the following actions associated with it:

- **onSlide:** This action is invoked by the platform when there is a change in the default selected value.
  - For touch based devices, this action is triggered when you stop sliding the thumb icon.
  - For non touch based devices, this action is triggered when the left or right key is released.
- **onSelection:** This action is invoked by the platform when the user makes a selection.
  - For touch based devices, this action is triggered when you stop sliding the thumb icon.
  - For non touch based devices, this action is triggered when the left or right key is released.

For more information on using the above actions, refer to section [Action Editor](#).

## 5.19.6 Placement inside a Widget

A slider can be directly added inside the following widgets:

| Widget    | Slider placement inside a widget                          |
|-----------|---|
| Form      | Yes   |
| HBox      | Yes   |
| VBox      | Yes   |
| ScrollBar | Horizontal Orientation - Yes<br>Vertical Orientation- Yes |
| Tab       | Yes   |
| Segment   | No  |

|          |                                  |
|----------|----------------------------------|
| Widget   | Slider placement inside a widget |
| Popup    | Yes                              |
| Template | Header- No<br>Footer- No         |

### 5.19.7 Widget Appearance on Platforms

The appearance of the Slider widget on various platforms is as follows:

| Platform | Appearance  |
|----------|---|
| Android  |  A horizontal slider with a yellow track bar and a grey thumb. |
| iPhone   |  A horizontal slider with a blue track bar and a white thumb. |

## 5.20 TextArea2

TextArea is used to provide an editable field for the user to enter text which spans over multiple lines . You can use the TextArea widget to provide a field where a user can enter multiple lines of text.

For example, in the **Feedback** section of an Application, you can place a TextArea widget and instruct the users to enter their comments.

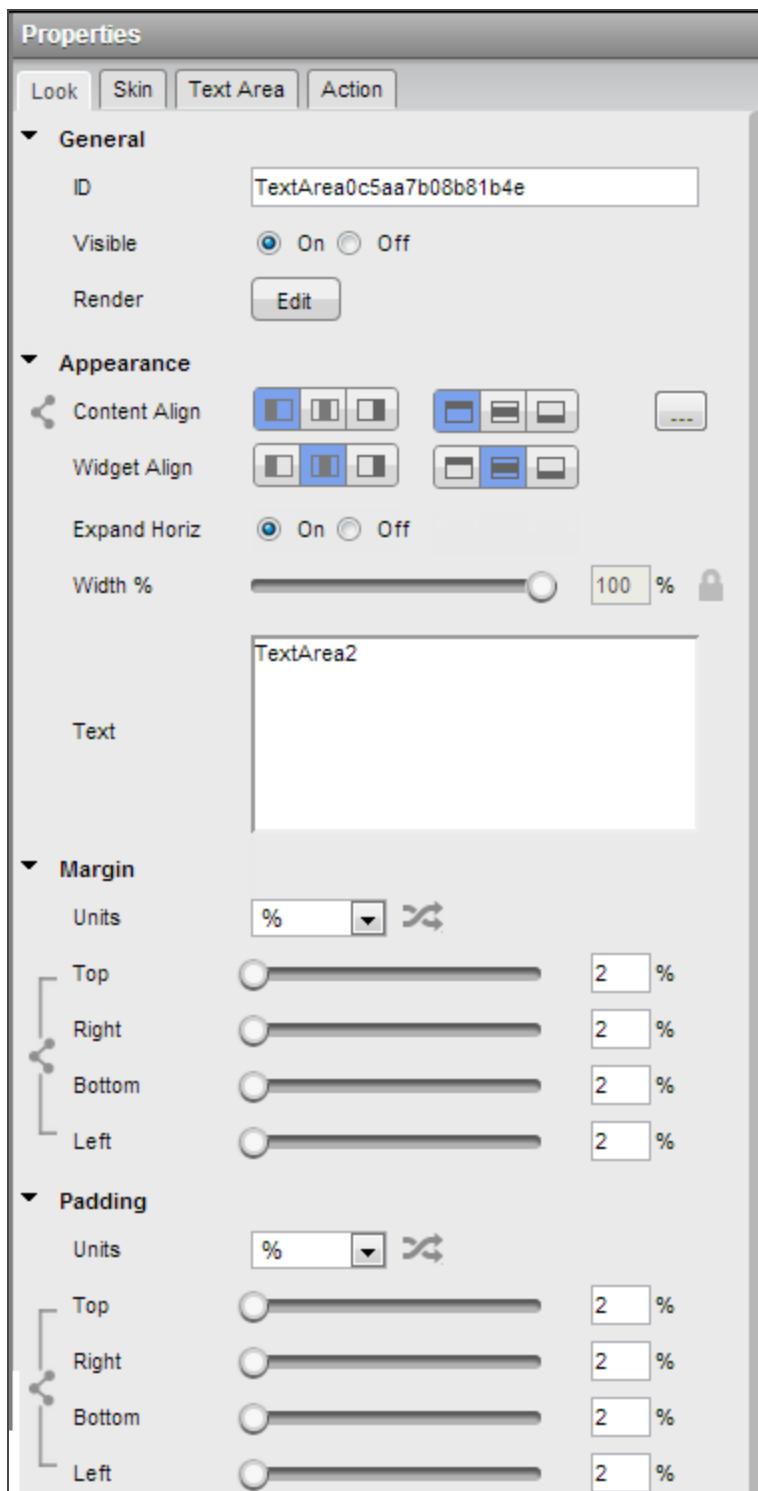
### 5.20.1 Important Considerations

The following are important considerations for a TextArea Widget.

- Editing on devices with small form factor takes place in a new screen.
- Editing on devices with medium or large form factor takes place in the same screen.
- In Mobile web, some browsers by default enable a scroll bar (vertical/horizontal) for the text area, even if the number of lines are less than [Number of visible Lines](#). The Mobile Web platform has no control on deciding whether the scroll bar should appear or not.

### 5.20.2 Look

This section details the **Properties > Look** tab of a **TextArea2** widget.



To specify the look properties of a TextArea2 widget, refer section: [Look](#).

### 5.20.3 Skin

Skin defines properties such as background color, borders and shadows for a widget. They also, define the widget font (applicable only if text is displayed on a widget.)

For the TextArea widget, you can apply a skin for the following states:

- Normal: Specifies the look and feel of the widget when not in focus.
- Focus: Specifies the skin that is applied when there is focus on a widget.
- Blocked UI: Specifies the skin that must be used to block the interface until the action in progress (for example, a service call) is completed.

**Note:** Blocked UI is available only for SPA platforms.

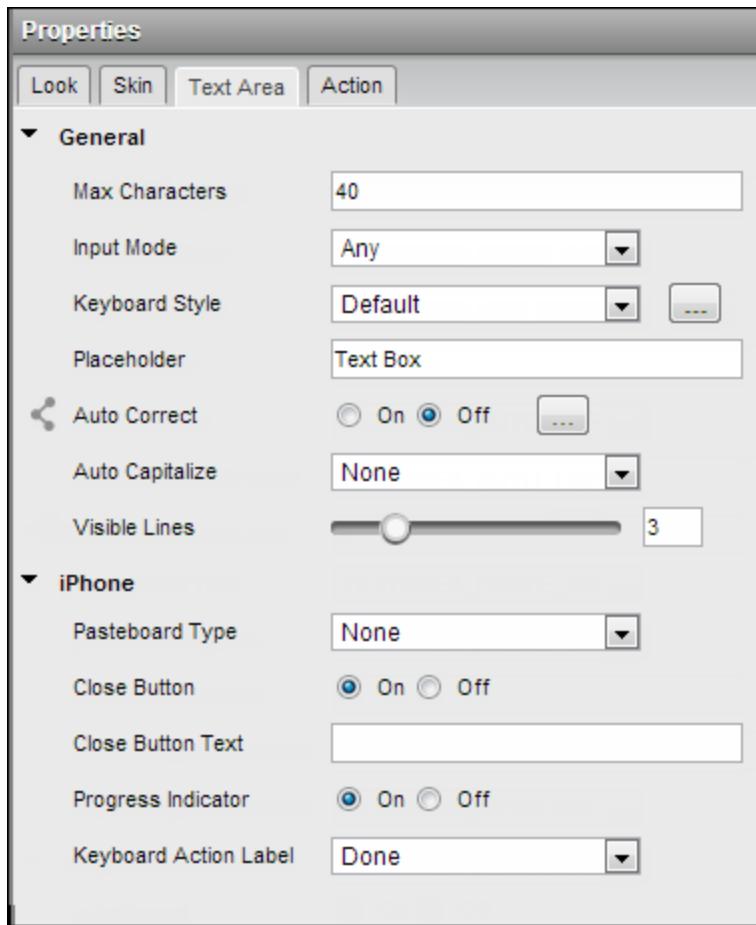
- Placeholder: Specifies the skin to be applied to the placeholder text in the TextArea widget. Only the font color skin attribute is applicable.
- Hover: Specifies the look and feel of a widget when the cursor hovers on the widget.

**Note:** Hoover Skins is available only on the Windows (native) Tablet platform.

Refer to section [Working with Skins](#) to know more about adding a skin to a widget.

### 5.20.4 Platform Specific Properties

**Note:** In this section, the properties that can be forked are identified by an icon ( located to the left of the property. For more details, refer to section [Forking a Widget property](#).



#### 5.20.4.1 Max Characters

Specifies the maximum number of characters that the text field can accept.

If you specify a number for this property, the number of input characters cannot exceed the specified number.

#### 5.20.4.2 Input Mode

Specifies the type of input characters that a user can enter into the **TextArea2** widget. You can use this property to restrict the input characters to only numbers or a combination of alphabets, numbers, and special characters.

Following are the available options:

- Any: Specifies that the input characters can be letters, numbers, special characters, or a combination of all three of them.
- Numeric Only: Specifies that the input characters must be numbers only. This option is not supported on Server side Mobile Web platforms.

#### 5.20.4.3 Key Board Style

When you interact with a TextBox widget, a keyboard is displayed. You can use this property to select the type of keyboard that you want to display.

The following are the available keyboard types :

- Decimal
- Number Pad
- Phone Pad
- Signed Decimal Number
- Signed Number

**Note:** This property is specific to the Android platform.

#### 5.20.4.4 Placeholder

The placeholder attribute specifies a short hint that describes the expected value of an input field (example, a sample value or a short description of the expected format). The hint is displayed in the input field when it is empty, and disappears when the field gets focus.

For example, for the Username field, you can enter the placeholder text as *Enter User ID* or *Email Address*. The user then clicks on the TextArea2 widget and enters the Username.

#### 5.20.4.5 Auto Correct

This property determines whether auto-correction is enabled or disabled during typing. With auto-correction enabled, the text object tracks unknown words and suggests a more suitable replacement to the user.

- If **On** is selected, the auto correction option is enabled.
- If **Off** is selected, the auto correction option is not enabled.

You can choose to provide a platform specific Auto Correct property by clicking ellipsis (...) button next to the Auto Correct **Off** radio button.

In the **AutoCorrect** dialog box, select the platform and from the corresponding value field, select the check box to enable or disable auto correct property.

**Note:** You cannot execute this property on App Canvas.

**Note:** This property is specific to the iOS and the SPA platform.

#### 5.20.4.6 Auto Capitalize

Specifies the character capitalization behavior.

Following are the options available:

- None: If you leave this option unchanged, no action takes place on the input string.

Example : This is sample text.

- Words: This option changes the first character of all the words to uppercase. (Not supported on Mobile Web)

Example : This Is Sample Text.

- Sentences: This option changes the first character of all the sentences to uppercase.

Example : This is sample text.

- All: This option changes all the characters to uppercase. (Not supported on Mobile Web)

Example : THIS IS SAMPLE TEXT.

#### 5.20.4.7 Visible Lines

Number of lines to be displayed at a given time in the device of the TextArea widget. This essentially decides the height of the text area.

Minimum number of lines displayed are three.

#### 5.20.4.8 Pasteboard Type

This property enables an application to share data within the application or with another application using system-wide or application-specific paste boards.

Typically, an object in the application writes data to a pasteboard when the user requests a copy or cut operation on a selection in the user interface. Another object in the same or different application then reads that data from the pasteboard and presents it to the user at a new location; this usually happens when the user requests a paste operation.

**Note:** You can only paste the text to a textbox with the same pasteboard type as that of the source textbox. For example, if you set the Pasteboard type as *App Level Persistent*, you can paste the text only to another textbox whose pasteboard type is also set to *App Level Persistent*.

The different pasteboard types are as follows:

- None: If you select this option, the value selected in the application properties gets applied.
- System Level
- App Level Persistent

- App Level Non-Persistent
- Default

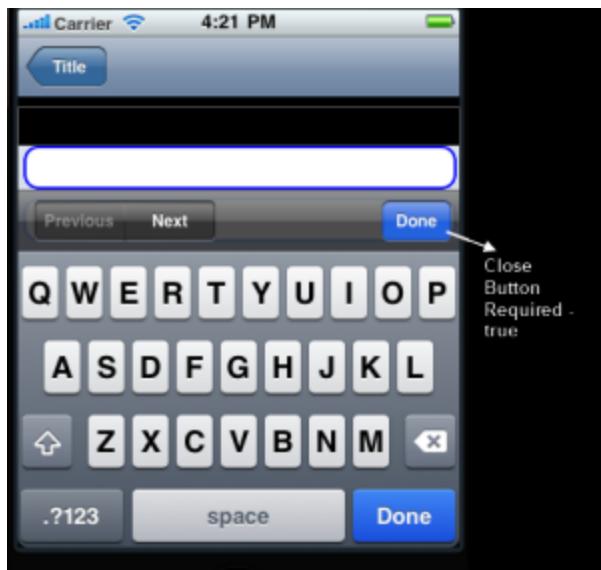
**Note:** This property is specific to the iOS platform.

#### 5.20.4.9 Close Button

Specifies if the **Done** button that appears in the keypad (opens when you select text box) must be visible or not.

- If set to **Off**, the **Done** button is not visible on the textbox.
- If set to **On**, the **Done** button is visible on the textbox.

The following image illustrates the Keypad when the property is set to true:



**Note:** This property is specific to the iOS platform.

#### 5.20.4.10 Close Button Text

Specifies the text to replace the "Done" button that appears on the Keypad (opens when you select a textbox). This property is available only when [Show Close Button](#) is enabled.

**Note:** This property is specific to the iOS platform.

#### 5.20.4.11 Progress Indicator

Specifies if there must be an indication to the user that the widget content is being loaded.

You can use this property typically for forms that require network calls during post show.

- If set to **Off**, the progress indicator is not displayed.
- If set to **On**, the progress indicator is displayed.

The following image illustrates the progress indicator on iPhone:



**Note:** This property is specific to the iOS platform.

#### 5.20.4.12 Keyboard Action Label

This property specifies if the text is to be displayed in the action key of the keyboard.

**Default:** Done

The following are the available options:

- Done
- Go
- Search

- Next
- Send
- Google
- Join
- Route
- Yahoo
- Call

**Note:** This property is specific to the iOS platform.

#### 5.20.4.13 Tool Tip

Specifies the hint text when the cursor hovers over a widget, without clicking it. The text entered in the tooltip appears as a small box when the cursor hovers over a widget.

**Note:** This property is specific to the Windows Tablet platform.

### 5.20.5 Actions

TextArea widget has the following actions associated with it:

- onTextChange: This action is triggered when text in the TextArea widget changes. This event is not fired when the text is changed programmatically.
- onDone: This action is triggered when the user is done with entering text in TextArea widget and click or touch the **Go** or **Enter** option.

For more information on using the above actions, refer to section [Action Editor](#).

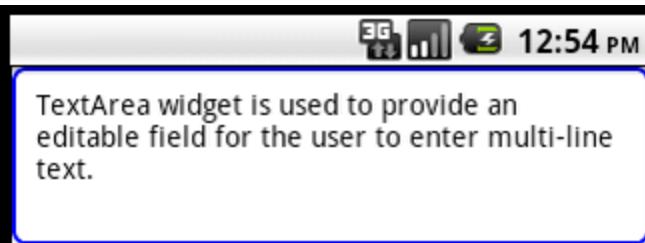
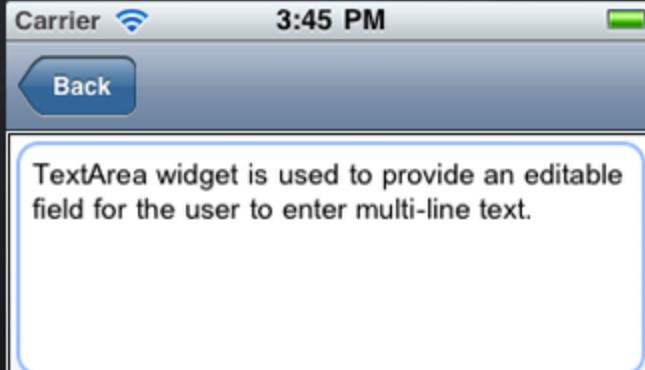
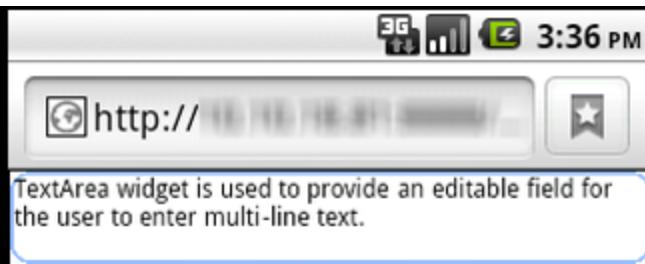
## 5.20.6 Placement Inside a Widget

A TextArea2 can be directly added inside the following widgets:

| Widget    | TextArea2 placement inside a widget                       |
|-----------|---|
| Form      | Yes   |
| HBox      | Yes   |
| VBox      | Yes   |
| ScrollBox | Horizontal Orientation - Yes<br>Vertical Orientation- Yes |
| Tab       | Yes   |
| Segment   | No  |
| Popup     | Yes   |
| Template  | Header- No<br>Footer- No                                  |

## 5.20.7 Widget Appearance on Platforms

The appearance of the TextArea widget with a specified text on various platforms is as follows:

| Platform | Appearance   |
|----------|--|
| Android  |  <p>TextArea widget is used to provide an editable field for the user to enter multi-line text.</p>   |
| iOS      |  <p>TextArea widget is used to provide an editable field for the user to enter multi-line text.</p>  |
| SPA      |  <p>TextArea widget is used to provide an editable field for the user to enter multi-line text.</p> |

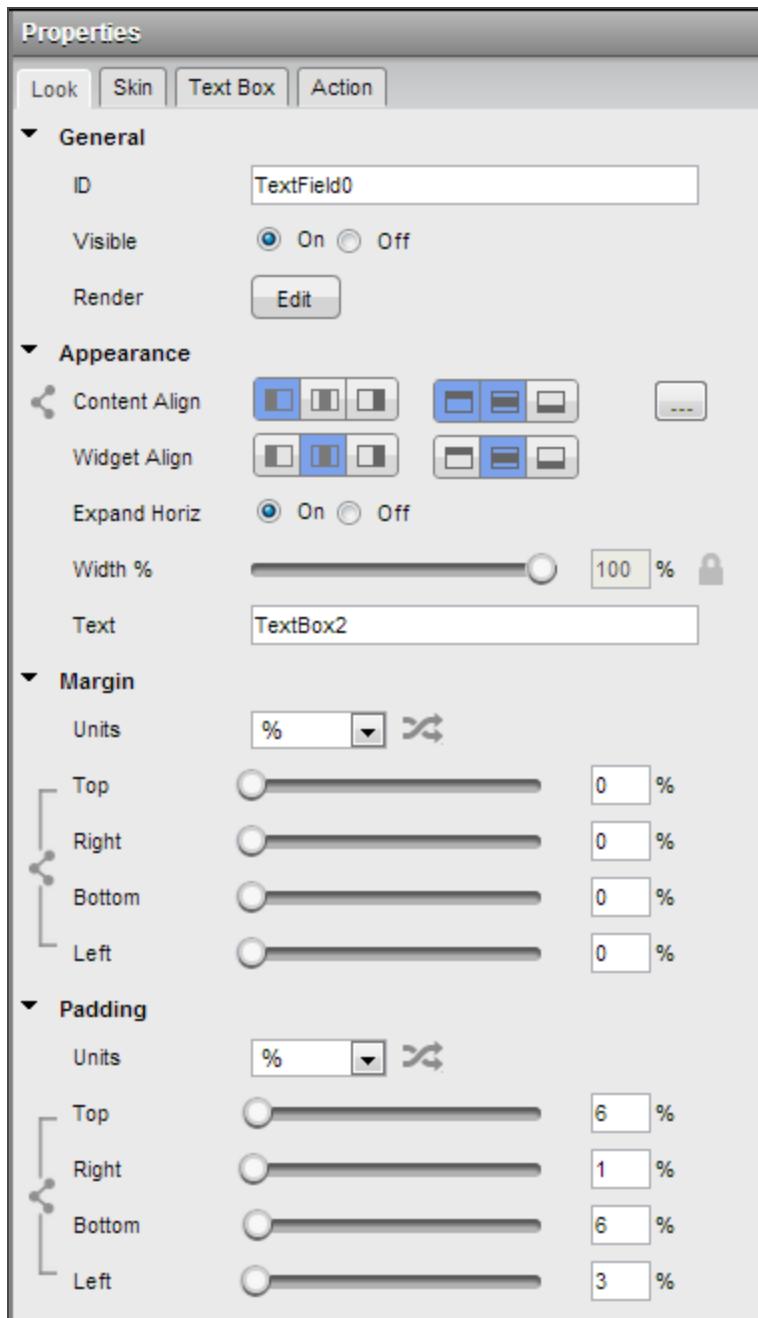
## 5.21 TextBox2

TextBox widget is an editable text component that can be placed on a Form and is used to obtain an input from a user. You can use the TextBox widget to provide a field where a user can enter input text.

For example, in the "Login" page of an Application, you can place two TextBox widgets for Login and Password fields and instruct the users to enter their login credentials in those fields.

### 5.21.1 Look

This section details the **Properties > Look** tab of a **TextBox2** widget.



To specify the look properties of a TextBox2 widget, refer section: [Look](#).

### 5.21.2 Skin

Skin defines properties such as background color, borders and shadows for a widget. They also, define the widget font (applicable only if text is displayed on a widget.)

For the TextBox widget, you can apply a skin for the following states:

- Normal: Specifies the look and feel of the widget when not in focus.
- Focus: Specifies the skin that is applied when there is focus on a widget.
- Blocked UI: Specifies the skin that must be used to block the interface until the action in progress (for example, a service call) is completed.

**Note:** Blocked UI is available only for SPA platforms.

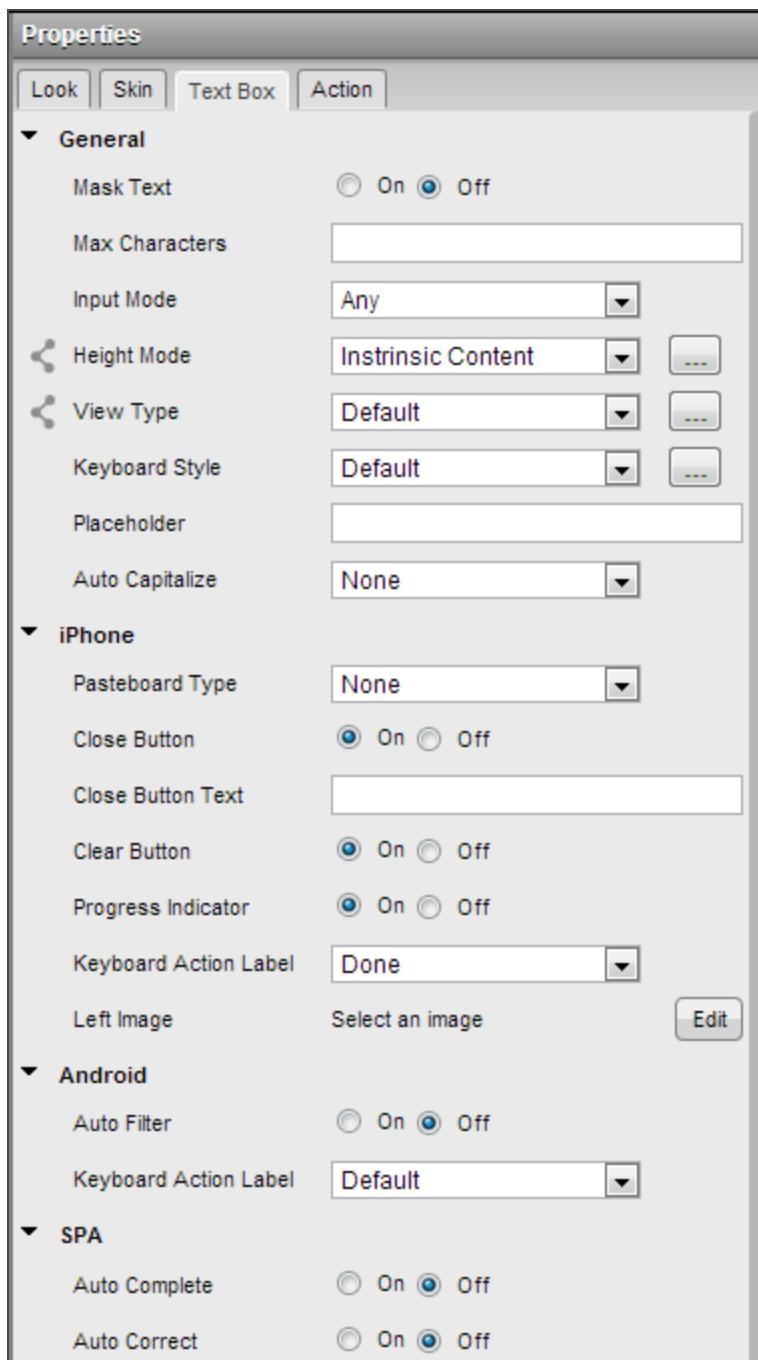
- Placeholder: Specifies the skin to be applied to the placeholder text in the TextBox widget. Only the font color skin attribute is applicable.
- Hover: Specifies the look and feel of a widget when the cursor hovers on the widget.

**Note:** Hoover Skins is available only on the Windows (native) Tablet platform.

Refer to section [Working with Skins](#) to know more about adding a skin to a widget.

### 5.21.3 Platform Specific Properties

**Note:** In this section, the properties that can be forked are identified by an icon ( located to the left of the property. For more details, refer to section [Forking a Widget property](#).



#### 5.21.3.1 Mask Text

Specifies whether the text entered by the user will be secured using a mask character, such as asterisk or dot. This is typically set to true for a password field.

Default: Off (Mask Text is disabled)

- If set to **On**, the text in the TextBox will be masked.
- If set to **Off**, the text in the TextBox will not be masked.

#### 5.21.3.2 Max Characters

Specifies the maximum number of characters that the text field can accept.

If you specify a number for this property, the number of input characters cannot exceed the specified number.

#### 5.21.3.3 Input Mode

Specifies the type of input characters that a user can enter into the TextBox widget. You can use this property to restrict the input characters to only numbers or a combination of alphabets, numbers, and special characters.

Following are the available options:

- Any: Specifies that the input characters can be letters, numbers, special characters, or a combination of all three of them.
- Numeric: Specifies that the input characters must be numbers only. .

#### 5.21.3.4 Height Mode

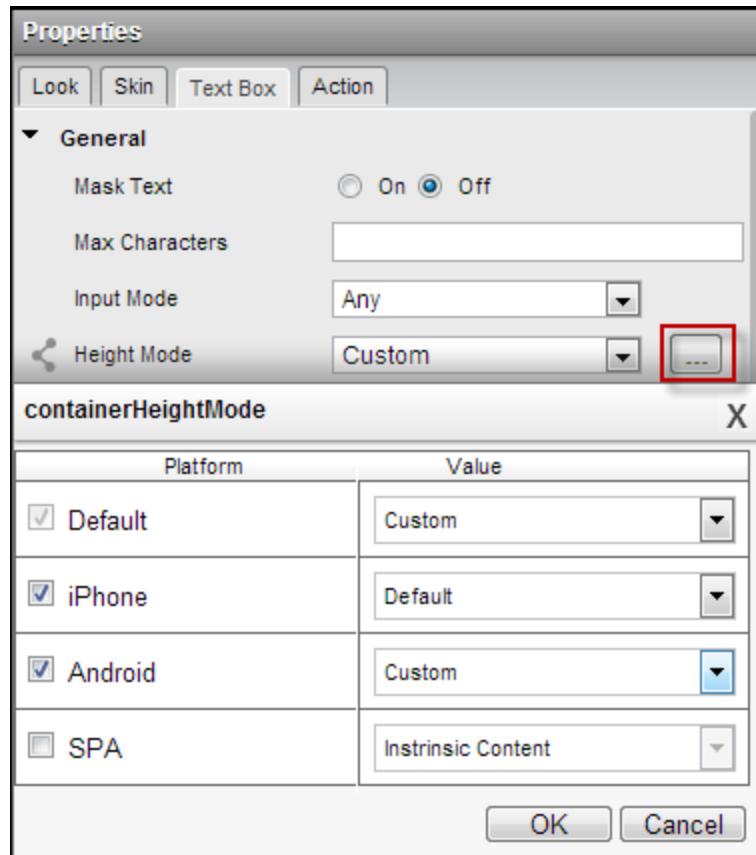
Specifies the widget height based on the selected option.

The following are the available options:

- Intrinsic Content: This is default option for iPhone and SPA platform and is available on those platforms only.
- Custom: This option is supported by all platforms and accepts height in percentage (%). The percentage is evaluated using the ContainerHeight.

The option selected here becomes the default **Height Mode** for all the platforms.

To provide a platform specific or even a default value for the Height Mode, click the Ellipsis button (...) next to the **Height Mode** drop-down field to open the **containerHeightMode** dialog box.



From here, you can select the desired platform, and from the corresponding **Value** drop-down field, select a value.

#### 5.21.3.5 Container Height

Specifies the height of the TextBox2 widget. The height is calculated based on the selected [Height Reference](#) option. This property is enabled when you set the [Height Mode](#) value as Custom.

**Note:** This property is unavailable on Flex Forms.

#### 5.21.3.6 Height Reference

The TextBox2 height percentage is calculated based on the option selected.

- Form Reference: The TextBox2 height percentage is calculated based on the height of the form excluding headers and footers. This option is not respected if TextBox2 is placed inside a popup or in templates.
- Parent Width: This option is used if the TextBox2 is placed inside a popup or in templates. The width is calculated based on the width of the parent container.

**Note:** This property is unavailable on Flex Forms.

#### 5.21.3.7 View Type

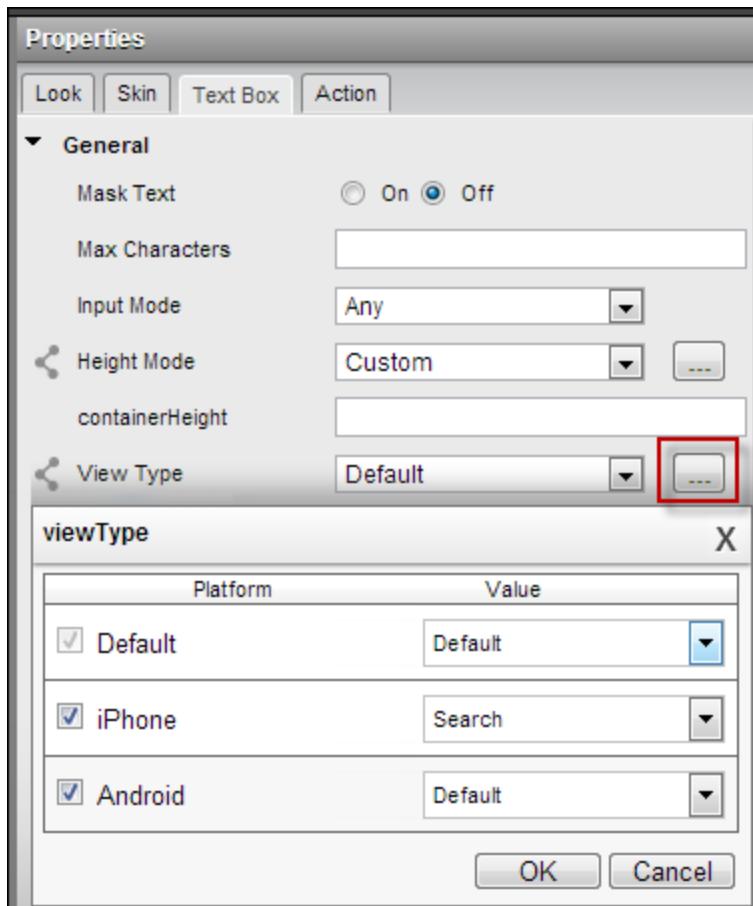
Specifies the appearance of the TextBox widget either as a Search box or as a text box.

You can select one of the following views:

- Default: This is the default selection. This view allows you to use the TextBox widget as a text field.
- Search: This view allows you to use the TextBox widget as a Search field.

The View Type selected here becomes the default for all the platforms.

To provide a platform specific value or even a default value for the View Type, click the Ellipsis button (...) next to the **View Type** drop-down field to open the **View Type** dialog box.



From here, you can select a desired platform, and from the corresponding **Value** drop-down field, select a value.

#### 5.21.3.8 keyBoard Style

When you interact with a TextBox widget, a keyboard is displayed. You can use this property to select the type of keyboard that you want to display.

The following are the available keyboard types when you select [Input Mode](#) as Any.

- Default: Specifies the default keyboard in respective platforms
- Email: Specifies the keyboard to enter email address.

- URL: Specifies the keyboard to enter URL address.
- Chat: Specifies the keyboard type which is helpful for chatting.

**Note:** This property is specific to the Android platform.

#### 5.21.3.9 Placeholder

The placeholder attribute specifies a short hint that describes the expected value of an input field (example, a sample value or a short description of the expected format). The hint is displayed in the input field when it is empty, and disappears when the field gets focus.

For example, for the Username field, you can enter the placeholder text as *Enter User ID* or *Email Address*. The user then clicks on the TextBox widget and enters the Username.

#### 5.21.3.10 Auto Capitalize

Specifies the character capitalization behavior.

Following are the options available:

- None: If you leave this option unchanged, no action takes place on the input string.

Example : This is sample text.

- Words: This option changes the first character of all the words to uppercase. (Not supported on Mobile Web)

Example : This Is Sample Text.

- Sentences: This option changes the first character of all the sentences to uppercase.

Example : This is sample text.

- All: This option changes all the characters to uppercase. (Not supported on Mobile Web)

Example : THIS IS SAMPLE TEXT.

### 5.21.3.11 Pasteboard Type

This property enables an application to share data within the application or with another application using system-wide or application-specific paste boards.

Typically, an object in the application writes data to a pasteboard when the user requests a copy or cut operation on a selection in the user interface. Another object in the same or different application then reads that data from the pasteboard and presents it to the user at a new location; this usually happens when the user requests a paste operation.

**Note:** You can only paste the text to a textbox with the same pasteboard type as that of the source textbox. For example, if you set the Pasteboard type as *App Level Persistent*, you can paste the text only to another textbox whose pasteboard type is also set to *App Level Persistent*.

The different pasteboard types are as follows:

- Default: If you select this option, the value selected in the application properties gets applied.
- System Level
- App Level Non-Persistent
- None: If you select this option, if you want to disable the content to be copied from a TextBox.

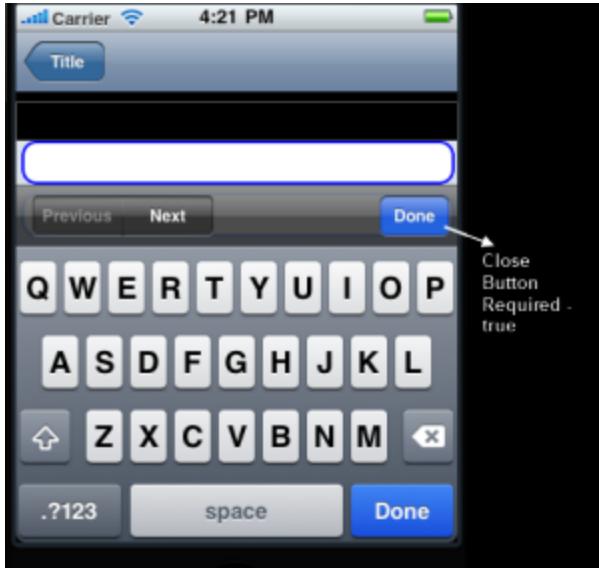
**Note:** This property is specific to the iOS platform.

### 5.21.3.12 Close Button

Specifies if the "Done" button that appears in the keypad (opens when you select text box) must be visible or not.

- If set to **Off**, the "Done" button is not visible on the textbox.
- If set to **On**, the "Done" button is visible on the textbox.

The following image illustrates the Keypad when the property is set to true:



**Note:** This property is specific to the iOS platform.

#### 5.21.3.13 Close Button Text

Specifies the text to replace the "Done" button that appears in the Keypad (opens when you select a text box). This property is available only when [Close Button](#) is enabled.

**Note:** This property is specific to the iOS platform.

#### 5.21.3.14 Clear Button

Specifies if a "Clear" button must be provided. You can use the "Clear" button to erase the text in the TextBox widget.

- If set to **Off**, the clear button is not provided to the textbox.
- If set to **On**, the clear button is provided to the textbox

**Note:** This property is specific to the iOS platform.

### 5.21.3.15 Progress Indicator

Specifies if there must be an indication to the user that the widget content is being loaded.

You can use this property typically for forms that require network calls during post show.

- If set to **Off**, the progress indicator is not displayed.
- If set to **On**, the progress indicator is displayed.

The following image illustrates the progress indicator on iPhone:

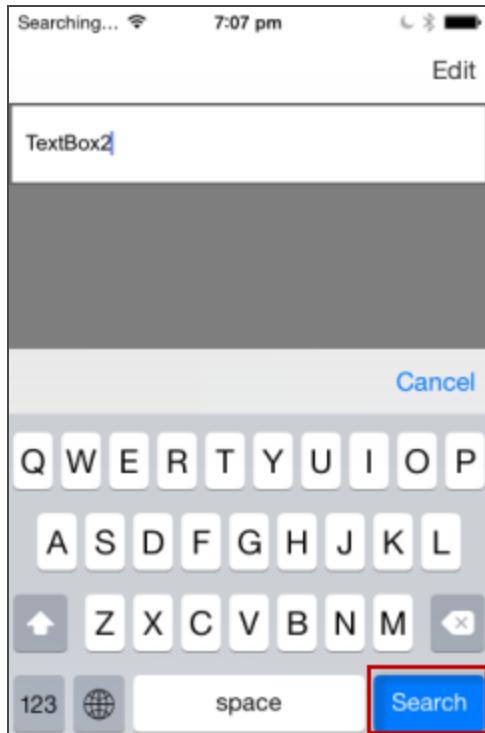


**Note:** This property is specific to the iOS platform.

### 5.21.3.16 Keyboard Action Label

Specifies if the text to be displayed in action key of the keyboard.

**Default:** Done



The following are the available options:

- Done
- Go
- Search
- Next
- Send
- Google
- Join
- Route
- Yahoo
- Call

**Note:** This property is specific to the iOS platform.

#### 5.21.3.17 Left Image

Specifies the image that must be displayed on the left-hand side within a TextBox widget. For example, if you want a magnifying glass image to be displayed to indicate "Search" option, you can use this property to display the image.

##### Placeholder Text



For inserting a left view image, click the **Edit** button against the **Left Image** field to open the **LeftViewImage** dialog box.

You can either:

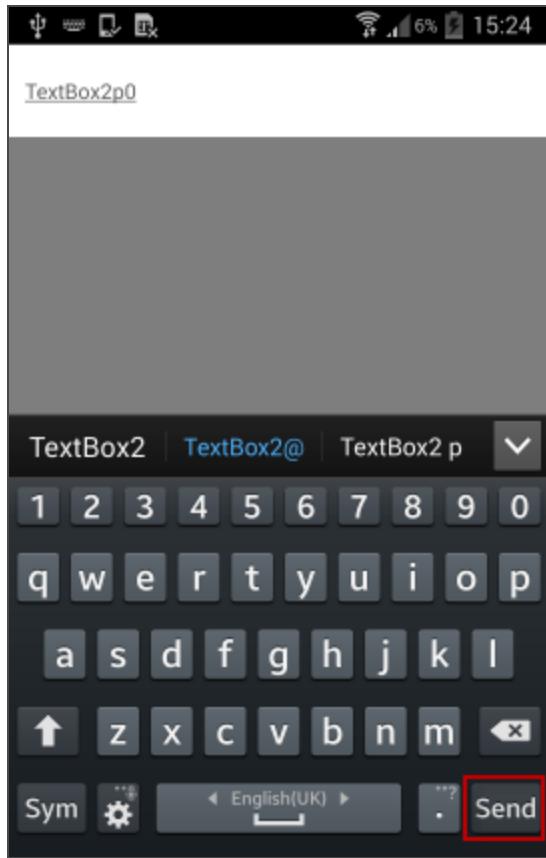
- Select a available image.
- Provide an image URL.

**Note:** This property is specific to the iOS platform.

#### 5.21.3.18 Keyboard Action Label

Specifies if the text to be displayed in action key of the keyboard.

**Default:** Done



The following are the available options:

- Done
- Go
- Search
- Next
- Send
- Google
- Join
- Route

- Yahoo
- Call

**Note:** This property is specific to the Android platform.

#### 5.21.3.19 Auto Filter

Specifies if the input characters you enter in the TextBox widget must be matched against the Filter List and possible matches be displayed.

**Default:** Off ( indicating that the input characters are not matched against the filterlist)

If you want the input characters to be matched against the filterlist and possible matches to be displayed, select the On Radio button.

**Note:** This property is specific to the Android platform.

#### 5.21.3.20 Auto Complete

Enables users to quickly find and select from a pre-populated list of values as you type, leveraging searching and filtering.

- If **On** is selected, the word suggestion is enabled.
- If **Off** is selected, the word suggestion is not enabled.

**Note:** This property is specific to the SPA platform.

#### 5.21.3.21 autoCorrect

This property determines whether auto-correction is enabled or disabled during typing. With auto-correction is enabled, the text object tracks unknown words and suggests a more suitable replacement to the user.

- If **On** is selected, the auto correction option is enabled.
- If **Off** is selected, the auto correction option is not enabled.

You can choose to provide a platform specific autoCorrect property by clicking ellipsis (...) button next to the autoCorrect **Off** radio button.

In the **AutoCorrect** dialog box, select the platform and from the corresponding value field, select the check box to enable or disable auto correction property.

**Note:** You cannot execute this property on App Canvas.

**Note:** This property is specific to the SPA platform.

#### 5.21.3.22 Tool Tip

Specifies the hint text when the cursor hovers over a widget, without clicking it. The text entered in the tooltip appears as a small box when the cursor hovers over a widget.

**Note:** This property is specific to the Windows Tablet platform.

#### 5.21.4 Actions

TextBox widget has the following actions associated with it:

- **onTextChange:** This action is triggered when text in the TextBox widget changes. This event is not fired when the text is changed programmatically.
- **onDone:** This action is a callback that is invoked by the platform when the user performs a click action on the **Go** or **Enter** button.
- **onBeginEditing:** This action is invoked by the platform when the user clicks within the TextBox and is about to start editing.

- **onEndEditing:** This action that is invoked by the platform when the user performs one of the below actions:
  - Click on any other focusable widget(for example, another TextBox)
  - Click on the Done button on the Next Previous bar.
  - Click on the Done button on the keypad.

When you click on the Done button of the keypad the following events take place in a sequence:

- onendediting
- ondone



For more information on using the above actions, refer to section [Action Editor](#).

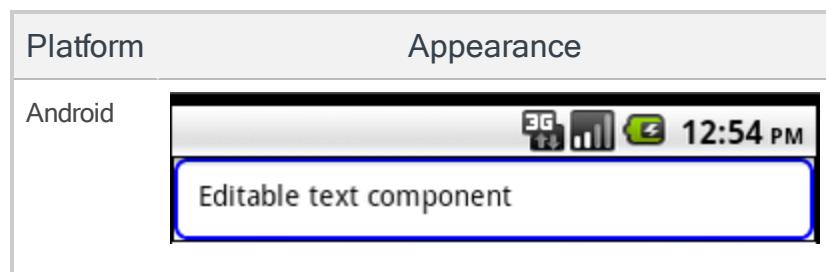
### 5.21.5 Placement Inside a Widget

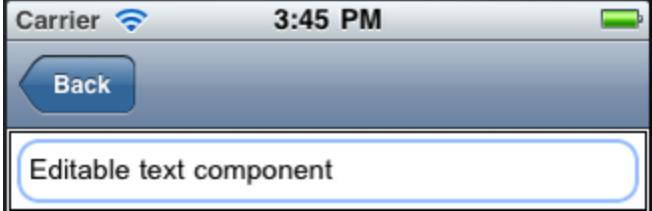
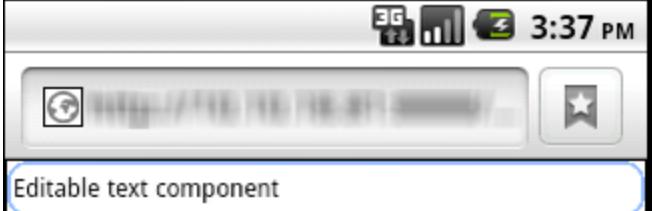
A TextBox2 can be directly added inside the following widgets:

| Widget     | TextBox2 placement inside a widget                        |
|------------|---|
| Form       | Yes   |
| HBox       | Yes   |
| VBox       | Yes   |
| ScrollView | Horizontal Orientation - Yes<br>Vertical Orientation- Yes |
| Tab        | Yes   |
| Segment    | No  |
| Popup      | Yes   |
| Template   | Header- No<br>Footer- No                                  |

### 5.21.6 Widget Appearance on Platforms

The appearance of the TextBox widget with the Text "Editable text component" on various platforms is as follows:



| Platform | Appearance   |
|----------|--|
| iOS      |  A screenshot of an iOS device interface. At the top, there's a status bar with "Carrier" and signal strength, followed by the time "3:45 PM" and a battery icon. Below the status bar is a blue "Back" button with a white arrow pointing left. The main content area contains a white rectangular box with rounded corners and a thin blue border, labeled "Editable text component". |
| SPA      |  A screenshot of a Single Page Application (SPA) interface. At the top, there's a header bar with icons for location, search, and notifications, along with the time "3:37 PM". Below the header is a white rectangular box with rounded corners and a thin blue border, labeled "Editable text component".   |

## 5.22 Browser

Use Browser widget to display HTML content of your application without navigating away from the application or opening the native browser. The HTML content can be either static or obtained from a URL.

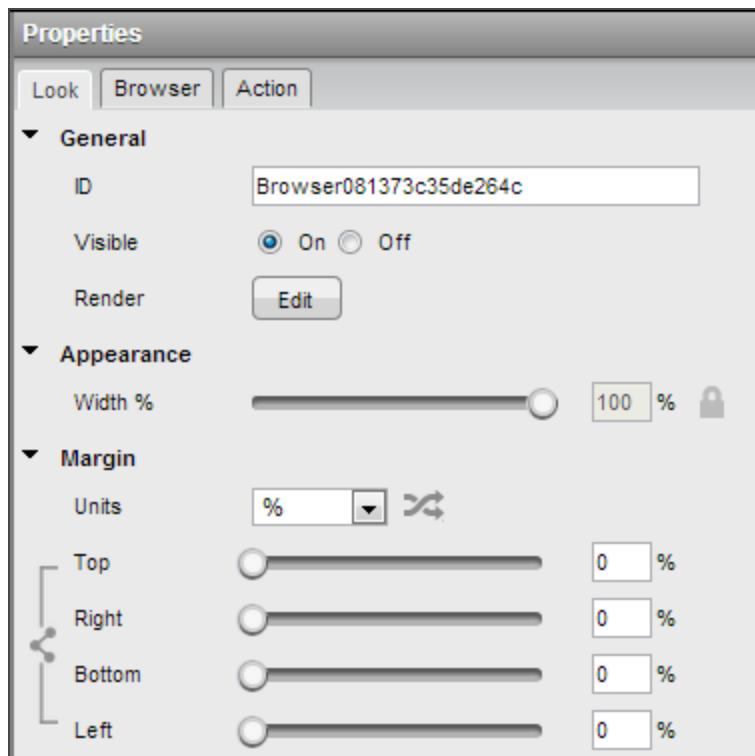
### 5.22.1 Important Considerations

The following are the important considerations:

- For iOS platform, when the Browser widget is set as non-screen level widget and you keep some widgets before browser and after browser then double scrolling issue will be seen which is native iOS issue. We recommend you to use browser widget as screen level widget and move other widgets to header or footer of the form.
- The Browser widget, unlike other widgets, is considered to be "heavy" widget as far as memory and performance is considered.
- The Browser widget uses a lot of initial RAM. The RAM usage grows in proportion to the number of images and static text rendered.
- If there are multiple instances of the Browser widget in the same application, there may be issues related to sharing of information. For example, cookies etc.
- You must not place multiple Browser widgets in a form. As a guideline, you must not have more than two Browser widgets in an application.
- You must not use the Browser widget as a RichText widget. It should only be used to display large HTML content.
- You must avoid using the Browser widget to create an application which looks and behaves like a mini web browser. Users normally expect to use the native browser to browse web content. Hence, replicating this functionality within your application is not recommended.

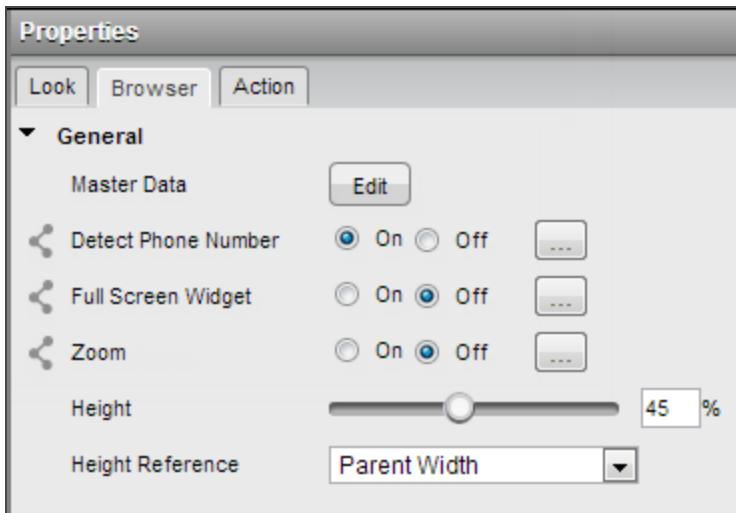
## 5.22.2 Look

This section details the **Properties > Look** tab of a **Browser** widget.



To specify the look properties of a Browser widget, refer section: [Look](#).

### 5.22.3 Platform Specific Properties



#### 5.22.3.1 Master Data

Click **Edit** button against **Master Data** to open the **Master Data** dialog box. Here, you have the following options:

- Content: When you select this option, a text field is displayed, where you can write the content to be displayed in the Browser widget.
- URL: Specifies the initial URL that must be requested from the server. The URL must begin with `http://`.

#### 5.22.3.2 Detect Phone Number

Specifies if the Browser widget must support the detection of phone numbers in the web page and display the phone numbers as clickable Phone links. If you click the Phone link, the Phone application launches and dials the number.

- Click **On** to enable telephone number detection.
- Click **Off** to disable telephone number detection.

The option you choose here becomes the default for all the platforms.

You can change this default option or provide a platform specific option, by forking the **Detect Tel Number** property. Refer to section [Forking a Widget Property](#) for more details.

#### 5.22.3.3 Full Screen Widget

Specifies whether the widget should occupy the whole form when your Browser widget has a large HTML content to display. You must enable this feature so that Browser widget occupy the complete Form and results in a good user experience.

If disabled, a scroll bar appears either on the Browser widget or on the Form.

- Click **On** to allow the widget occupy the whole container, when the Browser widget has a large HTML content to display.
- Click **Off** to disallow the widget occupy the whole container when the Browser widget has a large HTML content to display.

The option you choose here becomes the default for all the platforms.

You can change this default option or provide a platform specific option, by forking the **Screen Level Widget** property. Refer to section [Forking a Widget Property](#) for more details.

**Note:** You must not place more than one Browser widget as a screen level widget inside a Form. Also, if you choose to make a Browser widget a Screen Level Widget, you must place only the Browser widget in the Form and must not place any other widgets in the Form.

**Note:** Do not set the screen level widget property to true for more than one widget in the form. If you have multiple widgets with this property set as true, there may be issues with how information is displayed along with some scrolling issues.

**Note:** This property is unavailable on Flex Forms.

#### 5.22.3.4 Zoom

Specifies if Zoom (ability to change the scale of the view area) must be enabled.

- If set to **On**, the Zoom feature is enabled.
- If set to **Off**, the Zoom feature is disabled.

The option you choose here becomes the default for all the platforms.

You can change this default option or provide a platform specific option, by forking the **Enable Zoom** property. Refer to section [Forking a Widget Property](#) for more details.

#### 5.22.3.5 Height

Specifies the height of the Browser in terms of percentage. The percentage is with reference to the value of [Height Reference](#) property.

**Note:** This property is unavailable on Flex Forms.

#### 5.22.3.6 Height Reference

The Browser height percentage is calculated based on the option selected.

- Form Reference: The Browser height percentage is calculated based on the height of the form excluding headers and footers. This option is not respected if Browser is placed inside a popup or in templates.
- Parent Width: This option is used if the Browser is placed inside a popup or in templates. The width is calculated based on the width of the parent container.

**Note:** This property is unavailable on Flex Forms.

#### 5.22.4 Actions

Browser widget has the following actions associated with it:

- onFailure: This action is invoked by the platform when the given request URL is failed to load the data.

**Note:** This action is executed only for the given request URL, but not for the subsequent web navigation request failures.

- **onSuccess:** An action callback which gets invoked when the given request URL is successful in loading the data.

**Note:** This action is called only for the given request URL, but not for the subsequent web navigation requests.

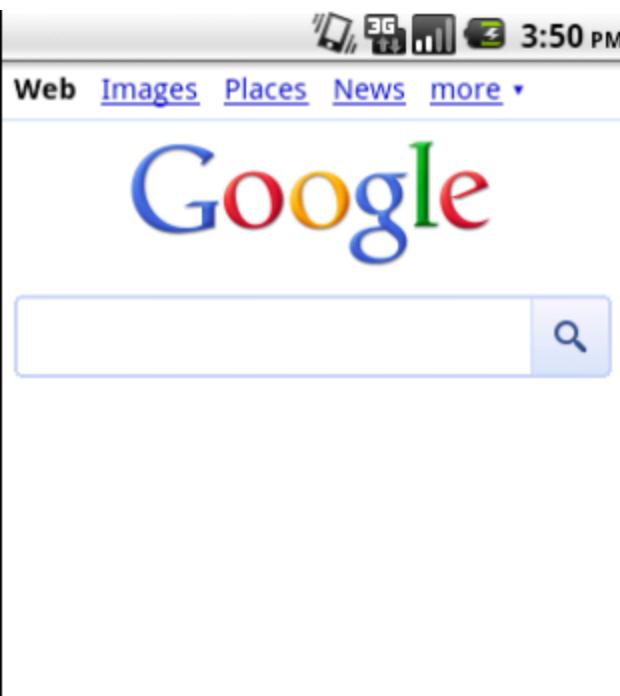
### 5.22.5 Placement Inside a Widget

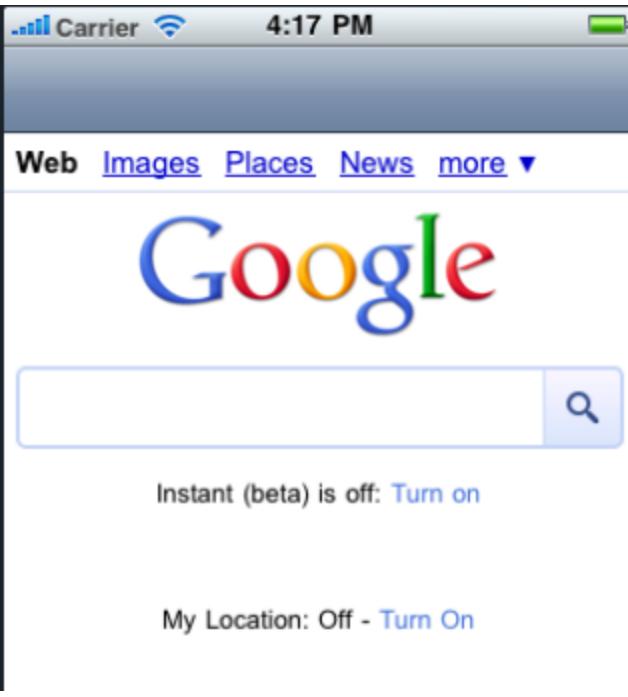
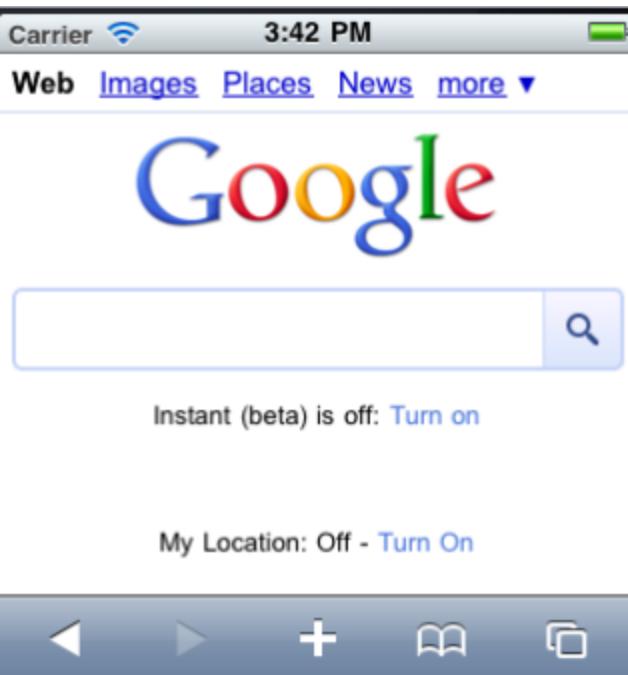
A browser can be directly added inside the following widgets:

| Widget    | Browser placement inside a widget                        |
|-----------|--|
| Form      | Yes  |
| HBox      | Yes  |
| VBox      | Yes  |
| ScrollBar | Horizontal Orientation -Yes<br>Vertical Orientation- Yes |
| Tab       | Yes  |
| Segment   | No   |
| Popup     | Yes  |
| Template  | Header- No<br>Footer- No                                 |

## 5.22.6 Widget Appearance on Platforms

The appearance of the Browser widget on various platforms is as follows:

| Platform | Appearance   |
|----------|--|
| Android  |  A screenshot of a Kony Visualizer browser widget on an Android device. The top status bar shows signal strength, battery level, and the time (3:50 PM). Below the status bar is a navigation bar with tabs for Web, Images, Places, News, and more. The main content area displays the Google homepage with the "Google" logo and a search bar. |

| Platform | Appearance   |
|----------|--|
| iOS      |  A screenshot of an iOS mobile browser. At the top, there's a status bar showing signal strength, 'Carrier', a Wi-Fi icon, the time '4:17 PM', and a battery icon. Below the status bar is a navigation bar with tabs: 'Web' (which is bolded), 'Images', 'Places', 'News', and 'more ▾'. The main content area displays the Google homepage with its signature multi-colored 'Google' logo. Below the logo is a search bar with a blue search button containing a white magnifying glass icon. At the bottom of the screen, there's a message: 'Instant (beta) is off: Turn on' and 'My Location: Off - Turn On'. The overall interface is clean and follows iOS design guidelines. |
| SPA      |  A screenshot of a Single Page Application (SPA) mobile browser. It has a similar layout to the iOS browser above it. The top features a status bar with 'Carrier', a Wi-Fi icon, the time '3:42 PM', and a battery icon. Below is a navigation bar with tabs: 'Web' (bolded), 'Images', 'Places', 'News', and 'more ▾'. The main content shows the Google homepage with its logo and search bar. A message at the bottom says 'Instant (beta) is off: Turn on' and 'My Location: Off - Turn On'. At the very bottom of the screen is a dark blue footer bar with five white icons: a left arrow, a right arrow, a plus sign, a bookmark, and a square.                             |

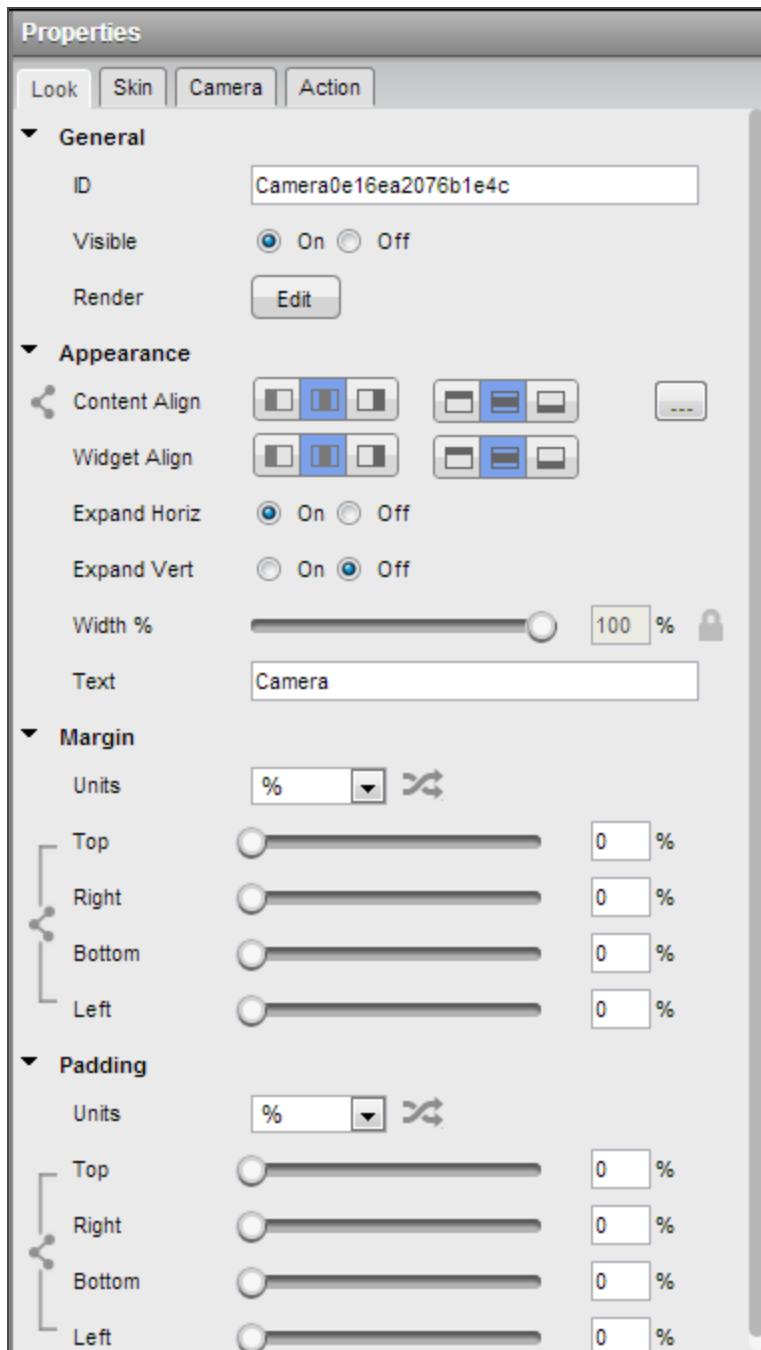
## 5.23 Camera

A Camera widget appears as a button in a form. If you select the Camera widget, the phone native camera application is invoked to capture an image (which you can choose to accept or discard) and is stored as a PNG (Portable Network Graphics) image by default with the original size

**Note:** The Camera widget is not supported on SPA platforms.

### 5.23.1 Look

This section details the **Properties > Look** tab of a Camera widget.



To specify the look properties of a Camera widget, refer section: [Look](#).

## 5.23.2 Skin

Skin defines properties such as background color, borders and shadows for a widget. They also, define the widget font (applicable only if text is displayed on a widget).

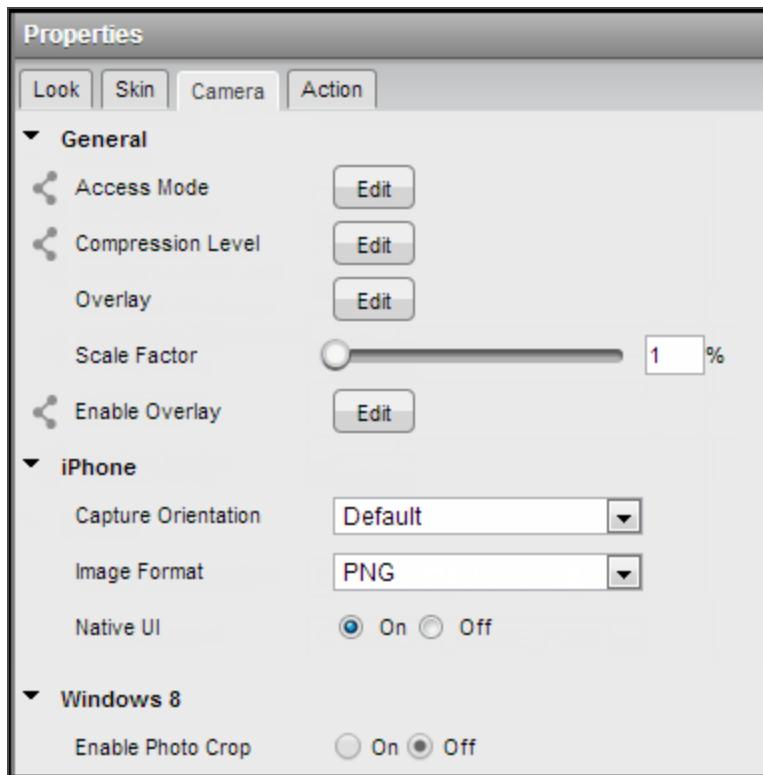
For the Camera widget, you can apply a skin for the following states:

- Normal: Specifies the look and feel of the widget when not in focus.
- Focus: Specifies the skin that is applied when there is focus on a widget.
- Hover Skin: Specifies the look and feel of a widget when the cursor hovers on the widget. This skin is available only on Windows Tablet platform.

Refer to section [Working with Skins](#) to know more about adding a skin to a widget.

## 5.23.3 Platform Specific Properties

**Note:** In this section, the properties that can be forked are identified by an icon ( located to the left of the property. For more details, refer to section [Forking a Widget property](#).



### 5.23.3.1 Access Mode

Specifies how the captured image must be stored.

Following are the available options:

- Public: The captured image is stored on the device as a Image and is accessible to all the applications on the device. For example, the captured images are accessible in ImageGallery.
- Private: This is the default option for Windows. The captured image is stored as an Image on the device and will not be accessible to any other application on the device and remain private to the application.
- In-Memory: The captured camera image is stored in memory and is never written to the disk. These captured images are lost when the application is closed.

To specify the access mode, click the **Edit** button against the **Access Mode** field to open the **Access Mode** dialog box.

Default: Public

- For changing default access mode: From the **value** list (corresponding to the Default platform), select a desired access mode.
- For providing platform specific access mode: Select the platform and from the corresponding drop-down **value** field, select desired access mode.

#### 5.23.3.2 Compression Level

Specifies the compression level or picture quality with which the captured image must be stored. You can specify the compression level value between 0 (best picture quality) and 100 (low picture quality).

To specify the compression level, click the **Edit** button against the **Compression Level** field to open the **Compression Level** dialog box.

Default: 0 ( The image is stored with the best picture quality)

- For changing default compression level: In the **value** field (corresponding to the Default platform), enter a new compression level value.
- For providing platform specific compression level: Select the platform and in the corresponding **value** field, enter a compression level value.

**Note:** Applicable only when the image format is jpeg.

#### 5.23.3.3 Overlay

Specifies the overlay configuration parameters for overlaying a form.

The following are the configurable properties available for various platforms:

iOS

- Overlay Form : Specifies the reference of the form to be rendered over the camera view. When this option is set, the [Capture Orientation](#) property is not respected.

Default : None

- Cropping Reference Image : Specifies the reference of the Image widget in the Overlay Form which guides the camera to crop the captured image to the Reference Image Dimensions.

Default : None

## Android

- Overlay Form : Specifies the reference of the form to be rendered over the camera view. When this option is set, the [Capture Orientation](#) property is not respected.

Default : None

- Cropping Reference Image : Specifies the reference of the Image widget in the Overlay Form which guides the camera to crop the captured image to the Reference Image Dimensions.

Default : None

- Capture Button Skin : Specifies the skin for a captured button.
- Capture Button Text : Specifies the text for a captured button.
- Tap Anywhere: Specifies to capture an image with a tap on the camera overlay view.

Default : false

## Windows 8

- Cropping Reference Image : Specifies the reference of the Image widget in the Overlay Form which guides the camera to crop the captured image to the Reference Image Dimensions.

Default : None

- Overlay Form : Specifies the reference of the form to be rendered over the camera view. When this option is set, the [Capture Orientation](#) property is not respected.

Default : None

- Tap Anywhere: Specifies to capture an image with a tap on the camera overlay view.

Default : false

#### 5.23.3.4 Scale Factor

Specifies the ratio by which a captured image is scaled down. You can set the scale factor between 10 and 100. If you set the scale factor as 100, no reduction takes place and the actual image is returned. If you set the value as 10, an image which is scaled down to 10% of the actual captured image is returned.

#### 5.23.3.5 Enable Overlay

A camera is launched with capability of over-lay a Form UI over the camera view.

To provide values for **Enable Overlay** property, click the **Edit** button against the **Enable Overlay** field. The **Enable Overlay** dialog box appears.

**Default:** Enable Overlay is disabled.

- To enable overlay by default for all the platforms, click the **Value** check box (corresponding to the **Default** platform).
- To enable overlay property for a specific platform, select a platform check box, and then click the check box of the corresponding **Value** field.

#### 5.23.3.6 Capture Orientation

Specifies the orientation of the captured image.

**Default:** Default

The following are the available options:

- Landscape: On the device the camera is always turned sideways so that the height of the screen becomes width.
- Portrait: On the device the camera is always displayed such that the horizontal sides are shorter than vertical sides.

**Note:** This property is specific to the iOS platform.

### 5.23.3.7 Image Format

Specifies if the image must be stored as a PNG (Portable Network Graphics) or a JPEG (Joint Photographic Experts Group) image.

**Default:** PNG

The following are the available options:

- PNG : When you select this option the image is always stored in PNG format.
- JPEG : When you select this option the image is always stored as JPEG format.

**Note:** This property is specific to the iOS platform.

### 5.23.3.8 Native UI

Specifies if the camera must have the native interface for camera view (an interface with the default platform controls for camera) or the user interface with custom options.

**Default:** enabled

- If disabled, the user interface with custom options is displayed.
- If enabled, the native interface of camera in respective platforms is displayed

**Note:** This property is specific to the iOS platform.

### 5.23.3.9 Enable Photo Crop

Enables you to crop the captured image manually.

**Default:** Off

- Click **On** to enable the photo crop feature.
- Click **Off** to disable the photo crop feature.

**Note:** This property is specific to the Windows platform.

#### 5.23.3.10 Tool Tip

Specifies the hint text when the cursor hovers over a widget, without clicking it. The text entered in the tooltip appears as a small box when the cursor hovers over a widget.

**Note:** This property is specific to the Windows Tablet devices.

#### 5.23.4 Actions

Camera widget has the following action associated with it:

- onCapture: This action is invoked when the user captures a picture.

For more information on using the above action, refer to section [Action Editor](#).

#### 5.23.5 Placement inside a Widget

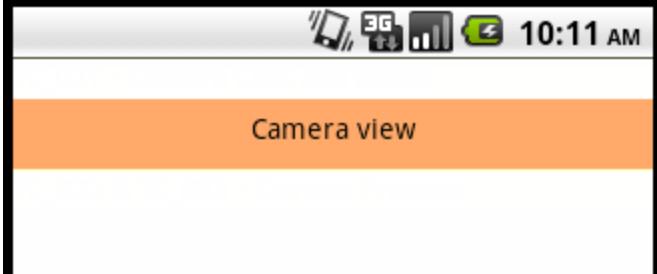
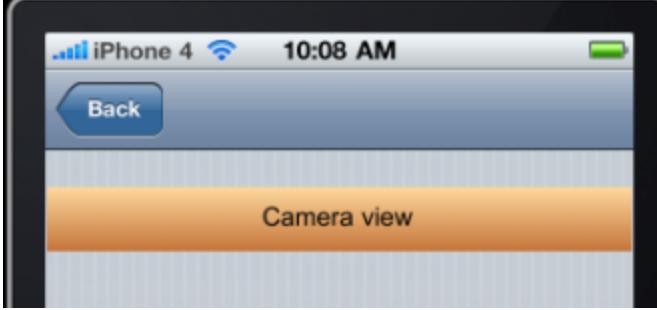
A camera can be directly added inside the following widgets:

| Widget    | Camera placement inside a widget                         |
|-----------|--|
| Form      | Yes  |
| HBox      | Yes  |
| VBox      | Yes  |
| ScrollBar | Horizontal Orientation -Yes<br>Vertical Orientation- Yes |
| Tab       | Yes  |
| Segment   | No   |

|          |                                  |
|----------|----------------------------------|
| Widget   | Camera placement inside a widget |
| Popup    | Yes                              |
| Template | Header- No<br>Footer- No         |

### 5.23.6 Widget Appearance on Platforms

The appearance of the Camera widget on various platforms is as follows:

| Platform | Appearance  |
|----------|---|
| Android  |   |
| iOS      |  |

## 5.24 Horizontal Image Strip2

HorizontalImageStrip also called as Hz Image Strip displays a list of images which are aligned side-by-side in horizontal direction. You can scroll through the Hz Image Strip to view the next or previous set of images.

You can use an Hz Image Strip to display a set of images to give an idea to the user about a products or a location (for example, in a Travel Application, you can add images of popular tourist destinations or add images of the different suites available in a hotel).

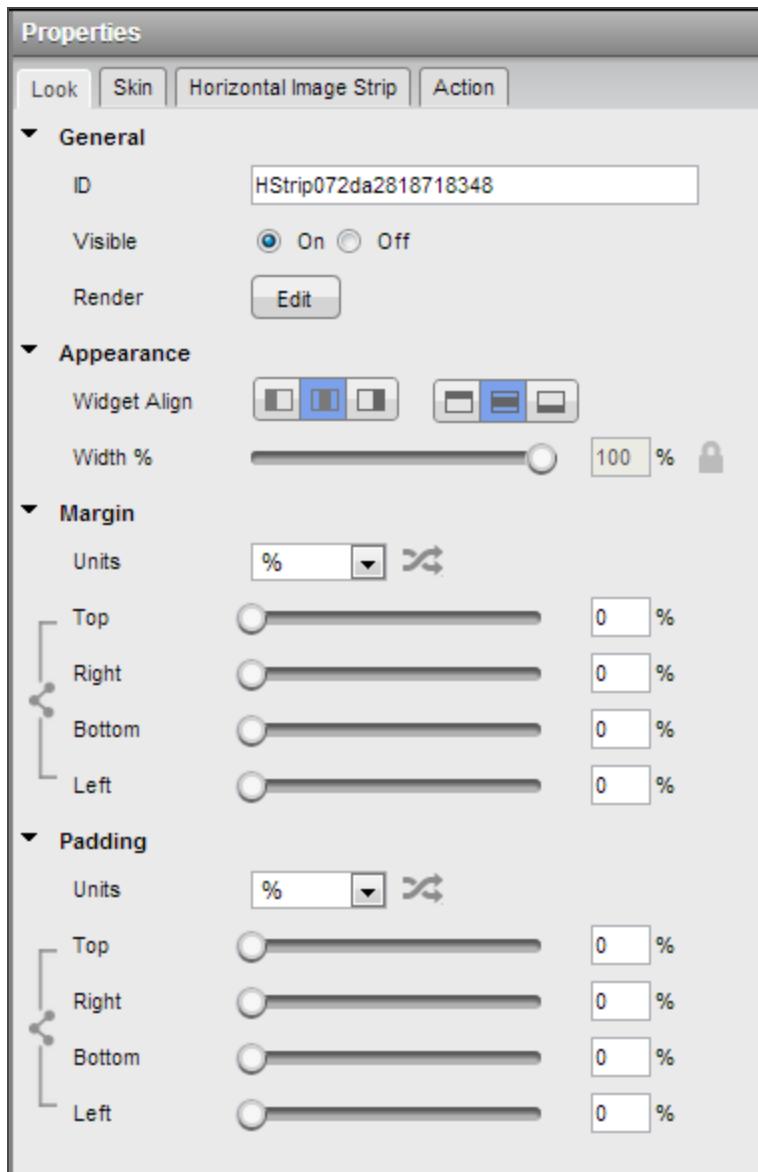
### 5.24.1 Important Considerations

The following are important considerations for a HorizontalImageStrip widget.

- A HorizontalImageStrip2 widget is available only for a Flex Form.
- For a good user experience, you must add images of the same width and height.
- You can scroll through one or more images at a time (platform dependent).

### 5.24.2 Look

This section details the **Properties > Look** tab of a HorizontalImageStrip2 widget.



To specify the look properties of a `HorizontalImageStrip2` widget, refer section: [Look](#).

### 5.24.3 Skin

Skin defines properties such as background color, borders and shadows for a widget. They also, define the widget font (applicable only if text is displayed on a widget.)

For the `HorizontalImageStrip2` widget, you can apply a skin for the following states:

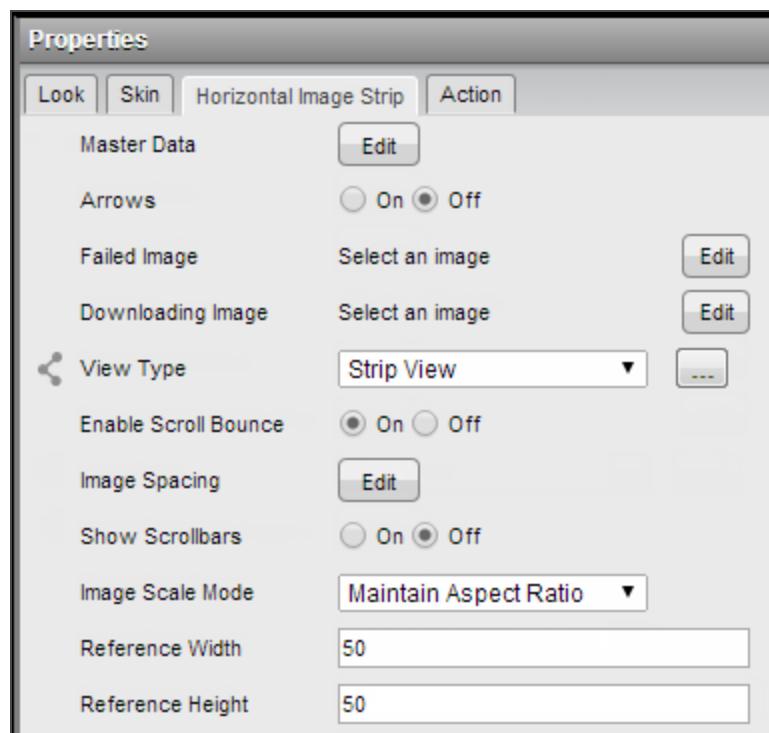
- Normal: Specifies the look and feel of the widget when not in focus.
- Focus: Specifies the skin that is applied when there is focus on a widget.
- Hover Skin: Specifies the look and feel of a widget when the cursor hovers on the widget.

**Note:** Hoover Skin is available only on Windows Tablet platform.

Refer to section [Working with Skins](#) to know more about adding a skin to a widget.

#### 5.24.4 Platform Specific Properties

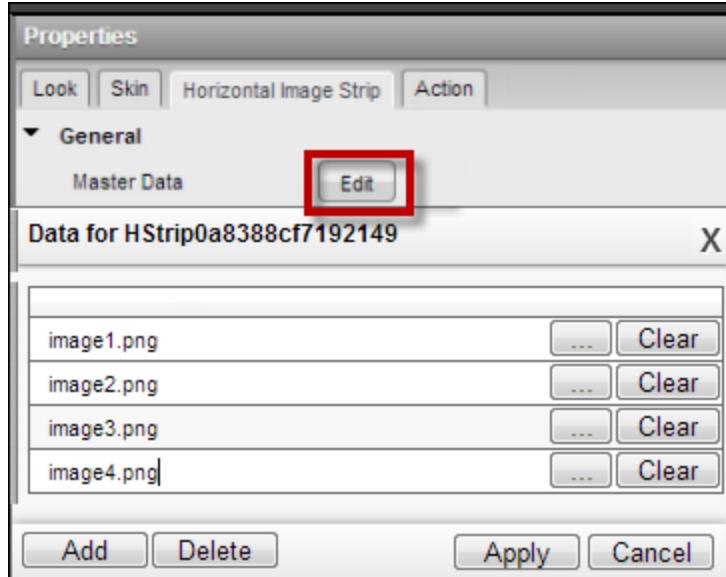
**Note:** In this section, the properties that can be forked are identified by an icon ( ) located to the left of the property. For more details, refer to section [Forking a Widget property](#).



##### 5.24.4.1 Master Data

You can specify all the images that you wanted to place inside this widget.

For providing masterData, click the **Edit** button against **Master Data** to open **Data for HStrip** dialog box.



Click ellipses (...) to open the **Select Image** dialog box. Select an image or provide an url for the image, and click **OK**.

After adding the images in the Master Data, click **Apply** to insert these images.

#### 5.24.4.2 Arrows

Specifies whether the arrow images must be displayed on the left and right edges of the HorizontalImageStrip.

**Default:** disabled

- If enabled, the arrows are displayed.
- If disabled, the arrows are not displayed.

#### 5.24.4.3 Failed Image

Specifies the image to be displayed when the remote resource is unavailable.

To specify an image, click the **Edit** button against the **Failed Image** field to open the **Failed Image** dialog box, from where you can locate an image to be displayed when the remote resource is unavailable.

#### 5.24.4.4 Downloading Image

Specifies the image to be displayed when the remote source is still being downloaded.

To specify an image, click the **Edit** button against the **Downloading Image** field. This results in displaying **Downloading Image** dialog box from where you can locate an image to be displayed when the remote resource is still being downloaded.

#### 5.24.4.5 View Type

Specifies the type of view of HorizontalImageStrip. It displays two different views:

- Strip View
- Slot View

The option selected here becomes the default view type for all the platforms.

For changing the default view type or to provide a platform specific view type, click the ellipsis (...) button next to the **View Type** list. The **View Type** dialog box appears.

Select a platform check box, and from its corresponding **Value** list, select a desired view type.

#### 5.24.4.6 Image Spacing

Specifies the space between the images in the horizontal image strip.

For proving **Image Spacing** value, click the **Edit** button against the **Image Spacing** field to open the **Image Spacing** dialog box.

For providing the value specific to a platform, select a platform check box and type a number in the corresponding **Value** field.

#### 5.24.4.7 Show Scrollbars

Specifies if the scrollbars must be visible all the time.

Default: As per native platform behavior.

- Click **On** to display the scrollbars.
- Click **Off** to hide the scrollbars.

#### 5.24.4.8 Image Scale Mode

The value specifies how an image rendered. Image Widget represents the underlying native widget which renders (and applies alignment to) the Source Image.

Following are the available options:

- Fit to Dimensions: The source image is resized to fill the ImageWidget dimensions.
- Maintain Aspect Ratio: This mode resizes the source image to fit in the ImageWidget dimensions while it preserves its native aspect ratio. In case,
- Crop: This scale mode preserves the original size of the Source Image.

#### 5.24.4.9 Reference Width

It is the reference image width in pixels. These are device independent Pixels specified against 163 dpi.

#### 5.24.4.10 Reference Height

It is the reference image height in pixels. These are device independent Pixels specified against 163 dpi.

#### 5.24.4.11 Tool Tip

Specifies the hint text when the cursor hovers over a widget, without clicking it. The text entered in the tooltip appears as a small box when the cursor hovers over a widget.

**Note:** This property is specific to the Windows Tablet platform.

### 5.24.5 Actions

HorizontalImageStrip2 widget has the following action associated with it:

- **onSelection:** This action is invoked by the platform when an Image is selected in HorizontalImageStrip widget.

For more information on using the above action, refer to section [Action Editor](#).

### 5.24.6 Placement Inside a Widget

A HorizontalImagestrip2 can be directly added inside the following widgets:

| Widget    | HorizontalImagestrip2 placement inside a widget          |
|-----------|--|
| Form      | Yes  |
| HBox      | Yes  |
| VBox      | Yes  |
| ScrollBar | Horizontal Orientation -Yes<br>Vertical Orientation- Yes |
| Tab       | Yes  |
| Segment   | No   |
| Popup     | Yes  |
| Template  | Header- No<br>Footer- No                                 |

## 5.25 ImageGallery2

ImageGallery represents a set of images adjacent to each other. If the images exceed the row size, they are pushed to the next line.

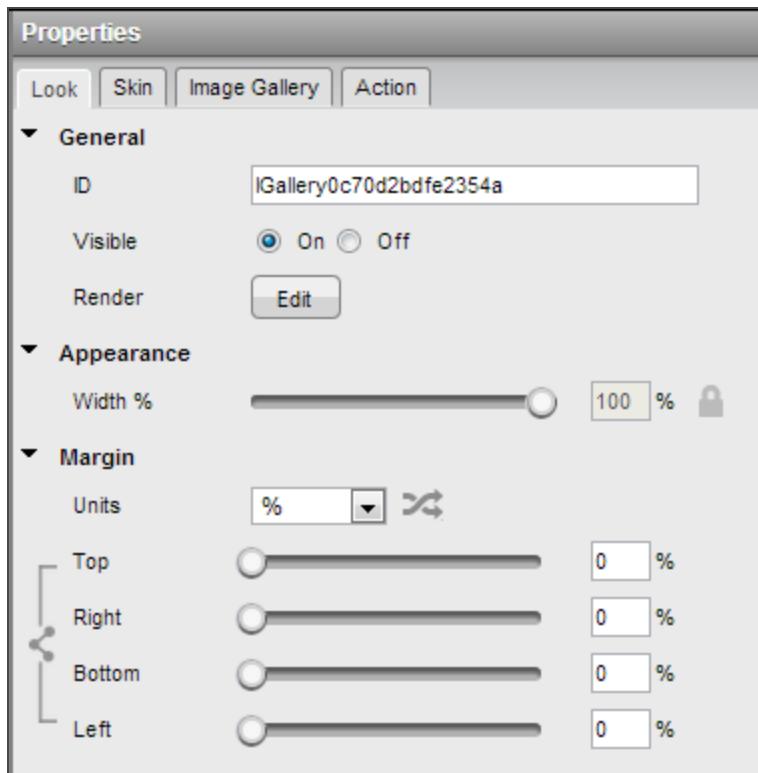
### 5.25.1 Important Considerations

The following are important considerations for an ImageGallery2 Widget.

- A ImageGallery2 widget is available only for a Flex Form.
- The Image Gallery occupies 100% of the screen width.
- On devices which have a navigation key, you can use the up or down keys to navigate through the images.

### 5.25.2 Look

This section details the **Properties > Look** tab of an ImageGallery2 widget.



To specify the look properties of an ImageGallery2 widget, refer section: [Look](#).

### 5.25.3 Skin

Skin defines properties such as background color, borders and shadows for a widget. They also, define the widget font (applicable only if text is displayed on a widget.)

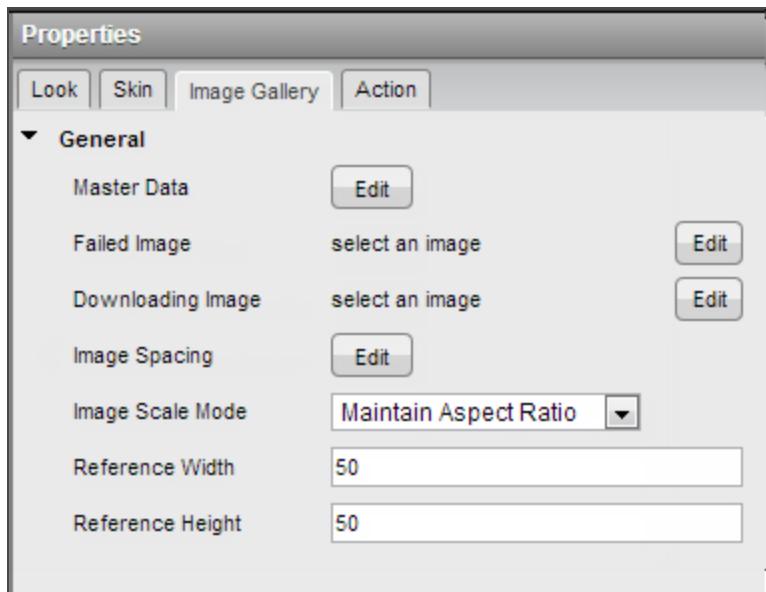
For the ImageGallery2 widget, you can apply a skin for the following states:

- Normal: Specifies the look and feel of the widget when not in focus.
- Focus: Specifies the skin that is applied when there is focus on a widget.

Refer to section [Working with Skins](#) to know more about adding a skin to a widget.

## 5.25.4 Platform Specific Properties

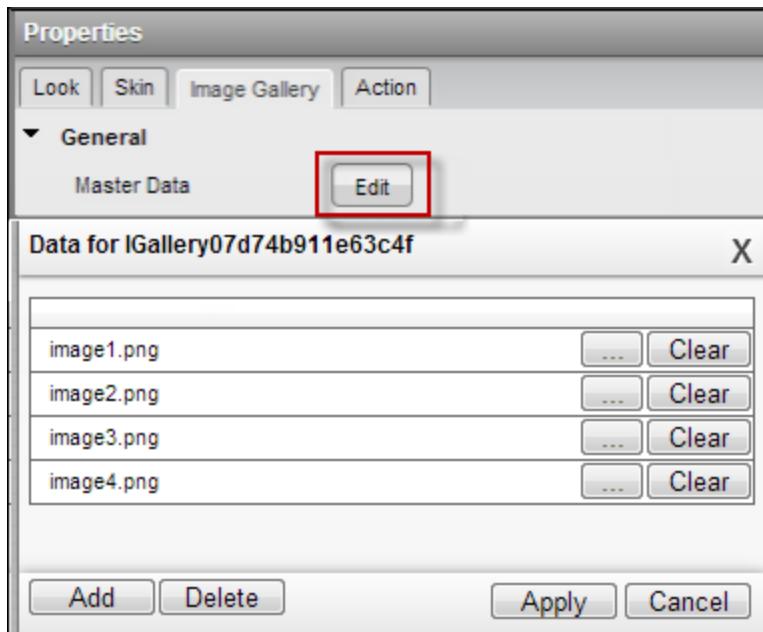
**Note:** In this section, the properties that can be forked are identified by an icon ( located to the left of the property. For more details, refer to section [Forking a Widget property](#).



### 5.25.4.1 Master Data

You can specify all the images that you wanted to place inside this widget.

For providing Master Data, click the **Edit** button against **Master Data** to open **Data for IGallery** dialog box.



Click ellipses (...) to open the **Select Image** dialog box. Select an image or provide an url for the image and click **OK**.

After adding the images in the masterData, click **Apply** to insert these images.

#### 5.25.4.2 Failed Image

Specifies the image to be displayed when the remote resource is unavailable.

For specifying an image, click the **Edit** button against the **Failed Image** field to open the **Failed Image** dialog box from where you can locate an image to be displayed when the remote resource is unavailable.

#### 5.25.4.3 Downloading Image

Specifies the image to be displayed when the remote source is still being downloaded.

For specifying an image, click the **Edit** button against the **Downloading Image** field to open the **Downloading Image** dialog box from where you can locate an image to be displayed when the remote resource is still being downloaded.

#### 5.25.4.4 Images Spacing

Specifies the space between the images in the ImageGallery.

For provide this value, click the **Edit** button against the **Image Spacing** field to open **Image Space** dialog box.

You can change the default value by typing a new number inside the **Value** field corresponding to **Default** platform.

For providing the value specific to a platform, select the platform and type a new number in the corresponding **Value** field.

#### 5.25.4.5 Image Scale Mode

A value of this property specifies how the rendered image's width and height are identified if those of the source image varies from the Image widget itself. Image Widget represents the underlying native widget which renders (and applies alignment to) the Source Image.

Following are the available options:

- Fit to Dimensions: The source image is resized to fill the ImageWidget dimensions.
- Maintain Aspect Ratio: This mode resizes the source image to fit in the ImageWidget dimensions while it preserves its native aspect ratio. In case,
- Crop: This scale mode preserves the original size of the Source Image.

#### 5.25.4.6 Reference Width

It is the reference image width in pixels. These are device independent Pixels specified against 163 dpi.

#### 5.25.4.7 Reference Height

It is the reference image height in pixels. These are device independent Pixels specified against 163 dpi.

## 5.25.5 Actions

ImageGallery widget has the following action associated with it:

- onSelection: This action is invoked by the platform when an Image is selected in ImageGallery.

For more information on using the above action, refer to section [Action Editor](#).

## 5.25.6 Placement Inside a Widget

A ImageGallery2 can be directly added inside the following widgets:

| Widget    | Image Gallery placement inside a widget                  |
|-----------|--|
| Form      | Yes  |
| HBox      | Yes  |
| VBox      | Yes  |
| ScrollBox | Horizontal Orientation -Yes<br>Vertical Orientation- Yes |
| Tab       | Yes  |
| Segment   | No   |
| Popup     | Yes  |
| Template  | Header- No<br>Footer- No                                 |

## 5.26 Map

A Map widget provides you the capability to display pre-defined locations (latitude and longitude) on an onscreen map. Platforms like iPhone (above 3.0) and Android provide a native map widget that can be displayed as part of the application.

On platforms where a native map widget is not available, Map widget integrates with Google Maps and displays the static image with zoom and pan controls. You can customize the map view if you do not want to use the default view.

**Note:**

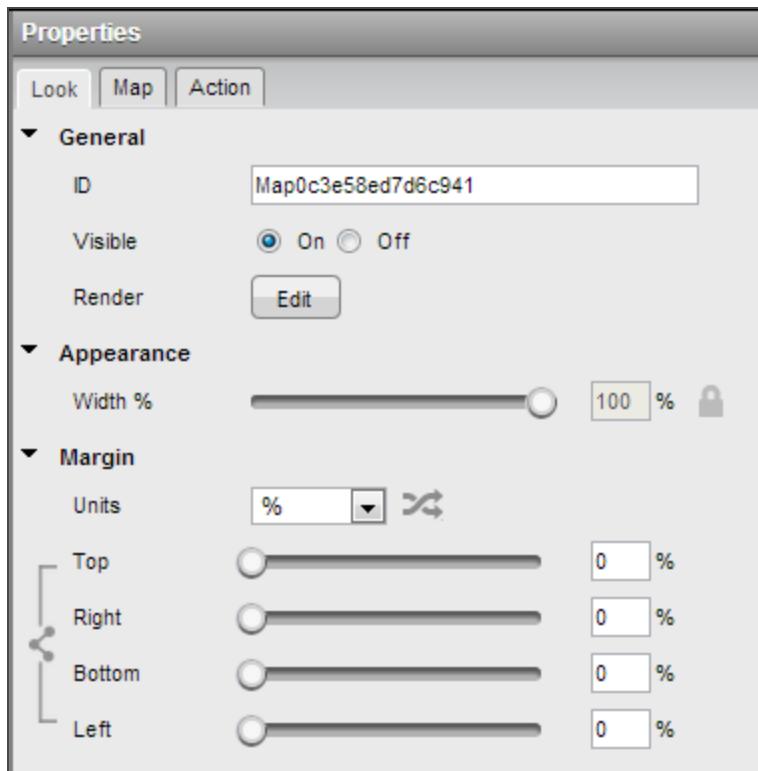
1. The platforms Mobile Web (basic) and Non-Touch HTML devices supports only static maps.
2. On Android platform, Map widget does not work in popup.

The Map widget renders a map using the mapping service provided by the platform. The following table shows the list of platforms and the available mapping services:

| Platform              | Mapping Service   |
|-----------------------|---|
| Android               | Google Maps   |
| iPhone                | Google Maps   |
| Mobile Web (advanced) | Google Static Maps, Native maps of the device, and interactive maps (Java script) |

### 5.26.1 Look

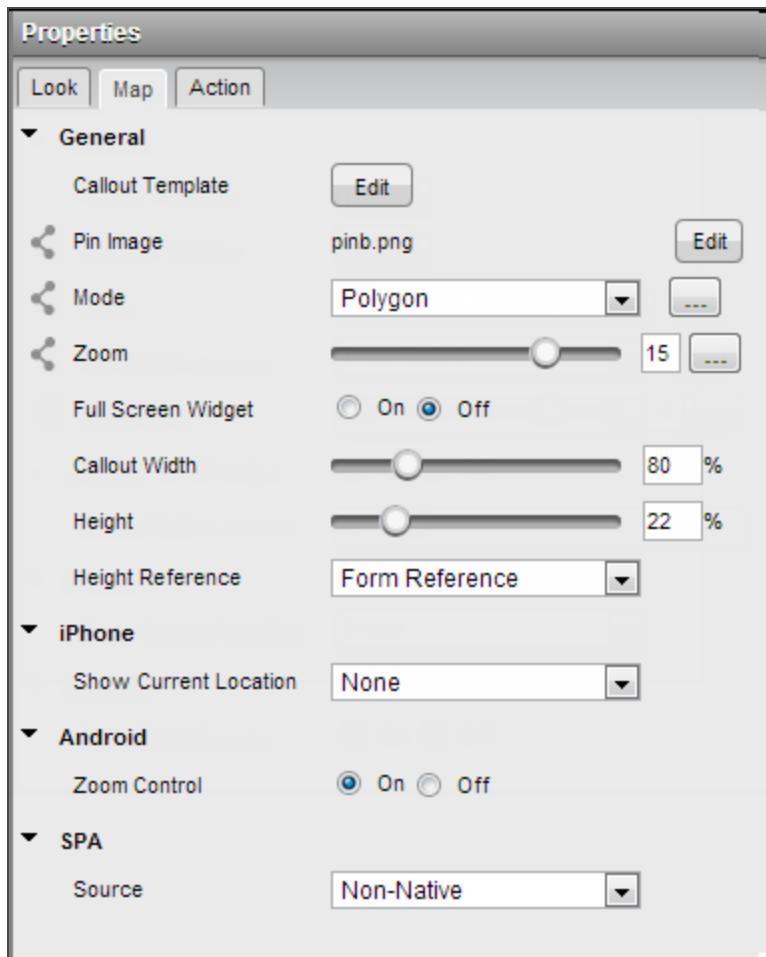
This section details the **Properties > Look** tab of a **Map** widget.



To specify the look properties of a Map widget, refer section: [Look](#).

### 5.26.2 Platform Specific Properties

**Note:** In this section, the properties that can be forked are identified by an icon ( ) located to the left of the property. For more details, refer to section [Forking a Widget property](#).



### 5.26.2.1 Callout Template

Accepts reference to a box widget which represents a UI template for a custom callout. The box template is allowed to have only Label, Link, RichText, Button and Image widgets.

**Note:** If template is not provided, it will fallback to the platform specific default callout for backward compatibility.

You can create a callout template by referring to section: [Templates:Maps](#)

For inserting a template, click the **Edit** against the **Callout Template** field to open the **Callout Templates** dialog box.

From the available templates, choose a template to be used for callouts.

### 5.26.2.2 Pin Image

The default map pin image to be used to indicate a location on map.

For proving an image, click the **Edit** button against the **Pin Image** field to open the **Pin Image** dialog box.

A default image is provide by the system. To replace this image, click inside the **Value** field corresponding to the **Default** platform to open the **Select Image** dialog box. Select a new image and then click **OK**.

To provide a platform specific image, select a platform and click inside the corresponding **Value** field to open the **Select Image** dialog box. Select a new image and then click **Ok**.

### 5.26.2.3 Mode

Specifies the map viewing mode.

- Normal: Traditional depiction of roads, parks, borders etc.
- Satellite: Map showing aerial imagery.
- Street: Navigate within street-level imagery.
- Hybrid: Street map superimposed on Satellite map.
- Terrain: Map showing the surface of the land in 3D view.
- Polygon: Map showing the polygonal area as specified in locationdata property.
- Traffic: Specifies the streets with different colors to indicate traffic information on the map. The color green indicates low traffic, orange indicates medium traffic and red indicates heavy traffic.

From the **Mode** list you can select a default mode that is applied for all the platforms. Alternatively, to provide a platform specific mode, fork the **Mode** property. Refer to the section [Forking a Widget Property](#) for more details.

#### 5.26.2.4 Source

Specifies the source of the maps.

You can choose one of the following available sources:

- Native: If you select this option, the application uses the mapKey and provider details to fetch the map. The fetched map is interactive with an ability to zoom and pan.

**Note:** Polygon view on advanced Mobile Web platform is available only when the source is set to non-native.

- Non Native: If you select this option, the application uses the map that is on the device.

**Note:** SPA platforms support only Google Static Maps as a source. Static maps are directly requested from Google for a given latitude and longitude. Kony does not support any other option because the size of the get request URL can be bigger than 256 characters leading to the request not being served.

From the **Map Source** list you can select a default map source that is applied for all the platforms. Alternatively, to provide a platform specific map source, fork the **Map Source** property. Refer to the section [Forking a Widget Property](#) for more details.

#### 5.26.2.5 Zoom Level

Sets the zoom level for the current map view. The range varies from platform to platform.

You can specify a zoom level by clicking the **Edit** button against the **Zoom Level** to open the **Zoom Level** dialog box.

Select the platform and from its corresponding **value** list, select a desired view type.

#### 5.26.2.6 Full Screen Widget

Specifies whether the widget should occupy the whole container excluding space for headers and footers, if any.

- If **On** is selected, the map occupies the space as set in the IDE or layout properties.
- If **Off** is selected, the map occupies the whole space on the container.

#### 5.26.2.7 Callout Width

Specifies the width of the callout on the map. It accepts a number between 1 to 100 in percentage relative to the map widget width. For example, 100% value means, callout width should fill its map widget width. If the value specified is less than 1 or more than 100, it should fallback to 80%.

#### 5.26.2.8 Height

Specifies the height of the Map in terms of percentage. The percentage is with reference to the value of [Height Reference](#) property.

**Note:** This property is unavailable on Flex Forms.

#### 5.26.2.9 Height Reference

The Map height percentage is calculated based on the option selected.

- Form Reference: The Map height percentage is calculated based on the height of the form excluding headers and footers. This option is not respected if Map is placed inside a popup or in templates.
- Parent Width: This option is used if the Map is placed inside a popup or in templates. The width is calculated based on the width of the parent container.

**Note:** This property is unavailable on Flex Forms.

#### 5.26.2.10 Show Current Location

Indicates the current location on map as a pin, circle or none.

Following are the available options:

- None
- Pin
- Circle

**Note:** This property is specific to the iOS platform.

#### 5.26.2.11 Zoom Control

Indicates if the zoom control is to be displayed on the map.

- If **On** is selected, the zoom control is displayed.
- If **Off** is selected, the zoom control is not displayed.

**Note:** This property is specific to the Android platform.

#### 5.26.3 Actions

Map widget has the following actions associated with it:

- **onPinClick:** This action is invoked by the platform when a map pin is clicked, passing the selected *locationdata* to the callback.
- **onSelection:** This action is invoked by the platform when the user clicks on a callout of the Map.

For more information on using the above actions, refer to section [Action Editor](#).

#### 5.26.4 Placement inside a Widget

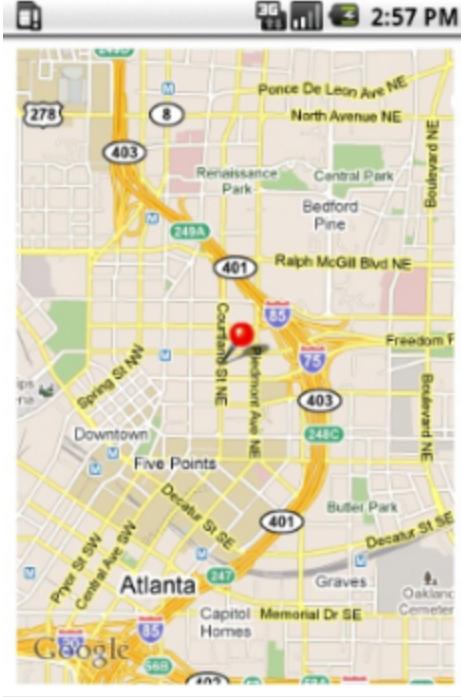
A map can be directly added inside the following widgets:

| Widget | Map placement inside a widget |
|--------|-------------------------------|
| Form   | Yes                           |

| Widget    | Map placement inside a widget                            |
|-----------|--|
| HBox      | Yes  |
| VBox      | Yes  |
| ScrollBar | Horizontal Orientation -Yes<br>Vertical Orientation- Yes |
| Tab       | Yes  |
| Segment   | No   |
| Popup     | Yes  |
| Template  | Header- No<br>Footer- No                                 |

### 5.26.5 Widget Appearance on Platforms

The appearance of the Map widget on various platforms is as follows:

| Platform | Appearance  |
|----------|---|
| Android  |   |
| iOS      |  |

| Platform | Appearance   |
|----------|--|
| SPA      |  |

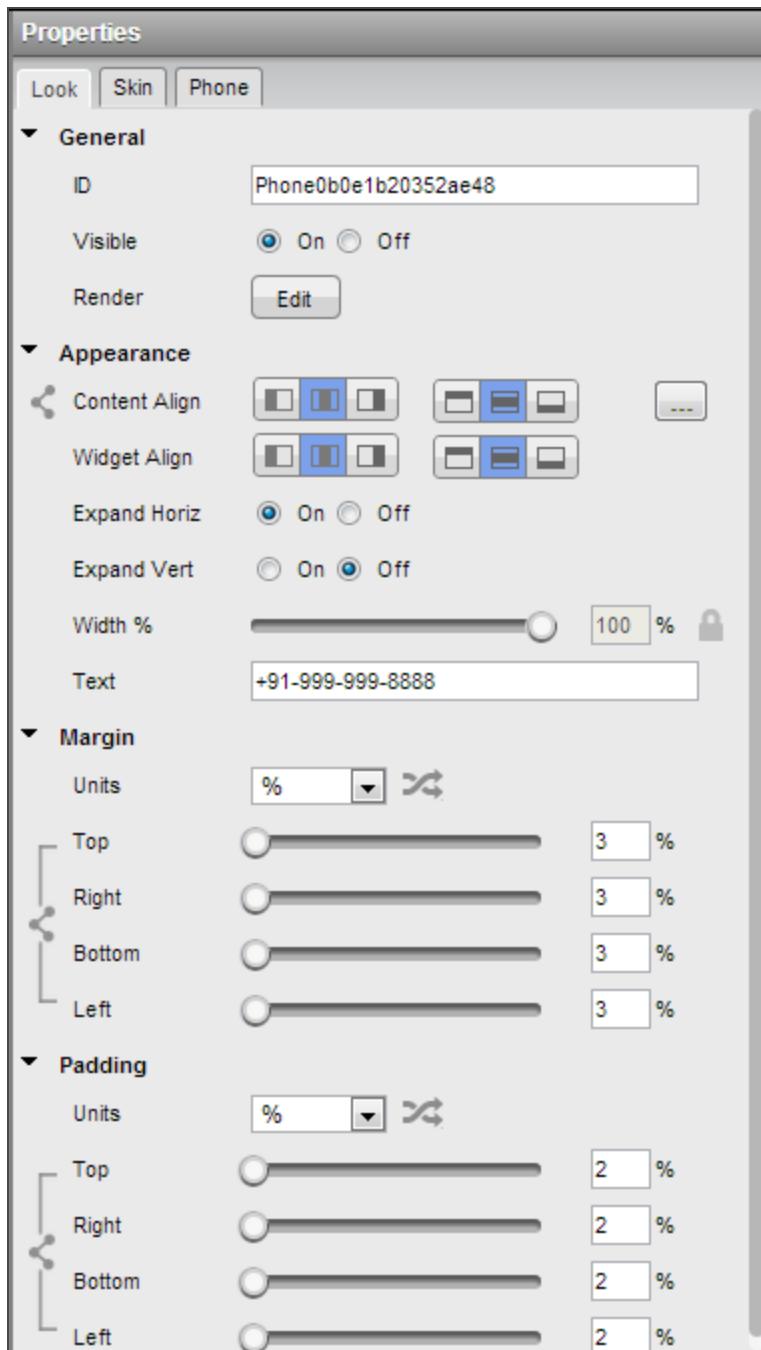
## 5.27 Phone

A Phone widget, when placed in an application, allows you to launch the native phone dialer and initiate a phone call to the number that is displayed on it. It appears as a button on the Form and the phone number is displayed on it either in the number format or the phone spell text. When the user selects the phone widget, the native dialer is launched to make a phone call.

**Note:** Phone widget is not applicable for Desktop Web.

### 5.27.1 Look

This section details the **Properties > Look** tab of a Phone widget.



To specify the look properties of a Phone widget, refer section: [Look](#).

## 5.27.2 Skin

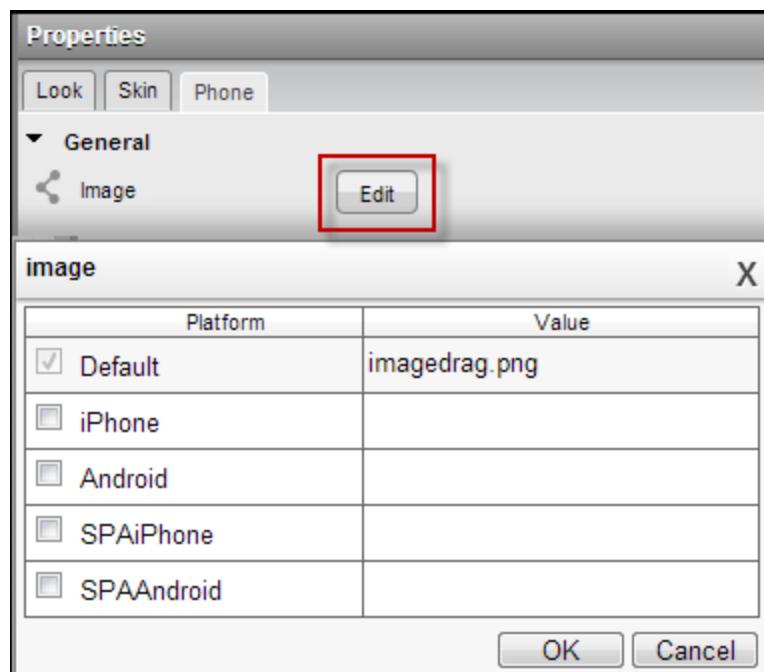
Skin defines properties such as background color, borders and shadows for a widget. They also, define the widget font (applicable only if text is displayed on a widget.)

For the Phone widget, you can apply a skin for the following states:

- Normal: Specifies the look and feel of the widget when not in focus.
- Focus: Specifies the skin that is applied when there is focus on a widget.

Refer to section [Working with Skins](#) to know more about adding a skin to a widget.

## 5.27.3 Platform Specific Properties



### 5.27.3.1 image

Specifies an Image for the Phone widget.

To specify an image, click the **Edit** button against the **Image**. This results in displaying **Image** dialog box from where you can specify a default image or platform specific image.

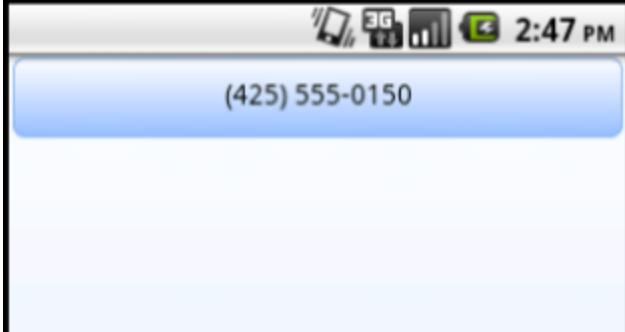
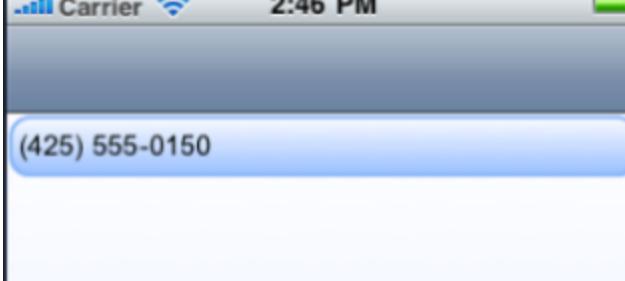
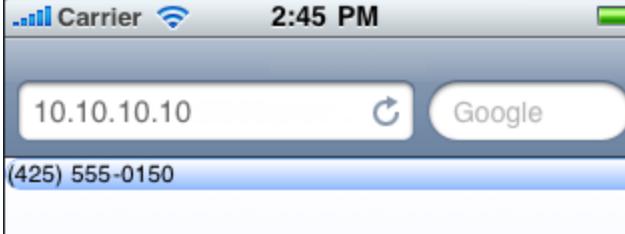
#### 5.27.4 Placement Inside a Widget

A phone can be directly added inside the following widgets:

| Widget    | Phone placement inside a widget                          |
|-----------|--|
| Form      | Yes  |
| HBox      | Yes  |
| VBox      | Yes  |
| ScrollBar | Horizontal Orientation -Yes<br>Vertical Orientation- Yes |
| Tab       | Yes  |
| Segment   | Yes  |
| Popup     | Yes  |
| Template  | Header- No<br>Footer- No                                 |

#### 5.27.5 Widget Appearance on Platforms

The appearance of the Phone widget on various platforms is as follows:

| Platform | Appearance  |
|----------|---|
| Android  |    |
| iOS      |   |
| SPA      |  |

## 5.28 PickerView

A PickerView widget uses a spinning wheel metaphor to display multiple sets of values and allows you to select a single combination of values. You can select a single combination of values by rotating the wheels and aligning the desired row of values with the selection indicator.

PickerView can have multiple components and each component comprises of keys. Users can choose the keys from different components and make the choices . useful in grouping the multiple choices that user can make in different categories related to concept. For example: color, model, year of manufacturing all these three can be modeled as components with different possible values so that user can make his choice using this single widget.

**Note:** PickerView widget is not supported on SPA platforms.

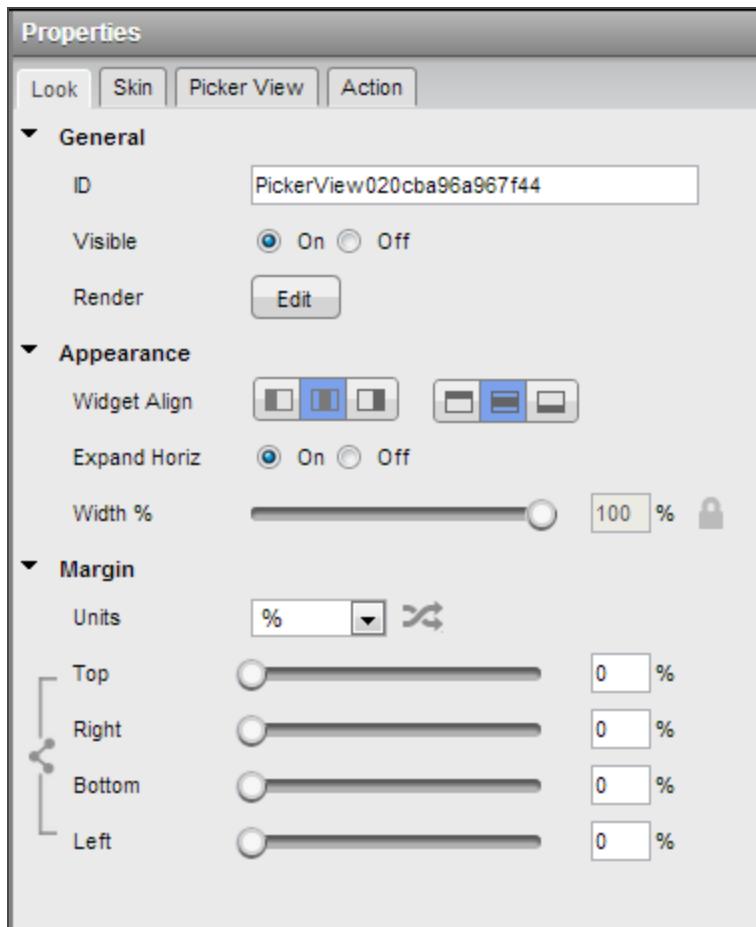
### 5.28.1 Important Considerations

The following are important considerations for a PickerView Widget.

- PickerView is used to make multiple selections and arrive at a combination of the selected values. For example, for Flight Bookings, if you want the user to select one of the available dates, time, and flight number; you can create lists for date, time, and flight numbers and display them in a PickerView widget.

### 5.28.2 Look

This section details the Properties > Look tab of a PickerView widget.



To specify the look properties of a PickerView widget, refer section: [Look](#).

### 5.28.3 Skin

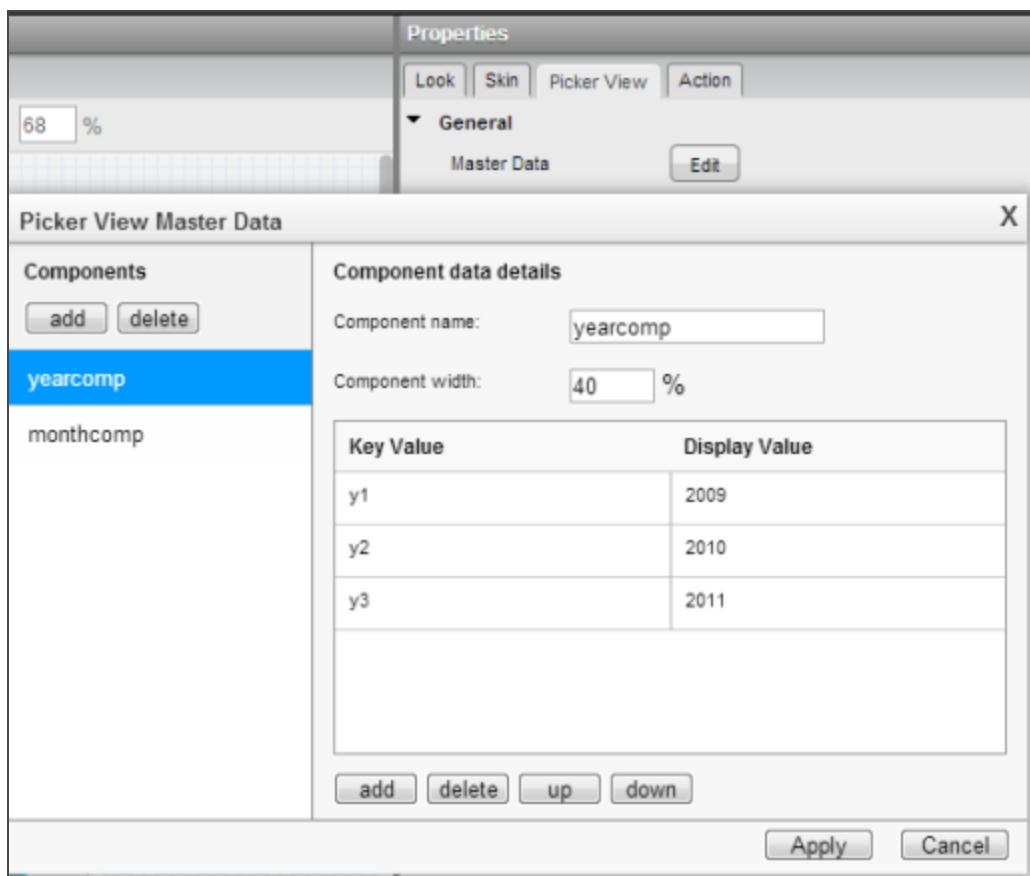
Skin defines properties such as background color, borders and shadows for a widget. They also, define the widget font (applicable only if text is displayed on a widget.)

For the PickerView widget, you can apply a skin for the following states:

- Skin: Specifies the look and feel of the widget when not in focus.
- Focus: Specifies the skin that is applied when there is focus on a widget.

Refer to section [Working with Skins](#) to know more about adding a skin to a widget.

## 5.28.4 Platform Specific Properties



### 5.28.4.1 Master Data

Specifies the set of values that must be displayed for the user to make a selection from the available choices.

To specify the set of values, click the **Edit** button against the **Master Data** field to open the **PickerView Master Data** dialog box.

In the Master Data window, you can specify the Key and Display Value. You can add the components by clicking the **add** button.

**Note:** Key Value and Display Value combined width should be 100%

### 5.28.5 Actions

PickerView widget has the following action associated with it:

- onSelection: This action is invoked by the platform when the component selection changes.

For more information on using the above action, refer to section [Action Editor](#).

### 5.28.6 Placement Inside a Widget

The appearance of the PickerView widget on various platforms is as follows:

| Widget     | PickerView placement inside a widget                      |
|------------|---|
| Form       | Yes   |
| HBox       | Yes   |
| VBox       | Yes   |
| Scroll Box | Horizontal Orientation - Yes<br>Vertical Orientation- Yes |
| Tab        | Yes   |
| Segment    | No  |
| Popup      | Yes   |
| Template   | Header- No<br>Footer- No                                  |

### 5.28.7 Widget Appearance on Platforms

The appearance of the PickerView widget on various platforms is as follows

| Platform | Appearance   |
|----------|--|
| Android  |  A screenshot of a PickerView Widget on an Android device. The screen shows a title "PickerView Widget" at the top. Below it is a 4x3 grid of data. The first column contains numbers 24, 25, 26, and 27. The second column contains times 12:00, 13:00, 14:00, and 15:00. The third column contains categorical names: Business, Economy, Frequent, and Discounted. The entire grid has a dark grey background.  |
| iOS      |  A screenshot of a PickerView Widget on an iOS device. The screen shows a title "PickerView Widget" above a navigation bar with a "Back" button. Below is a 4x3 grid of data. The first column contains numbers 24, 25, 26, and 27. The second column contains times 12:..., 13:..., 14:..., and 15:.... The third column contains categorical names: Business, Economy, Frequent..., and Discounte... (with ellipses). The grid has a light blue gradient background. |

## 5.29 Segment2

A Segment2 consists of multiple segments (rows or records) and each segment (row or record) can have multiple child widgets.

### 5.29.1 Important Considerations

The following are important considerations for an Segment2 Widget:

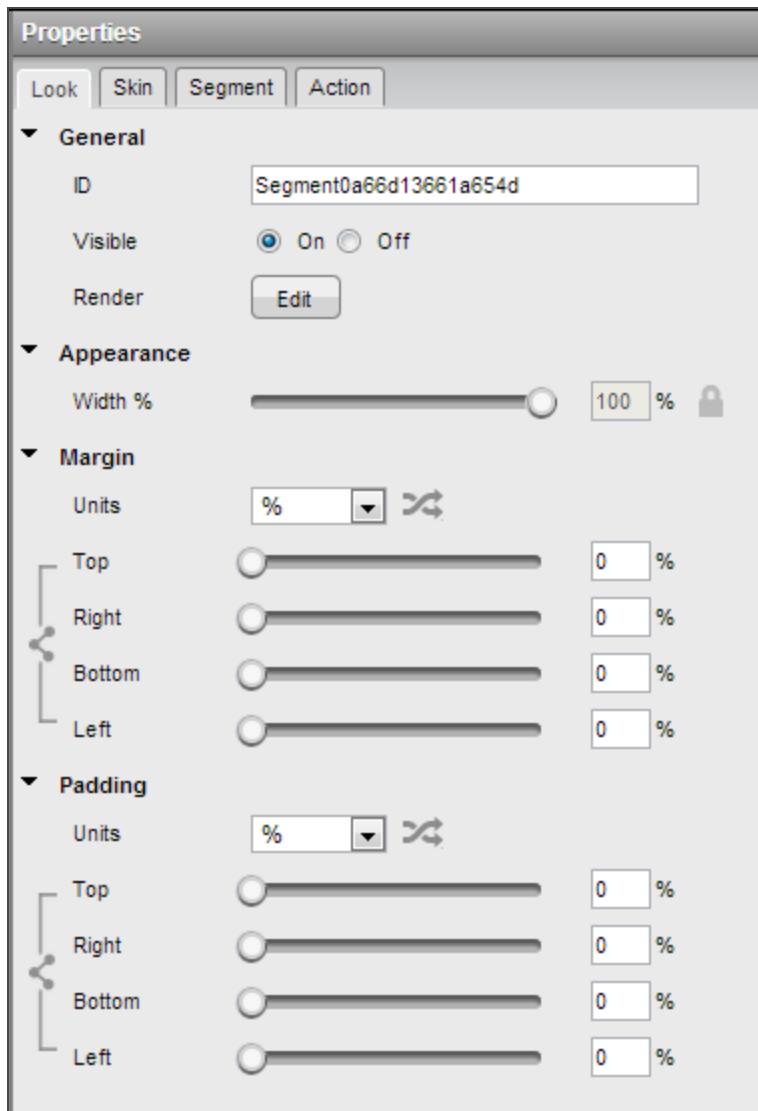
- Segment2 occupies memory from two perspectives:
  - Amount of data required from number of rows. For example, if you set data for 100 rows, memory for all the 100 records will be in memory.
  - View hierarchy (Box and other supported widgets) in each segment row. If the View hierarchy is complex, the memory usage is high.

**Note:** On iPhone and Android, and Windows platforms, if your segment has large data sets (more than 20 records with each record having more than 15 widgets), set the segment as a Screen Level Widget.

- You cannot add any elements to the widgets dynamically. But, you can hide any elements, if you do not provide any data for that element.
- You can dynamically change the skin of the widgets in the segment.
- A Segment2 can be placed in a ScrollBox only if the orientation of the ScrollBox is vertical.
- The height of the Segment2 is determined by the content of the widget. If you set the Screen Level Widget as true, then the height of the Segment2 widget is the form height excluding headers and footers.

### 5.29.2 Look

This section details the **Properties > Look** tab of a Segment2 widget.



To specify the look properties of an Segment2 widget, refer section: [Look](#).

### 5.29.3 Skin

Skin defines properties such as background color, borders and shadows for a widget. They also, define the widget font (applicable only if text is displayed on a widget.)

For the Segment2 widget, you can apply a skin for the following states:

- Row: Specifies the skin that is applied for each row.
- Row - Focus: Specifies the skin that is applied when user focuses on a row.
- Section Header: Specifies the skin that is applied to the Section Header of Segment2 widget.
- Widget: Specifies the skin that is applied to the entire Segment2 .
- Row - Alternate: Specifies the skin that is applied to every alternate *even numbered* row in the segment.
- Blocked UI: Specifies the skin that is to block the interface until the action in progress (for example, a service call) is completed.

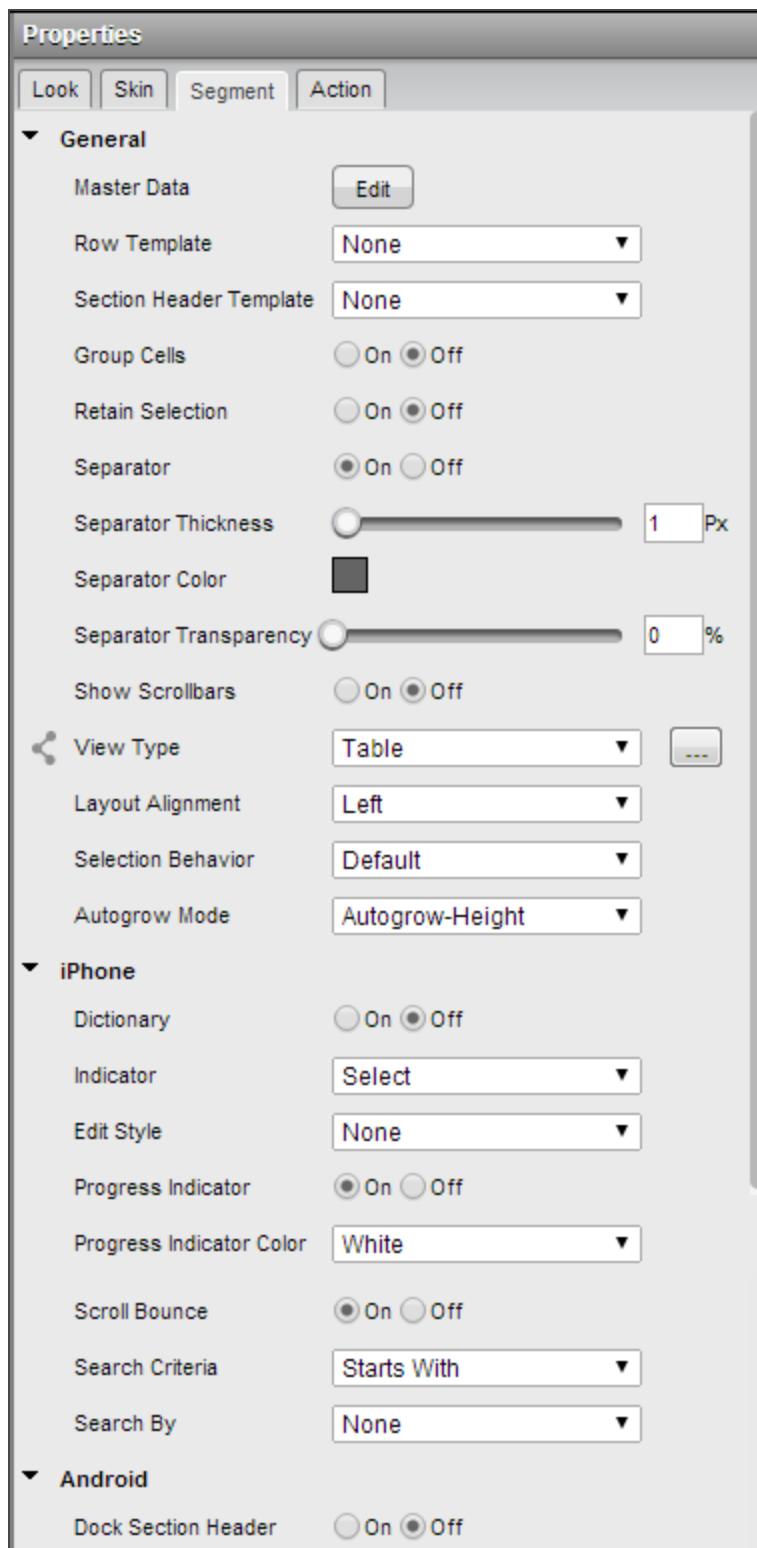
**Note:** Blocked UI is available only for SPA platforms.

- Pressed: Specifies the skin to indicate that the row of the segment is pressed or clicked.
- Refresh - Pull: Specifies the look and feel of a widget when the scroll bar is pulled.
- Refresh - Push: Specifies the look and feel of a widget when the scroll bar is pushed.

Refer to section [Working with Skins](#) to know more about adding a skin to a widget.

#### 5.29.4 Platform Specific Properties

**Note:** In this section, the properties that can be forked are identified by an icon ( ) located to the left of the property. For more details, refer to section [Forking a Widget property](#).

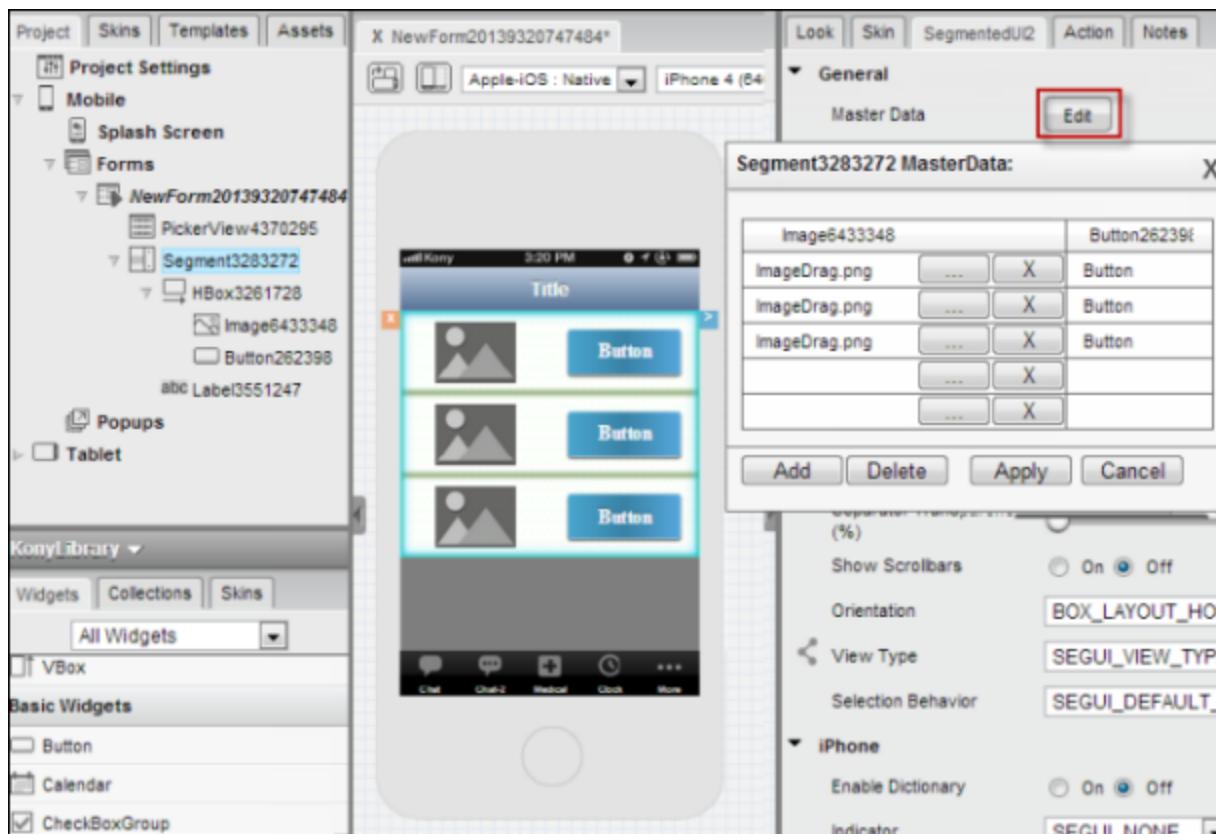


#### 5.29.4.1 Master Data

The master data is enabled only when the widgets are added to the **Segment2** widget.

The below example illustrates the Master Data feature.

In this example , we add an Image and a Button to the Segment2 widget. Following are the steps:



1. Drag and drop an HBox container widget inside the **Segment2** widget.
2. Within this HBox container, drag and drop Image and Button widget.
3. From the **Segment2** widget properties, click **Segment** tab.
4. Click the **Edit** button against the Mater Data to open the **Master Data** dialog box.

5. Provide the image and button widget details as shown in the above image.

6. Click OK.

#### 5.29.4.2 Row Template

Indicates the common template to be used for each row while creating the row and filling the data. This can be overridden at the row level when setting the data using the template key.

**Note:** Only those templates that are created from Project Explorer >Templates > Segements are visible under Row Template drop-down.

#### 5.29.4.3 Section Header Template

Specifies the common template to be used for each section while creating the section header and filling the data. This is optional parameter and if not provided the default template provided by each platform will be used. It can also be provided at each section level when setting the data.

**Note:** Only those templates that are created from Project Explorer >Templates > Segements are visible under Row Header Template drop-down.

**Note:** When a Section Header is provided along with the rows/items, the Section Header is "clamped" to the top of the scrollable area (on the Form) as one scrolls through a long list of items (for example, if you have a long list of contacts that all begin with the letter "A", the "A" header will be fixed at the top until you scroll down past the last "A" item). This behavior can be clearly seen iPhone's Contacts application.

This behavior of Section Headers is available on iOS and Android platform and is enabled when the Screen Level Widget has been set to true.

#### 5.29.4.4 Group Cells

Specifies if all the rows in a segment should be grouped using a rounded corner background and border.

- If **On** is selected, the cells will not have rounded border.
- If **Off** is selected, the cells will have a rounded border.

#### 5.29.4.5 Retain Selection

Specifies if the segment should retain the selection made when the user navigates out of the form and revisits the form.

- If **On** is selected, the selection is retained when the user navigates to different form.
- If **Off** is selected, the selection is not retained.

#### 5.29.4.6 Full Screen Widget

Specifies whether the widget should occupy the whole container or not. You must set the value to true if your segment has large data sets (more than 20 records with each record having more than 15 widgets) to facilitate a better reuse of the widgets and a different scrolling behavior.

If **On** is selected, the widget occupies the whole container and there is a reduction in load time of the Segment2 as only few rows are loaded at the load time. The rest of the rows are loaded as the user scrolls through the widget. But the scrolling speed reduces.

If **Off** is selected, the widget does not occupy the whole container and load time of Segment2 increases because all the rows are loaded at the beginning. But the scrolling speed improves.

#### 5.29.4.7 Separator

Specifies if the segment should display the separator between the rows.

- If **On** is selected, the separator is displayed.
- If **Off** is selected, the separator is not displayed.

#### 5.29.4.8 Separator Thickness

Specifies the thickness of the separator in pixels.

#### 5.29.4.9 Separator Color

Specifies the color of the separator between rows of Segment2. Click the color sampler to open the color picker from where you can select a separator color.

#### 5.29.4.10 Separator Transparency

Provide the desired transparency for the separator.

#### 5.29.4.11 Show Scrollbars

Specifies if the scrollbars of the segment must be visible all the time.

- If **On** is selected, the scrollbars are displayed.
- If **Off** is selected, the scrollbars are not displayed.

#### 5.29.4.12 Orientation

Specifies how you can stack the widgets within the Segment2. You can set the orientation of the Segment2 as horizontal or vertical.

Following are the available options:

- Horizontal: Enables you to stack the content within the Segment2 horizontally.
- Vertical: Enables you to stack the content within the Segment2 vertically.

#### 5.29.4.13 View Type

You can use this property to select the view type of a segment. The following are the available view types that you can select and their appearances on iPhone native client:

- Table: The rows of the segment appear in a table as a list.



- Page: The rows of the segment appear in pages and you need to scroll through the pages to view the rows.



#### 5.29.4.14 Layout Alignment

This specifies the direction in which the widgets are laid out.

**Default:** LEFT

The options are:

- Left: The widgets placed inside a Segment2 are aligned left.
- Center: The widgets placed inside a Segment2 are aligned center.
- Right: The widgets placed inside a Segment2 are aligned right.

#### 5.29.4.15 Selection Behavior

Specifies if the segment will support single or multiple selection.

Following are the available options:

- Default: Indicates that the segment does not support either single or multiple selection. This option allows you to define an onRowClick event for the segment.
- Single Select: Indicates that you can make one selection when you have many choices in the segment (the behavior is similar to a RadioButtonGroup).
- Multi Select: Indicates that you can make more than one selection when you have many choices in the segment (the behavior is similar to a CheckBoxGroup).

#### 5.29.4.16 Autogrow Mode

This property is applicable only when the segment is placed inside a flex container and View Type is set as Table. It specifies the segment to grow when the new content is added.

Following are the options:

- None: Auto growth of a Segment is disabled.
- Autogrow-Height: Auto growth of a Segment is enabled.

#### Rules and Priorities of Autogrow-Mode property

- If the height of the Segment2 is not computable and **Autogrow Mode** property is configured as **Autogrow-Height**, then the height of the Segment2 will be the Preferred Height, and min and max constraints are applied on top of the Preferred Height computed.

**Note:** Preferred Height in the above statement refers to the cumulative height of the segment contents (rows - defined using templates, section headers / footers, separators etc.).

- If the height of the Segment2 is not computable and **Autogrow Mode** property is configured as **None**, then the height of the Segment2 will be the default value and min/max constraints are applied on top of the default value.

- The Autogrow Mode property gets preference, when the height of the Segment2 is specified as preferred and Autogrow Mode property is configured Autogrow-Height.
- If the Autogrow Mode property is not specified or the specified value is invalid, then the default value of the Autogrow Mode property is equal to specifying **None**.
- If Autogrow Mode property is specified, and the templates defined are specified in percentage, then the height of the row, which uses percentage template will be considered as the default row height of the template, which is 100dp.
- All the above rules are applied even if the template specified is a box template.
- When a Segment2 is generated using single bucket constructor, height property must be set as undefined for Autogrow Mode property to work.
- If the row template height is not specified and the Segment2 Autogrow Mode property is set as Autogrow-Height, then row height of the Segment2 is 220dp (preferred height of a flex container).

#### 5.29.4.17 Height

Specifies the height of the Segment2 in terms of percentage. The percentage is with reference to the value of [Height Reference](#) property.

**Note:** This property is unavailable on Flex Forms.

#### 5.29.4.18 Height Reference

The Segment2 height percentage is calculated based on the option selected.

- Form Reference: The Segment2 height percentage is calculated based on the height of the form excluding headers and footers. This option is not respected if Segment2 is placed inside a popup or in templates.
- Parent Width: This option is used if the Segment2 is placed inside a popup or in templates. The

width is calculated based on the width of the parent container.

**Note:** This property is unavailable on Flex Forms.

#### 5.29.4.19 Dictionary

Specifies if dictionary must be enabled for easy navigation.

If the dictionary property is enabled, alphabets from A to Z appear on the screen and when you select any alphabet, all the corresponding results that start with the selected alphabet are displayed.

**Note:** This property is applicable if Screen Level Widget property is set to true and the section headers have been set.

- If **On** is selected, the dictionary is available.
- If **Off** is selected, the dictionary is not available.

**Note:** This property is specific to the iOS platform.

#### 5.29.4.20 Indicator

Specifies the indicator type as rowSelect, rowClick, or none. Based on your selection, the behavior is exhibited:

Following are the available options:

If the user selects the indicator, the related content appears in the next screen .

- Click: Specifies the disclosure button. The button appears as follows:



If the user selects the disclosure button, the detailed content appears.

- None: No indicator or button is displayed.

**Note:** This property is specific to the iOS platform.

#### 5.29.4.21 Edit Style

Specifies the way in which the edit feature of Segment2 can be enabled.

Following are the available options:

- Icon: An icon will be displayed on the left hand side of each row.
- Swipe: A delete or insert button will be shown on the right hand side of each row when the user performs a SWIPE gesture on the row. Whether an insert button or delete button is to be shown is controlled by the editmode property that is set using the data property of the Segment2.
- None: No special edit styles are applied.

**Note:** This property is specific to the iOS platform.

#### 5.29.4.22 Progress Indicator

Specifies if the progress indicator must be displayed.

Default: true (the progress indicator is displayed on the widget)

- If **On** is selected, the progress indicator is displayed on the widget.
- If **Off** is selected, the progress indicator is not displayed on the widget.

**Note:** This property is specific to the iOS platform.

#### 5.29.4.23 Progress Indicator Color

Specifies the color of the progress indicator as white or grey.

- White: The progress indicator is white in color.
- Grey: The progress indicator is grey in color.

**Note:** This property is specific to the iOS platform.

#### 5.29.4.24 Scroll Bounces

Specifies whether the scroll view bounces past the edge of the content and back again.

- If **On** is selected, the scroll view bounce is applied.
- If **Off** is selected, the scroll view bounce is not applied.

**Note:** This property is specific to the iOS platform.

#### 5.29.4.25 Search Criteria

This specifies the search criteria to be applied when searching has been enabled.

**Note:** This property applies only when the [Full Screen Widget](#) property is enabled, [View Type](#) is set to **Table**, and a template is selected for [Search By](#) property.

The options are:

- Starts With : The search is performed on the strings that start with the input string.
- Ends With: The search is performed on the strings that end with the input string.
- Contains: The search is performed on the strings that contain the input string.

**Note:** This property is specific to the iOS platform.

#### 5.29.4.26 Search By

Indicates the identifier of the widget placed inside the row of the Segment2. Search will be performed against the content present inside the widget.

**Note:** Note: This property is applicable only when [Full Screen Widget](#) property is enabled, [View Type](#) is set to **Table**, and a template is selected for [Row Template](#) property.

**Note:** This property is specific to the iOS platform.

#### 5.29.4.27 Dock Section Header

The docking header property enables you to dock or place the section header at the top of the segment while scrolling the section content. If you are scrolling the segment data, the next section header will be docked on top of the segment

**Note:** This property applies only when the [Full Screen Widget](#) property is enabled and [View Type](#) is set to **Table**.

For example, if you scroll the segment data shown in the following figure, as the segment data scrolls up, the Samsung Phones docked header moves out of the Samsung header view, and is replaced with the HTC Phones section header, which is now docked.



**Note:** This property is specific to the Android platform.

## 5.29.5 Actions

Segment2 widget has the following actions associated with it:

- onRowClick: This action is triggered when the user click any row of the Segment2 .
- onPull: This action is called when Segment2 is pulled from top.
- onPush: This action is called when Segment2 is pushed from bottom.
- onReachingBeginning : This action is called when scrolling reaches the beginning of the Segment2 widget.
- onReachingEnd: This action is called when scrolling reaches the end of the Segment2 widget.
- onSwipe: This event is triggered when you swipe a row in a segment. This event is available only when the View Type is set to page view.
- onEditing: This action is triggered when a user indicates his desire to edit the row (delete or insert). This action is only triggered if the Edit Style is set to SEGUI\_EDITING\_STYLE\_ICON or SEGUI\_EDITING\_STYLE\_SWIPE.

**Note:** This action is applicable only on iOS platform.

For more information on using the above actions, refer to section [Action Editor](#).

## 5.29.6 Placement Inside a Widget

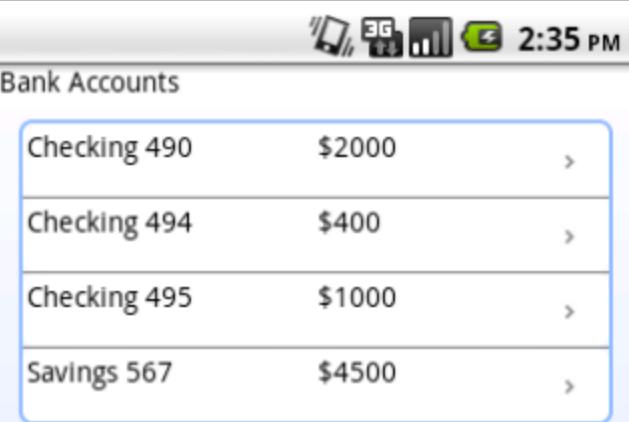
A Segment2 can be directly added inside the following widgets:

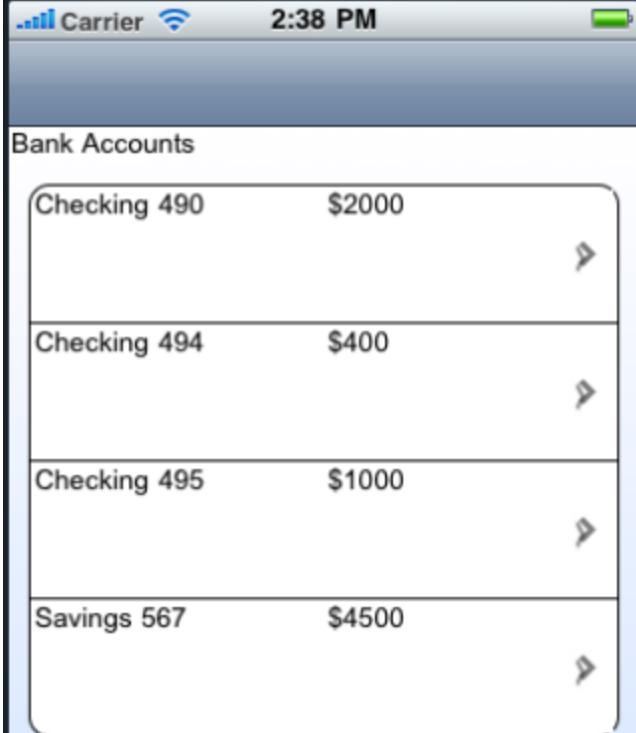
| Widget | Segment2 placement inside a widget |
|--------|------------------------------------|
| Form   | Yes                                |
| HBox   | No                                 |

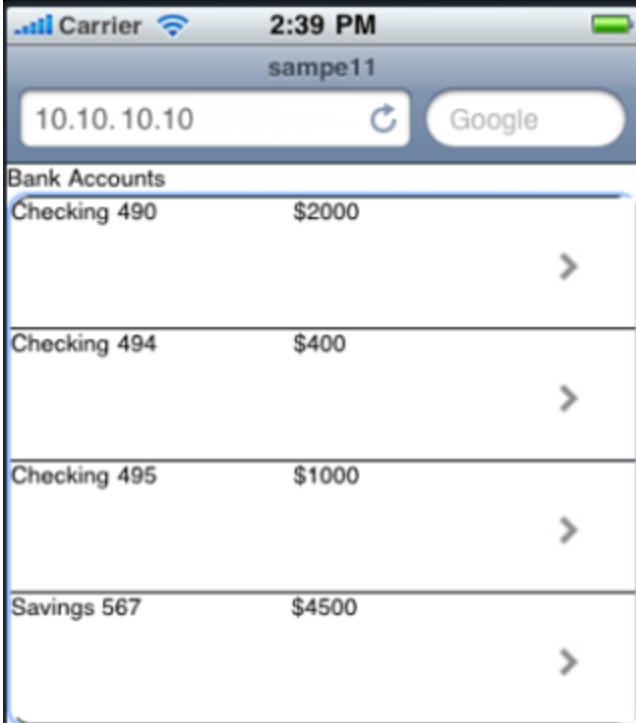
|           |  |
|-----------|--|
| Widget    | Segment2 placement inside a widget                       |
| VBox      | Yes  |
| ScrollBar | Horizontal Orientation - No<br>Vertical Orientation- Yes |
| Tab       | Yes  |
| Segment   | No   |
| Popup     | Yes  |
| Template  | Header- No<br>Footer- No                                 |

### 5.29.7 Widget Appearance on Platforms

The appearance of the Segment2 on various platforms is as follows:

| Platform | Appearance   |
|----------|--|
| Android  |  |

| Platform      | Appearance   |
|---------------|--|
| iOS           |   |
| Windows Phone |  |

| Platform | Appearance   |
|----------|--|
| SPA      |  <p>The screenshot shows a mobile application interface. At the top, there is a header bar with the text "Carrier" and "2:39 PM". Below the header, the screen displays a list titled "Bank Accounts". The list contains four items, each representing a bank account with its name and balance:</p> <ul style="list-style-type: none"><li>Checking 490 \$2000</li><li>Checking 494 \$400</li><li>Checking 495 \$1000</li><li>Savings 567 \$4500</li></ul> <p>Each list item has a right-pointing arrow icon to its right, indicating it is a link to another screen.</p> |

## 5.30 Switch

The Switch widget is identical to the Switch Control (on-off switch which is non customizable) in iPhone and presents two mutually exclusive choices or states.

The Switch widget displays the value that is currently in effect. You must slide the control to select (or reveal) the other value.

**Note:** The Switch widget is supported on iOS (Native and SPA) and Android (SPA) platforms.

### 5.30.1 Important Considerations

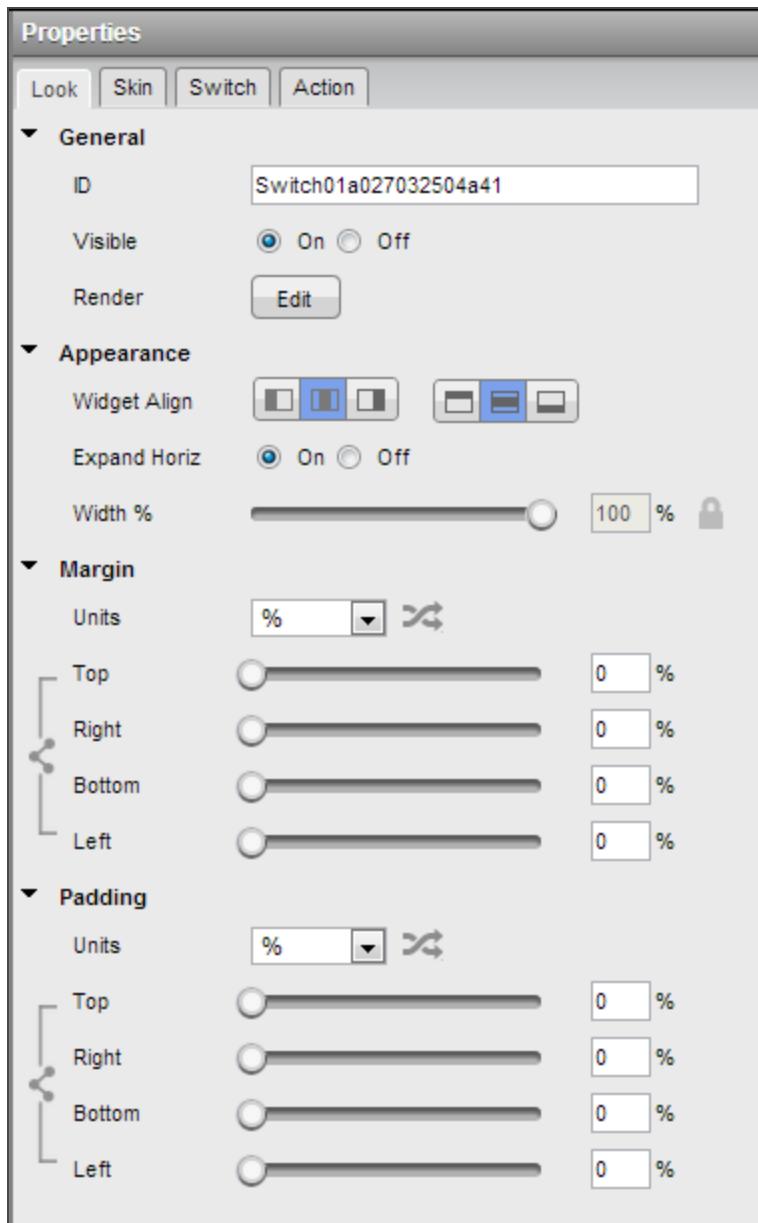
The following are important considerations for an Switch Widget:

- For a good user experience, use a predictable pair of values so that the users do not have to slide the Switch to know the other value.
- You can also consider using the Switch widget to change the state of other UI elements in the view. Depending upon the choice that the user makes, changes in the UI must take place.

For example, in an Airline application booking screen, based on the Switch widget selection as "One Way" or "Round Trip", the required UI elements must change.

### 5.30.2 Look

This section details the **Properties > Look** tab of a Switch widget.



To specify the look properties of an Switch widget, refer section: [Look](#).

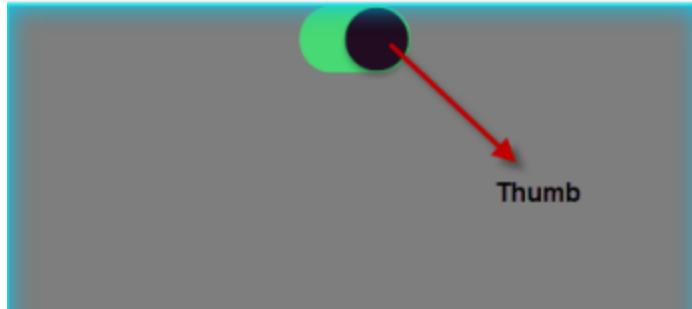
### 5.30.3 Skin

Skin defines properties such as background color, borders and shadows for a widget. They also, define the widget font (applicable only if text is displayed on a widget.)

For the Switch widget, you can apply skin for the following state:

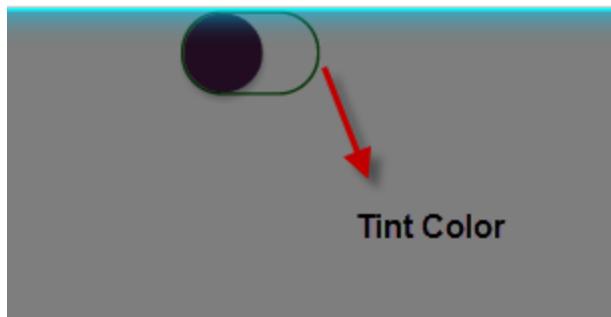
- Normal: Specifies the look and feel of the widget when not in focus.

Thumb Color: This property is available under BackGround only when you fork the skin for iPhone native. It specifies the color of the thumb.



**Note:** This property is specific to the iOS7 platform.

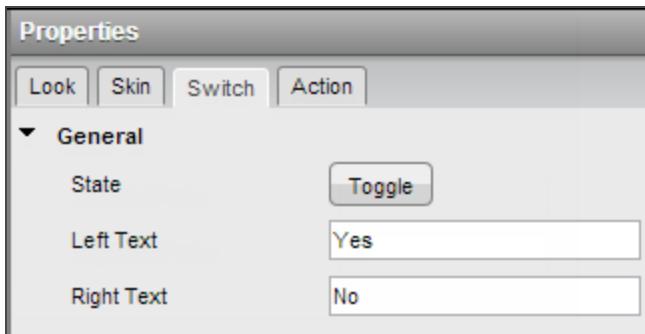
Tint Color: This property is available under BackGround only when you fork the skin for iPhone native.



**Note:** This property is specific to the iOS7 platform.

Refer to section [Working with the Skins](#) to know more about adding a skin to a widget.

## 5.30.4 Platform Specific Properties



### 5.30.4.1 State

Allows you to swap the colors.

### 5.30.4.2 Left Text

Specifies the text to be displayed on the left portion of the Switch.

### 5.30.4.3 Right Text

Specifies the text to be displayed on the right portion of the Switch.

## 5.30.5 Actions

Switch widget has the following action associated with it:

- **onSlide:** This action is invoked by the platform when there is a change in the default selected value.

For more information on using the above action, refer to section [Action Editor](#).

## 5.30.6 Placement Inside a Widget

A switch can be directly added inside the following widgets:

|           |   |
|-----------|---|
| Widget    | Switch placement inside a widget                          |
| Form      | Yes   |
| HBox      | Yes   |
| VBox      | Yes   |
| ScrollBox | Horizontal Orientation - yes<br>Vertical Orientation- Yes |
| Tab       | Yes   |
| Segment   | Yes   |
| Popup     | Yes   |
| Template  | Header- No<br>Footer- No                                  |

## 6. VBox Form

Form is a visual area (basic application screen) that holds other widgets. You can use a form to set a title and scroll content (similar to a web browser). The entire contents of the form except the headers and footers scroll together. A form is the top most container widget. A form can contain any number of widgets but cannot contain another form.

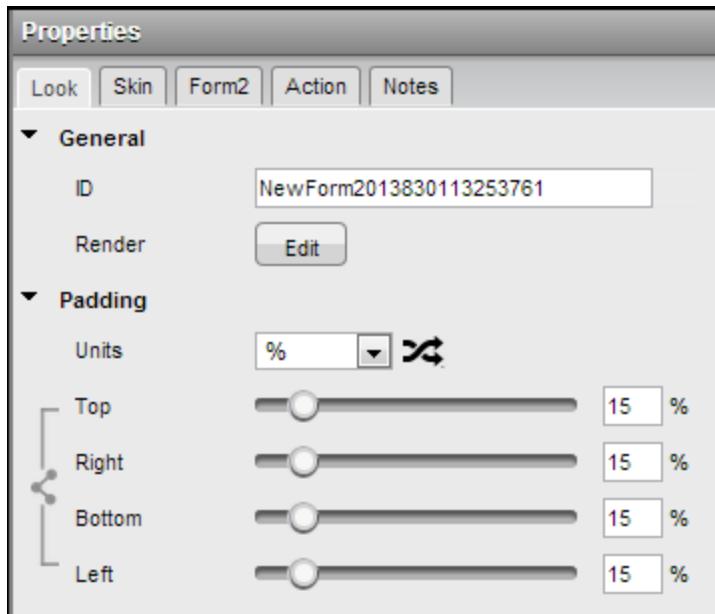
The screen width occupied by the form is the total available width of the mobile device; the virtual height is determined by the number of widgets on the form (the total height of the form is sum of the virtual heights of its first level visible child widgets).

### 6.1 Form Properties

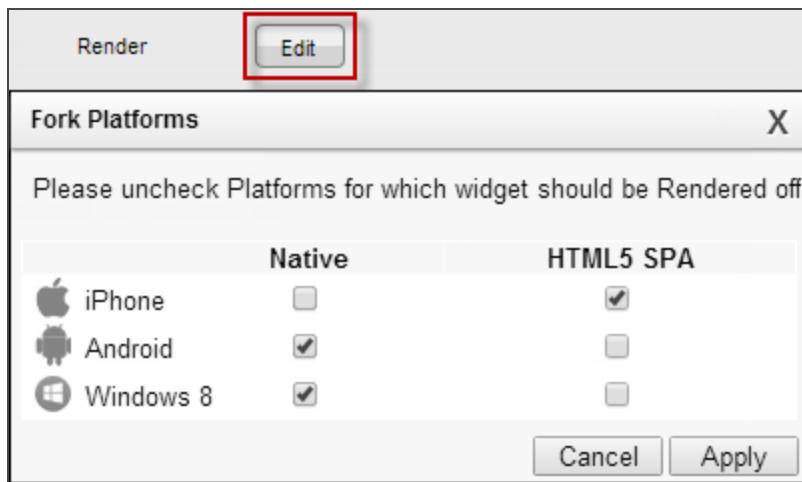
Form properties allows you to modify the design look and feel, provide platform specific properties, and attach various actions events.

### 6.2 Look

Following properties can be modified from this tab.



1. **General > ID:** It is the unique identifier of the form. When a form is added, system assigns a unique ID (or name). You can rename the form ID, if required.
2. **Render:** Defines whether a widget is displayed on a specific platform. By default, a widget is rendered for all the platforms. If you do not want to render a widget for a specific platform, click the **Edit** button against the **Render** field to open the **Fork Platforms** dialog box.



Clear the check box of the platforms for which the widget should not be rendered.

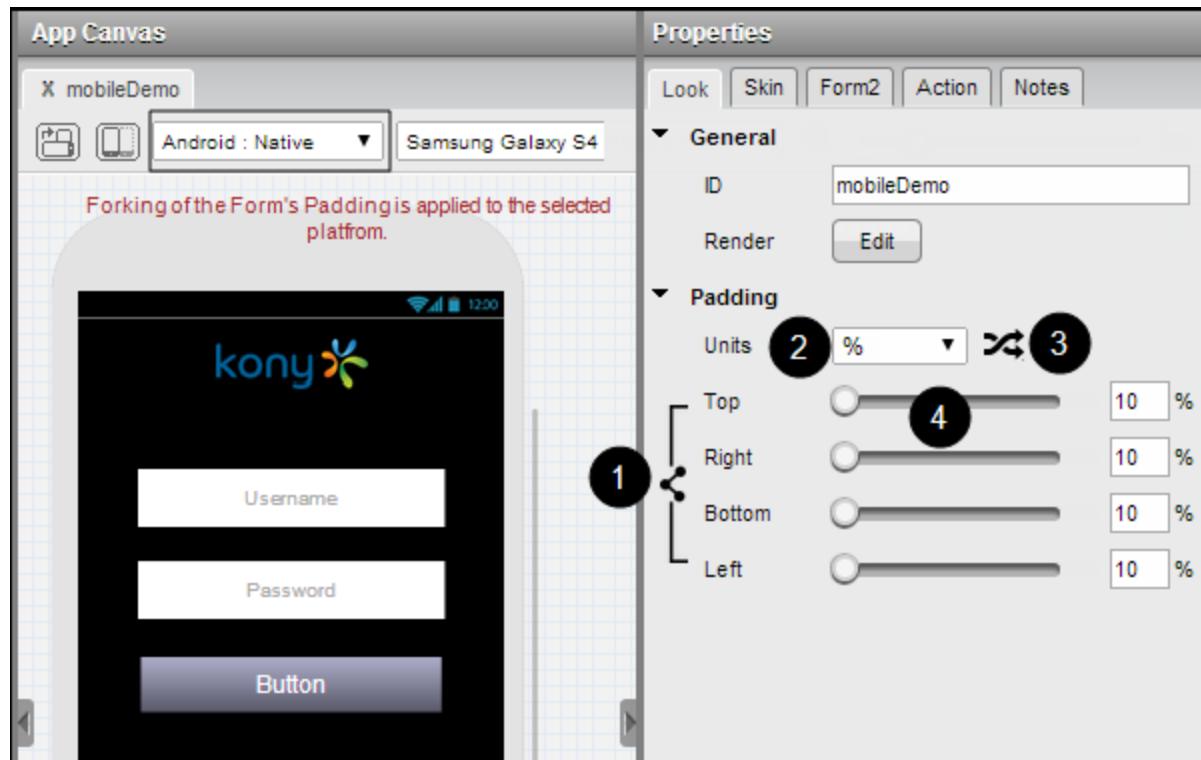
3. **Padding:** Defines the space between the content of the form and the form boundaries. Following padding options are available for the form.

| Property | Definition     | Action  |
|----------|----------------|---|
| Top      | Top padding    | Move the slider to adjust the top padding of the widget.    |
| Bottom   | Bottom padding | Move the slider to adjust the bottom padding of the widget. |
| Left     | Left padding   | Move the slider to adjust the left padding of the widget.   |
| Right    | Right padding  | Move the slider to adjust the right padding of the widget.  |

**Note:** Padding can be applied simultaneously for all the four sides by clicking the  next to the Units list.

## Forking a Form's Padding

To fork a form's padding, follow these steps:



1. Click the fork icon.
2. Select the padding units from the **Units** drop-down.
3. To apply the same values across the four sides, click uniform values icon. Platform
4. Adjust the values of the padding.

**Note:** The forking of a form's padding is applied only to the platform selected in the App Canvas.

## 6.3 Skin

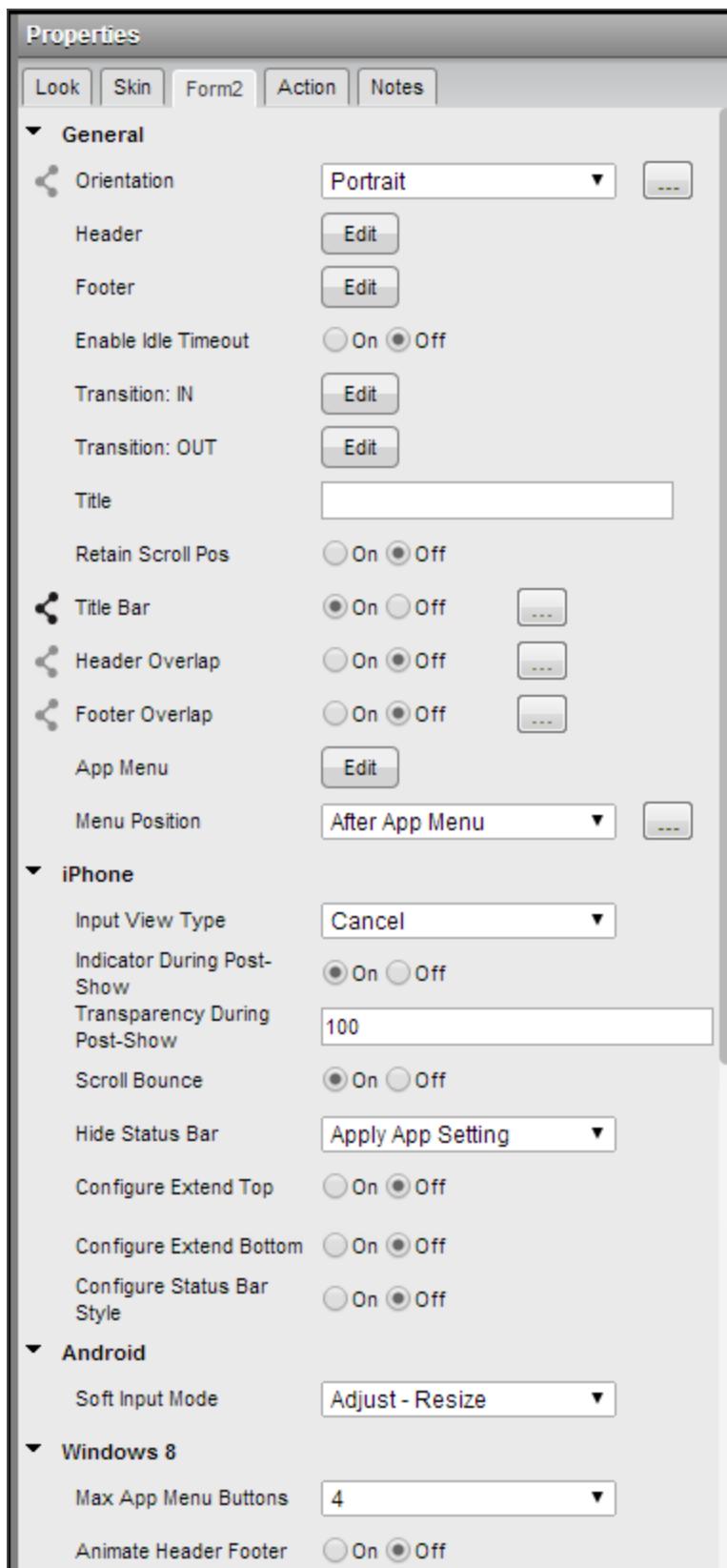
Skin define properties such as background color, borders and shadows for a form.

For a Form widget, you can apply a skin for the following states:

- Normal: Specifies a background skin for Form widget.
- Menu Normal: Specifies the look and feel of a menu items when not in focus.
- Menu Focus: Specifies the look and feel of the Menu Item when focused.

For more details on skin properties, refer to section: [Widget - Skins.](#)

## 6.4 Platform Specific properties



#### 6.4.1 Orientation

This selection decides forms orientation. To provide a platform specific orientation, click the ellipsis(...) next to the **Orientation** list. From the Orientation window,

- To assign a default orientation, from the **Value** list corresponding to **Default** platform, select an orientation from the list.
- To assign a platform specific orientation, select a platform check box, and the from the **Value** list corresponding to the selected platform, select an orientation.

#### 6.4.2 Header

To insert a header to a form, click the **Edit** button against the **Header** field. From the **Header** window, select a header.

**Note:** A header template should be created for a project before inserting the Header into a form.

Refer section [Headers](#) for steps to create a header template.

#### 6.4.3 Footer

To insert a footer to a form, click the **Edit** button against the **Footer**. From the **Footer** window, select a footer.

**Note:** A footer template should be created for a project before inserting the footer into a form.

Refer section [Footer](#) for steps to create a footer template.

#### 6.4.4 Enable Idle Timeout

Idle time indicates the amount of time that a user has not interacted with an application. Some applications require a notification to be raised when a user has not interacted with a form for a

specified amount of time. For example, a banking app might require a notification after five minutes of inactivity by the user. At the same time, applications also must be able to withhold this notification for certain forms in the application. For example, an ATM Locator form in a banking app will raise the notification after a specific period of inactivity. if **Enable Idle Timeout** property is selected.

- Click **On** to enable idle timeout.
- Click **Off** to disable idle timeout.

#### 6.4.5 Transition: IN

Specifies the configuration to be used when the user arrives on this form. Click the **Edit** button against **Transition: IN**. The **Transition: IN** dialog appears. Choose the desired transition direction and transaction effect for specific platforms.

##### 6.4.5.1 iOS Properties

**Transition Direction:** Specifies the direction from which the Form is displayed. The available options are:

- None - Use this option if you do not want to specify a transition direction.
- From Right - Specifies that the Form must appear from the right.
- From Left - Specifies that the Form must appear from the left.
- From Bottom - Specifies that the Form must appear from the bottom.
- From Top - Specifies that the Form must appear from the top.

**Transition Effect:** Specifies the effect from which the form is displayed. The available options are:

- None - Use this option if you do not want to specify a transition direction.
- Fade - Specifies that the form must fade when it is transitioned to a hidden or an invisible state.
- MoveIn - Specifies that the form must slide over the existing content in the direction as specified in the transition direction.

- Push - Specifies that the form must push the existing content in the direction as specified in the transition direction and take its place.
- Flip - Specifies that the form must be rotated along the Y-axis as the content is being displayed. It supports transition directions from right and left.
- Reveal - Specifies that the form must be revealed gradually in the direction as specified in the transition direction.
- Curl - Specifies that the form must be curled or folded (look and feel is similar to turning of a page in a book) as the content is being displayed. It supports transition directions from top and bottom only.
- Two Split Horizontal In - Specifies the form which is split horizontally into two parts rejoins when the transition takes place.
- Two Split Vertical In - Specifies the form which is split vertically into two parts rejoins when the transition takes place.
- Four Split In - Specifies the form which is split in four parts rejoins when the transition takes place.
- Four Split Rotate In - Specifies the form which is split in four parts rejoins by rotating the parts when the transition takes place.
- Two Split Horizontal Out - Specifies the form which is split horizontally into two parts and move away when the transition takes place.
- Two Split Vertical Out - Specifies the form which is split vertically into two parts and move away when the transition takes place.
- Four Split Rotate Out - Specifies the form which is split in four parts move away by rotating the parts when the transition takes place.
- Switch Left - Specifies that the form must go out of the view in 3D circular space along the Y-axis towards left as the content is being displayed.

- Switch Right - Specifies that the form must go out of the view in 3D circular space along the Y-axis towards right as the content is being displayed.
- Cloth - Specifies the present form must go out of screen animating as if a cloth is removed.
- Flip Right - Specifies that the form must be rotated along the Y-axis as the content is being displayed giving an illusion of a page is turned towards right in a book.
- Flip Left - Specifies that the form must be rotated along the Y-axis as the content is being displayed giving an illusion of a page is turned towards left in a book.
- Door - Specifies that the form must be revealed giving an illusion of a opening a door.
- Rotate Exchange - Specifies that the form must be rotated along the X-axis as the content is being displayed.

#### 6.4.5.2 Android Properties

**Transition Effect:** Specifies the effect from which the form is displayed. The available options are:

- None - The default device effect is applied or none of the effect is applied.
- Bottom Top - It specifies that the form must slide-in from the bottom and proceed towards the top.
- From Left - It specifies that the form must slide-in from the left with a fade effect.
- From Right - It specifies that the form must slide-in from the right with a fade effect.
- To Right - It specifies that the form must slide-out to the right with a fade effect.
- To Left - It specifies that the form must slide-out to the left with a fade effect.
- From Center - It specifies that the form must grow from the center with a fade effect.
- Top Right Bottom - It specifies that the form must slide-in from the top-right corner and proceed towards the bottom.

- Bottom Left Top - It specifies that the form must slide-in from the bottom-left corner and proceed towards the top.
- Bottom Top 2 - It specifies that the form must shrink from the bottom towards the top.
- Top Down - It specifies that the form must slide-in from the top and proceed towards the bottom.

#### 6.4.5.3 Windows Phone 8 Properties

**Transition Mode:** Specifies the direction from which the Form is displayed. The available options are:

- Default - The default device effect is applied or none of the effect is applied.
- Sequential- The transition of the Form going out of the view completes first and then the transition of the Form coming into the view takes place.
- Parallel- The transition of the Form going out of the view and the transition of the Form coming into the view takes place simultaneously.

**Transition Key** - Specifies the effect from which the Form is displayed. The available options are:

- Default - Specifies that the Form must slide horizontally into the view.
- Rotate 3D Single - Specifies that the Form must be rotated along the center Y-Axis when coming into the view.
- Rotate 3D Dual - Specifies that the Form must be shown making a circle around the screen from the background before coming into the view.
- Slide - Specifies that the Form must slide horizontally into the view.
- Pop - Specifies that the Form must emerge from center-bottom of the screen and gradually occupy the complete screen.
- Squeeze - Specifies that the Form must be expanded horizontally from an initial width of zero.

**Transition Speed** - Specifies the speed at which the Form is transitioned. The value must be specified in seconds.

#### 6.4.5.4 SPA Properties

**Transition Effect:** Specifies the effect from which the form is displayed. The available options are:

- None- Use this option if you do not want to specify a transition direction.
- Top Center - Specifies that the form must appear from the top center.
- Bottom Center - Specifies that the form must appear from the bottom center.
- Right Center - Specifies that the form must appear from the right center.
- Left Center - Specifies that the form must appear from the left center.

#### 6.4.6 Transition: OUT

Specifies the configuration to be used when the user arrives on this form. Click the **Edit** button against **Transition: OUT** field. From the **Transcation: OUT** window, choose the desired transition direction and transaction effect for specific platforms.

#### 6.4.6.1 iOS Properties

**Transition Direction:** Specifies the direction from which the Form is displayed. The available options are:

- None - Use this option if you do not want to specify a transition direction.
- From Right - Specifies that the Form must appear from the right.
- From Left - Specifies that the Form must appear from the left.
- From Bottom - Specifies that the Form must appear from the bottom.
- From Top - Specifies that the Form must appear from the top.

**Transition Effect:** Specifies the effect from which the form is displayed. The available options are:

- None - Use this option if you do not want to specify a transition direction.
- Fade - Specifies that the form must fade when it is transitioned to a hidden or an invisible state.
- MoveIn - Specifies that the form must slide over the existing content in the direction as specified in the transition direction.
- Push - Specifies that the form must push the existing content in the direction as specified in the transition direction and take its place.
- Flip - Specifies that the form must be rotated along the Y-axis as the content is being displayed. It supports transition directions from right and left.
- Reveal - Specifies that the form must be revealed gradually in the direction as specified in the transition direction.
- Curl - Specifies that the form must be curled or folded (look and feel is similar to turning of a page in a book) as the content is being displayed. It supports transition directions from top and bottom only.
- Two Split Horizontal In - Specifies the form which is split horizontally into two parts rejoins when the transition takes place.
- Two Split Vertical In - Specifies the form which is split vertically into two parts rejoins when the transition takes place.
- Four Split In - Specifies the form which is split in four parts rejoins when the transition takes place.
- Four Split Rotate In - Specifies the form which is split in four parts rejoins by rotating the parts when the transition takes place.
- Two Split Horizontal Out - Specifies the form which is split horizontally into two parts and move away when the transition takes place.
- Two Split Vertical Out - Specifies the form which is split vertically into two parts and move away when the transition takes place.

- Four Split Rotate Out - Specifies the form which is split in four parts move away by rotating the parts when the transition takes place.
- Switch Left - Specifies that the form must go out of the view in 3D circular space along the Y-axis towards left as the content is being displayed.
- Switch Right - Specifies that the form must go out of the view in 3D circular space along the Y-axis towards right as the content is being displayed.
- Cloth - Specifies the present form must go out of screen animating as if a cloth is removed.
- Flip Right - Specifies that the form must be rotated along the Y-axis as the content is being displayed giving an illusion of a page is turned towards right in a book.
- Flip Left - Specifies that the form must be rotated along the Y-axis as the content is being displayed giving an illusion of a page is turned towards left in a book.
- Door - Specifies that the form must be revealed giving an illusion of a opening a door.
- Rotate Exchange - Specifies that the form must be rotated along the X-axis as the content is being displayed.

#### 6.4.6.2 Android Properties

**Transition Effect:** Specifies the effect from which the form is displayed. The available options are:

- None - The constant value is 0. The default device effect is applied or none of the effect is applied.
- Bottom Top - The constant value is 1. It specifies that the form must slide-in from the bottom and proceed towards the top.
- From Left - The constant value is 2. It specifies that the form must slide-in from the left with a fade effect.
- From Right - The constant value is 3. It specifies that the form must slide-in from the right with a fade effect.

- To Right - The constant value is 4. It specifies that the form must slide-out to the right with a fade effect.
- To Left - The constant value is 5. It specifies that the form must slide-out to the left with a fade effect.
- From Center - The constant value is 6. It specifies that the form must grow from the center with a fade effect.
- Top Right Bottom - The constant value is 7. It specifies that the form must slide-in from the top-right corner and proceed towards the bottom.
- Bottom Left Top - The constant value is 8. It specifies that the form must slide-in from the bottom-left corner and proceed towards the top.
- Bottom Top 2 - The constant value is 9. It specifies that the form must shrink from the bottom towards the top.
- Top Down - The constant value is 10. It specifies that the form must slide-in from the top and proceed towards the bottom.

#### 6.4.6.3 Windows Phone 8 Properties

**Transition Mode:** Specifies the direction from which the Form is displayed. The available options are:

- Default - The default device effect is applied or none of the effect is applied.
- Sequential- The transition of the Form going out of the view completes first and then the transition of the Form coming into the view takes place.
- Parallel- The transition of the Form going out of the view and the transition of the Form coming into the view takes place simultaneously.

**Transition Key** - Specifies the effect from which the Form is displayed. The available options are:

- Default - Specifies that the Form must slide horizontally into the view.
- Rotate 3D Single - Specifies that the Form must be rotated along the center Y-Axis when coming into the view.
- Rotate 3D Dual - Specifies that the Form must be shown making a circle around the screen from the background before coming into the view.
- Slide - Specifies that the Form must slide horizontally into the view.
- Pop - Specifies that the Form must emerge from center-bottom of the screen and gradually occupy the complete screen.
- Squeeze - Specifies that the Form must be expanded horizontally from an initial width of zero.

**Transition Speed** - Specifies the speed at which the Form is transitioned. The value must be specified in seconds.

#### 6.4.6.4 SPA Properties

**Transition Effect:** Specifies the effect from which the form is displayed. The available options are:

- None- Use this option if you do not want to specify a transition direction.
- Top Center - Specifies that the form must appear from the top center.
- Bottom Center - Specifies that the form must appear from the bottom center.
- Right Center - Specifies that the form must appear from the right center.
- Left Center - Specifies that the form must appear from the left center.

#### 6.4.7 Title

Specifies the title of the form and is displayed in the Title Bar.

To update Title property, refer to [Title Bar](#).

#### 6.4.8 Retain Scroll Pos

Specifies if the Form must remember the scroll position when the user revisits the Form.

#### 6.4.9 Title Bar

Specifies whether the title bar must be displayed on a form.

- Click **On** to display the titleBar.
- Click **Off** to hide the titleBar.

The selected option becomes the default for all the platforms.

If you wanted to choose a different **Title Bar** option for a specific platform, click the ellipses (...) button next to the **Title Bar Off** option to open **TitleBar** dialog box.

Select a platform and from the corresponding **Value** field , select the check box to enable or disable displaying the title bar.

#### 6.4.10 Header Overlap

Specifies if the header must overlap the form. For example, every time you scroll the form, the header is fixed and the header overlaps the form as specified in the header overlap field. If this field is selected, the form scrolls behind the header background and a part of the header background is transparent.

- Click **On** to allow header overlap the form.
- Click **Off** to disallow header overlap the form.

If you wanted to choose a different **Header Overlap** option for a specific platform, click the ellipses (...) button next to the **Header Overlap Off** option to open **Header Overlap** dialog box.

Select a platform and from the corresponding **Value** field , select the check box to allow or disallow header overlap.

#### 6.4.11 Footer Overlap

Specifies if the footer must overlap the form. For example, every time you scroll the form, the footer is fixed and the footer overlaps the form as specified in the Footer Overlap field. If this field is selected, the form scrolls behind the footer background and a part of the footer background is transparent.

- Click **On** to allow footer overlap the form.
- Click **Off** to disallow footer overlap the form.

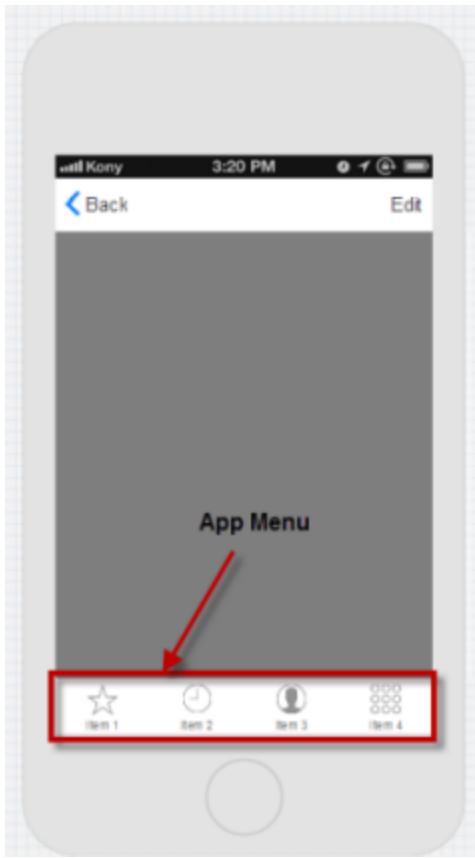
If you wanted to choose a different **Footer Overlap** option for a specific platform, click the ellipses (...) button next to the **Footer Overlap Off** option to open **Footer Overlap** dialog box.

Select a platform and from the corresponding **Value** field , select the check box to allow or disallow footer overlap.

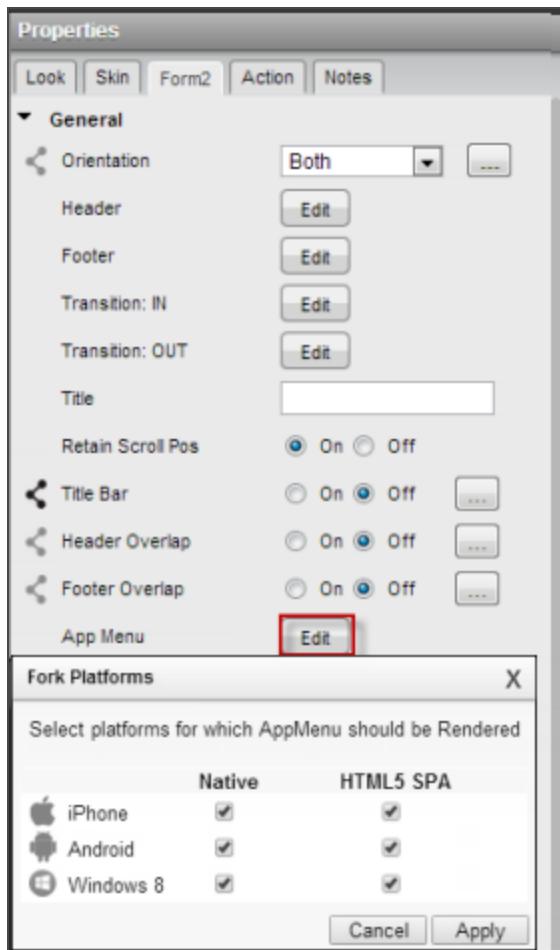
#### 6.4.12 App Menu

The Application Menu (App Menu) contains features that apply to the application as a whole, rather than to a specific application screen or window.

You can choose to display App Menu on every form or on specific forms of the application.

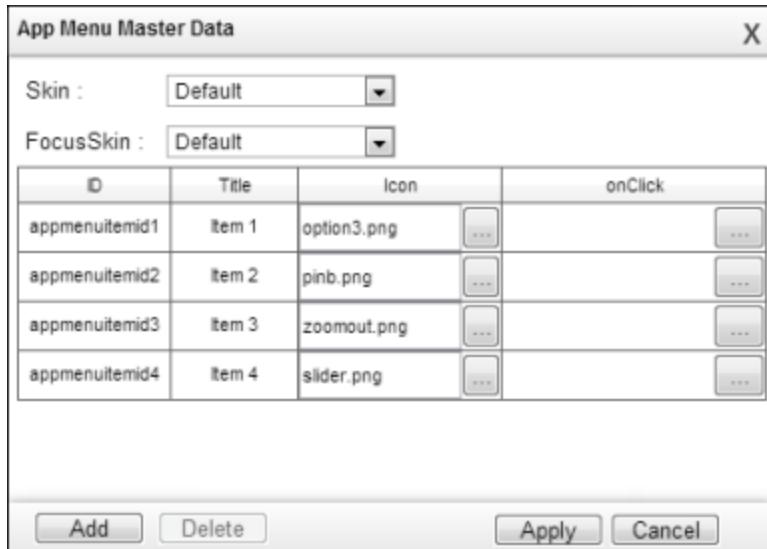


By default, App Menu is viable on all the platforms. To hide App Menu for a specific platforms, click the **Edit** button next to the **App Menu** to open the **Fork Platforms** dialog box. Clear a platform check box on which you wish to hide App Menu.



To customize App Menu items, follow below steps:

1. Click inside the App Menu in the App Canvas to display a right arrow.
2. Click the right arrow, and then click **Action: Edit Master Data** to open App Menu Master Data.



3. You can customize the following properties:
  - a. Skin: Select a skin from the list of available skins.
  - b. FocusSkin: Select a skin from the list of available skins.
  - c. ID: Unique identifier for the menu item.
  - d. Title: name of the menu item
  - e. Icon: An icon is displayed above the Title. To assign an icon, click the ellipsis (...) button in the cell, and then select an icon.
  - f. onClick: You can assign onClick event by clicking the ellipsis (...) button in the cell and creating an event in the **Action Editor** dialog box.

#### 6.4.13 Menu Position

Specifies if the menu is shown or hidden in the application. In some platforms, form menu items appear along with app menu items. This property allows developer to configure whether to append at the end of application menu list or inserted at beginning of the application menus.

Following are the available options:

- After App Menu
- Before App Menu

The selected menu position here becomes the default for all the platforms.

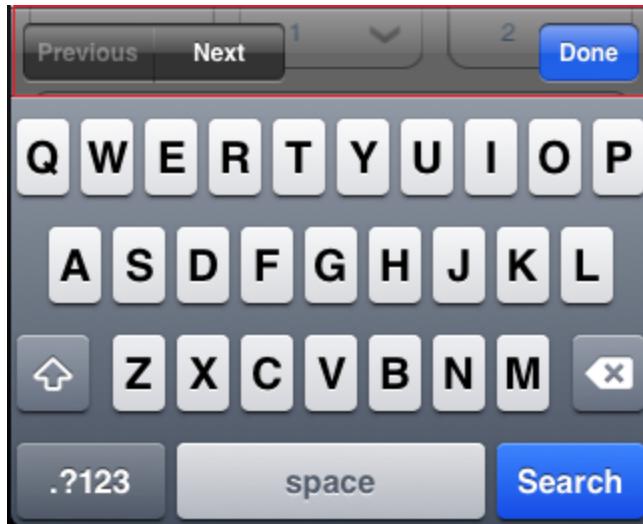
For specifying a platform specific menu position, click ellipsis (...) button next to the **Menu Postion** list to open **Menu Postion** dialog box.

Select the platform, and from its corresponding **Value** list, choose a desired menu position.

#### 6.4.14 Input View Type

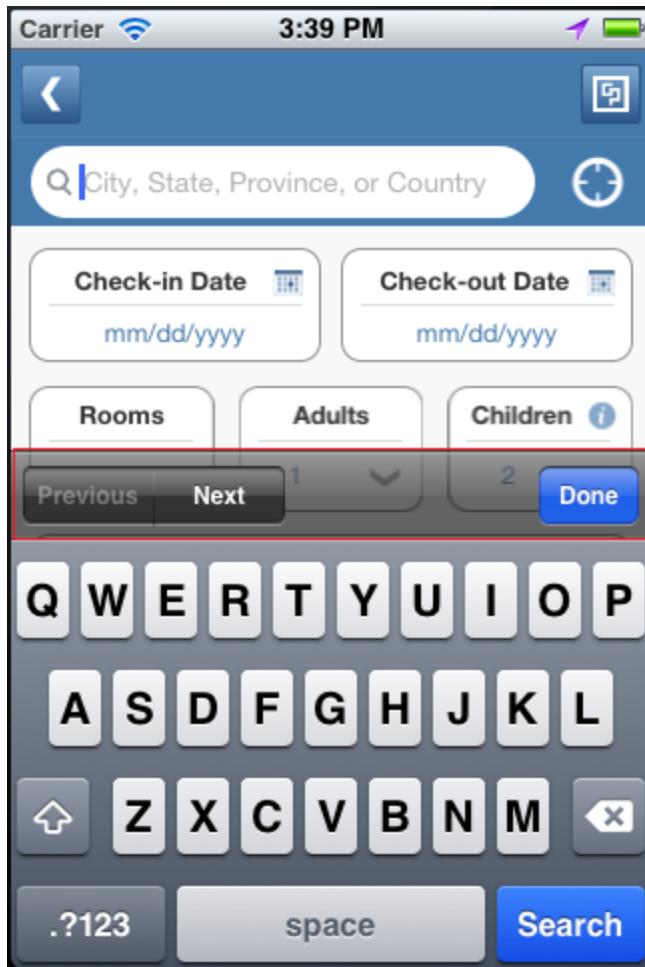
When building iPhone applications that support or provide text input, it's often necessary to create some extra buttons (or other controls) beyond the ones provided by the default keyboard interface. Kony Platform by default, adds the Previous, Next and Done buttons to the applicable input controls. These buttons allow specific operations needed by your application, such as moving to the next or previous text field, make the keyboard disappear. The area above the keyboard is known as Input Accessory View.

This property, allows you to specify the type of accessory view that will be shown for all the input controls on this form.



Following are the available options:

- None: Use this option if you do not want to specify the toolbar. This option should be used carefully, as setting this option for widgets like calendar leaves the user with no option to select and drop-down a wheel calendar.
- Default: Specifies that the toolbar that is defined in the Application level settings. To set the Application level settings, right-click on the project and navigate to **Properties > Native App > iPhone/iPad**.
- Next / Previous: Specifies the navigation options as Next, Previous, and Done for a form. The below image illustrates the **next prev toolbar** set for a **Textbox**. The highlighted toolbar is achieved by setting the **Keyboard Type** as **Default** for a Textbox and **Input Accessory View Type** as **nextprevtoolbar** to the Form.



- Cancel: Specifies that the input accessory view has a cancel button. This option does not trigger any events.

**Note:** This property is specific to the iOS platform.

#### 6.4.15 Indicator During Post-Show

This property specifies if there must be a progress indicator to show that the form content is being loaded. You can use this property for forms that require network calls during post-show.

- Click **On** to display the progress indicator.
- Click **Off** to hide the progress indicator.

The following image illustrates the loading indicator:



**Note:** This property is specific to the iOS platform.

#### 6.4.16 Transparency During Post-Show

Normally, when a form is loading, the user interface is blocked. You can use this property to specify the percentage of transparency.

Default: 100

**Note:** This property is specific to the iOS platform.

#### 6.4.17 Scroll Bounces

Specifies whether the scroll view bounces past the edge of the content and back again.

- If **On** option is selected, the scroll view bounce is allowed.
- If **Off** option is selected, the scroll view bounce is disallowed.

**Note:** This property is specific to the iOS platform.

#### 6.4.18 Configure Extend Top

- If **On** option is selected, the property Extend Top is displayed.
- If **Off** option is selected, the property Extend Top is not displayed.

**Note:** This property is specific to the iOS platform.

#### 6.4.19 Configure Extend Bottom

- If **On** option is selected, the property Extend Bottom is displayed.
- If **Off** option is selected, the property Extend Bottom is not displayed.

**Note:** This property is specific to the iOS platform.

#### 6.4.20 Configure Status Bar Style

- If **On** option is selected, the property Status Bar Style is displayed.
- If **Off** option is selected, the property Status Bar Style is not displayed.

**Note:** This property is specific to the iOS platform.

#### 6.4.21 Extend Top

Specifies the form content to scroll under the App Menu. This property is supported in iOS7 and above only. This property is also applicable on the Application Level. The property set at form level takes precedence over Application level. If **On** option is selected, the form scroll under the App Menu.

**Note:** This property is specific to the iOS later platforms.

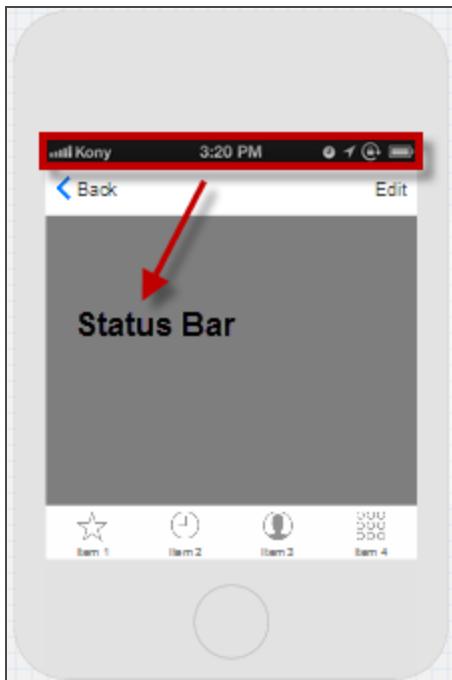
#### 6.4.22 Extend Bottom

This property enables you to Configure Extend Bottom property. This property is supported on iOS7 and above only. This property is also applicable on the Application Level. The property set at form level takes precedence over Application level. If **On** option is selected, the property extendBottom is displayed.

**Note:** This property is specific to the iOS platform.

#### 6.4.23 Status Bar Style

This property enables you to set the status bar style.



You can select one of the following styles for your status bar:

- Default: The status bar contents are displayed in black color suitable for a light background.
- Light Content: The status bar contents are displayed in white color suitable for a dark background.

**Note:** This property is specific to the iOS7 platform.

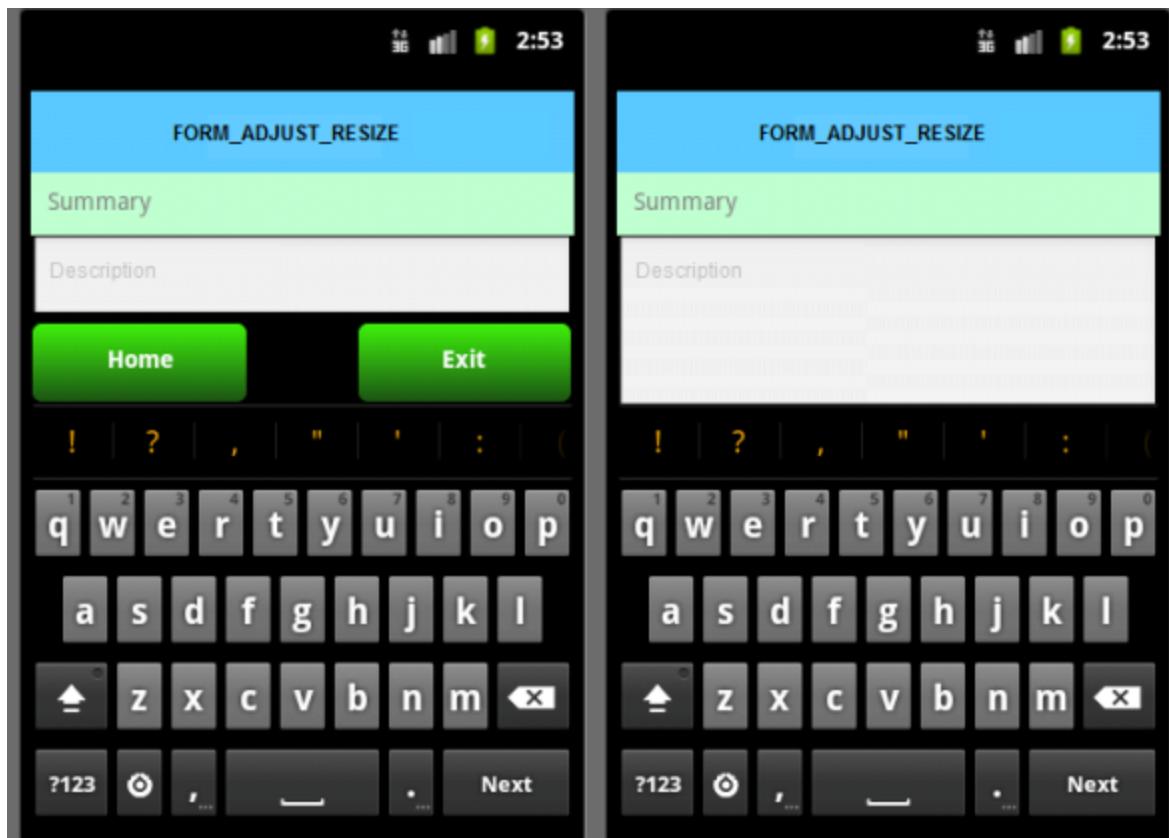
#### 6.4.24 Soft Input Mode

Specifies the soft keyboard for a widget that requires an input should be placed at the bottom of the form. For example, when a TextBox or a TextArea requires input from the user, the soft-keyboard pops up when you enter text.

Following are the available options:

- Adjust - Resize: Specifies the form is resized and when you click or start typing within the widget which requires an input, the form scrolls up and the widget which requires an input is not overlapped by the keypad or footer.
- Adjust - Pan: Specifies the widget which requires an input is placed at the bottom of the form is overlapped by the keypad.

The below image illustrates the above two options:



**Note:** This property is specific to the Android platform.

#### 6.4.25 Max App Menu Buttons

Specifies the number of app menu buttons that should be displayed on the screen.

**Note:** This property is specific to the Windows 8 platform.

#### 6.4.26 Animate Header Footer

Specifies if the headers and footers of the Form must slide up and down respectively away from the view when the Form is in transition. To enable animation, select the **On** option next to the **Animate Header Footer**.

**Note:** This property is specific to the Windows 8 platform.

### 6.5 Actions

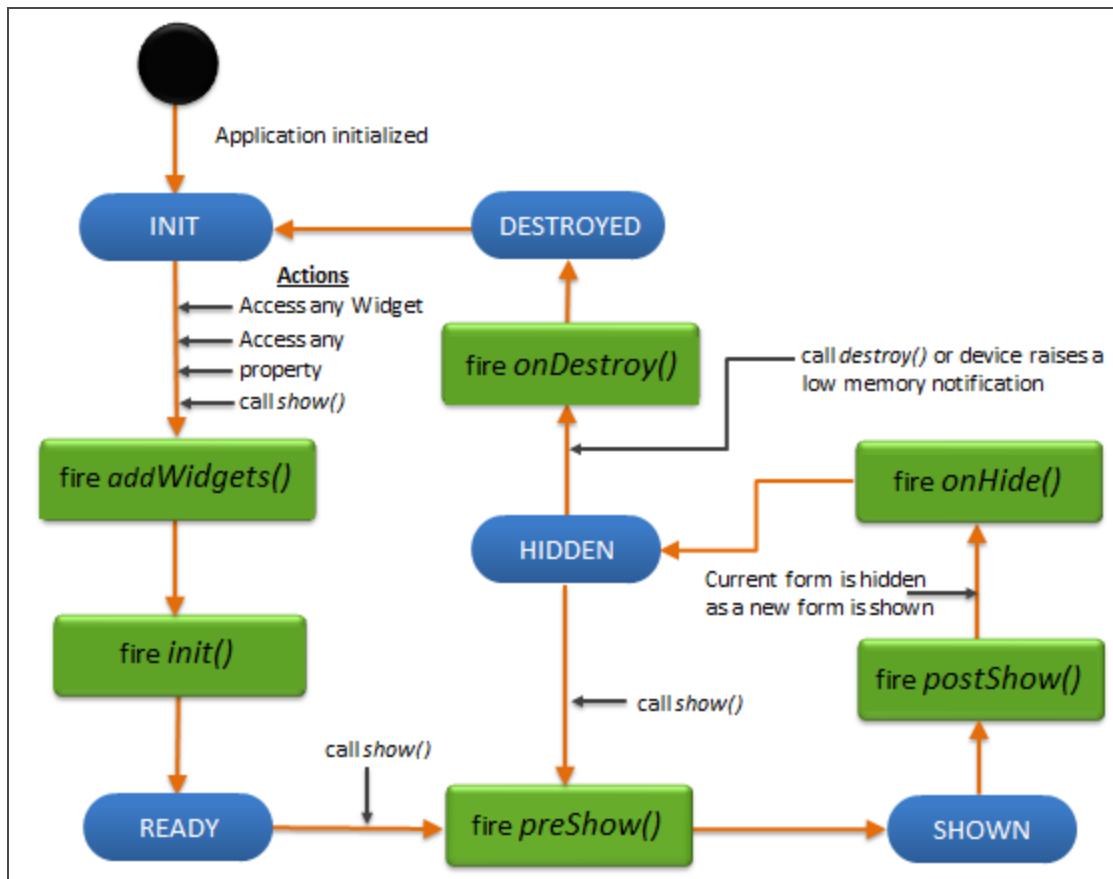
Form has the following actions associated with it:

- **init** - called immediately after an addWidgets event for any initializations required for the form.  
Init initializes the form and any widgets.

In case of Server side Mobile Web and SPA, addWidgets and init events gets called as soon as the form is created. In case of native platforms, as an optimization, these events are deferred until the first access.

- **preh3** - called just before a form is visible on the screen. A form can be made visible by explicitly calling the h3 method of the form.
- **postShow** - called immediately after the form is visible on the screen. A form is made visible by explicitly calling the show method of the form.
- **onHide** - called when the form goes out of the screen. A form can go out of the screen when another form is to be shown.
- **onDestroy** - called when a form is destroyed. A form is destroyed when the developer explicitly calls destroy and this event gets called before destroying the form.

The following image illustrates the lifecycle of a form:



- **onOrientationChange**: Specifies an Event which is triggered when there is a change in orientation of the form from portrait to landscape or vice versa.

**Note:** This property is specific to the iOS platform.

- **onDeviceBack**: Specifies an event which is triggered when the user uses the back button on the device.

**Note:** This property is specific to the Android platform.

- **onDeviceMenu**: Specifies an event which is triggered when the user uses the back button on the device.

**Note:** This property is specific to the Android platform.

- **onOrientationChange:** Specifies an event which is triggered when there is a change in orientation of the form from portrait to landscape or vice versa.

**Note:** This property is specific to the Android platform.

- **onDeviceBack:** Specifies an event which is triggered when the user uses the back button on the device.

**Note:** This property is specific to the Windows 8 platform.

- **onOrientationChange:** Specifies an event which is triggered when there is a change in orientation of the form from portrait to landscape or vice versa.

**Note:** This property is specific to the Windows platform.

- **onDeviceBack:** Specifies an event which is triggered when the user uses the back button on the device.

**Note:** This property is specific to the SPA platform.

- **onOrientationChange:** Specifies an event which is triggered when there is a change in orientation of the form from portrait to landscape or vice versa.

**Note:** This property is specific to the SPA platform.

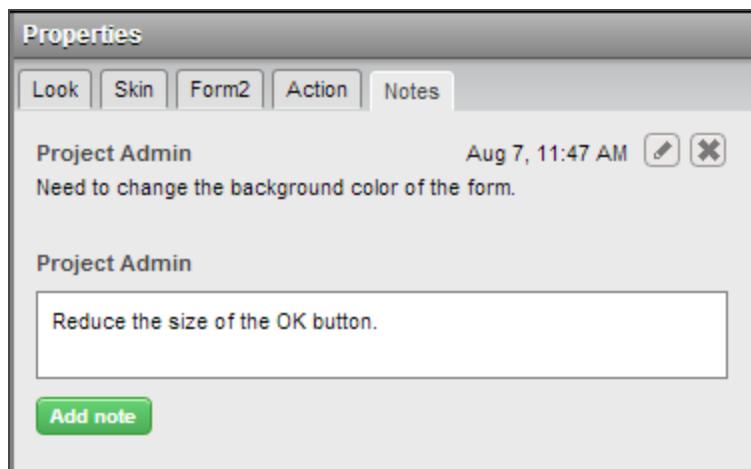
For more information on using the above action, refer to section [Action Editor](#).

## 6.6 Notes

**Notes** allow you to provide comments, feedback and suggestions about a form (or page). These notes are visible to all the users during function preview (cloud).

### 6.6.1 Adding a Note

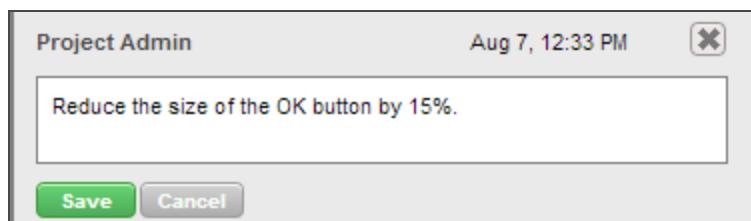
To add a note, follow these steps:



1. Navigate to the required form, and from its **Properties** tab, click **Notes**.
2. Type your note in the box.
3. Click **Add Notes**.

### 6.6.2 Editing a note

To edit a note, follow these steps:



1. Navigate to the required form, and from its **Properties** tab, click **Notes**.
2. Click  (edit icon) of a note.
3. Make necessary changes to the note.
4. Click **Save**.

**Note:** Only the author of a note can edit it.

### 6.6.3 Delete a Note

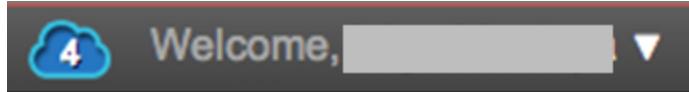
To delete a note, follow these steps:

1. Navigate to the required form, and from its **Properties** tab, click **Notes**.
2. Click the delete icon of a note.

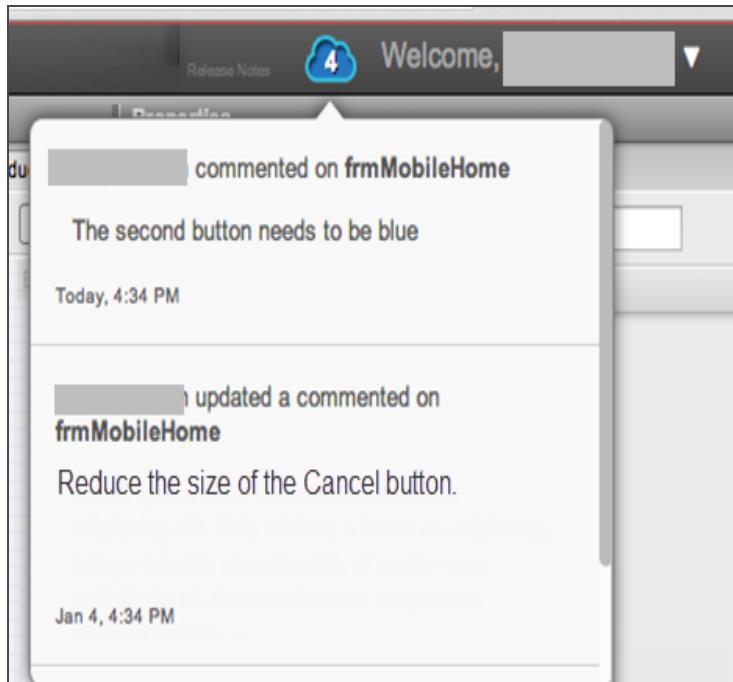
**Note:** Only the author of a note can delete the note.

### 6.6.4 Viewing Functional Preview Notes

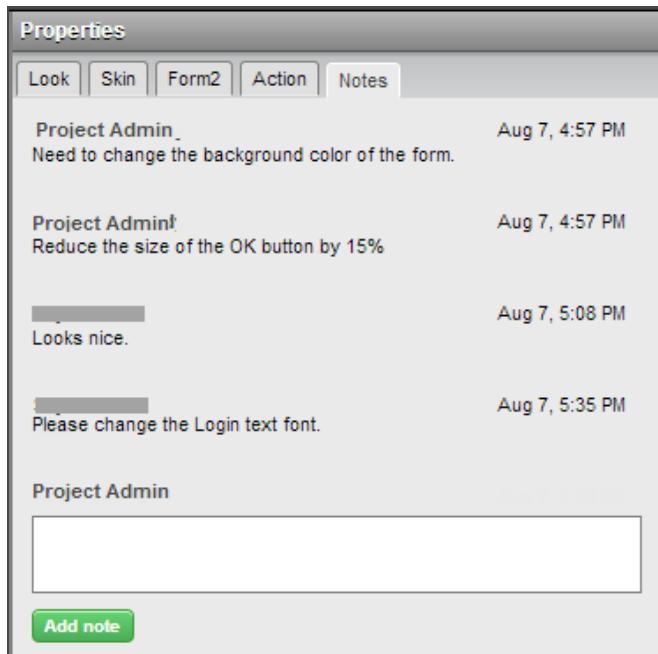
When a note is added during [Functional Preview \(Cloud\)](#) they will be available to you in your Kony Visualizer project. The cloud icon in your project will change to



Click the icon to view the added notes:



These notes are downloaded to your local project and are available in your form Notes tab.



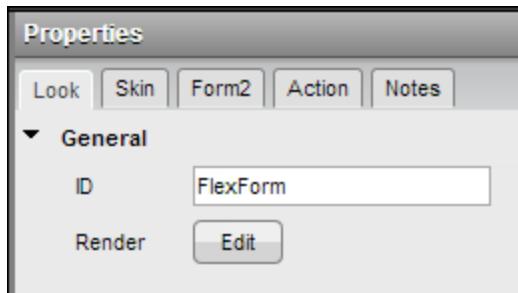
## 7. Flex Form

Form is a visual area (basic application screen) that holds other widgets. You can use a form to set a title and scroll content (similar to a web browser). The entire contents of the form except the headers and footers scroll together. A form is the top most container widget. A form can contain any number of widgets but cannot contain another form.

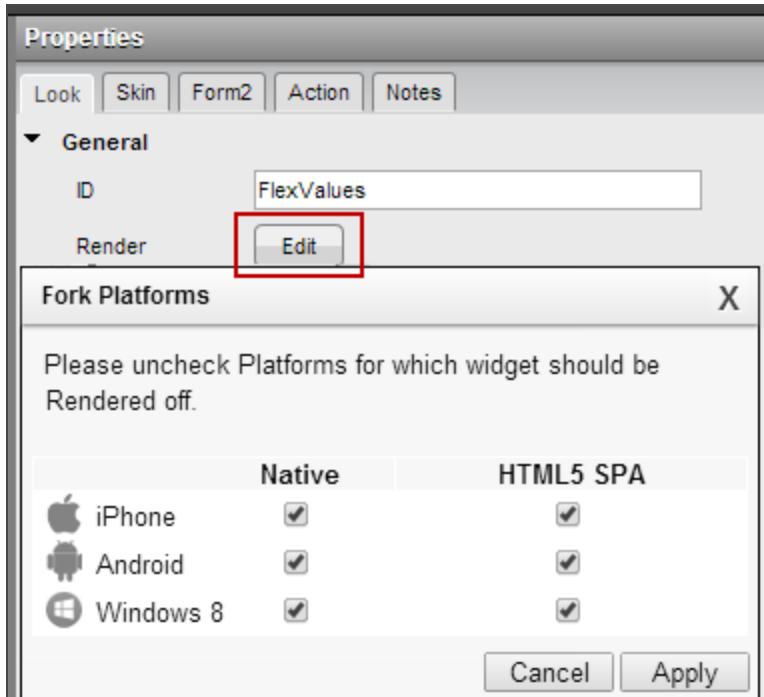
The screen width occupied by the form is the total available width of the mobile device; the virtual height is determined by the number of widgets on the form (the total height of the form is sum of the virtual heights of its first level visible child widgets).

### 7.1 Look

This section details the **Properties > Look** tab of a VBox widget. Here, you can define various widget properties that are explained in this section.



1. **ID:** It is the unique identifier of the form. When a form is added, system assigns a unique ID (or name). You can rename the form ID, if required.
2. **Render:** This property defines whether a form should be displayed on a specific platform. By default, a form is rendered for all the platforms.



If you do not want to render a form for a specific platform, click the **Edit** button against the **Render** to open the **Fork Platforms** dialog box. Clear a platform check box for which you do not want to render the form.

## 7.2 Skin

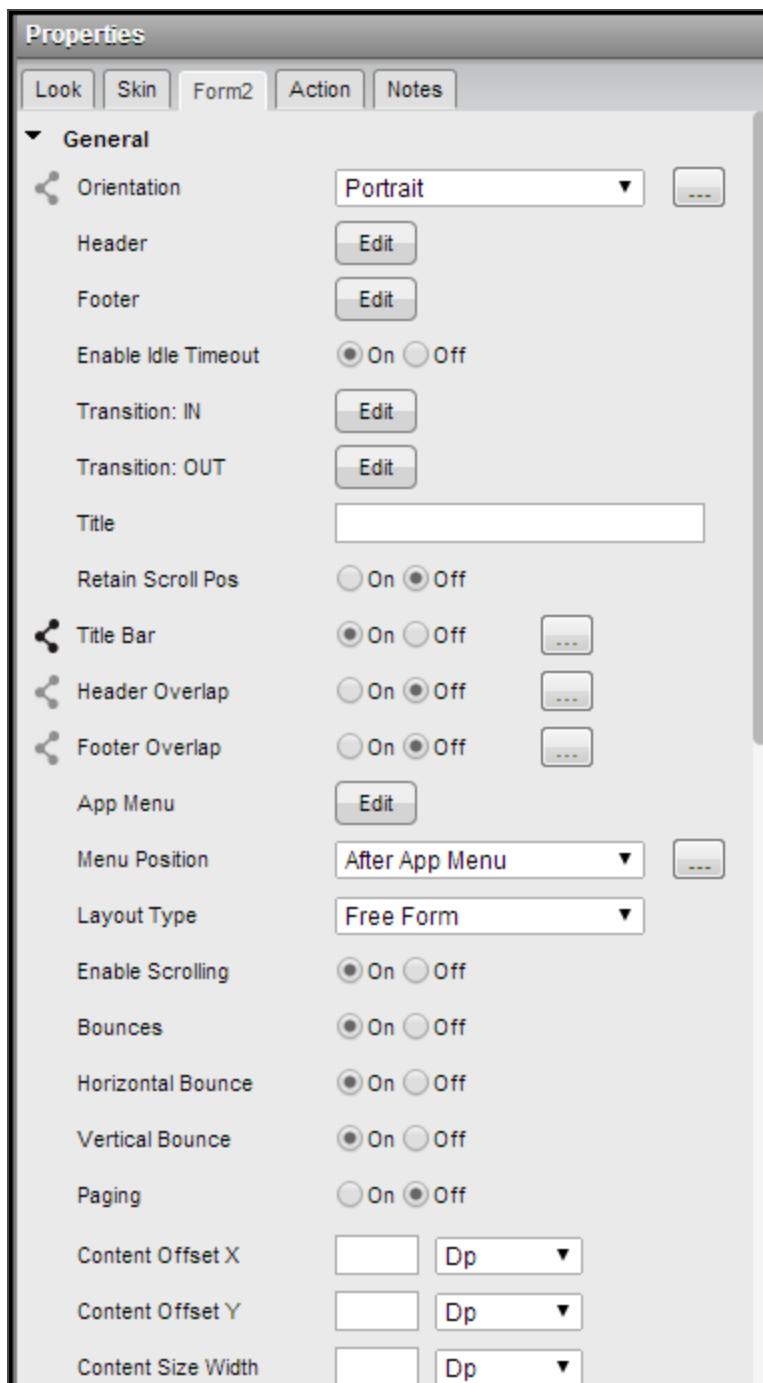
Skin define properties such as background color, borders and shadows for a form.

For a Form widget, you can apply a skin for the following states:

- Normal: Specifies a background skin for Form widget.
- Menu Normal: Specifies the look and feel of a menu items when not in focus.
- Menu Focus: Specifies the look and feel of the Menu Item when focused.

For more details on skin properties, refer to section: [Widget - Skins.](#)

## 7.3 Platform Specific properties



|                               |   |
|-------------------------------|---|
| Content Size Height           | <input type="text"/> Dp ▾                                     |
| Snap to Grid                  | <input checked="" type="radio"/> On <input type="radio"/> Off |
| Snap Grid Size                | <input type="text"/> 10                                       |
| Default Unit                  | Dp ▾  |
| <b>iPhone</b>                 |   |
| Input View Type               | <input type="text"/> Cancel ▾                                 |
| Indicator During Post-Show    | <input checked="" type="radio"/> On <input type="radio"/> Off |
| Transparency During Post-Show | <input type="text"/> 100                                      |
| Scroll Bounce                 | <input checked="" type="radio"/> On <input type="radio"/> Off |
| Hide Status Bar               | <input type="text"/> Apply App Setting ▾                      |
| Configure Extend Top          | <input type="radio"/> On <input checked="" type="radio"/> Off |
| Configure Extend Bottom       | <input type="radio"/> On <input checked="" type="radio"/> Off |
| Configure Status Bar Style    | <input type="radio"/> On <input checked="" type="radio"/> Off |
| Zoom Scale                    | <input type="text"/> 1.0                                      |
| Min Zoom Scale                | <input type="text"/> 1.0                                      |
| Max Zoom Scale                | <input type="text"/> 1.0                                      |
| <b>Android</b>                |   |
| Soft Input Mode               | <input type="text"/> Adjust - Resize ▾                        |
| <b>Windows 8</b>              |   |
| Max App Menu Buttons          | <input type="text"/> 4 ▾                                      |
| Animate Header Footer         | <input type="radio"/> On <input checked="" type="radio"/> Off |

### 7.3.1 Orientation

This selection decides forms orientation. To provide a platform specific orientation, click the ellipsis(...) next to the **Orientation** list. From the Orientation window,

- To assign a default orientation, from the **Value** list corresponding to **Default** platform, select an orientation from the list.
- To assign a platform specific orientation, select a platform check box, and the from the **Value** list corresponding to the selected platform, select an orientation.

### 7.3.2 Header

To insert a header to a form, click the **Edit** button against the **Header** field. From the **Header** window, select a header.

**Note:** A header template should be created for a project before inserting the Header into a form.

Refer section [Headers](#) for steps to create a header template.

### 7.3.3 Footer

To insert a footer to a form, click the **Edit** button against the **Footer**. From the **Footer** window, select a footer.

**Note:** A footer template should be created for a project before inserting the footer into a form.

Refer section [Footer](#) for steps to create a footer template.

### 7.3.4 Enable Idle Timeout

Idle time indicates the amount of time that a user has not interacted with an application. Some applications require a notification to be raised when a user has not interacted with a form for a

specified amount of time. For example, a banking app might require a notification after five minutes of inactivity by the user. At the same time, applications also must be able to withhold this notification for certain forms in the application. For example, an ATM Locator form in a banking app will raise the notification after a specific period of inactivity. if **Enable Idle Timeout** property is selected.

- Click **On** to enable idle timeout.
- Click **Off** to disable idle timeout.

### 7.3.5 Transition: IN

Specifies the configuration to be used when the user arrives on this form. Click the **Edit** button against **Transition: IN** field. From the **Transition: IN** window, choose the desired transition direction and transaction effect for specific platforms.

#### 7.3.5.1 iOS Properties

**Transition Direction:** Specifies the direction from which the Form is displayed. The available options are:

- None - Use this option if you do not want to specify a transition direction.
- From Right - Specifies that the Form must appear from the right.
- From Left - Specifies that the Form must appear from the left.
- From Bottom - Specifies that the Form must appear from the bottom.
- From Top - Specifies that the Form must appear from the top.

**Transition Effect:** Specifies the effect from which the form is displayed. The available options are:

- None - Use this option if you do not want to specify a transition direction.
- Fade - Specifies that the form must fade when it is transitioned to a hidden or an invisible state.
- MoveIn - Specifies that the form must slide over the existing content in the direction as specified in the transition direction.

- Push - Specifies that the form must push the existing content in the direction as specified in the transition direction and take its place.
- Flip - Specifies that the form must be rotated along the Y-axis as the content is being displayed. It supports transition directions from right and left.
- Reveal - Specifies that the form must be revealed gradually in the direction as specified in the transition direction.
- Curl - Specifies that the form must be curled or folded (look and feel is similar to turning of a page in a book) as the content is being displayed. It supports transition directions from top and bottom only.
- Two Split Horizontal In - Specifies the form which is split horizontally into two parts rejoins when the transition takes place.
- Two Split Vertical In - Specifies the form which is split vertically into two parts rejoins when the transition takes place.
- Four Split In - Specifies the form which is split in four parts rejoins when the transition takes place.
- Four Split Rotate In - Specifies the form which is split in four parts rejoins by rotating the parts when the transition takes place.
- Two Split Horizontal Out - Specifies the form which is split horizontally into two parts and move away when the transition takes place.
- Two Split Vertical Out - Specifies the form which is split vertically into two parts and move away when the transition takes place.
- Four Split Rotate Out - Specifies the form which is split in four parts move away by rotating the parts when the transition takes place.
- Switch Left - Specifies that the form must go out of the view in 3D circular space along the Y-axis towards left as the content is being displayed.

- Switch Right - Specifies that the form must go out of the view in 3D circular space along the Y-axis towards right as the content is being displayed.
- Cloth - Specifies the present form must go out of screen animating as if a cloth is removed.
- Flip Right - Specifies that the form must be rotated along the Y-axis as the content is being displayed giving an illusion of a page is turned towards right in a book.
- Flip Left - Specifies that the form must be rotated along the Y-axis as the content is being displayed giving an illusion of a page is turned towards left in a book.
- Door - Specifies that the form must be revealed giving an illusion of a opening a door.
- Rotate Exchange - Specifies that the form must be rotated along the X-axis as the content is being displayed.

#### 7.3.5.2 Android Properties

**Transition Effect:** Specifies the effect from which the form is displayed. The available options are:

- None - The constant value is 0. The default device effect is applied or none of the effect is applied.
- Bottom Top - The constant value is 1. It specifies that the form must slide-in from the bottom and proceed towards the top.
- From Left - The constant value is 2. It specifies that the form must slide-in from the left with a fade effect.
- From Right - The constant value is 3. It specifies that the form must slide-in from the right with a fade effect.
- To Right - The constant value is 4. It specifies that the form must slide-out to the right with a fade effect.
- To Left - The constant value is 5. It specifies that the form must slide-out to the left with a fade effect.

- From Center - The constant value is 6. It specifies that the form must grow from the center with a fade effect.
- Top Right Bottom - The constant value is 7. It specifies that the form must slide-in from the top-right corner and proceed towards the bottom.
- Bottom Left Top - The constant value is 8. It specifies that the form must slide-in from the bottom-left corner and proceed towards the top.
- Bottom Top 2 - The constant value is 9. It specifies that the form must shrink from the bottom towards the top.
- Top Down - The constant value is 10. It specifies that the form must slide-in from the top and proceed towards the bottom.

#### 7.3.5.3 Windows Phone 8 Properties

**Transition Mode:** Specifies the direction from which the Form is displayed. The available options are:

- Default - The default device effect is applied or none of the effect is applied.
- Sequential- The transition of the Form going out of the view completes first and then the transition of the Form coming into the view takes place.
- Parallel- The transition of the Form going out of the view and the transition of the Form coming into the view takes place simultaneously.

**Transition Key** - Specifies the effect from which the Form is displayed. The available options are:

- Default - Specifies that the Form must slide horizontally into the view.
- Rotate 3D Single - Specifies that the Form must be rotated along the center Y-Axis when coming into the view.
- Rotate 3D Dual - Specifies that the Form must be shown making a circle around the screen from the background before coming into the view.
- Slide - Specifies that the Form must slide horizontally into the view.

- Pop - Specifies that the Form must emerge from center-bottom of the screen and gradually occupy the complete screen.
- Squeeze - Specifies that the Form must be expanded horizontally from an initial width of zero.

**Transition Speed** - Specifies the speed at which the Form is transitioned. The value must be specified in seconds.

#### 7.3.5.4 SPA Properties

**Transition Effect:** Specifies the effect from which the form is displayed. The available options are:

- None- Use this option if you do not want to specify a transition direction.
- Top Center - Specifies that the form must appear from the top center.
- Bottom Center - Specifies that the form must appear from the bottom center.
- Right Center - Specifies that the form must appear from the right center.
- Left Center - Specifies that the form must appear from the left center.

#### 7.3.6 Transition: OUT

Specifies the configuration to be used when the user arrives on this form. Click the **Edit** button against **Transition: OUT** field. From the **Transcation: OUT** window, choose the desired transition direction and transaction effect for specific platforms.

##### 7.3.6.1 iOS Properties

**Transition Direction:** Specifies the direction from which the Form is displayed. The available options are:

- None - Use this option if you do not want to specify a transition direction.
- From Right - Specifies that the Form must appear from the right.
- From Left - Specifies that the Form must appear from the left.

- From Bottom - Specifies that the Form must appear from the bottom.
- From Top - Specifies that the Form must appear from the top.

**Transition Effect:** Specifies the effect from which the form is displayed. The available options are:

- None - Use this option if you do not want to specify a transition direction.
- Fade - Specifies that the form must fade when it is transitioned to a hidden or an invisible state.
- MoveIn - Specifies that the form must slide over the existing content in the direction as specified in the transition direction.
- Push - Specifies that the form must push the existing content in the direction as specified in the transition direction and take its place.
- Flip - Specifies that the form must be rotated along the Y-axis as the content is being displayed. It supports transition directions from right and left.
- Reveal - Specifies that the form must be revealed gradually in the direction as specified in the transition direction.
- Curl - Specifies that the form must be curled or folded (look and feel is similar to turning of a page in a book) as the content is being displayed. It supports transition directions from top and bottom only.
- Two Split Horizontal In - Specifies the form which is split horizontally into two parts rejoins when the transition takes place.
- Two Split Vertical In - Specifies the form which is split vertically into two parts rejoins when the transition takes place.
- Four Split In - Specifies the form which is split in four parts rejoins when the transition takes place.
- Four Split Rotate In - Specifies the form which is split in four parts rejoins by rotating the parts when the transition takes place.

- Two Split Horizontal Out - Specifies the form which is split horizontally into two parts and move away when the transition takes place.
- Two Split Vertical Out - Specifies the form which is split vertically into two parts and move away when the transition takes place.
- Four Split Rotate Out - Specifies the form which is split in four parts move away by rotating the parts when the transition takes place.
- Switch Left - Specifies that the form must go out of the view in 3D circular space along the Y-axis towards left as the content is being displayed.
- Switch Right - Specifies that the form must go out of the view in 3D circular space along the Y-axis towards right as the content is being displayed.
- Cloth - Specifies the present form must go out of screen animating as if a cloth is removed.
- Flip Right - Specifies that the form must be rotated along the Y-axis as the content is being displayed giving an illusion of a page is turned towards right in a book.
- Flip Left - Specifies that the form must be rotated along the Y-axis as the content is being displayed giving an illusion of a page is turned towards left in a book.
- Door - Specifies that the form must be revealed giving an illusion of a opening a door.
- Rotate Exchange - Specifies that the form must be rotated along the X-axis as the content is being displayed.

#### 7.3.6.2 Android Properties

**Transition Effect:** Specifies the effect from which the form is displayed. The available options are:

- None - The constant value is 0. The default device effect is applied or none of the effect is applied.
- Bottom Top - The constant value is 1. It specifies that the form must slide-in from the bottom and proceed towards the top.

- From Left - The constant value is 2. It specifies that the form must slide-in from the left with a fade effect.
- From Right - The constant value is 3. It specifies that the form must slide-in from the right with a fade effect.
- To Right - The constant value is 4. It specifies that the form must slide-out to the right with a fade effect.
- To Left - The constant value is 5. It specifies that the form must slide-out to the left with a fade effect.
- From Center - The constant value is 6. It specifies that the form must grow from the center with a fade effect.
- Top Right Bottom - The constant value is 7. It specifies that the form must slide-in from the top-right corner and proceed towards the bottom.
- Bottom Left Top - The constant value is 8. It specifies that the form must slide-in from the bottom-left corner and proceed towards the top.
- Bottom Top 2 - The constant value is 9. It specifies that the form must shrink from the bottom towards the top.
- Top Down - The constant value is 10. It specifies that the form must slide-in from the top and proceed towards the bottom.

#### 7.3.6.3 Windows Phone 8 Properties

**Transition Mode:** Specifies the direction from which the Form is displayed. The available options are:

- Default - The default device effect is applied or none of the effect is applied.
- Sequential- The transition of the Form going out of the view completes first and then the transition of the Form coming into the view takes place.
- Parallel- The transition of the Form going out of the view and the transition of the Form coming into the view takes place simultaneously.

**Transition Key** - Specifies the effect from which the Form is displayed. The available options are:

- Default - Specifies that the Form must slide horizontally into the view.
- Rotate 3D Single - Specifies that the Form must be rotated along the center Y-Axis when coming into the view.
- Rotate 3D Dual - Specifies that the Form must be shown making a circle around the screen from the background before coming into the view.
- Slide - Specifies that the Form must slide horizontally into the view.
- Pop - Specifies that the Form must emerge from center-bottom of the screen and gradually occupy the complete screen.
- Squeeze - Specifies that the Form must be expanded horizontally from an initial width of zero.

**Transition Speed** - Specifies the speed at which the Form is transitioned. The value must be specified in seconds.

#### 7.3.6.4 SPA Properties

**Transition Effect**: Specifies the effect from which the form is displayed. The available options are:

- None- Use this option if you do not want to specify a transition direction.
- Top Center - Specifies that the form must appear from the top center.
- Bottom Center - Specifies that the form must appear from the bottom center.
- Right Center - Specifies that the form must appear from the right center.
- Left Center - Specifies that the form must appear from the left center.

#### 7.3.7 Title

Specifies the title of the form and is displayed in the Title Bar.

To update Title property, refer to [Title Bar](#).

### 7.3.8 Retain Scroll Pos

Specifies if the Form must remember the scroll position when the user revisits the Form.

### 7.3.9 Title Bar

Specifies whether the title bar must be displayed on a form.

- Click **On** to display title bar.
- Click **Off** to hide title bar.

The selected option becomes the default for all the platforms.

If you wanted to choose a different **Title Bar** option for a specific platform, click the ellipses (...) button next to the **Title Bar Off**. The **TitleBar** dialog box appears.

Select a platform and from the corresponding **Value** field , select the check box to enable or disable displaying the title bar.

### 7.3.10 Header Overlap

Specifies if the header must overlap the form. For example, every time you scroll the form, the header is fixed and the header overlaps the form as specified in the header overlap field. If this field is selected, the form scrolls behind the header background and a part of the header background is transparent.

- Click **On** to allow header overlap the form.
- Click **Off** to disallow header overlap the form.

If you wanted to choose a different **Header Overlap** option for a specific platform, click the ellipses (...) button next to the **Header Overlap Off**. The **Header Overlap** dialog box appears.

Select a platform and from the corresponding **Value** field , select the check box to allow or disallow header overlap.

### 7.3.11 Footer Overlap

Specifies if the footer must overlap the form. For example, every time you scroll the form, the footer is fixed and the footer overlaps the form as specified in the Footer Overlap field. If this field is selected, the form scrolls behind the footer background and a part of the footer background is transparent.

- Click **On** to allow footer overlap the form.
- Click **Off** to disallow footer overlap the form.

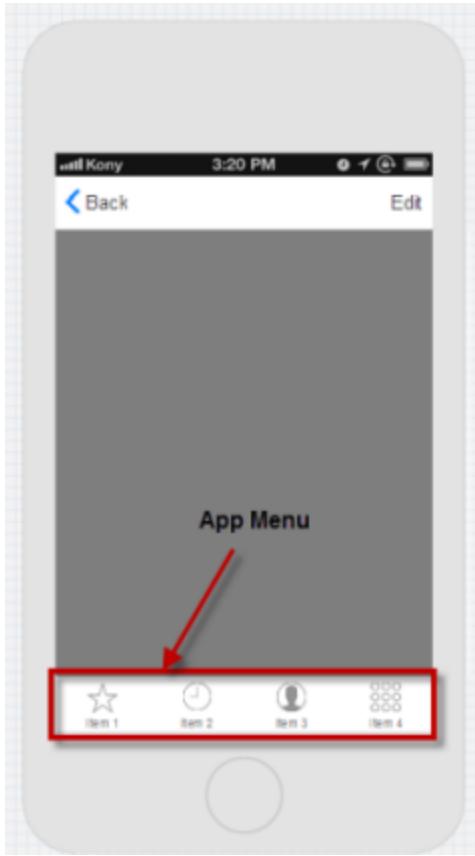
If you wanted to choose a different **Footer Overlap** option for a specific platform, click the ellipses (...) button next to the **Footer Overlap Off**. The **Footer Overlap** dialog box appears.

Select a platform and from the corresponding **Value** field , select the check box to allow or disallow footer overlap.

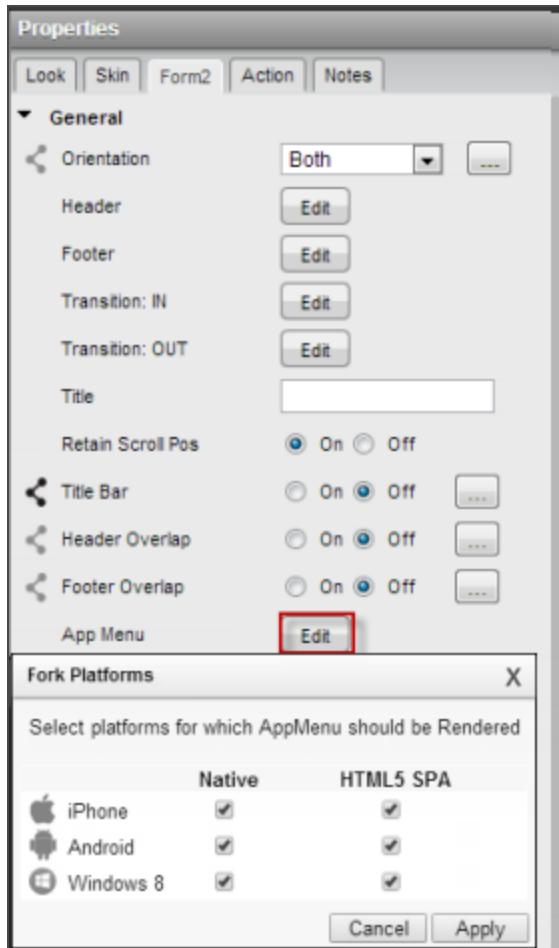
### 7.3.12 App Menu

The Application Menu (App Menu) contains features that apply to the application as a whole, rather than to a specific application screen or window.

You can choose to display App Menu on every form or on specific forms of the application.

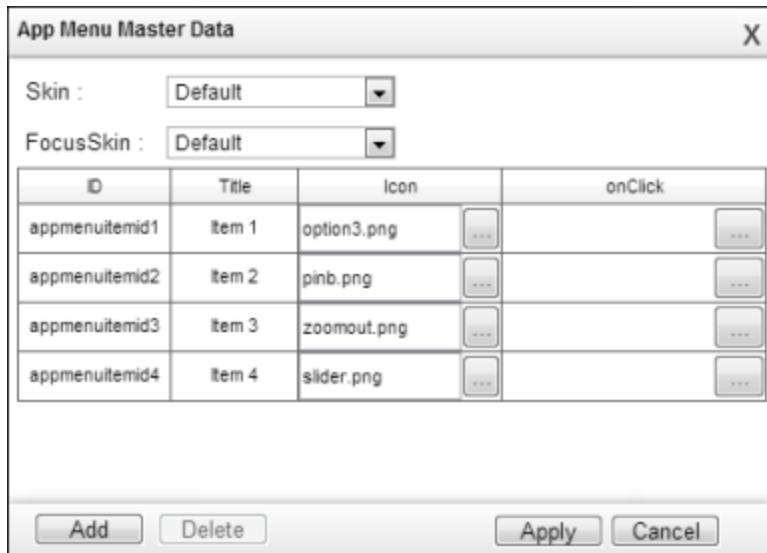


By default, App Menu is viable on all the platforms. To hide App Menu for a specific platforms, click the **Edit** button next to the **App Menu**. The **Fork Platforms** dialog box appears. Clear a platform check box on which you wish to hide App Menu.



To customize App Menu items, follow below steps:

1. Click inside the App Menu in App Canvas to display a right arrow.
2. Click the right arrow, and then click Action: **Edit Master Data**. The App Menu Master Data dialog box appears.



3. You can customize the following properties:
  - a. Skin: From the list of available skins, select a required skin.
  - b. FocusSkin: From the list of available skins, select a required skin.

From within the table:

  - c. ID: Unique identifier for the menu item.
  - d. Title: name of the menu item
  - e. Icon: An icon is displayed above the Title. To assign an icon, click the ellipsis (...) button in the cell, and then select an icon.
  - f. onClick: You can assign onClick event by clicking the ellipsis (...) button in the cell, and creating an event in the **Action Editor** dialog box.

### 7.3.13 Menu Position

Specifies if the menu is shown or hidden in the application. In some platforms, form menu items appear along with app menu items. This property allows developer to configure whether to append at the end of application menu list or inserted at beginning of the application menus.

Following are the available options:

- After App Menu
- Before App Menu

The selected menu position here becomes the default for all the platforms.

For specifying a platform specific menu position, click ellipsis (...) button next to the **Menu Postion** list to open **Menu Postion** dialog box.

Select the platform, and from its corresponding **Value** list, choose a desired menu position.

### 7.3.14 Layout Type

Specifies if the arrangement of the widgets either in free form or horizontal or vertical direction.

The following are the layout types of flex forms:

- Free Form: Enables you to place the widgets in any order. That is, you can align the widgets horizontally or vertically.
- Flow Vertically: Enables you to align the widgets vertically.

### 7.3.15 Enable Scrolling

Specifies whether the scrolling is enabled on the container or not.

- To enable scrolling, click **On**.
- To disable scrolling, click **Off**.

### 7.3.16 Bounces

Specifies whether the scroll bounce is enabled or disabled.

- To enable scroll bounce, click **On**.
- To disable scroll bounce, click **Off**.

### 7.3.17 Horizontal Bounce

Specifies whether the scroll bounce is enabled or disabled in the horizontal direction.

- To enable horizontal scroll bounce, click **On**.
- To disable horizontal scroll bounce, click **Off**.

### 7.3.18 Vertical Bounce

Specifies whether the scroll bounce is enabled or disabled in the vertical direction.

- To enable vertical scroll bounce, click **On**.
- To disable vertical scroll bounce, click **Off**.

### 7.3.19 Paging

Specifies the whether the paging is enabled for the container. If this property is set to true, the scroll view stops on multiples of the scroll view's bounds when the user scrolls.

### 7.3.20 Content Offset X

Specifies the x coordinate of the top-left of the scrollable region. When the value is set, the container scrolls even if the scrolling is disabled.

### 7.3.21 Content Offset Y

Specifies the y coordinate of the top-left of the scrollable region. When the value is set, the container scrolls even if the scrolling is disabled.

### 7.3.22 Content Size Width

Specifies the width of the container to accommodate all the widgets placed on a form.

### 7.3.23 Content Size Height

Specifies the height of the container to accommodate all the widgets placed on a form.

### 7.3.24 Snap to Grid

A widget aligns to the nearest intersection of lines in the grid, or other widgets.

- Click **On** to allow snapping the widgets to a grid.
- Click **Off** to disable snapping the widgets to a grid.

### 7.3.25 Snap Grid Size

Specifies the value of the grid size. This option is available only when you enable **Snap to Grid** property.

### 7.3.26 Default unit

Specifies the default unit to be used for interpretation of numbers with no qualifiers when passed to layout properties.

Following are the available options:

- dp: Specifies the values in terms of device independent pixels.
- px: Specifies the values in terms of device hardware pixels.
- %: Specifies the values in percentage relative to the parent dimensions.

**Note:** The default unit assigned for a form becomes the default unit for all the flex widgets except for the widgets FlexContainer and FlexScrollContainer.

### 7.3.27 Input View Type

When building iPhone applications that support or provide text input, it's often necessary to create some extra buttons (or other controls) beyond the ones provided by the default keyboard interface. Kony Platform by default, adds the Previous, Next and Done buttons to the applicable input controls. These buttons allow specific operations needed by your application, such as moving to the next or previous text field, make the keyboard disappear. The area above the keyboard is known as Input Accessory View.

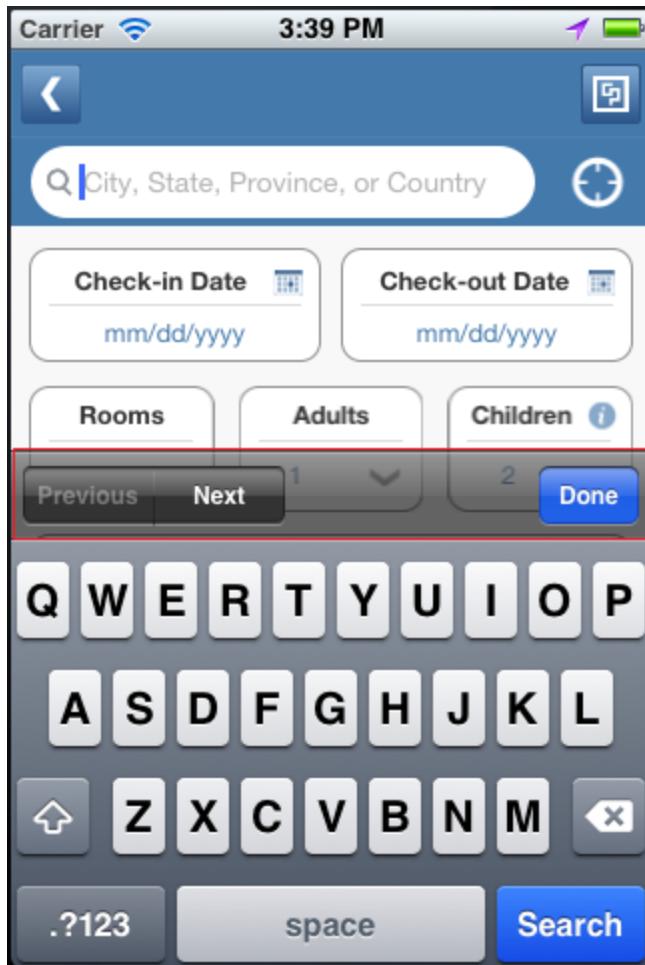
This property, allows you to specify the type of accessory view that will be shown for all the input controls on this form.



Following are the available options:

- None: Use this option if you do not want to specify the toolbar. This option should be used carefully, as setting this option for widgets like calendar leaves the user with no option to select and drop-down a wheel calendar.
- Default: Specifies that the toolbar that is defined in the Application level settings. To set the Application level settings, right-click on the project and navigate to Properties > Native App > iPhone / iPad.
- Next / Previous: Specifies the navigation options as Next, Previous, and Done for a form. The

below image illustrates the **next prev toolbar** set for a **Textbox**. The highlighted toolbar is achieved by setting the **Keyboard Type** as **Default** for a Textbox and **Input Accessory View Type** as **nextprevtoolbar** to the Form.



- Cancel: Specifies that the input accessory view has a cancel button. This option does not trigger any events.

**Note:** This property is specific to the iOS platform.

### 7.3.28 Indicator During Post-Show

This property specifies if there must be a progress indicator to show that the form content is being loaded. You can use this property for forms that require network calls during post-show.

- Click **On** to display the progress indicator.
- Click **Off** to hide the progress indicator.

The following image illustrates the loading indicator:



**Note:** This property is specific to the iOS platform.

### 7.3.29 Transparency During Post-Show

Normally, when a form is loading, the user interface is blocked. You can use this property to specify the percentage of transparency.

Default: 100

**Note:** This property is specific to the iOS platform.

### 7.3.30 Scroll Bounces

Specifies whether the scroll view bounces past the edge of the content and back again.

- Click **On** to allow the scroll view bounce..
- Click **Off** to disable the scroll view bounce.

**Note:** This property is specific to the iOS platform.

### 7.3.31 Hide Status Bar

Specifies the status of the status bar.

Following are the available options:

- ApplyApp Settings
- True
- False

### 7.3.32 Configure Extend Top

- Click **On** to display the **Extend Top** property.
- Click **Off** to hide the **Extend Top** property.

**Note:** This property is specific to the iOS platform.

### 7.3.33 Configure Extend Bottom

- Click **On** to display the **Extend Bottom** property.
- Click **Off** to hide the **Extend Bottom** property.

**Note:** This property is specific to the iOS platform.

### 7.3.34 Configure Status Bar Style

- Click **On** to display the **Status Bar Style** property.
- Click **Off** to hide the **Status Bar Style** property.

**Note:** This property is specific to the iOS platform.

### 7.3.35 Extend Top

Specifies the form content to scroll under the App Menu. This property is supported in iOS7 and above only. This property is also applicable on the Application Level. The property set at form level takes precedence over Application level. If **On**option is selected, the form scroll under the App Menu.

**Note:** This property is specific to the iOS later platforms.

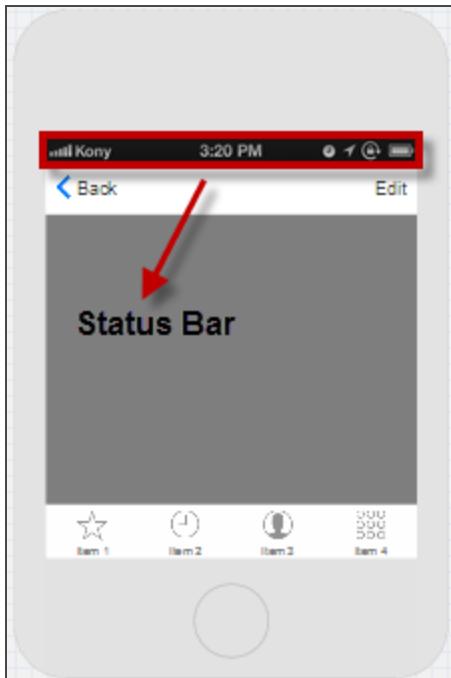
### 7.3.36 Extend Bottom

This property enables you to Configure Extend Bottom property. This property is supported on iOS7 and above only. This property is also applicable on the Application Level. The property set at form level takes precedence over Application level. If **On**option is selected, the property extendBottom is displayed.

**Note:** This property is specific to the iOS platform.

### 7.3.37 Status Bar Style

This property enables you to set the status bar style.



You can select one of the following styles for your status bar:

- Default: The status bar contents are displayed in black color suitable for a light background.
- Light Content: The status bar contents are displayed in white color suitable for a dark background.

**Note:** This property is specific to the iOS7 platform.

### 7.3.38 Zoom Scale

Specifies the current scale factor applied to the form content.

Default: 1

### 7.3.39 Min Zoom Scale

Specifies the minimum zoom scale factor that can be applied to the form.

Default: 1

### 7.3.40 Max Zoom Scale

Specifies the maximum zoom scale factor that can be applied to the form.

Default: 1

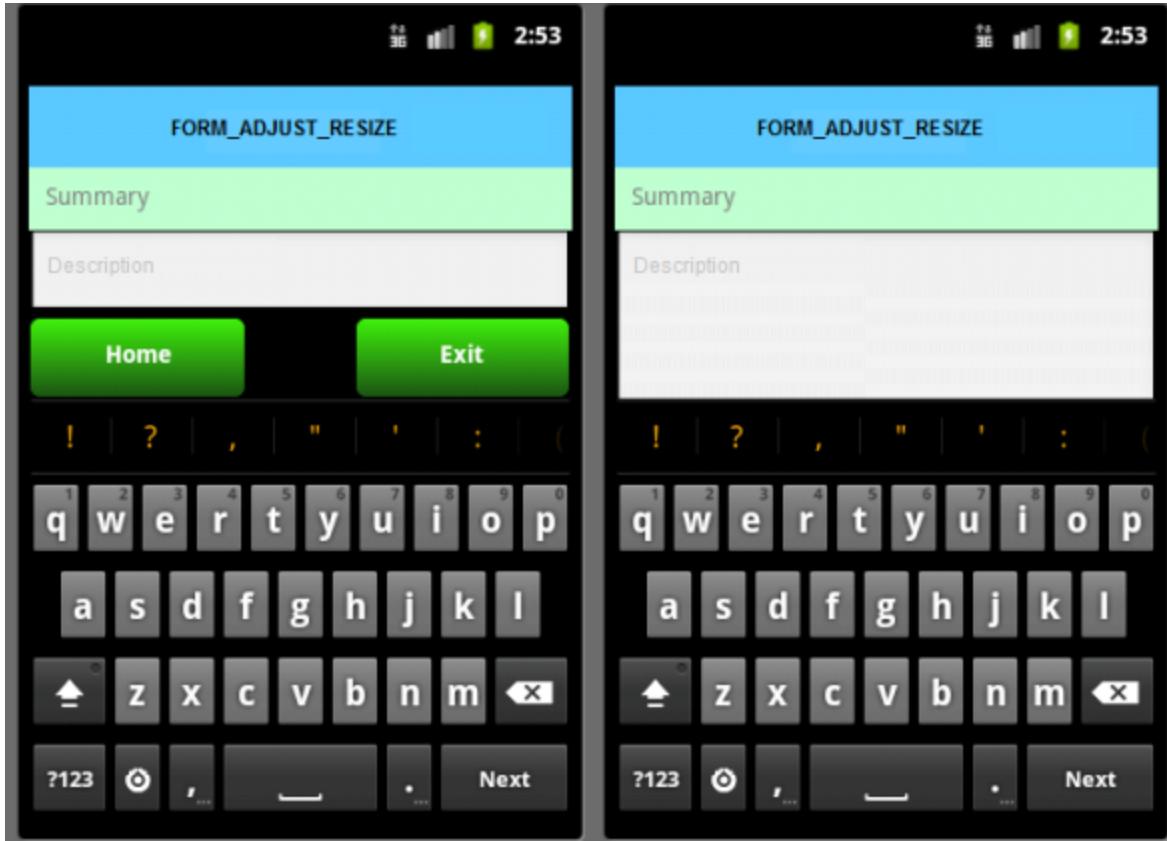
### 7.3.41 Soft Input Mode

Specifies the soft keyboard for a widget that requires an input should be placed at the bottom of the form. For example, when a TextBox or a TextArea requires input from the user, the soft-keyboard pops up when you enter text.

Following are the available options:

- Adjust - Resize: Specifies the form is resized and when you click or start typing within the widget which requires an input, the form scrolls up and the widget which requires an input is not overlapped by the keypad or footer.
- Adjust - Pan: Specifies the widget which requires an input is placed at the bottom of the form is overlapped by the keypad.

The below image illustrates the above two options:



**Note:** This property is specific to the Android platform.

#### 7.3.42 Max App Menu Buttons

Specifies the number of app menu buttons that should be displayed on the screen.

**Note:** This property is specific to the Windows 8 platform.

#### 7.3.43 Animate Header Footer

Specifies if the headers and footers of the Form must slide up and down respectively away from the view when the Form is in transition. To enable animation, select the **On** option next to the **Animate Header Footer**.

**Note:** This property is specific to the Windows 8 platform.

## 7.4 Actions

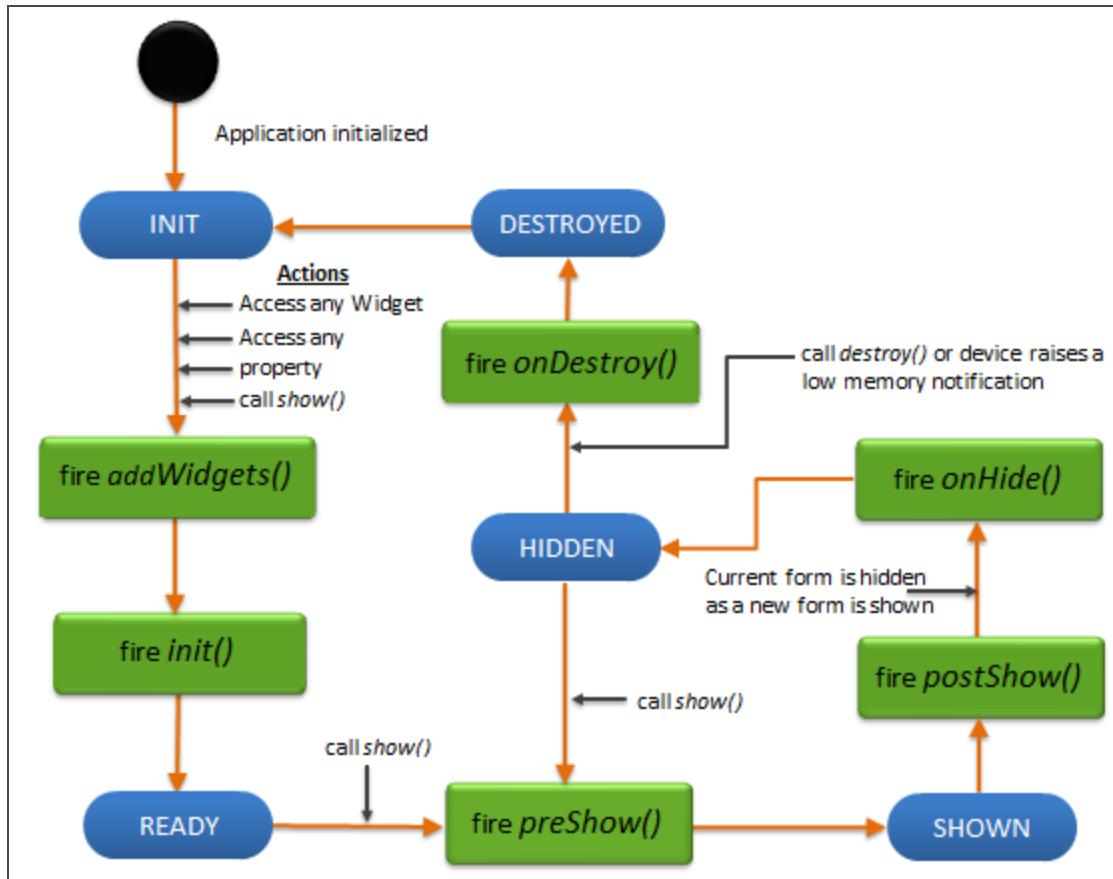
Form has the following actions associated with it:

- **init** - called immediately after an addWidgets event for any initializations required for the form.  
Init initializes the form and any widgets.

In case of Server side Mobile Web and SPA, addWidgets and init events gets called as soon as the form is created. In case of native platforms, as an optimization, these events are deferred until the first access.

- **preShow** - called just before a form is visible on the screen. A form can be made visible by explicitly calling the show method of the form.
- **postShow** - called immediately after the form is visible on the screen. A form is made visible by explicitly calling the show method of the form.
- **onHide** - called when the form goes out of the screen. A form can go out of the screen when another form is to be shown.
- **onDestroy** - called when a form is destroyed. A form is destroyed when the developer explicitly calls destroy and this event gets called before destroying the form.

The following image illustrates the lifecycle of a form:



- **onScrollStart**: An event callback is invoked by the platform when the user starts scrolling the content. This event is invoked asynchronously.
- **onScrollTouchReleased**: An event callback is invoked by the platform when the user touch is released from the touch surface. This event is invoked asynchronously.
- **onScrolling**: An event callback is invoked by the platform when the scrolling is in progress. This event is invoked asynchronously.
- **onDecelerationStarted**: An event callback is invoked by the platform when the user stops scrolling but the content still moves before the content actually stop.
- **onScrollEnd**: An event callback is invoked by the platform when the scrolling is ended. This event is invoked asynchronously.

- **onOrientationChange:** Specifies an Event which is triggered when there is a change in orientation of the form from portrait to landscape or vice versa.

**Note:** This property is specific to the iOS platform.

- **widgetToZoom:** An event callback is invoked by the platform to return one of the child widgets of source to zoom. The returning source itself may not result in zooming the entire source. The container will not zoom, if a null value is returned. This event is invoked asynchronously.

**Note:** This property is specific to the iOS platform.

- **onZoomStart:** An event callback is invoked by the platform when the container is about to zoom. This event is invoked asynchronously.

**Note:** This property is specific to the iOS platform.

- **onZooming:** An event callback is invoked by the platform when the container is zooming. This event is invoked asynchronously.

**Note:** This property is specific to the iOS platform.

- **onZoomEnd:** An event callback is invoked by the platform when the zooming has ended. This event is invoked asynchronously.

**Note:** This property is specific to the iOS platform.

- **onDeviceBack:** Specifies an event which is triggered when the user uses the back button on the device.

**Note:** This property is specific to the Android platform.

- **onDeviceMenu:** Specifies an event which is triggered when the user uses the back button on

the device.

**Note:** This property is specific to the Android platform.

- **onOrientationChange:** Specifies an event which is triggered when there is a change in orientation of the form from portrait to landscape or vice versa.

**Note:** This property is specific to the Android platform.

- **onDeviceBack:** Specifies an event which is triggered when the user uses the back button on the device.

**Note:** This property is specific to the Windows 8 platform.

- **onOrientationChange:** Specifies an event which is triggered when there is a change in orientation of the form from portrait to landscape or vice versa.

**Note:** This property is specific to the Windows platform.

- **onDeviceBack:** Specifies an event which is triggered when the user uses the back button on the device.

**Note:** This property is specific to the SPA platform.

- **onOrientationChange:** Specifies an event which is triggered when there is a change in orientation of the form from portrait to landscape or vice versa.

**Note:** This property is specific to the SPA platform.

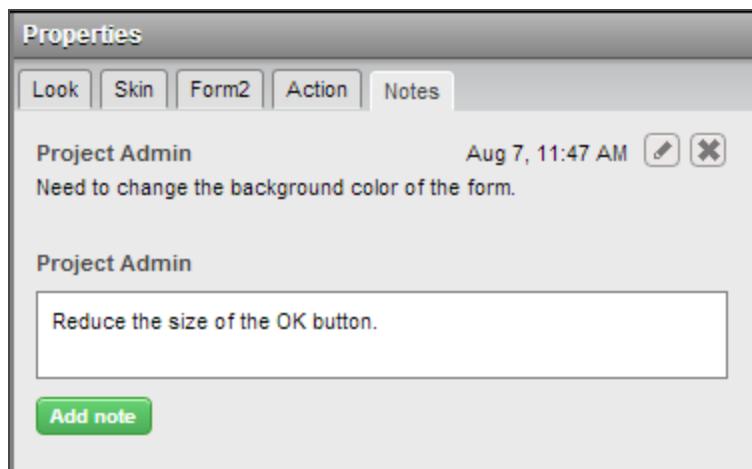
For more information on using the above action, refer to section [Action Editor](#).

## 7.5 Notes

**Notes** allow you to provide comments, feedback and suggestions about a form (or page). These notes are visible to all the users during function preview (cloud).

### 7.5.1 Adding a Note

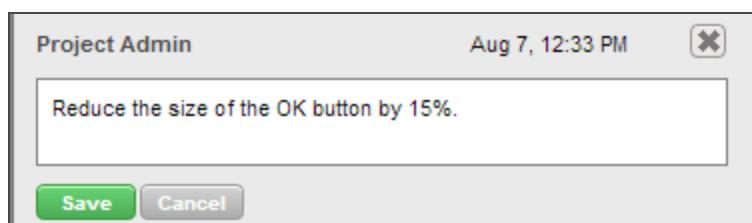
To add a note, follow these steps:



1. Navigate to the required form, and from its **Properties** tab, click **Notes**.
2. Type your note in the box.
3. Click **Add Notes**.

### 7.5.2 Editing a note

To edit a note, follow these steps:



1. Navigate to the required form, and from its **Properties** tab, click **Notes**.
2. Click  (edit icon) of a note.
3. Make necessary changes to the note.
4. Click **Save**.

**Note:** Only the author of a note can edit it.

### 7.5.3 Delete a Note

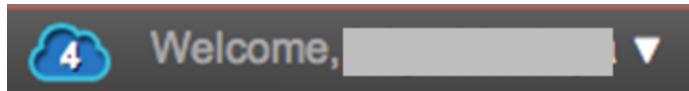
To delete a note, follow these steps:

1. Navigate to the required form, and from its **Properties** tab, click **Notes**.
2. Click the delete icon of a note.

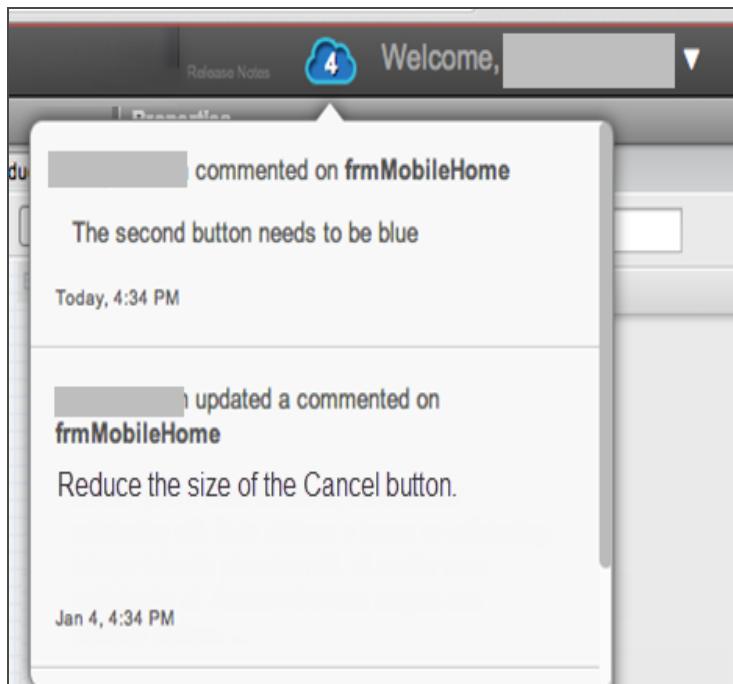
**Note:** Only the author of a note can delete the note.

### 7.5.4 Viewing Functional Preview Notes

When a note is added during [Functional Preview \(Cloud\)](#) they will be available to you in your Kony Visualizer project. The cloud icon in your project will change to



Click the icon to view the added notes:



These notes are downloaded to your local project and are available in your form Notes tab.

A screenshot of the 'Properties' panel in Kony Visualizer. The 'Notes' tab is selected. The panel displays a list of notes from different users. The notes are as follows:

- Project Admin, Aug 7, 4:57 PM: Need to change the background color of the form.
- Project Admin!, Aug 7, 4:57 PM: Reduce the size of the OK button by 15%
- [Redacted], Aug 7, 5:08 PM: Looks nice.
- [Redacted], Aug 7, 5:35 PM: Please change the Login text font.

At the bottom of the panel, there is a text input field and a green 'Add note' button.

## 8. Flex Container Rules

Widgets placed on a Flex Container adhere to some rules for width, height, and positioning of the widget with respect to other widgets and parent container.

- [Height and Width Rules](#)
- [Preferred Width and Preferred Height Rules](#)
- [Widget Layout Properties - Priority Rules](#)

### 8.0.1 Height and Width Rules

Height and width of a widget is determined by the explicit values specified to width and height properties. There are cases where you will not be able to give an explicit width and height, due to dynamic nature of the application.

The properties Min Height, Max Height, Min Width, and Max Width of the widget are used to determine the width and height of the widgets.

Below are some of the cases that explain the behavior of the widget with respect to the properties specified:

| Condition                         | Behavior   |
|-----------------------------------|--|
| Width of the widget is specified  | The specified width is taken for layout purpose. Specifying the width explicitly will make Min Width and Max Width ineffective.            |
| Height of the widget is specified | The specified height is taken for layout purpose. Specifying the height explicitly will make Min Height and Max Height values ineffective. |

| Condition  | Behavior   |
|--|--|
| Width is undefined and Min Width and Max Width are specified     | <ul style="list-style-type: none"><li>The Min Width and Max Width values are used to determine the width of the widget.</li><li>Widget is asked for its preferred width and the final width of widget is calculated based on Min Width and Max Width specified and preferred width of the widget.</li><li>Always the equation <code>Min Width &lt;= Calculated Width &lt;= Max Width</code> is maintained.</li></ul>             |
| Height is undefined, and Min Height and Max Height are specified | <ul style="list-style-type: none"><li>The Min Height and Max Height values are used to determine the height of the widget.</li><li>Widget is asked for its preferred height and the final height of the widget is calculated based on Min Height and Max Height values, and preferred height of the widget.</li><li>Always the equation <code>Min Height &lt;= calculatedHeight &lt;= Max Height</code> is maintained.</li></ul> |
| Width and Min Width are undefined, and Max Width is specified    | <ul style="list-style-type: none"><li>The Max Width vale is used to determine the width of the widget.</li><li>Widget is asked for its preferred width, and the final width of the widget is calculated based on Max Width specified and preferred width of the widget.</li><li>Always the equation <code>0 &lt;= Calculated Width &lt;= Max Width</code> is maintained.</li></ul>   |

| Condition  | Behavior  |
|--|---|
| Width and Max Width are undefined, and Min Width is specified    | <ul style="list-style-type: none"> <li>The Min Width value is used to determine the width of the widget.</li> <li>Widget is asked for its preferred width, and the final width of the widget is calculated based on Min Width specified and preferred width of the widget.</li> <li>Always the equation <math>\text{Min Width} \leq \text{Calculated Width} \leq \text{Preferred Width}</math> is maintained.</li> </ul>          |
| Height and Min Height are undefined, and Max Height is specified | <ul style="list-style-type: none"> <li>The Max Height value is used to determine the height of the widget.</li> <li>Widget is asked for its preferred height, and the final height of the widget is calculated based on Max Height specified and preferred height of the widget.</li> <li>Always the equation <math>0 \leq \text{Calculated Height} \leq \text{Max Height}</math> is maintained.</li> </ul>                       |
| Height and Max Height are undefined, and Min Height is specified | <ul style="list-style-type: none"> <li>The Min Height value is used to determine the height of the widget.</li> <li>Widget is asked for its preferred height, and the final height of the widget is calculated based on Min Height specified and preferred height of the widget.</li> <li>Always the equation <math>\text{Min Height} \leq \text{Calculated Height} \leq \text{Preferred Height}</math> is maintained.</li> </ul> |
| Min Width > Max Width  | Min Width specified is ignored. For width calculations only Max Width is considered assuming Min Width as zero.   |
| Min Height > Max Height  | Min Height specified is ignored. For height calculations only Max Height is considered assuming Min Height as zero.   |

## 8.0.2 Preferred Width and Preferred Height Rules

The preferred width and preferred height will vary per widget from platform to platform. Widgets are categorized with respect to their preferred width and preferred height calculations:

| Content Driven Widgets | Fixed Width in Individual Platforms | Greedy Widgets         |
|------------------------|-------------------------------------|------------------------|
| Label                  | Switch                              | Segment2               |
| Button                 | TextArea                            | TabPane                |
| Image                  | TextBox                             | Horizontal Image Strip |
| Calendar               | Slider                              | Image Gallery          |
| CheckBox Group         | Line                                | DataGrid               |
| ListBox                |                                     | FlexContainer          |
| Link                   |                                     | FlexScrollView         |
| ComboBox               |                                     | Browser                |
| RadioButton Group      |                                     | Map                    |
| RichText               |                                     |                        |
| Camera                 |                                     |                        |
| PickerView             |                                     |                        |
| Phone                  |                                     |                        |

**Note:** Padding is considered as part of the content when deciding the Preferred Width and Preferred Height for the content driven widgets.

### 8.0.3 Widget Layout Properties - Priority Rules

For Flex Layout it is not necessary to set all the layout properties except that determine the position and dimension of the widget without any ambiguity.

| Positioning            | Priority Rules  |
|------------------------|---|
| Horizontal positioning | <ul style="list-style-type: none"><li>• Center X property has more priority over left.</li><li>• Left property has more priority over right.</li><li>• When none of the horizontal properties are defined, widget is positioned such a way that left of the widget is placed at the left boundary of the parent.</li></ul>  |
| Vertical positioning   | <ul style="list-style-type: none"><li>• Center Y property has more priority over top.</li><li>• Top property has more priority over bottom.</li><li>• When none of the vertical properties are defined, widget is positioned such a way that top of the widget is placed at the top boundary of the parent.</li></ul>   |
| Width                  | <ul style="list-style-type: none"><li>• Width property has more priority over than implicit width calculations.</li><li>• Implicit width calculations when width is not specified, will rely on Min Width and Max Width values.</li><li>• Implicit width calculations when width related properties are not specified, will rely on positioning properties.<ul style="list-style-type: none"><li>■ In the order of Center X, left, and right properties are used to determine width.</li></ul></li><li>• When implicit calculated values for width results in negative then values will be defaulted to zero.</li></ul> |

| Positioning | Priority Rules  |
|-------------|---|
| Height      | <ul style="list-style-type: none"><li>• Height property has more priority over than implicit height calculations.</li><li>• Implicit height calculations when height is not specified, will rely on Min Height and Max Height values.</li><li>• Implicit height calculations when height related properties are not specified, will rely on positioning properties.<ul style="list-style-type: none"><li>■ In the order of Center Y, top, and bottom properties are used to determine height.</li></ul></li><li>• When implicit calculated values for height results in negative then values will be defaulted to zero.</li></ul> |
|             |   |

**Additional information:**

- When any of the width or height related parameters given then width will be determined based on the [width and height rules](#).
- If none of the width or height related properties are specified then layout manager as a second priority does implicit calculations and determines the width and height based on the positional properties provided as per the rules laid out in the above table.
- When implicit calculated values for width and height results in negative then values will be defaulted to zero.
- There will not be any implicit calculations to determine the width and height happen in case of Flow Horizontal or Flow Vertical layout types.
- If positional properties are not good enough to perform implicit calculations then layout manager purely goes by widgets preferred dimensions (preferred width and preferred height).

- The properties like Screen Level Widget, Container Height, and Container Height Reference are not supported for the widgets placed inside flex containers.

## 9. Popup

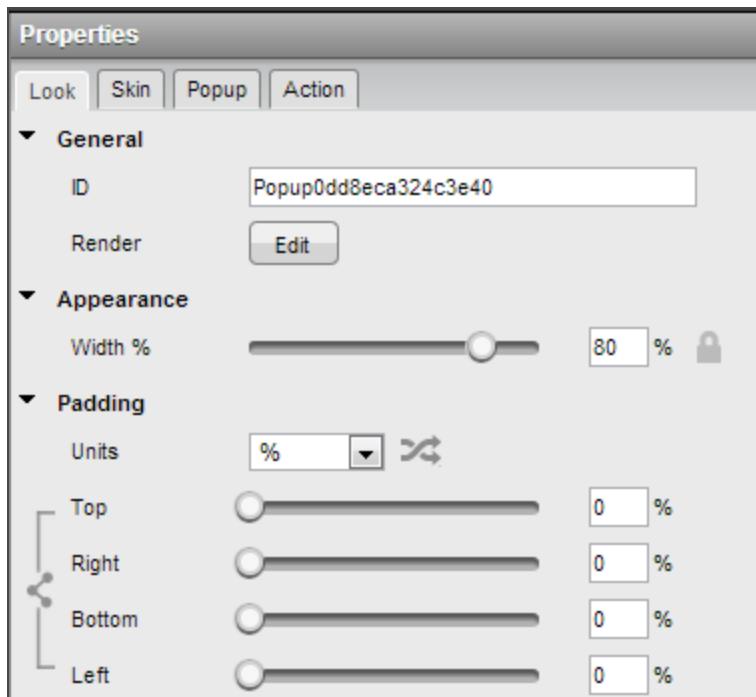
A pop-up usually appears in the center of the screen on top of the form from which you have invoked the pop-up. It does not span the entire screen width. A pop-up allows you to partition UI design into smaller parts.

### 9.1 Popup Properties

You can update a pop-up properties such as look and feel, provide platform specific properties, and attach various actions events.

### 9.2 Look

Following properties can be modified from this tab.



1. **General > ID:** It is the unique identifier of the pop-up. When a pop-up is added , system assigns a unique ID (or name). You can rename the pop-up ID, if required.

2. **Render:** Defines whether a pop-up should be displayed on a specific platform. By default, a pop-up is rendered for all the platforms. If you do not want to render a pop-up for a specific platform, click the **Edit** button against the **Render** field to open the **Fork Platforms** dialog box. Clear the platform check box for which you do not want to render the pop-up.
3. **Appearance - Width%:** Specifies the width of a widget. You can lock the percentage of the width occupied by a widget within a container.
4. **Padding:** Defines the space between the content of the popup and the popup boundaries. Following padding options are available for the popup.

| Property | Definition     | Action  |
|----------|----------------|---|
| Top      | Top padding    | Move the slider to adjust the top padding of the widget.    |
| Bottom   | Bottom padding | Move the slider to adjust the bottom padding of the widget. |
| Left     | Left padding   | Move the slider to adjust the left padding of the widget.   |
| Right    | Right padding  | Move the slider to adjust the right padding of the widget.  |

### 9.3 Skin

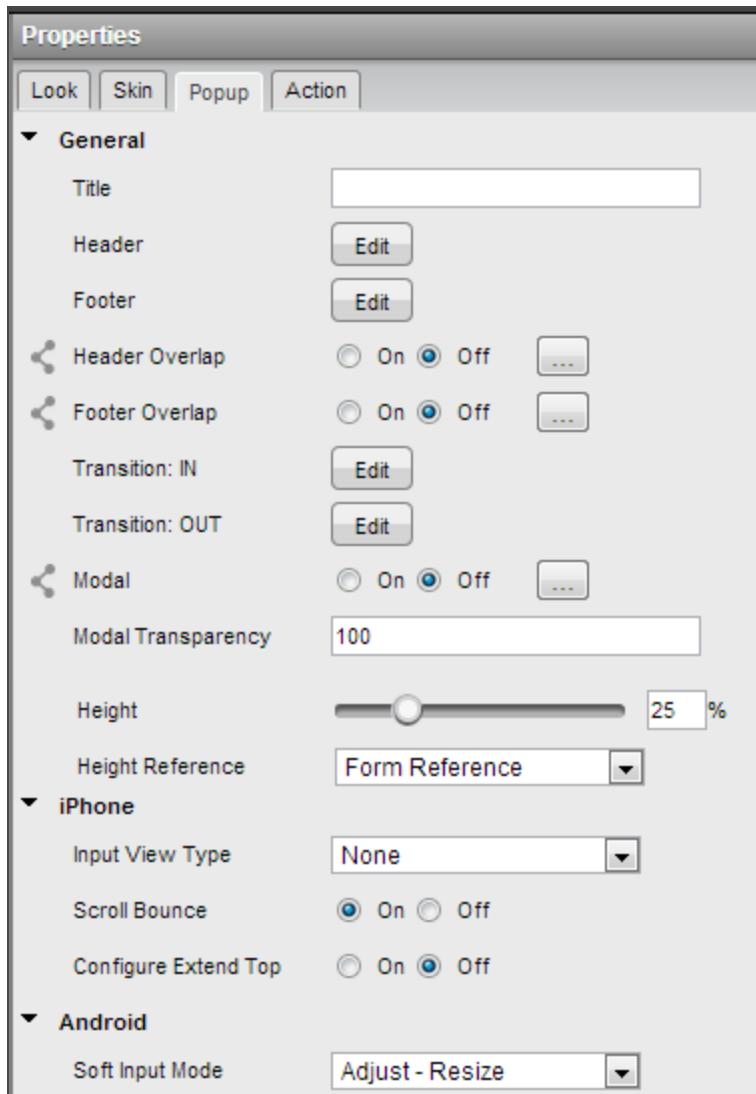
Skin define properties such as background color, borders and shadows for a popup.

For a popup widget, you can apply a skin for the following states:

- Normal: Specifies a background skin for popup widget.

For more details on skin properties, refer to section: [Widget - Skins](#).

## 9.4 Platform Specific properties



### 9.4.1 Title

Specifies the name of the popup.

#### 9.4.2 Header

For inserting a header to a popup, click the **Edit** button against the **Headers** field. From the **Headers** window, you can select a header.

**Note:** A header template should be created for a project before inserting the Header into a popup. Refer to section [Headers](#) for steps to create a Header.

#### 9.4.3 Footer

For inserting a footer to a popup, click the **Edit** button against the **Footers** field. From the **Footers** window, you can select a footer.

**Note:** A footer template should be created for a project before inserting the footer into a popup. Refer to section [Footer](#) for steps to create a footer.

#### 9.4.4 Header Overlap

Specifies if the header must overlap the popup. For example, every time you scroll the popup, the header is fixed and the header overlaps the popup as specified in the header overlap field. If this field is selected, the popup scrolls behind the header background and a part of the header background is transparent.

- Click **On** to allow header overlap the popup.
- Click **Off** to disallow header overlap the popup.

If you wanted to choose a different **Header Overlap** option for a specific platform, click the ellipses (...) button next to the **Header Overlap Off** option to open **Header Overlap** dialog box.

Select a platform and from the corresponding **value** field , select the check box to allow or disallow header overlap.

#### 9.4.5 Footer Overlap

Specifies if the footer must overlap the popup. For example, every time you scroll the pop-up, the footer is fixed and the footer overlaps the popup as specified in the Footer Overlap field. If this field is selected, the popup scrolls behind the footer background and a part of the footer background is transparent.

- Click **On** option to allow footer overlap the popup.
- Click **Off** option to disallow footer overlap the popup.

If you wanted to choose a different **Footer Overlap** option for a specific platform, click the ellipses (...) button next to the **Footer Overlap Off** option to open **Footer Overlap** dialog box.

Select a platform and from the corresponding **value** field , select the check box to allow or disallow footer overlap.

#### 9.4.6 Transition: IN

Specifies the configuration to be used when the user arrives on this pop-up. Click the **Edit** button against **Transition: IN**. The **Transition: IN** dialog appears. Choose the desired transition direction and transaction effect for specific platforms.

##### 9.4.6.1 iOS Properties

**Transition Direction:** Specifies the direction from which the pop-up is displayed. The available options are:

- None - Use this option if you do not want to specify a transition direction.
- From Right - Specifies that the pop-up must appear from the right.
- From Left - Specifies that the pop-up must appear from the left.
- From Bottom - Specifies that the pop-up must appear from the bottom.
- From Top - Specifies that the pop-up must appear from the top.

**Transition Effect:** Specifies the effect from which the pop-up is displayed. The available options are:

- None - Use this option if you do not want to specify a transition direction.
- Fade - Specifies that the pop-up must fade when it is transitioned to a hidden or an invisible state.
- MoveIn - Specifies that the pop-up must slide over the existing content in the direction as specified in the transition direction.
- Push - Specifies that the pop-up must push the existing content in the direction as specified in the transition direction and take its place.
- Flip - Specifies that the pop-up must be rotated along the Y-axis as the content is being displayed. It supports transition directions from right and left.
- Reveal - Specifies that the pop-up must be revealed gradually in the direction as specified in the transition direction.
- Curl - Specifies that the pop-up must be curled or folded (look and feel is similar to turning of a page in a book) as the content is being displayed. It supports transition directions from top and bottom only.
- Two Split Horizontal In - Specifies the pop-up which is split horizontally into two parts rejoins when the transition takes place.
- Two Split Vertical In - Specifies the pop-up which is split vertically into two parts rejoins when the transition takes place.
- Four Split In - Specifies the pop-up which is split in four parts rejoins when the transition takes place.
- Four Split Rotate In - Specifies the pop-up which is split in four parts rejoins by rotating the parts when the transition takes place.
- Two Split Horizontal Out - Specifies the pop-up which is split horizontally into two parts and move away when the transition takes place.

- Two Split Vertical Out - Specifies the pop-up which is split vertically into two parts and move away when the transition takes place.
- Four Split Rotate Out - Specifies the pop-up which is split in four parts move away by rotating the parts when the transition takes place.
- Switch Left - Specifies that the pop-up must go out of the view in 3D circular space along the Y-axis towards left as the content is being displayed.
- Switch Right - Specifies that the pop-up must go out of the view in 3D circular space along the Y-axis towards right as the content is being displayed.
- Cloth - Specifies the present pop-up must go out of screen animating as if a cloth is removed.
- Flip Right - Specifies that the pop-up must be rotated along the Y-axis as the content is being displayed giving an illusion of a page is turned towards right in a book.
- Flip Left - Specifies that the pop-up must be rotated along the Y-axis as the content is being displayed giving an illusion of a page is turned towards left in a book.
- Door - Specifies that the pop-up must be revealed giving an illusion of a opening a door.
- Rotate Exchange - Specifies that the pop-up must be rotated along the X-axis as the content is being displayed.

#### 9.4.6.2 Android Properties

**Transition Effect:** Specifies the effect from which the form is displayed. The available options are:

- None - The default device effect is applied or none of the effect is applied.
- Bottom Top - It specifies that the pop-up must slide-in from the bottom and proceed towards the top.
- Bottom Left Top - It specifies that the pop-up must slide-in from the bottom-left corner and proceed towards the top.

- Bottom Right Top -It specifies that the pop-up must slide-in from the bottom-right corner and proceed towards the top.
- Top Right Bottom - It specifies that the pop-up must slide-in from the top-right corner and proceed towards the bottom.
- From Center - It specifies that the pop-up must grow from the center with a fade effect.
- Slide In Left - It specifies that the pop-up must slide-in from the left with a fade effect.
- Slide In Right - It specifies that the pop-up must slide-in from the right with a fade effect.
- Top Down - It specifies that the pop-up must slide-in from the top and proceed towards the bottom.

#### 9.4.6.3 Windows Phone 8 Properties

**Transition Mode:** Specifies the direction from which the From is displayed. The available options are:

- Default - The default device effect is applied or none of the effect is applied.
- Sequential- The transition of the pop-up going out of the view completes first and then the transition of the pop-up coming into the view takes place.
- Parallel- The transition of the pop-up going out of the view and the transition of the pop-up coming into the view takes place simultaneously.

**Transition Key** - Specifies the effect from which the From is displayed. The available options are:

- Default - Specifies that the pop-up must slide horizontally into the view.
- Rotate 3D Single - Specifies that the pop-up must be rotated along the center Y-Axis when coming into the view.
- Rotate 3D Dual - Specifies that the pop-up must be shown making a circle around the screen from the background before coming into the view.
- Slide - Specifies that the pop-up must slide horizontally into the view.

- Pop - Specifies that the pop-up must emerge from center-bottom of the screen and gradually occupy the complete screen.
- Squeeze - Specifies that the pop-up must be expanded horizontally from an initial width of zero.

**Transition Speed** - Specifies the speed at which the From is transitioned. The value must be specified in seconds.

#### 9.4.6.4 SPA Properties

**Transition Effect:** Specifies the effect from which the form is displayed. The available options are:

- None- Use this option if you do not want to specify a transition direction.
- Top Center - Specifies that the pop-up must appear from the top center.
- Bottom Center - Specifies that the pop-up must appear from the bottom center.
- Right Center - Specifies that the pop-up must appear from the right center.
- Left Center - Specifies that the pop-up must appear from the left center.

#### 9.4.7 Transition: OUT

Specifies the configuration to be used when the user arrives on this pop-up. Click the **Edit** button against **Transition: OUT** field. From the **Transcation: OUT** window, choose the desired transition direction and transaction effect for specific platforms.

##### 9.4.7.1 iOS Properties

**Transition Direction:** Specifies the direction from which the pop-up is displayed. The available options are:

- None - Use this option if you do not want to specify a transition direction.
- From Right - Specifies that the pop-up must appear from the right.
- From Left - Specifies that the pop-up must appear from the left.

- From Bottom - Specifies that the pop-up must appear from the bottom.
- From Top - Specifies that the pop-up must appear from the top.

**Transition Effect:** Specifies the effect from which the pop-up is displayed. The available options are:

- None - Use this option if you do not want to specify a transition direction.
- Fade - Specifies that the pop-up must fade when it is transitioned to a hidden or an invisible state.
- MoveIn - Specifies that the pop-up must slide over the existing content in the direction as specified in the transition direction.
- Push - Specifies that the pop-up must push the existing content in the direction as specified in the transition direction and take its place.
- Flip - Specifies that the pop-up must be rotated along the Y-axis as the content is being displayed. It supports transition directions from right and left.
- Reveal - Specifies that the pop-up must be revealed gradually in the direction as specified in the transition direction.
- Curl - Specifies that the pop-up must be curled or folded (look and feel is similar to turning of a page in a book) as the content is being displayed. It supports transition directions from top and bottom only.
- Two Split Horizontal In - Specifies the pop-up which is split horizontally into two parts rejoins when the transition takes place.
- Two Split Vertical In - Specifies the pop-up which is split vertically into two parts rejoins when the transition takes place.
- Four Split In - Specifies the pop-up which is split in four parts rejoins when the transition takes place.
- Four Split Rotate In - Specifies the pop-up which is split in four parts rejoins by rotating the parts when the transition takes place.

- Two Split Horizontal Out - Specifies the pop-up which is split horizontally into two parts and move away when the transition takes place.
- Two Split Vertical Out - Specifies the pop-up which is split vertically into two parts and move away when the transition takes place.
- Four Split Rotate Out - Specifies the pop-up which is split in four parts move away by rotating the parts when the transition takes place.
- Switch Left - Specifies that the pop-up must go out of the view in 3D circular space along the Y-axis towards left as the content is being displayed.
- Switch Right - Specifies that the pop-up must go out of the view in 3D circular space along the Y-axis towards right as the content is being displayed.
- Cloth - Specifies the present pop-up must go out of screen animating as if a cloth is removed.
- Flip Right - Specifies that the pop-up must be rotated along the Y-axis as the content is being displayed giving an illusion of a page is turned towards right in a book.
- Flip Left - Specifies that the pop-up must be rotated along the Y-axis as the content is being displayed giving an illusion of a page is turned towards left in a book.
- Door - Specifies that the pop-up must be revealed giving an illusion of a opening a door.
- Rotate Exchange - Specifies that the pop-up must be rotated along the X-axis as the content is being displayed.

#### 9.4.7.2 Android Properties

**Transition Effect:** Specifies the effect from which the form is displayed. The available options are:

- None - The default device effect is applied or none of the effect is applied.
- Bottom Top - It specifies that the pop-up must slide-in from the bottom and proceed towards the top.

- Bottom Left Top - It specifies that the pop-up must slide-in from the bottom-left corner and proceed towards the top.
- Bottom Right Top - It specifies that the pop-up slide-in from the bottom-right corner and proceed towards the top.
- Top Right Bottom - It specifies that the pop-up must slide-in from the top-right corner and proceed towards the bottom.
- From Center - It specifies that the pop-up must grow from the center with a fade effect.
- Slide In Left - It specifies that the pop-up must slide-in from the left with a fade effect.
- Slide In Right - It specifies that the pop-up must slide-in from the right with a fade effect.
- Top Down - It specifies that the pop-up must slide-in from the top and proceed towards the bottom.

#### 9.4.7.3 Windows Phone 8 Properties

**Transition Mode:** Specifies the direction from which the From is displayed. The available options are:

- Default - The default device effect is applied or none of the effect is applied.
- Sequential- The transition of the pop-up going out of the view completes first and then the transition of the pop-up coming into the view takes place.
- Parallel- The transition of the pop-up going out of the view and the transition of the pop-up coming into the view takes place simultaneously.

**Transition Key** - Specifies the effect from which the From is displayed. The available options are:

- Default - Specifies that the pop-up must slide horizontally into the view.
- Rotate 3D Single - Specifies that the pop-up must be rotated along the center Y-Axis when coming into the view.
- Rotate 3D Dual - Specifies that the pop-up must be shown making a circle around the screen from the background before coming into the view.

- Slide - Specifies that the pop-up must slide horizontally into the view.
- Pop - Specifies that the pop-up must emerge from center-bottom of the screen and gradually occupy the complete screen.
- Squeeze - Specifies that the pop-up must be expanded horizontally from an initial width of zero.

**Transition Speed** - Specifies the speed at which the From is transitioned. The value must be specified in seconds.

#### 9.4.7.4 SPA Properties

**Transition Effect:** Specifies the effect from which the form is displayed. The available options are:

- None- Use this option if you do not want to specify a transition direction.
- Top Center - Specifies that the pop-up must appear from the top center.
- Bottom Center - Specifies that the pop-up must appear from the bottom center.
- Right Center - Specifies that the pop-up must appear from the right center.
- Left Center - Specifies that the pop-up must appear from the left center.

#### 9.4.8 Modal

In user interface design, a modal window, which is a child window that requires users to interact with it before they can return to operating the parent application, thus preventing the workflow on the application main window.

This property indicates whether the popup is to be shown as modal window or a non-modal window. The selected option becomes default.

To provide a platform specific style, click the ellipsis (...) button next to the Off option of Modal property to open the **Modal** dialog box.

from here, you can either change the default value or provide a platform specific value.

#### 9.4.9 Modal Transparency

You can specify the opacity of the Modal.

#### 9.4.10 Height

Specifies the height of the Pop-up in terms of percentage. The percentage is with reference to the value of [Height Reference](#) property.

**Note:** This property is unavailable on Flex Forms.

#### 9.4.11 Height Reference

The Pop-up height percentage is calculated based on the option selected.

- Form Reference: The Pop-up height percentage is calculated based on the height of the Pop-up excluding headers and footers.
- Parent Width: The width is calculated based on the width of the Pop-up.

**Note:** This property is unavailable on Flex Forms.

#### 9.4.12 Input View Type

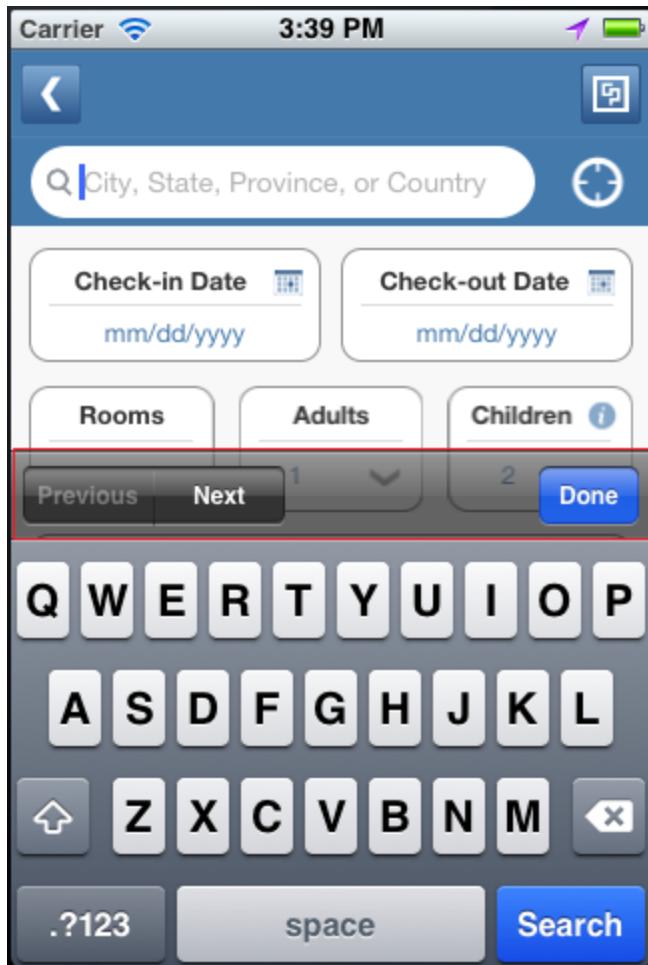
When building iPhone applications that support or provide text input, it's often necessary to create some extra buttons (or other controls) beyond the ones provided by the default keyboard interface. Kony Platform by default, adds the Previous, Next and Done buttons to the applicable input controls. These buttons allow specific operations needed by your application, such as moving to the next or previous text field, make the keyboard disappear. The area above the keyboard is known as Input Accessory View.

This property, allows you to specify the type of accessory view that will be shown for all the input controls on this popup.



Following are the available options:

- None: Use this option if you do not want to specify the toolbar. This option should be used carefully, as setting this option for widgets like calendar leaves the user with no option to select and drop-down a wheel calendar.
- Default: Specifies that the toolbar that is defined in the Application level settings. To set the Application level settings, right-click on the project and navigate to **Properties > Native App > iPhone/iPad**.
- Next / Previous: Specifies the navigation options as Next, Previous, and Done for a popup. The below image illustrates the **next prev toolbar** set for a **Textbox**. The highlighted toolbar is achieved by setting the **Keyboard Type** as **Default** for a Textbox and **Input Accessory View Type** as **nextprevtoolbar** to the popup.



- Cancel: Specifies that the input accessory view has a cancel button. This option does not trigger any events.

**Note:** This property is specific to the iOS platform.

#### 9.4.13 Scroll Bounces

Specifies whether the scroll view bounces past the edge of the content and back again.

- If **On** option is selected, the scroll view bounce is allowed.
- If **Off** option is selected, the scroll view bounce is disallowed.

**Note:** This property is specific to the iOS platform.

#### 9.4.14 Configure Extend Top

- If **On** option is selected,
- If **Off** option is selected,

#### 9.4.15 Extend Top

Extend Top specifies whether the scrollbar should start from the header of a popup or the body of the popup. If a popup does not have a header, this property behaves the same when you choose either of the following options:

- If **On** option is selected, the scrollbar starts from the header of a popup.
- If **Off** option is selected, the scrollbar starts from the body of the popup.

**Note:** This property is specific to the iOS7 platform.

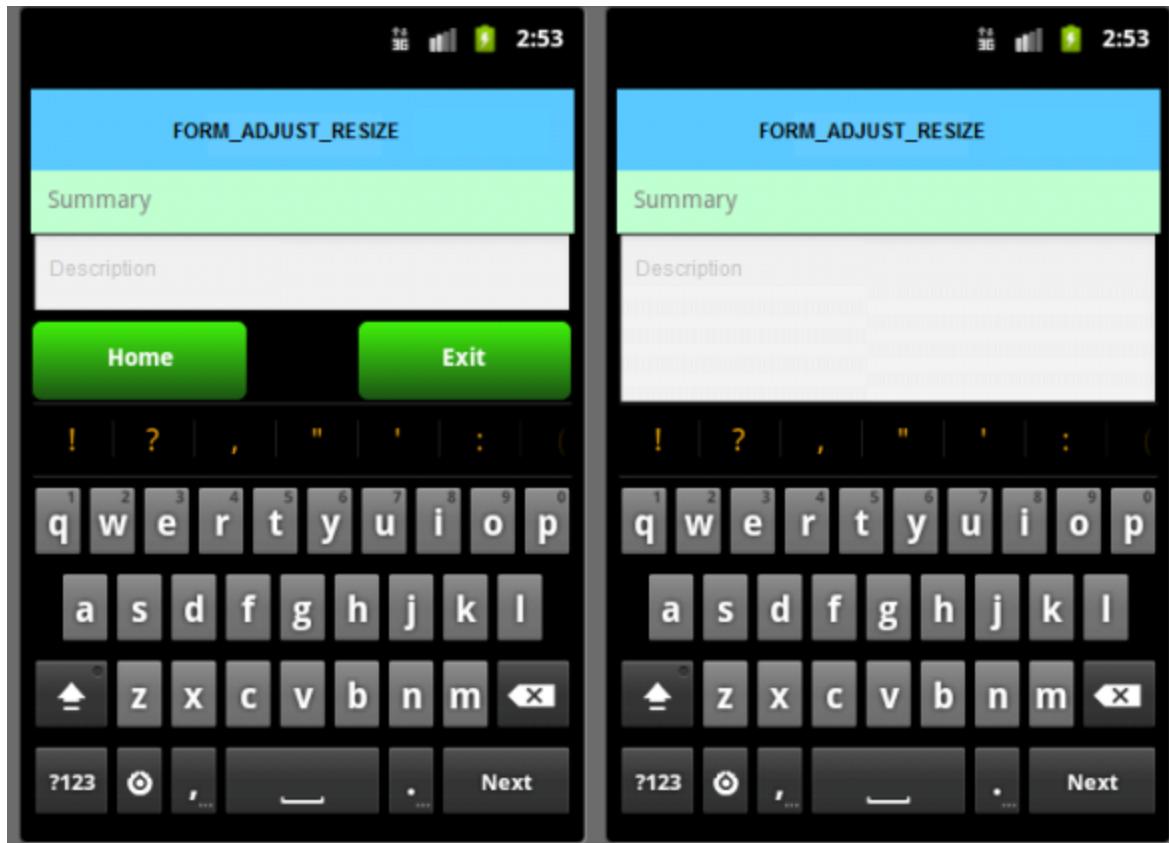
#### 9.4.16 Soft Input Mode

Specifies the soft key-board for a widget that requires an input should be placed at the bottom of the popup. For example, when a TextBox or a TextArea requires input from the user, the soft-keyboard pops up when you enter text.

Following are the available options:

- Adjust -, Resize: Specifies the popup is resized and when you click or start typing within the widget which requires an input, the popup scrolls up and the widget which requires an input is not overlapped by the keypad or footer.
- Adjust - Pan: Specifies the widget which requires an input is placed at the bottom of the popup is overlapped by the keypad.

The below image illustrates the above two options:



**Note:** This property is specific to the Android platform.

#### 9.4.17 Max App Menu Buttons

Specifies the number of app menu buttons that should be displayed on the device.

### 9.5 Actions

pop-up has the following actions associated with it:

- **init:** is called immediately after an addWidgets event for any initializations required for the pop-up. Init initializes the pop-up and any widgets.

In case of Server side Mobile Web and SPA, addWidgets and init events gets called as soon as the pop-up is created. In case of native platforms, as an optimization, these events are deferred until the first access.

- **onHide:** is called when the pop-up goes out of the screen. A pop-up can go out of the screen when another pop-up is to be shown.
- **onDeviceBack:** Specifies an event which is triggered when the user uses the back button on the device.

## 10. Common Properties

Widgets contain common properties that are explained in this section.

- [Widgets - Look](#)
- [Working with Skins](#)
- [Forking a Widget Property](#)

## 10.1 Look

This section details the **Properties > Look** tab of a widget.

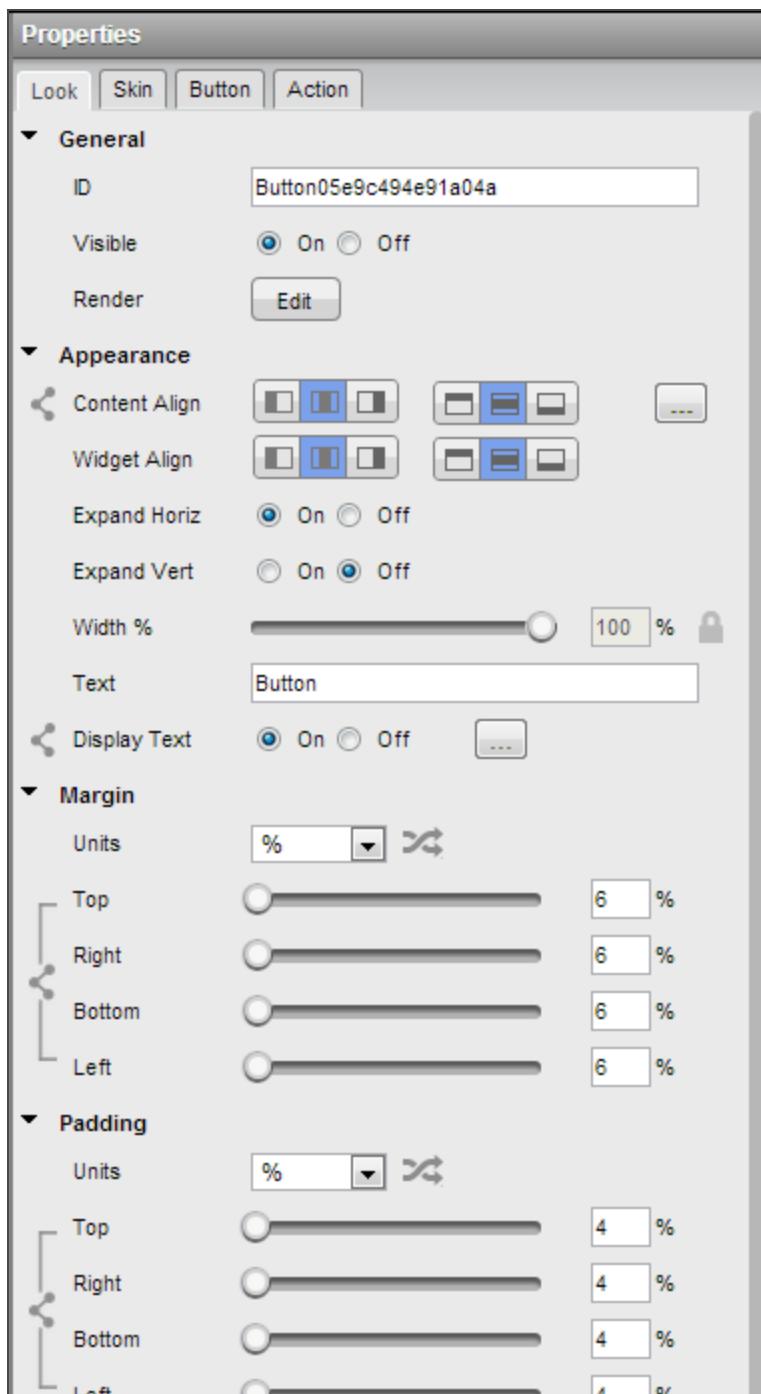
**Note:** Some of the properties discussed in this section may not be applicable for all the widgets.

The following are the two types of Look properties available for a widget:

- [VBox Form](#)
- [Flex Form](#)

### 10.1.1 VBox Form

The following are the widgets' **Look** properties available for a VBox Form:



#### 10.1.1.1 ID

Denotes the name of a widget. When a widget is added to a form, a unique name is assigned to the widget. You can rename a widget by typing a name in the ID box.

**Note:** You can also rename a widget from the Project Explorer by right-clicking a widget, and then clicking **Rename**.

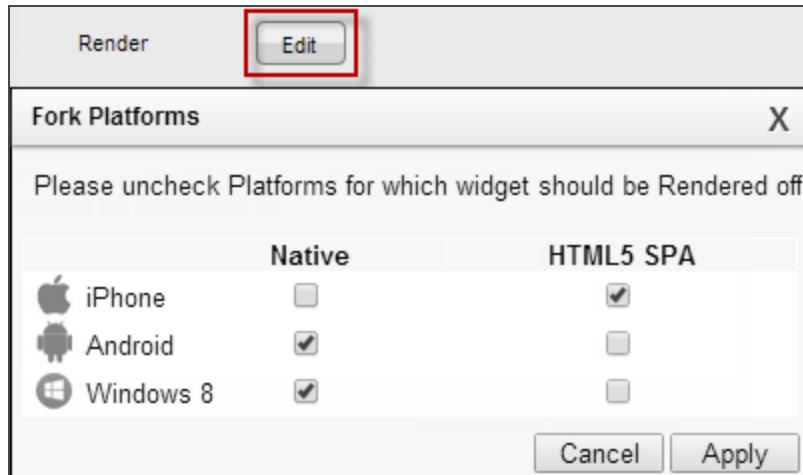
#### 10.1.1.2 Visible

Controls the visibility of a widget.

- To make a widget visible, click **On**.
- To make a widget invisible, click **Off**.

#### 10.1.1.3 Render

Defines whether a widget is displayed on a specific platform. By default, a widget is rendered for all the platforms. If you do not want to render a widget for a specific platform, click the **Edit** button against the **Render** field to open the **Fork Platforms** dialog box.



Clear the check box of the platforms for which the widget should not be rendered.

#### Difference between Visible and Render

- When a Widget is *not* rendered for a platform, it implies that the widget is hidden from that specific platform.

- Whereas, when a widget is set as invisible, it implies that the widget is available, but is invisible. This feature is useful when you wanted to display a widget based on certain conditions.

#### 10.1.1.4 Content Align

Allows you to define how the widget content is aligned. Following alignment options are available:

- a. Horizontal Alignment : Allows you to align the content horizontal.

**Note:** Horizontal alignment of a widget is possible only when the [Expand Horiz](#) property is enabled.

- b. Vertical Alignment : Allows you to align the content vertically. This type of alignment is possible only when:

- Two or more widgets are placed inside a container (with horizontal orientation) such as HBox.
- Widget whose content is to be aligned vertically should not occupy maximum height.
- The [Expand Vert](#) property is enabled for the widget whose content is to be aligned vertically.

**Note:** You can only view the behavior of the [Expand Vert](#) property during the [Quick Preview](#) and the [Functional Preview](#).

You can define the content alignment of a widget specific to a platform by forking **Content Align**. For more information on forking, refer section: [Forking a property](#).

#### 10.1.1.5 Widget Align

Specifies how a widget is aligned with respect to its boundaries and its neighboring widgets (within the same container widget). Following alignments options are available:

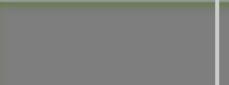
- a. Horizontal Alignment  : Allows you to align a widget horizontally within its boundaries. This type of alignment is possible only when:
- The [Expand Horiz](#) property is disabled.
  - The text entered in the [Text](#) box should occupy width less than the [Width%](#) set for that widget.

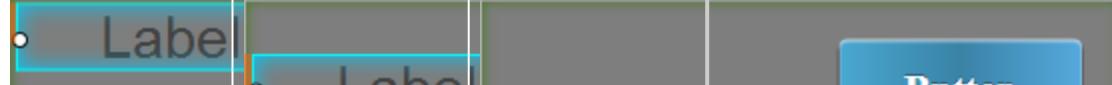
Below table displays how a **Button** widget is aligned within its boundaries.

| Left  | Center  | Right   |
|---|---|---|
|  |  |  |

- b. Vertical alignment  : Allows you to align a widget vertically with respect to its neighboring widget. This type of alignment is possible only when:
- Two or more widgets are placed inside a container widget (with horizontal orientation) such as HBox.
  - Widget that is being aligned vertically should not occupy maximum height.

For example, consider a **Label** widget and a **Button** widget are placed inside an **HBox** widget. Below table demonstrates how the **Label** widget is aligned with respect to the **Button** widget.

| Left  | Center  | Right   |
|---|---|---|
|  |  |  |



#### 10.1.1.6 Expand Horiz

Specifies whether widget is allowed to expand horizontally.

#### 10.1.1.7 Expand Vert

: Specifies whether widget is allowed to expand vertically.

#### 10.1.1.8 Width %

Specifies the width of a widget.

- If the widget is directly placed on a form, the widget occupies 100 percent width.
- If a widget is placed inside another widget, you can reduce the width of the widget. You can also lock the percentage of the width occupied by the widget.

#### 10.1.1.9 Text

Specifies the text displayed on a widget.

#### 10.1.1.10 Display Text

Specifies whether the text should be displayed on a widget.

- To display text on a widget, click **On**.
- To hide the text on a widget, click **Off**.

You can choose to display the text specific to a platform by forking **Display Text**. For more information on forking, refer section: [Forking a property](#).

#### 10.1.1.11 Margin

Defines the space around a widget. You can use this option to assign the left, top, right, and bottom distance between the widget and the next element.

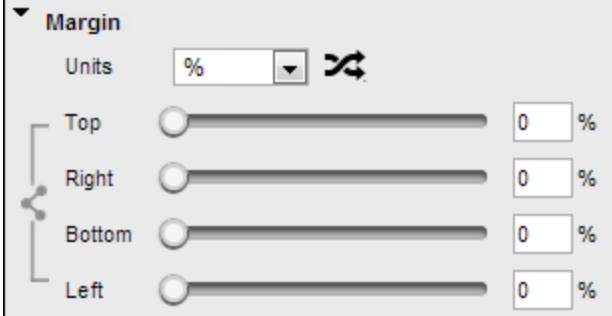
| Property | Definition    | Action   |
|----------|---------------|--|
| Top      | Top margin    | Move the slider to adjust the top margin of the widget.    |
| Right    | Right margin  | Move the slider to adjust the right margin of the widget.  |
| Bottom   | Bottom margin | Move the slider to adjust the bottom margin of the widget. |

| Property | Definition  | Action   |
|----------|-------------|--|
| Left     | Left margin | Move the slider to adjust the left margin of the widget. |

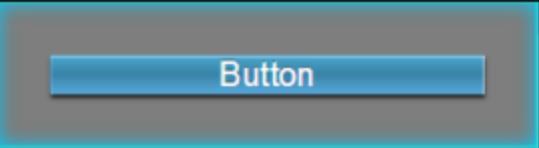
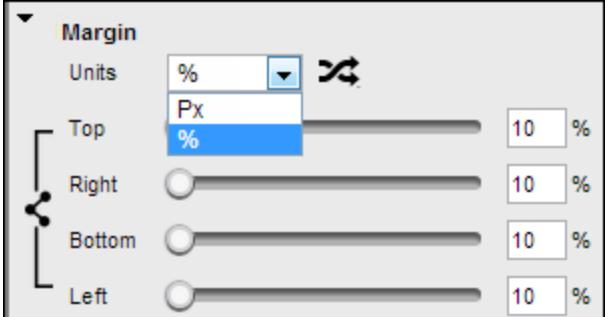
**Note:**

1. Uniform margins are applied across the directions by clicking .
2. When widget margins are modified, it will affect the margins of its parent widget as well as that of the child widgets.

Below image illustrates the Button widget forking:

| Before Applying Margins   |   |
|---|---|
| <b>App Canvas</b><br> | <b>Margin Properties</b><br> |

| After applying margins   |  |
|--|--|
| <b>App Canvas</b><br> | <b>Margin Properties</b><br> |

You can provide margins specify to a platform by forking **Margin**. For more information on forking, refer section: [Forking a property](#).

**Note:** When a **Margin** property is forked, you can apply margins in either Pixels or Percentage for native applications.

#### 10.1.1.12 Padding

Defines the space between the content of the widget and the widget boundaries. You can use this option to assign the top, left, right, and bottom distance between the widget content and the widget's boundaries.

| Property | Definition     | Action  |
|----------|----------------|---|
| Top      | Top padding    | Move the slider to adjust the top padding of the widget.    |
| Bottom   | Bottom padding | Move the slider to adjust the bottom padding of the widget. |
| Left     | Left padding   | Move the slider to adjust the left padding of the widget.   |
| Right    | Right padding  | Move the slider to adjust the right padding of the widget.  |

**Note:**

1. Uniform margins are applied across the directions .
2. When widget padding is modified, it will affect the padding of its parent widget as well as that of the child widgets.

Below image illustrates, Button widget forking:

| Before Applying Padding |                    |
|-------------------------|--------------------|
| App Canvas              | Padding Properties |

Before Applying Padding

Button >

Padding

Units: %

Top: 0 %

Right: 0 %

Bottom: 0 %

Left: 0 %

After applying paddings

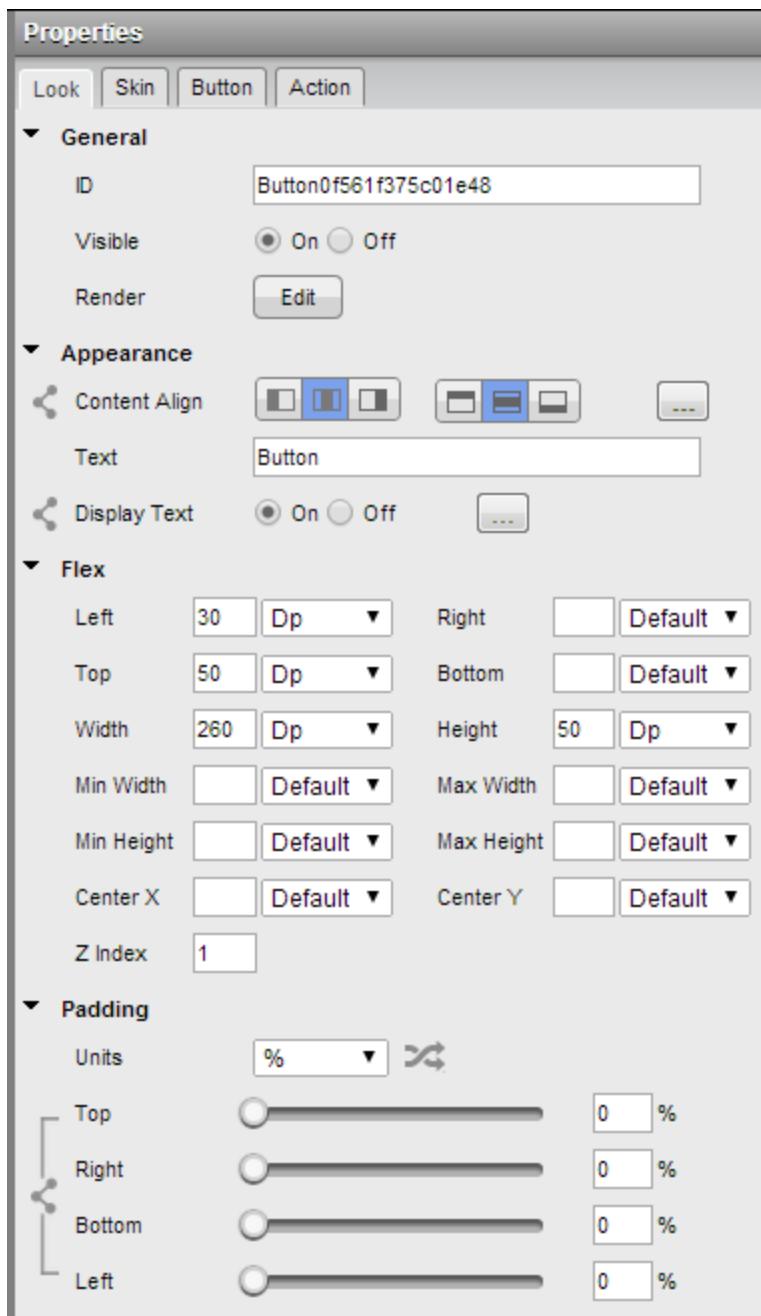
| App Canvas  | Margin Properties  |
|---|--|
|  | <p>Padding</p> <p>Units: Px <input type="button" value="▼"/> %</p> <p>Top: 10 %</p> <p>Right: 10 %</p> <p>Bottom: 10 %</p> <p>Left: 10 %</p> |

You can provide padding specify to a platform by forking **Padding**. For more information on forking, refer section: [Forking a property](#).

**Note:** When Padding property is forked, you can apply paddings in either Pixels or Percentage for native applications.

### 10.1.2 Flex Form

The following are the widgets' **Look** properties available for a Flex Form



#### 10.1.2.1 ID

Denotes the name of a widget. When a widget is added to a form, a unique name is assigned to the widget. You can rename a widget by entering a new name in the ID box.

**Note:** You can also rename a widget from the Project Explorer by right-clicking a widget, and then clicking **Rename**.

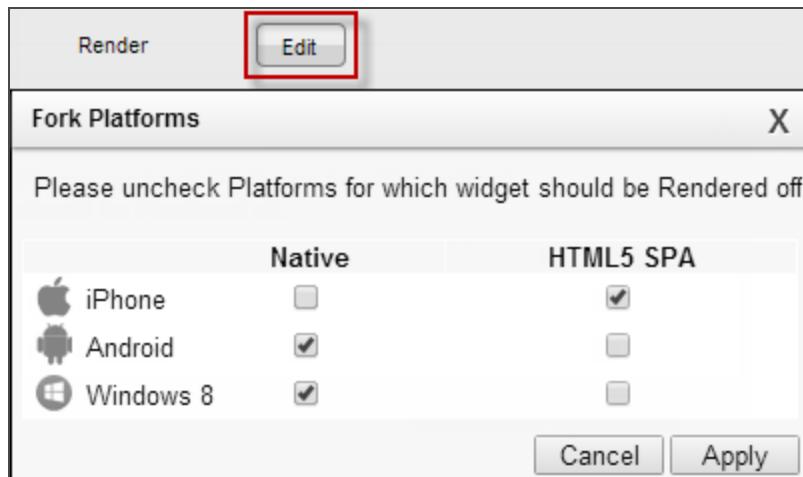
### 10.1.2.2 Visible

Controls the visibility of a widget.

- To make a widget visible, click **On**.
- To make a widget invisible, click **Off**.

### 10.1.2.3 Render

Defines whether a widget is displayed on a specific platform. By default, a widget is rendered for all the platforms. If you do not want to render a widget for a specific platform, click the **Edit** button against the **Render** field to open the **Fork Platforms** dialog box.



Clear the check box of the platforms for which the widget should not be rendered.

#### Difference between Visible and Render

- When a Widget is *not* rendered for a platform, it implies that the widget is hidden from that specific platform.

- Whereas, when a widget is set as invisible, it implies that the widget is available, but is invisible. This feature is useful when you wanted to display a widget based on certain conditions.

#### 10.1.2.4 Content Align

Allows you to define how the widget content is aligned. Following alignment options are available:

- a. Horizontal Alignment : Allows you to align the content horizontal.

**Note:** Horizontal alignment of a widget is possible only when the [Expand Horiz](#) property is enabled.

- b. Vertical Alignment : Allows you to align the content vertically. This type of alignment is possible only when:
  - Two or more widgets are placed inside a container (with horizontal orientation) such as HBox.
  - Widget whose content is to be aligned vertically should not occupy maximum height.

You can define the content alignment of a widget specific to a platform by forking **Content Align**. For more information on forking, refer section: [Forking a property](#).

#### 10.1.2.5 Text

Specifies the text displayed on a widget.

#### 10.1.2.6 Display Text

Specifies whether the text should be displayed on a widget.

- To display text on a widget, click **On**.
- To hide the text on a widget, click **Off**.

You can choose to display the text specific to a platform by forking **Display Text**. For more information on forking, refer section: [Forking a property](#).

### 10.1.2.7 Flex Properties

The Flex properties enables you to resize, move and position a widget.

#### 10.1.2.8 Left

Determines the left edge of the widget and measured from the left bound of the parent container.

#### 10.1.2.9 Right

Determines the right edge of the widget and measured from the right bound of the parent container.

#### 10.1.2.10 Top

Determines the top edge of the widget and measured from the top bounds of the parent container.

#### 10.1.2.11 Bottom

Determines the bottom edge of the widget and measured from the bottom bounds of the parent container.

#### 10.1.2.12 Width

It determines the width of the widget and measured along the x-axis.

Following are the available options that can be used as units of width:

- %: Specifies the values in percentage relative to the parent dimensions.
- px: Specifies the values in terms of device hardware pixels.
- dp: Specifies the values in terms of device independent pixels.
- Default: Specifies the unit that is considered as default. The default unit is specified in the Flex form's platform specific property: [Default Unit](#).

- Preferred Size: When this option is specified, the layout uses preferred height of the widget as height and preferred size of the widget is determined by the widget and may varies between platforms.

#### 10.1.2.13 Height

Height determines the height of the widget and measured along the y-axis (height of the parent). You can use any of the following options:

- %: Specifies the values in percentage relative to the parent dimensions.
- px: Specifies the values in terms of device hardware pixels.
- dp: Specifies the values in terms of device independent pixels.
- Default: Specifies the unit that is considered as default. The default unit is specified in the Flex form's platform specific property: [Default Unit](#).
- Preferred Size: When this option is specified, the layout uses preferred height of the widget as height and preferred size of the widget is determined by the widget and may varies between platforms.

#### 10.1.2.14 Min Width

Specifies the minimum width of the widget. This property is considered only when width property is not specified.

#### 10.1.2.15 Max Width

Specifies the maximum width of the widget. This property is considered only when width property is not specified.

#### 10.1.2.16 Min Height

Specifies the minimum height of the widget. This property is considered only when height property is not specified.

#### 10.1.2.17 Max Height

Specifies the maximum height of the widget. This property is considered only when height property is not specified.

#### 10.1.2.18 Center X

It determines the center of widget measured from left bounds of the parent container.

**Note:** If the Layout Type is set as Horizontal, Center X is measured from right edge of the left sibling widget.

#### 10.1.2.19 Center Y

It determines the center of the widget measured from top bounds of the parent container.

**Note:** If the Layout Type is set as Vertical, centerY is measured from bottom edge of the top sibling widget.

#### 10.1.2.20 Z Index

Specifies the stack order of the widgets. A widget with higher zIndex is in front of the one with lower zIndex.

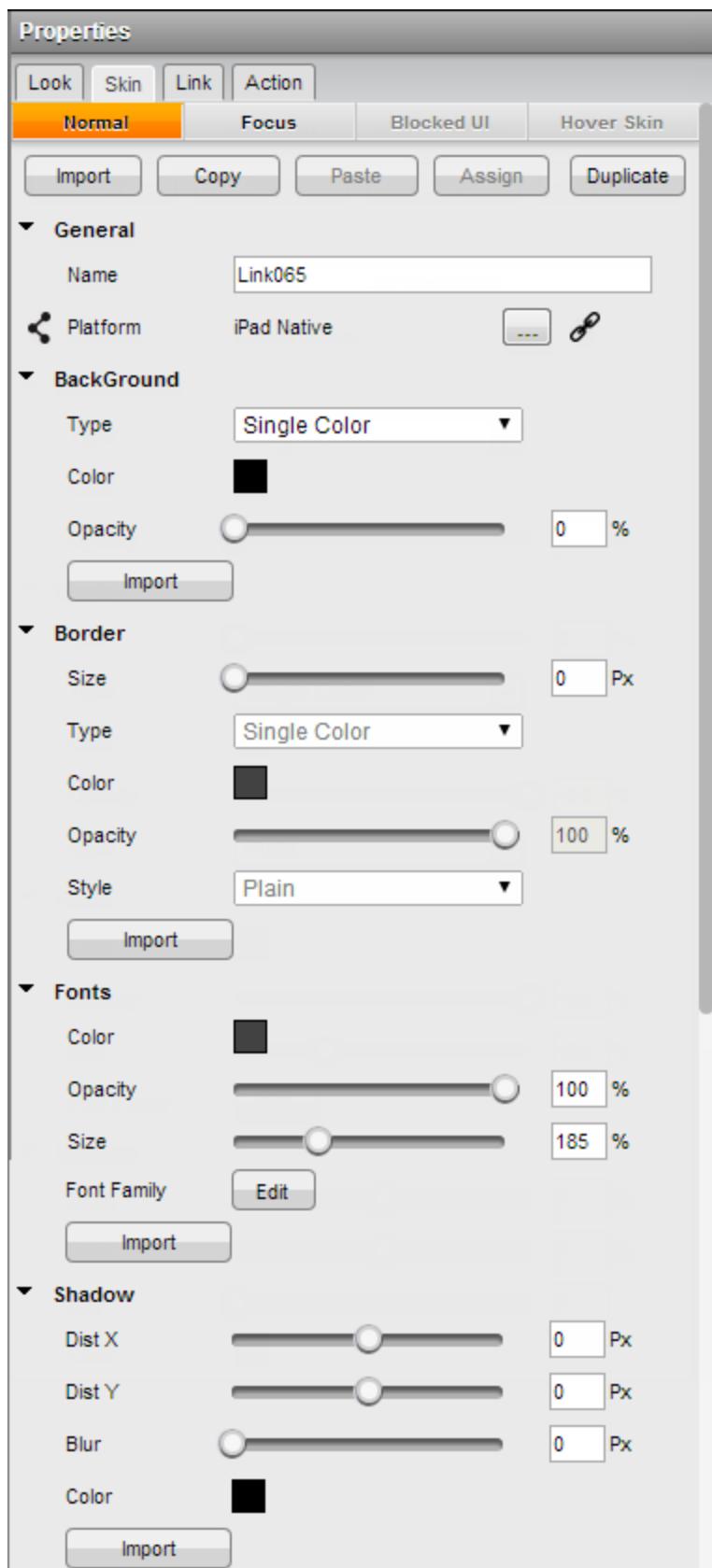
**Note:** Modifying the zIndex does not modify the order of widgets inside a flex container. If widgets are overlapping with each other with the same zIndex, then the widget order decides the order of overlapping. Last added widget will be displayed on top.

Default:1 (Indicates the widgets is lowest bottom in the layer hierarchy.)

## 10.2 Working with Skins

This section details the **Properties >Skin** tab of a widget.

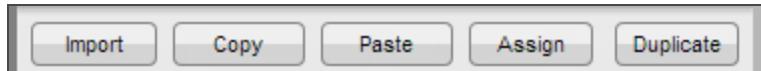
**Note:** Some of the below properties may not be applicable to all the widgets.



**Skins** tab allows you to control the background color and border color of a widget. From here, you can also modify the font styles, text shadows and widget shadows.

### 10.2.1 Skins Reusability

To facilitate reusability of a skin, Visualizer provides you with the following options:



- **Copy:** Allows you to copy the current skin of a widget to a clipboard.
- **Paste:** Allows you to paste the copied skin to a selected widget.

**Note:** When a skin is pasted, a copy of the original skin is created. The pasted skin becomes an independent skin. Any changes made to the original skin does not affect the pasted skin.

**Note:** Paste function can be applied across the widget states. For example, a Normal state skin of a Button widget can be copied, and pasted to Focus skin state of the Button widget.

- **Assign:** Allows you to assign the copied skin to a selected widget. All the properties of the original skin are applied to the selected widget.

**Note:** As the skin is assigned, any changes made to the original skin affects the widgets to which this skin is assigned.

**Note:** Assign function can be applied across the widget states. For example, a Normal state of a Button widget can be copied and assigned to Focus state of the Button widget.

- **Duplicate:** Allows you to create a duplicate of the current skin. A duplicated skin becomes an independent skin, and any changes made to the original skin does not affect the duplicated skin.

**Difference between Paste and Assign function:**

- When a skin is copied and pasted, a new skin is created. Whereas, when a skin is copied and assigned, no new skin is created.
- Changes made to the copied skins do not affect the pasted skin. Whereas changes made to the copied skin affects the assigned skin (as the copied and assigned skin are one and the same).

## 10.2.2 General

Following are the general properties of a skin:

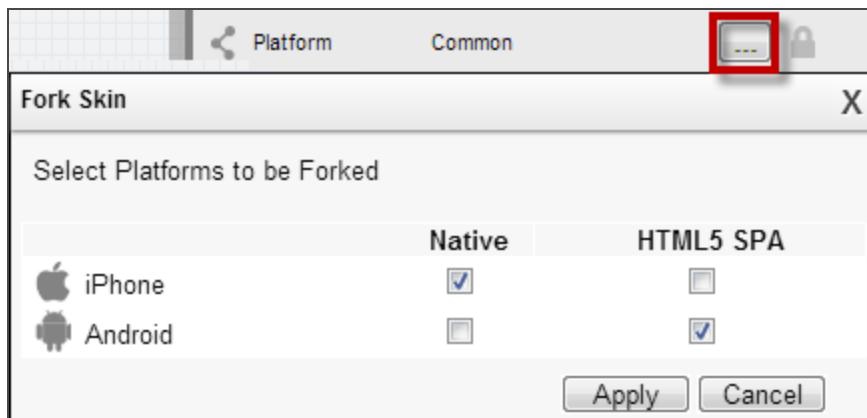


1. **Name:** This is the unique name assigned by the system. You are allowed to rename a skin:
  - When you fork it.
  - When you modify any of its properties.
2. **Platform:** Allows you to fork the skin widget for a platform.

### Important Considerations:

- Widgets that have background type as Multi Step Gradient are by default forked (for example, Button and Phone widgets).
- When you fork and lock a widget skin, changes made to a widget's skin in one platform gets applied to the remaining platforms, unless, you specifically, unlock a platform.
- To provide a unique skin for a platform, you apply fork and unlock properties for that platform.

- By default all the widgets (except widgets that have background type as Multi Step Gradient), will have a **Common** skin. Changes made to this skin will be applied across the platforms.
- You can select the platforms to be forked by clicking the ellipsis (...) button, and from the **Fork Skin** dialog box, select the desired platforms to be forked.



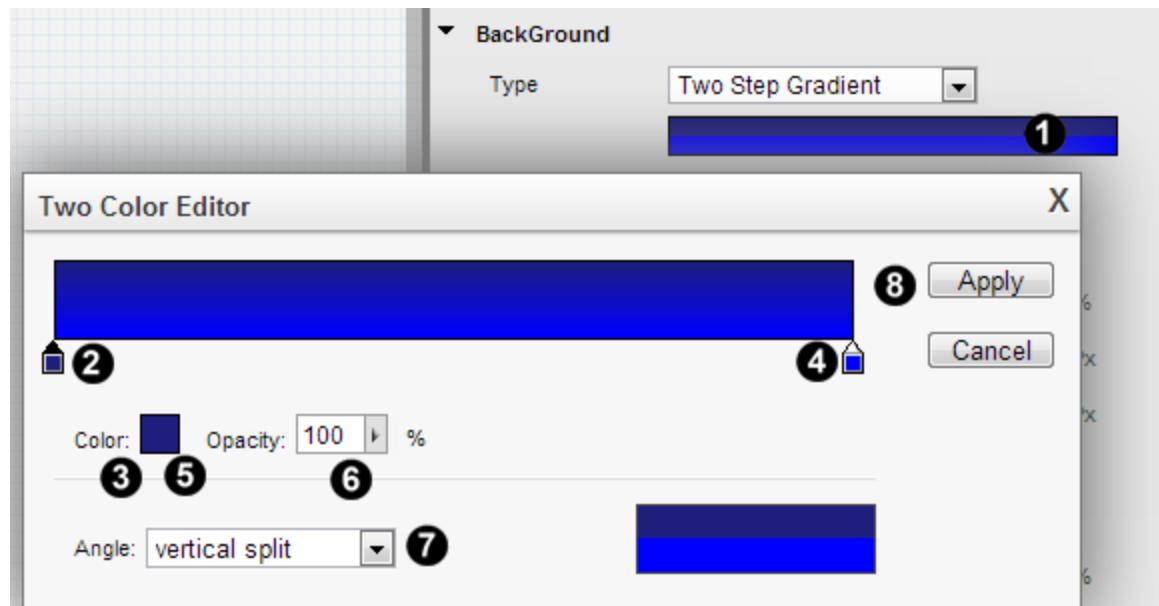
### 10.2.3 Background

**Note:** Not all widgets contain below referenced background types.

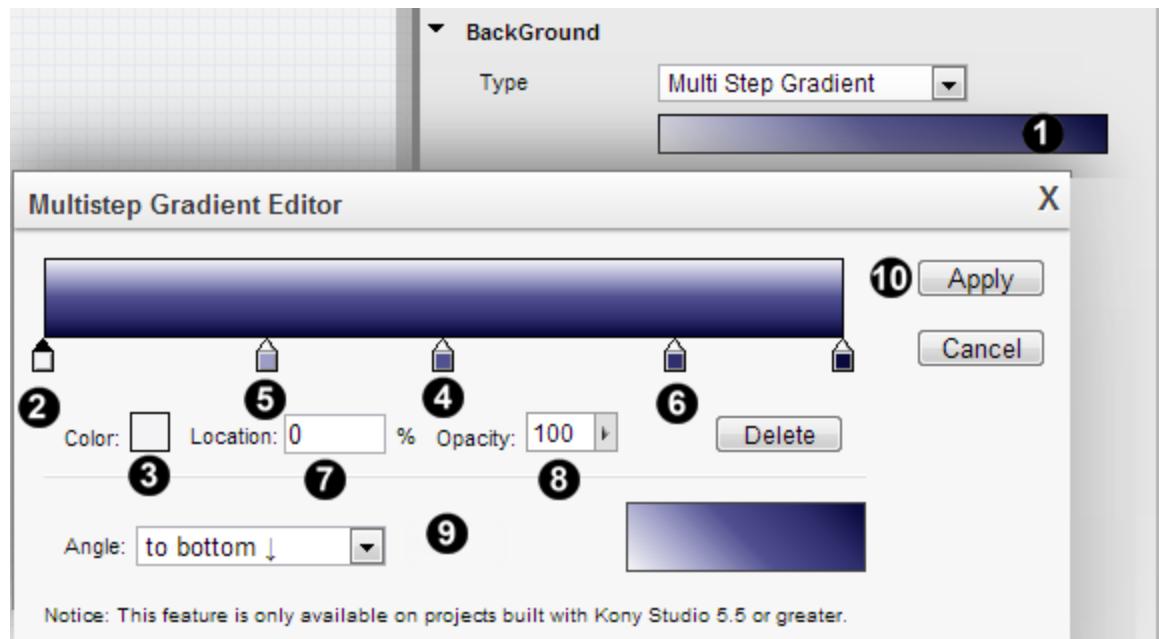
1. **Type:** You can change the background color of a widget by selecting one of the below options from the **Type** list.
  - a. **Single Color:** If you select **Single Color** as the type, you can select the color and also, set the transparency level for the background. To apply **Single Color**, follow the below steps:



1. Click the Color icon to open the Color picker window.
  2. Select a new color.
  3. Click the to apply color.
  4. Adjust the opacity of the color, if required.
- b. **Two Step Gradient:** If you select **Two step gradient** as the style, you can select top and bottom colors. To apply **Two Step Gradient**, follow the below steps:



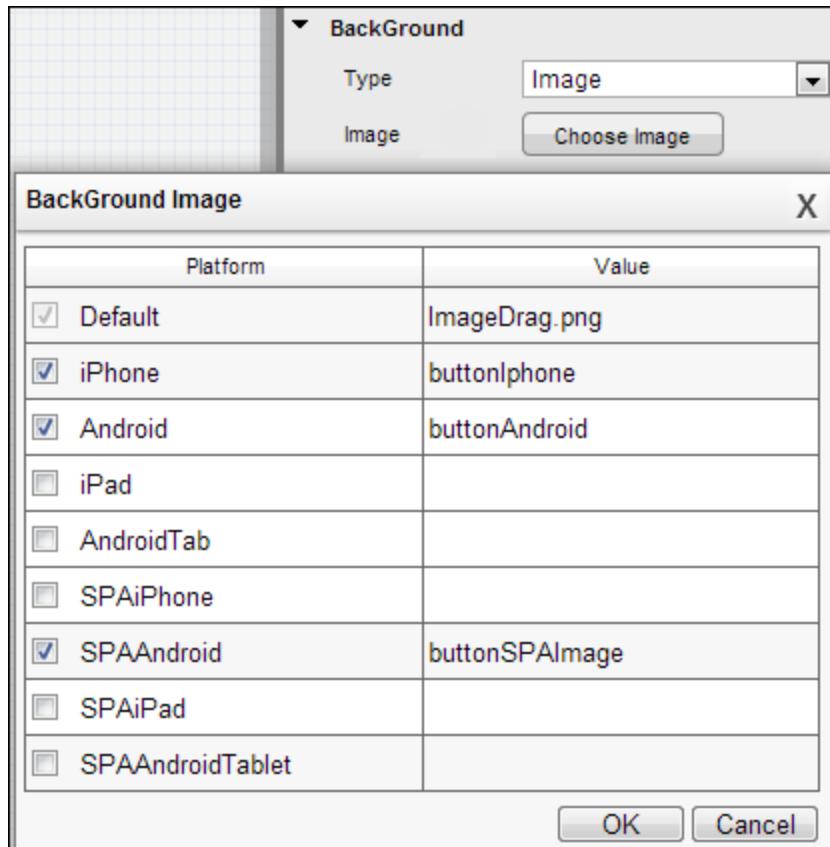
1. Click the color preview to display **Two Color Editor**.
  2. Click the left gradient icon.
  3. Click the Color picker icon to open the **Color Picker** window. Select a new color.
  4. Click the right gradient icon.
  5. Click the Color picker icon to open the **Color Picker** window. Select a new color.
  6. Adjust the opacity of the gradients, if required.
  7. Select the angle at which the gradient colors are to be displayed on the widget.
  8. Click **Apply** to save the changes made to the background color of the widget.
- c. **Multi Step Gradient:** If you select Multi Step Gradient as the style, you can use multiple colors for creating a skin background. To apply **Multi Step Gradient**, follow the below steps:



1. Click the color preview to display **Multistep Gradient Editor**.
2. Click the left gradient icon.
3. Click the Color picker icon to open the **Color Picker** window. Select a new color.
4. Click on the edge of the gradient picker, to create a new gradient step. Click the Color picker icon to open the **Color Picker** window. Select a new color.
5. Another example of a gradient step.
6. Another example of gradient step.
7. For each of the color stopper, you can provide the location at which the new color
8. Adjust the opacity of the color, if required.
9. Select an angle for the colors to be displayed on the widget.
10. Click **Apply** to save the changes made to the background color of the widget.

**Note:** At any point, you can delete a gradient step by clicking **Delete**.

- d. **Image:** If you select an image as a background for a widget, then the background skin will be replaced by an image. To apply **image**, follow the below steps:



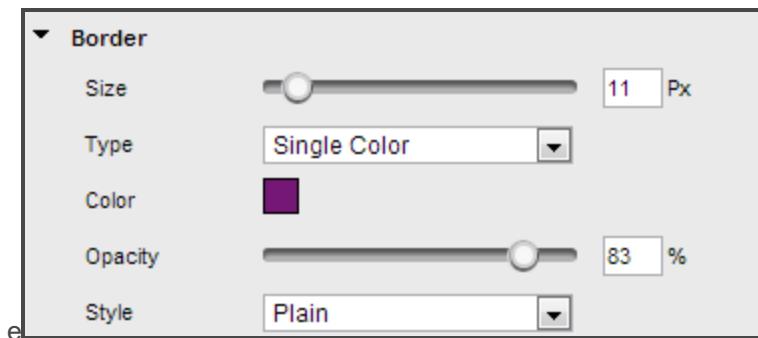
- i. Click **Choose Image** to display **BackGround Image** dialog box.
- ii. To provide a default image for all the platforms, click **Value** field corresponding to **Default** platform and from the **Search Image** window select an image.
- iii. To provide a platform specific image, click the platform check box and then click inside the corresponding **Value** field to display **Search Image** window from where you can select an image.

**Note:** Before using an image in a project, navigate to following location:

<konyworkspace>/<Name of your project>/resources and copy the image in the respective subfolder. For example, if you want an image to be available across all the channels, copy the image in the subfolder common.

Alternatively, if you wanted an image to be available only for a specific platform, for example, mobile channel > native platform, copy the image to mobile/native subfolder. After copying the image, ensure you click the refresh button (↻) from the **Project Explorer > Assets** tab (in Kony Visualizer) to bind the newly added image to the project.

#### 10.2.4 Border

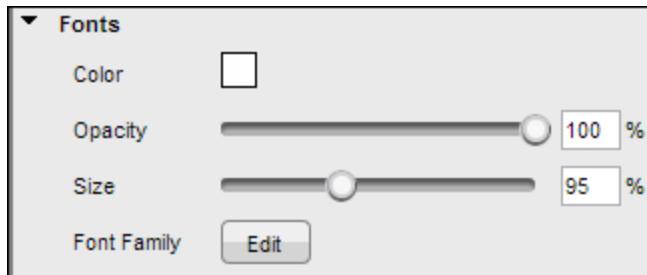


You can modify the following border properties of a widget:

- a. **Size:** Provide the Radius of the widget borders.
- b. **Type:** Type defines the border color of the widget. When the border size is above zero pixels, **Type** property is enabled. You can choose a single color or two as border color or select a multi-step gradient colors as border color.
- c. **Color:** Change the border color of a widget.
- d. **Opacity:** Change the opacity level of the border color.
- e. **Style:** When the border size is above zero pixels, **Styles** property is enabled. Following type of Styles are available for the widgets: Plain, Rounded Corner, Complete Rounded Corner and Custom style borders.

- Plain: Widget borders are in rectangular style.
- Rounded Corner: Widget borders will have a rounded corners.
- Complete Rounded Corner: To be updated.
- Custom: Select this option to provide a your own unique style by using the **Radius** slider.

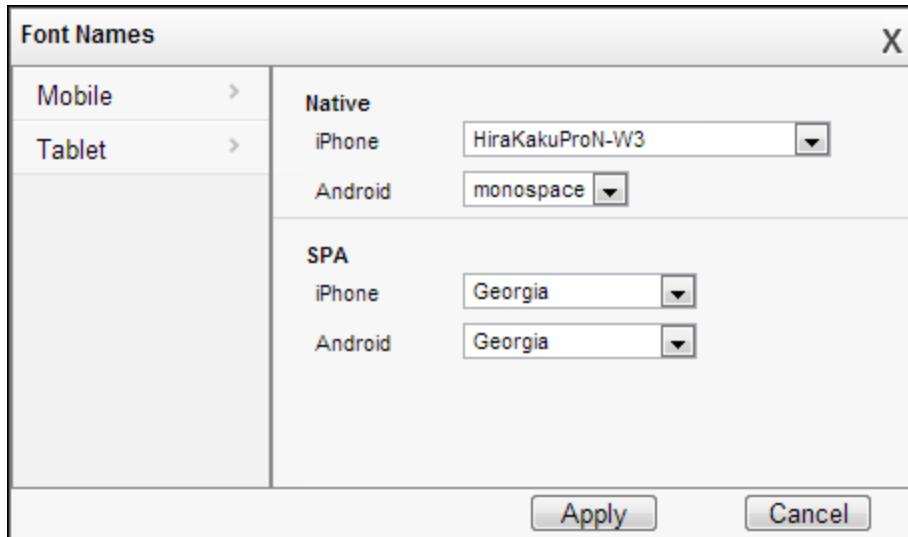
## 10.2.5 Fonts



You can modify the following font properties of a widget:

- a. **Color:** Select font color for the widget.
- b. **Opacity:** You can change the opacity level of the font color.
- c. **Size:** You can modify the font size between the ranges of 0 to 600% .

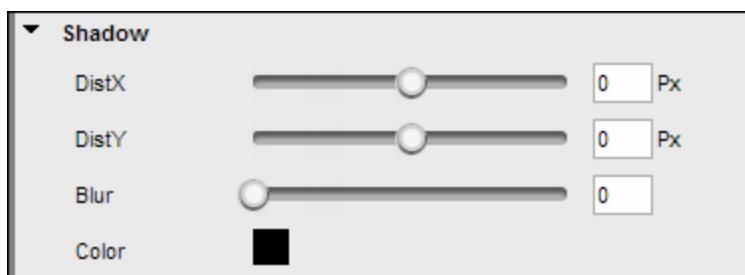
- d. **Font Family:** Click the **Edit** button against **Font Family** to open the **Font Names** window.



Here, you can select the fonts for individual platforms for both Mobile and Tablets.

- e. **Weight:** Select whether text on the widget should be normal or bold.
- f. **Style:** Select whether text on the widget should be underlined or italicized.

### 10.2.6 Shadow



**Note:** Shadow feature is available only a widget is forked.

You can modify the following shadow properties of a widget:

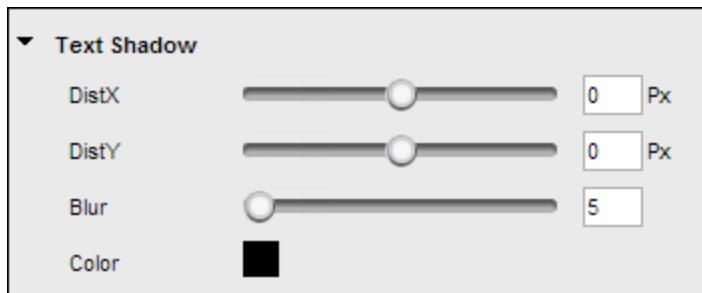
- a. **DistX:** Denotes the horizontal shadow distance away from the widget. When a positive **DisX**

value is entered, shadow is moved to right-side of a widget; when a negitve value is entered, shadow is moved to left-side of a widget.

- b. **DistY:** Denotes the vertical shadow distance away from the widget. When a positive **DisY** value is entered, shadow is moved to top-side of a widget; when a negitve value is entered, shadow is moved to bottom-side of a widget.
- c. **Blur:** Denotes the shadow blur of the widget by allowing you to soften shadow.
- d. **Color:** Denotes the color of the shadow. You can select a shadow color by clicking the **Color Picker**.
- e. **InnerShadow:** Click this check box if you wanted the shadow to be displayed inside the widget. Else, clear the check box to have the shadows displayed outside the widget.



### 10.2.7 Text Shadow



**Note:** Text Shadow feature is available only for specific widgets such as Button, Label and TextBox2.

You can modify the following text shadow properties of a widget:

- a. **DistX:** Denotes the horizontal shadow distance away from the widget text.
  - A positive value denotes that the text shadow is displayed to the right of the text.
  - A negative value denotes that the text shadow is displayed to the left of the text.
- b. **DistY:** Denotes the vertical shadow distance away from the widget.
  - A positive value denotes that the text shadow is displayed below the text.
  - A negative value denotes that the text shadow is displayed above the text.
- c. **Blur:** Denotes the shadow blur of the widget by allowing you to soften text shadow.
- d. **Color:** Denotes the color of the text shadow. You can select a text shadow color by clicking the **Color Picker**.

## 10.3 Forking

Forking allows you to customize properties uniquely for a platform. You can fork:

- A Form
- A Widget Property

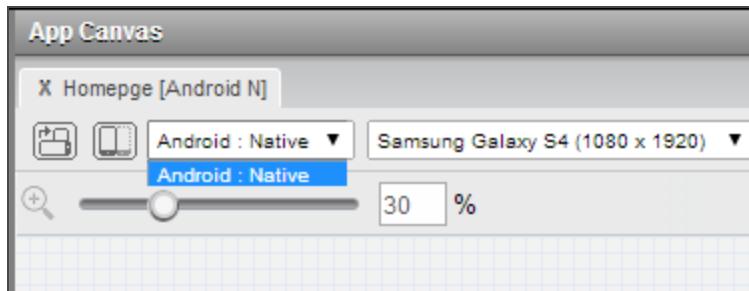
### 10.3.1 Forking a Form

Forking forms enables you to customize the position and availability of widgets on different platforms.

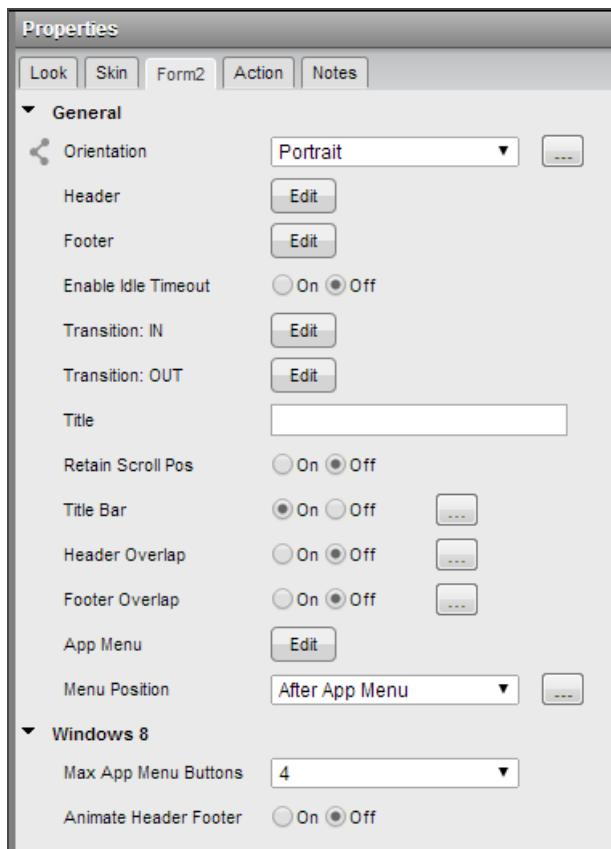
- You fork a form to:
  - position widgets differently on different platforms
  - make widgets available on one platform but not on another platform
  - replace a widget on one platform with a widget on another platform.

#### Important Considerations

1. You can fork Flex forms and VBox forms.
2. All the forked forms appear below their source forms in the Project Explorer.
3. Renaming a source form renames all its forked forms.
4. Deleting a source form deletes all its forked forms.
5. On **App Canvas**, the platform for which a form is forked appears on the platform list. For example, a form forked for Android Native platform will have only Android Native in the platform list.



6. A forked form only lists common properties and platform specific properties. For example, a form forked for Windows 8 Native lists only common properties and Windows 8 properties.



7. Default skins and user-defined skins are also forked during form forking.
8. Notes have only those comments made in the forked form's platform. For example, a form forked for Android Native, will have only the notes written on an Android device.

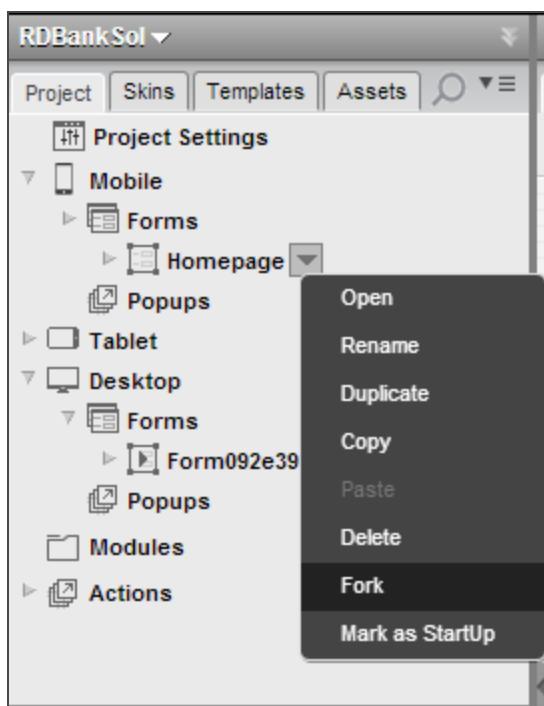
9. During forking, any unsupported widgets are ignored. For example, Camera widget is not supported on iOS web. If a form containing a Camera widget is forked for iOS web, the form will be forked, but the Camera widget is not forked.

#### 10.3.1.1 How to Fork a Form

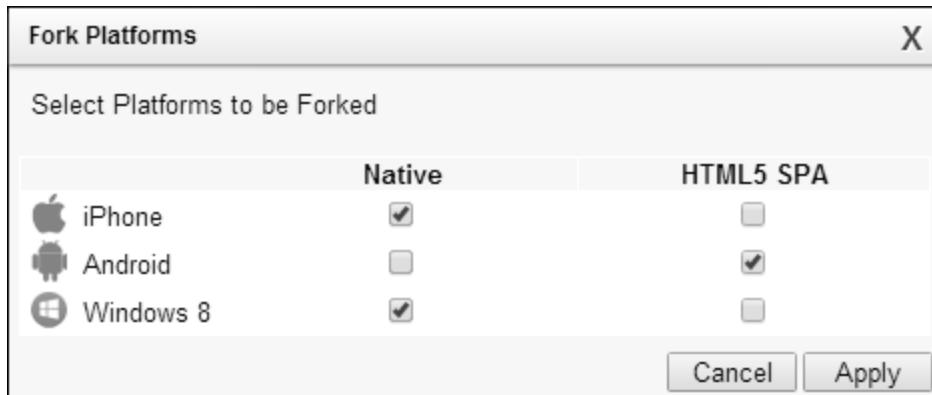
Forking allows you to customize a form specific to a platform. For example, you can fork a form to appear on an iOS (native) platform with green background and have the same form forked to appear on an Android (web) platform with yellow background.

To fork a form, follow these steps:

1. Go to Project Explorer > Project > Forms.

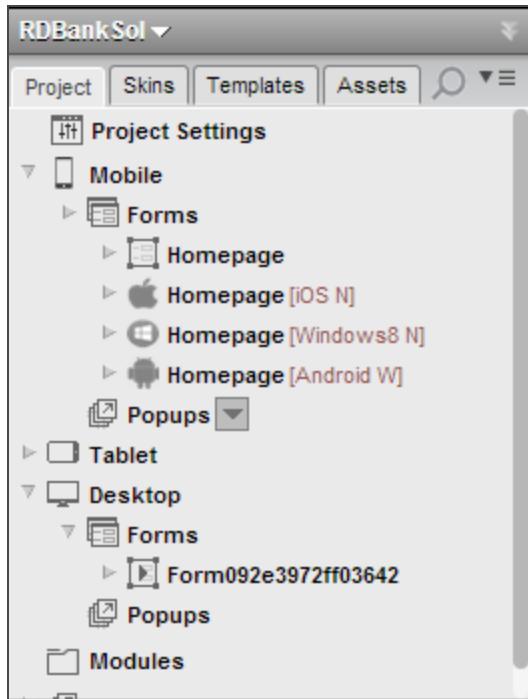


2. Right-click a form, and then click Fork.



The **Fork Platforms** window appears.

3. From the list of platforms, select the desired platforms, and then click **OK**.



The form is forked for the desired platforms.

### 10.3.2 Forking a Widget Property

Forking allows you to provide different values for the same property or skin, across the platforms.

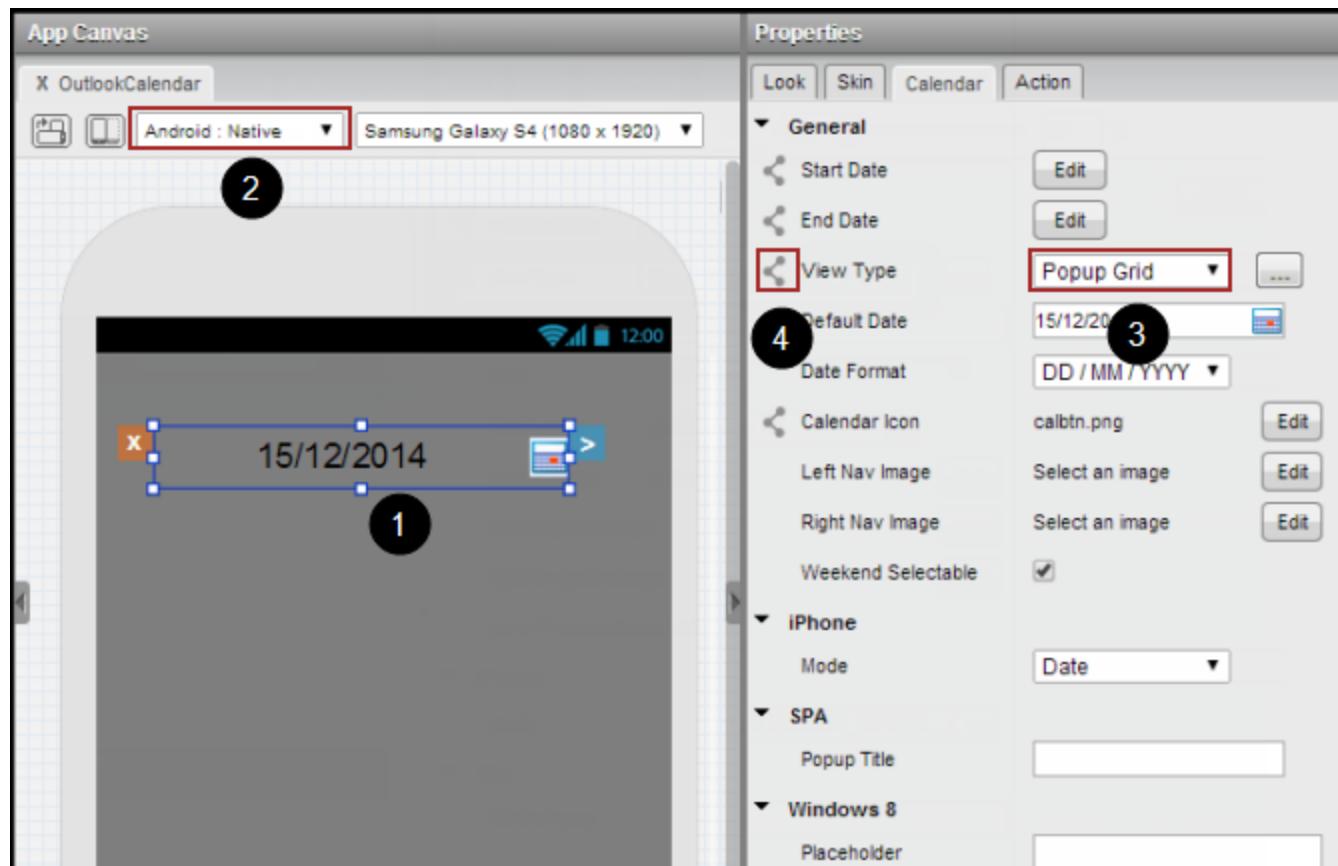
Only certain widget properties can be forked. You can identify whether a property can be forked by the presence of an icon ( ↗) to the left of the property's name. Two types of forking are available for you:

1. Simple Forking: Allows you to fork one platform at a time.
2. Complex Forking: Allows you to fork multiple platforms at the same time.

#### 10.3.2.1 Simple Forking

In this section, using **View Type** property of a **Calender** widget, let us understand simple forking. The below steps, however, can be used for forking any of the widget properties.

To fork **View Type** property of a Calender widget for **Android : Native** platform, follow these steps:



1. Select the **Calendar** widget from App Canvas.
2. Select **Android : Native** from the platform list.
3. Select **Popup Grid** from the **View Type** list.
4. Click the fork icon ( ) located to the left of the **View Type**. The icon changes to ( ), indicating that the **View Type** value is forked specifically for **Android : Native** platform.

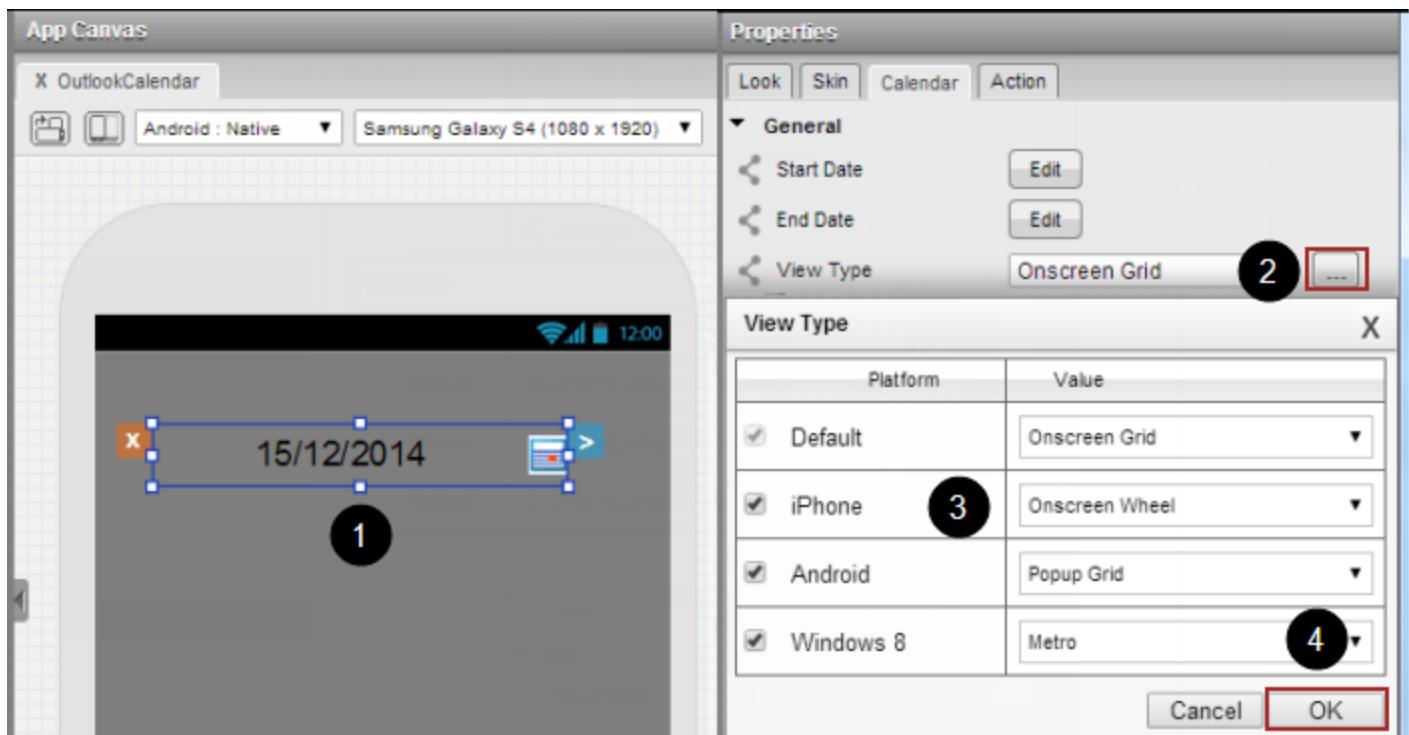
When you navigate to a different platform, the value of the forked property is replaced either with:

- A default value of the property.
- A forked property value, if the property is forked for that specific platform.

#### 10.3.2.2 Complex Forking

In this section, using **View Type** property of a **Calender** widget, let us understand complex forking. The below steps, however, can be used for forking any of the widget properties.

To fork **View Type** property of a **Calender** widget, follow these steps:



1. Select the **Calendar** widget from App Canvas.
2. Click the ellipsis (...) located to the right of **View Type** list. The View Type dialog box appears.
3. You can choose to:
  - a. Change the default **View Type**: From the **Value** list corresponding to the **Default** platform, select a new **View Type**.
  - b. Change the **View Type** of a specific platform: Select the platform check box, and from the corresponding **Value** list, choose a **View Type**.

**Note:** If you do not provide a **View Type** for a specific platform, the **Default** platform **View Type** value is applied.

## 11. Action Editor

The Action Editor allows you to assign actions to a form, pop-up, or a widget. Visualizer provides various predefined actions such as rotating a widget, updating widgets properties, and sending an email message. The actions assigned to a project are run in real-time environment or during functional preview.

This section details the following process:

- [Accessing Action Editor](#)
- [Adding and configuring an action](#)
- [List of Actions](#)

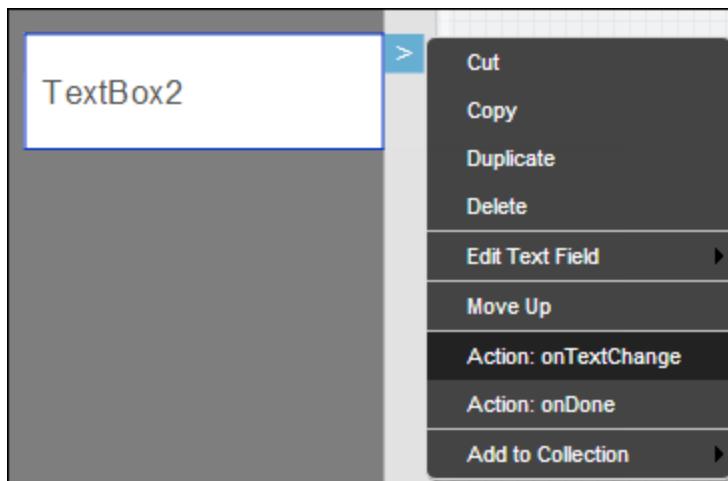
### 11.1 Important Considerations

- Animation actions are available only for flex widgets.
- You cannot run the actions on App Canvas.
- You cannot apply actions to the child widgets of a Segment2 widget.

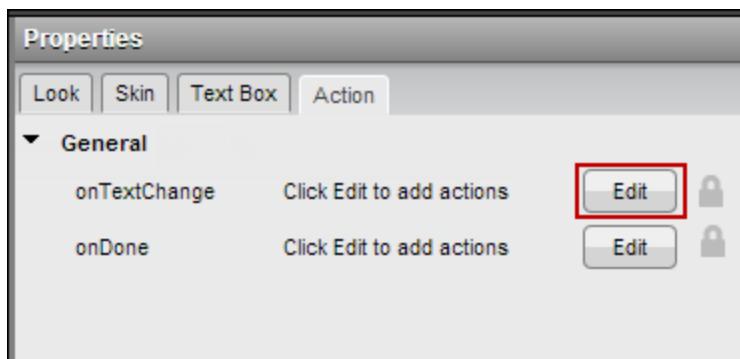
### 11.2 Accessing Action Editor

You can access the Action Editor dialog using one of the following options:

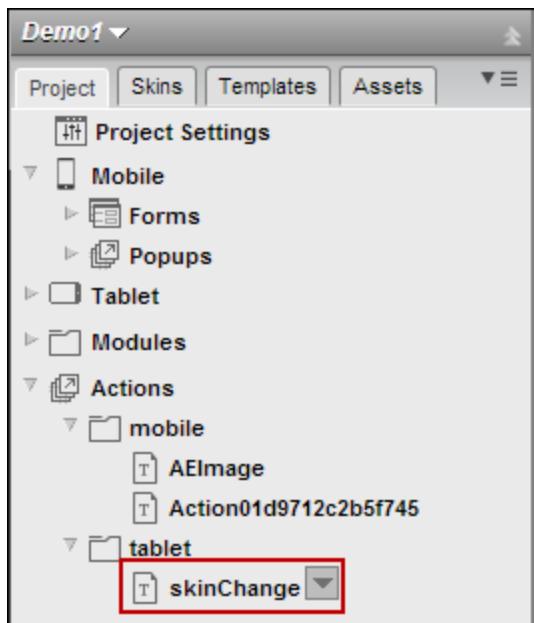
- **App Canvas:** Click on a required widget on App Canvas. A right-arrow appears (at the top-right of the selected widget.) Click this arrow, and then click the required action.



- **Action tab:** Click Action tab from a widget's Properties pane, and then click the Edit button against the required action.

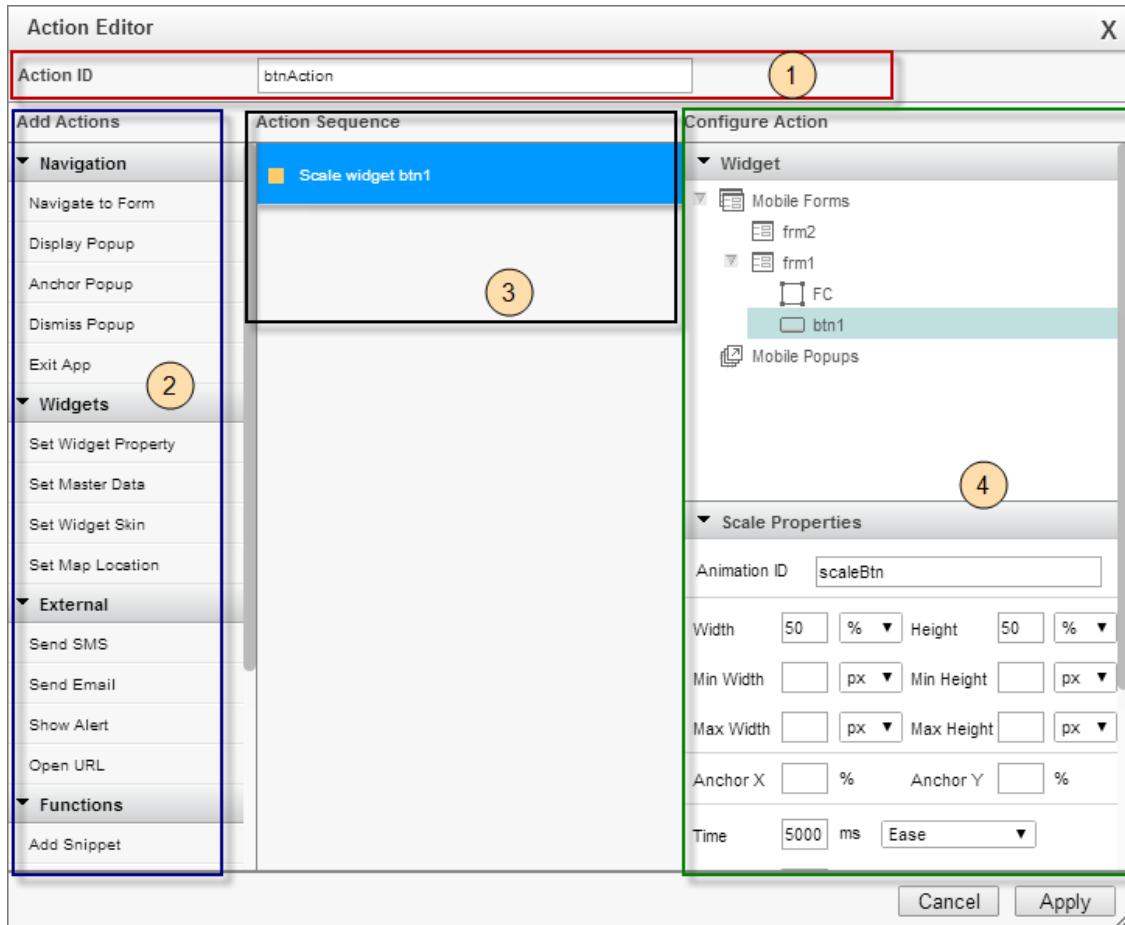


- Actions tab: Expand Actions > Mobile (or Tablet), and then click action.



### 11.3 Adding and configuring an action

To add and configure an action, follow these steps:



1. Action ID: By default, an ID is assigned to an action. You can rename the action ID, if required.

**Note:** If you are accessing the **Action Editor** dialog using the options [App Canvas](#) or [Action tab](#), and if you wanted an action to be available under [Actions](#) tab, you need to rename the Action ID.

2. Add Actions: From the list of available actions, you select the required action.
3. Action Sequence: You perform the following actions:
  - a. Structure the sequence of the actions, if required.
  - b. Click an action to open its configurable properties.

4. Configure Action: You perform the following actions:

- a. Select a form, pop-up, or a widget.
- b. Configure the properties.

### 11.3.1 Action ID

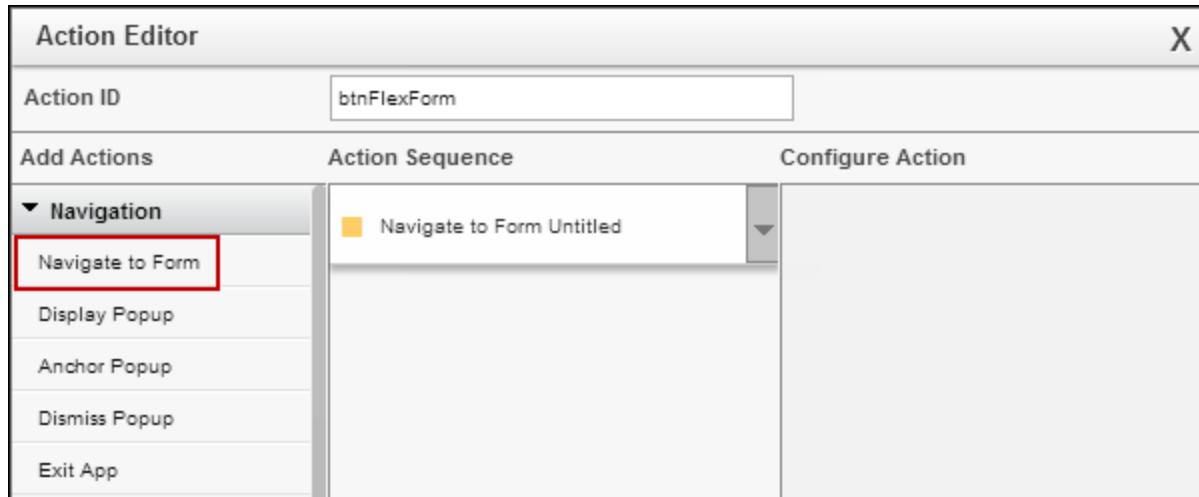
By default, an Action ID is assigned by the system.



Double-click inside the Action ID box to rename.

### 11.3.2 Add Actions

To add an action, from the left-pane of the Action Editor dialog, click the required action.

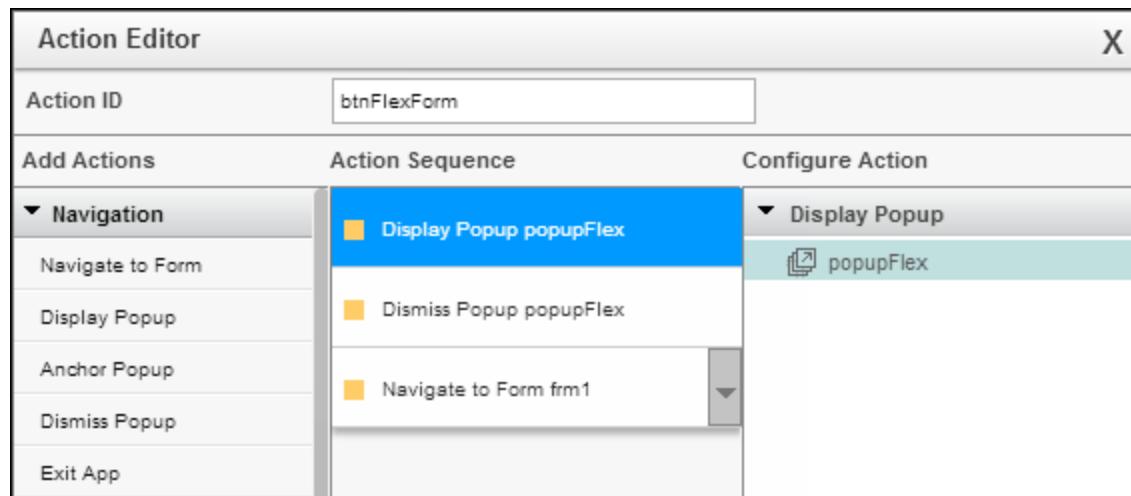


The action is added and appears under the Action Sequence.

For more details on actions, see [Actions](#).

### 11.3.3 Action Sequence

To configure an action, from the list of available actions, click an action.



Based on the action selected, **Configure Action** pane appears.

The following are the context menu items of an action:

- Move Up: Moves an action one-level above its existing position.

**Note:** This option is unavailable for the top action in the Action Sequence list.

- Move Down: Moves an action one-level below its existing position.

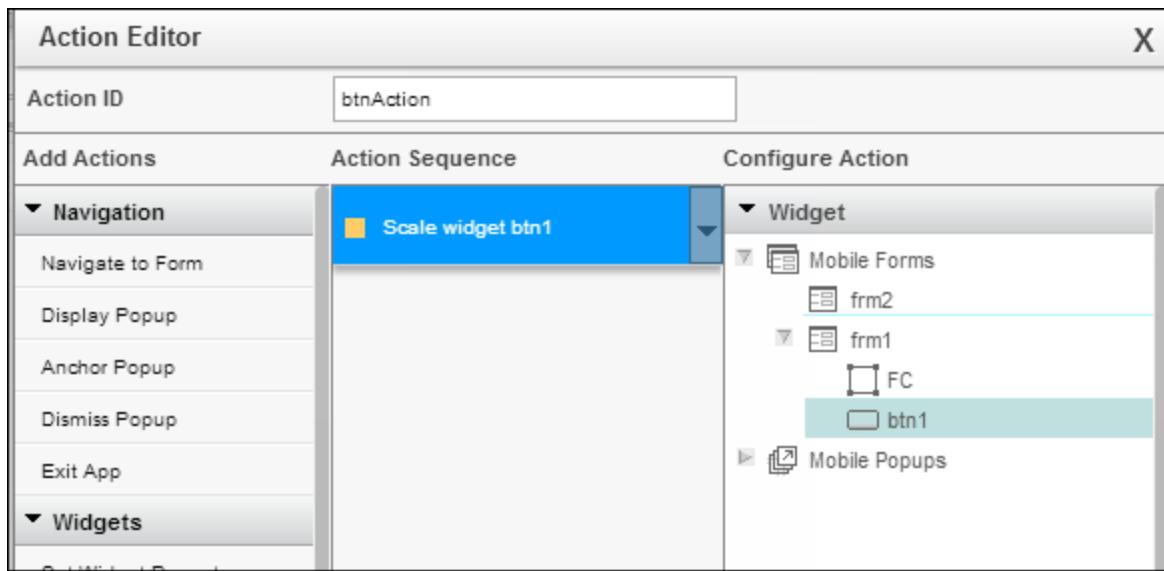
**Note:** This option is unavailable for the bottom action in the Action Sequence list.

- Delete: Deletes an action immediately.
- Duplicate: Duplicates an action.

### 11.3.4 Configure Actions

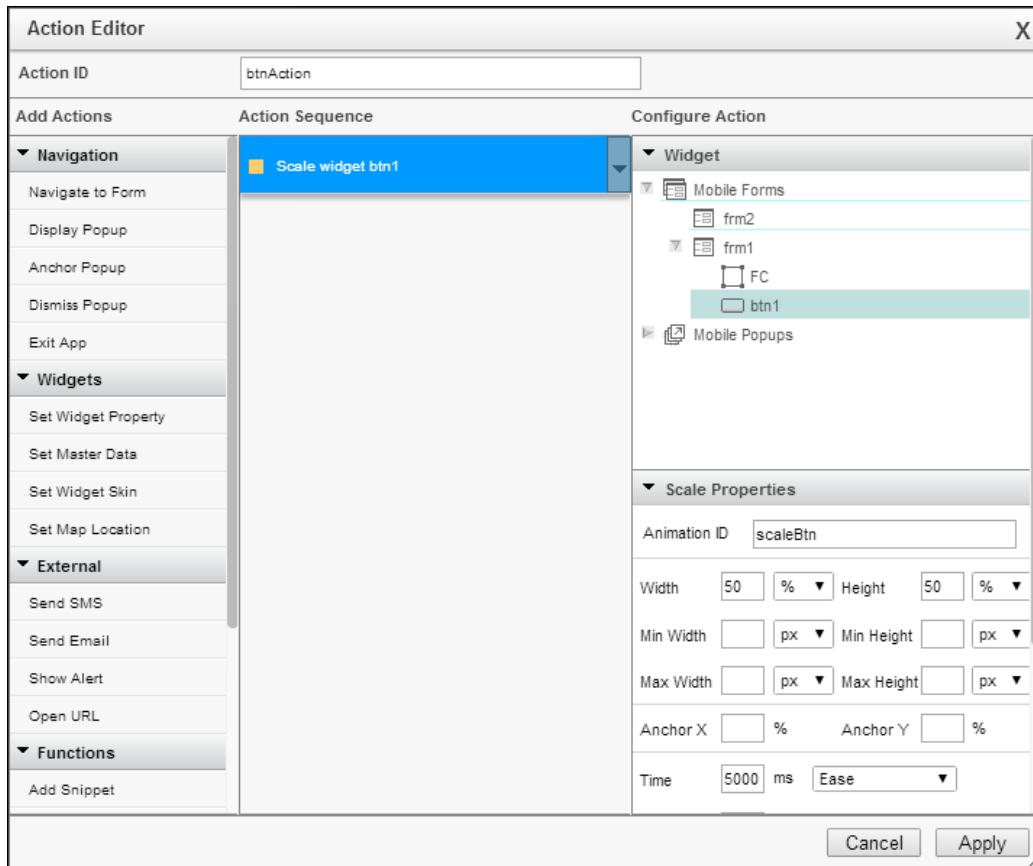
You need to perform the following action on this pane:

1. Select a form, pop-up, or widget.



**Note:** Instead of selecting a form, pop-up, or a widget, for certain actions such as Send SMS, Send Email, Show Alert, Open URL, Add Snippet, and Call Action, you can directly update the action properties.

2. Update the properties.



3. Click **Apply**.

For more details on updating properties for individual actions, see [Actions](#).

#### 11.4 List of Actions

The actions are grouped under the following types:

1. Navigation: These actions allow you to navigate from one form to other, display, anchor, dismiss a pop-up, or close an app.
2. Widgets: These actions allow you to set widget properties, update master data, replace a widget skin, or set a map location.

3. External: The actions allows you to send an SMS message, an email message, display an alert, or open a webpage.
4. Functions: These actions allows you to write code snippets or reuse an existing action.
5. Animations: These actions allows you to move, scale, rotate, transform, and change the background color of a flex widget.

### 11.4.1 Navigation

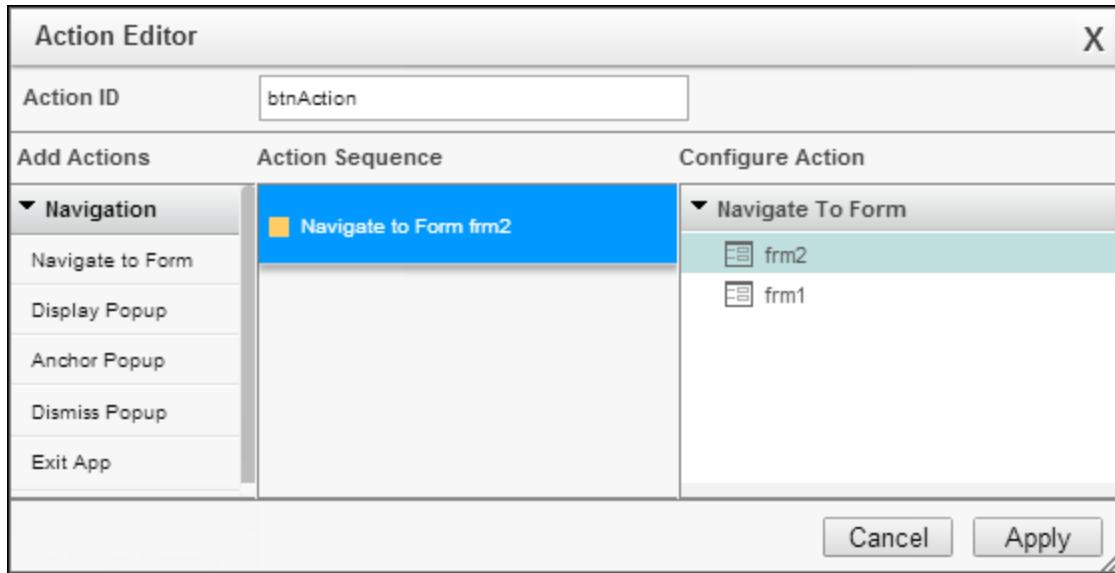
This section describes how to add the following actions:

1. [Navigate to a Form](#)
2. [Display a Pop-up](#)
3. [Anchor a Pop-up](#)
4. [Dismiss a Pop-up](#)
5. [Exit an App](#)

#### 11.4.1.1 Navigate to a Form

Allows you to navigate from one form to other.

**To navigate from one form to other, follow these steps:**

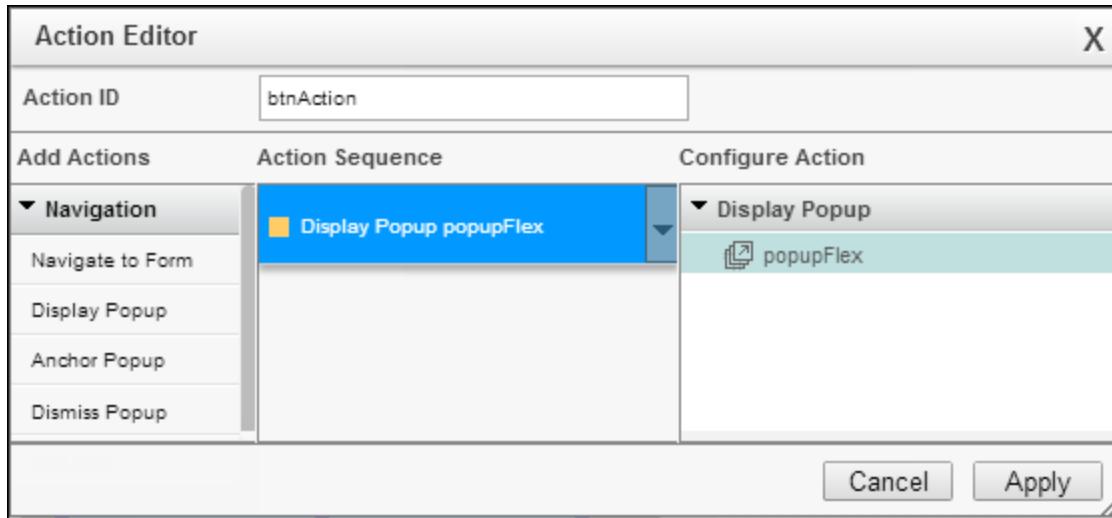


1. Click **Navigate to Form** from **Add Actions > Navigation**.
2. Click **Navigate to Form** from **Action Sequence**.
3. Select a form from the **Navigate To Form** tab.
4. Click **Apply**.

#### 11.4.1.2 Display a Pop-up

Allows you to display a pop-up.

To display a pop-up, follow these steps:



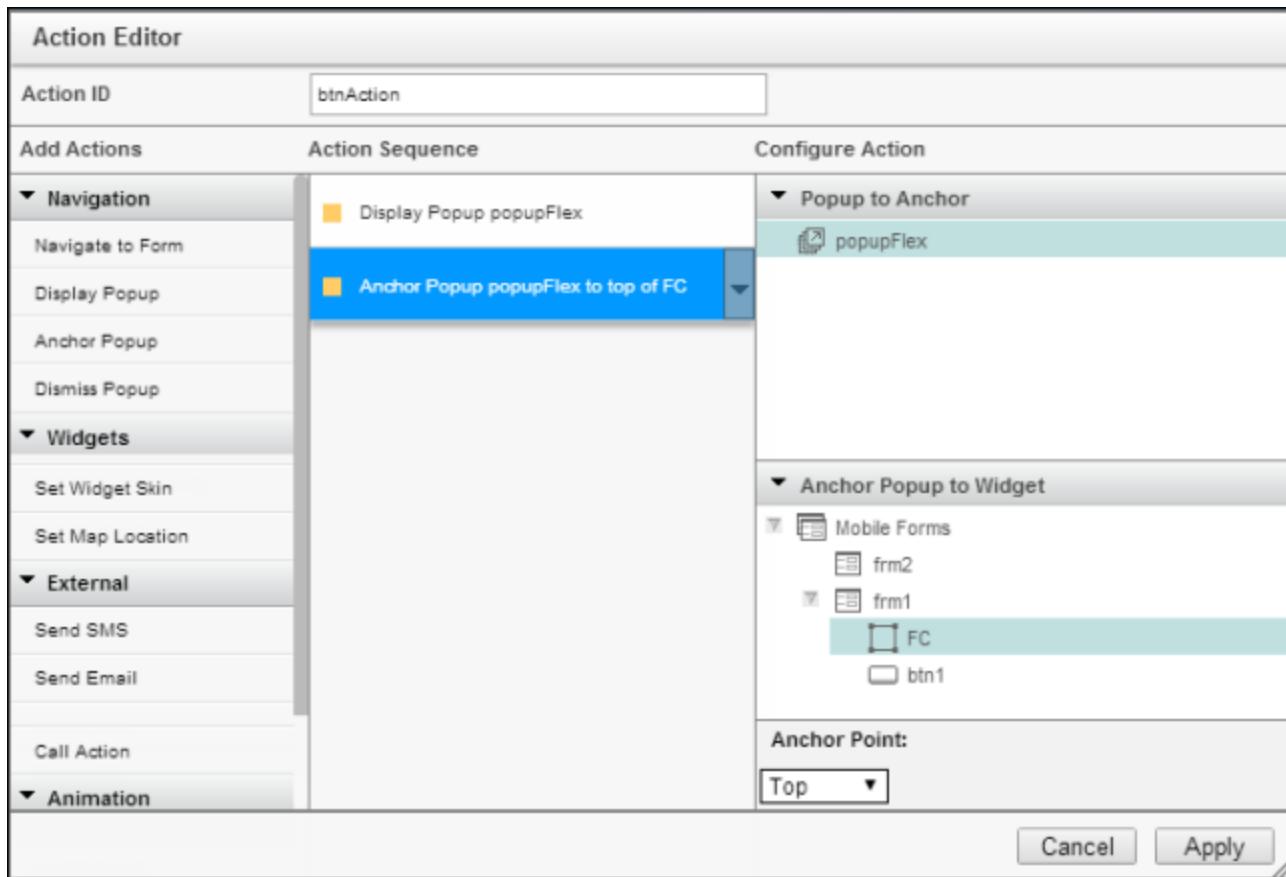
1. Click Display Popup from Add Actions > Navigation.
2. Click Display Popup from Action Sequence.
3. Select a pop-up from the Display popup tab.
4. Click Apply.

#### 11.4.1.3 Anchor a Pop-up

Allows you to anchor a pop-up.

**Note:** You have to [display a popup](#) before you anchor the pop-up.

To anchor a pop-up, follow these steps:



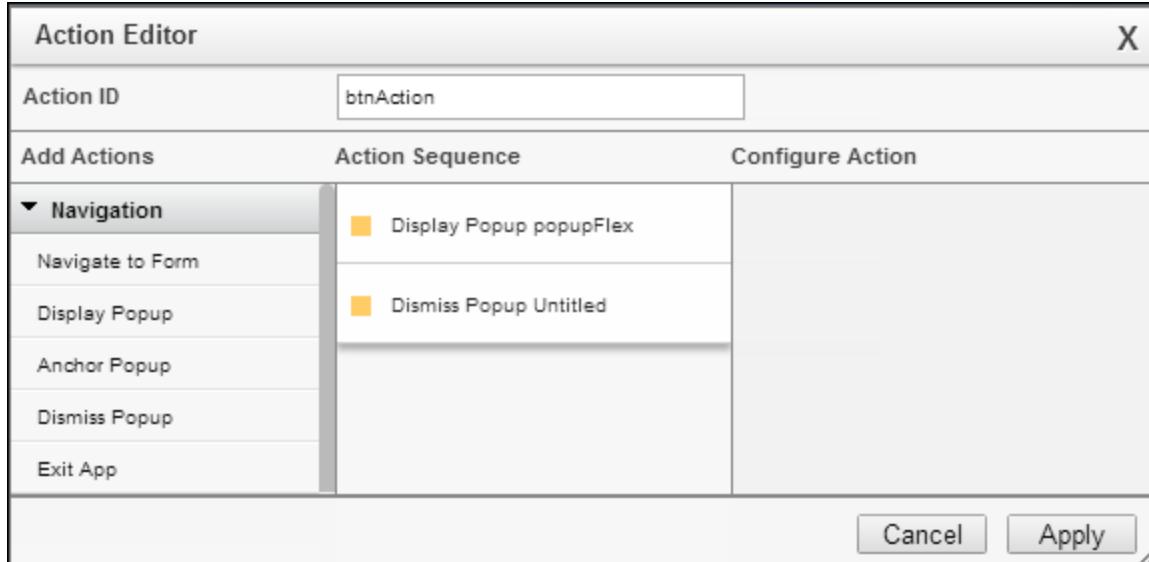
1. Click Anchor Popup from Add Actions > Navigation.
2. Click Anchor Popup from Action Sequence.
3. Select a pop-up from the Popup to Anchor tab.
4. Select a widget from the Anchor Popup to Widget tab.
5. Select an Anchor Point.
6. Click Apply.

#### 11.4.1.4 Dismiss a Pop-up

Allows you to close a pop-up.

**Note:** You have to [display a popup](#) before you can dismiss the pop-up.

To close a pop-up, follow these steps:

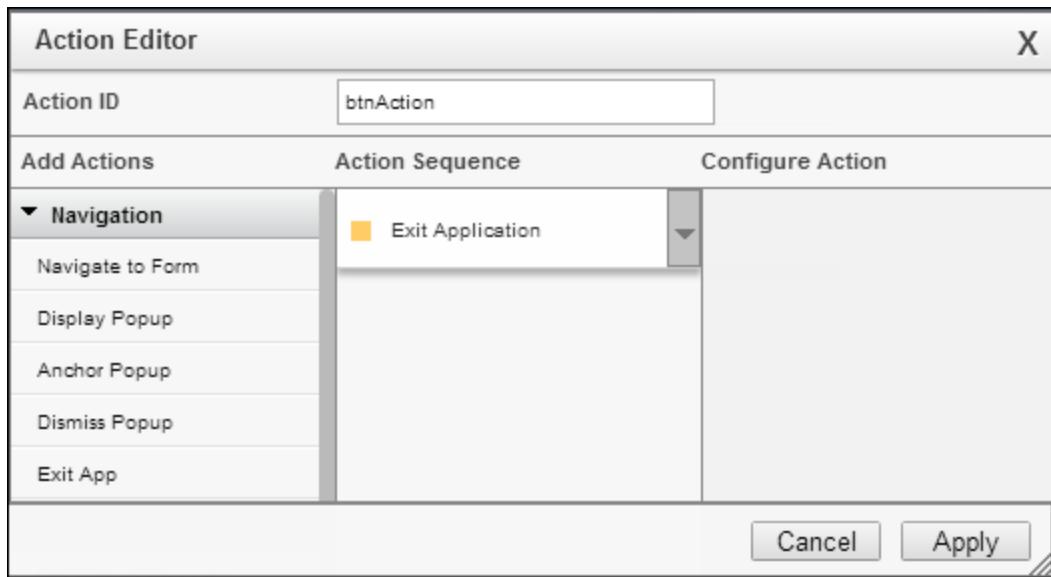


1. Click **Dismiss Popup** from **Add Actions > Navigation**.
2. Click **Dismiss Popup** from **Action Sequence**.
3. Select a pop-up from **Dismiss Popup**.
4. Click **Apply**.

#### 11.4.1.5 Exit an App

Allows you to close an application.

To close application, follow these steps:



1. Click **Exit App** from **Add Actions > Navigation**.
2. Click **Exit Application** from **Action Sequence**.
3. Click **Apply**.

## 11.4.2 Widgets

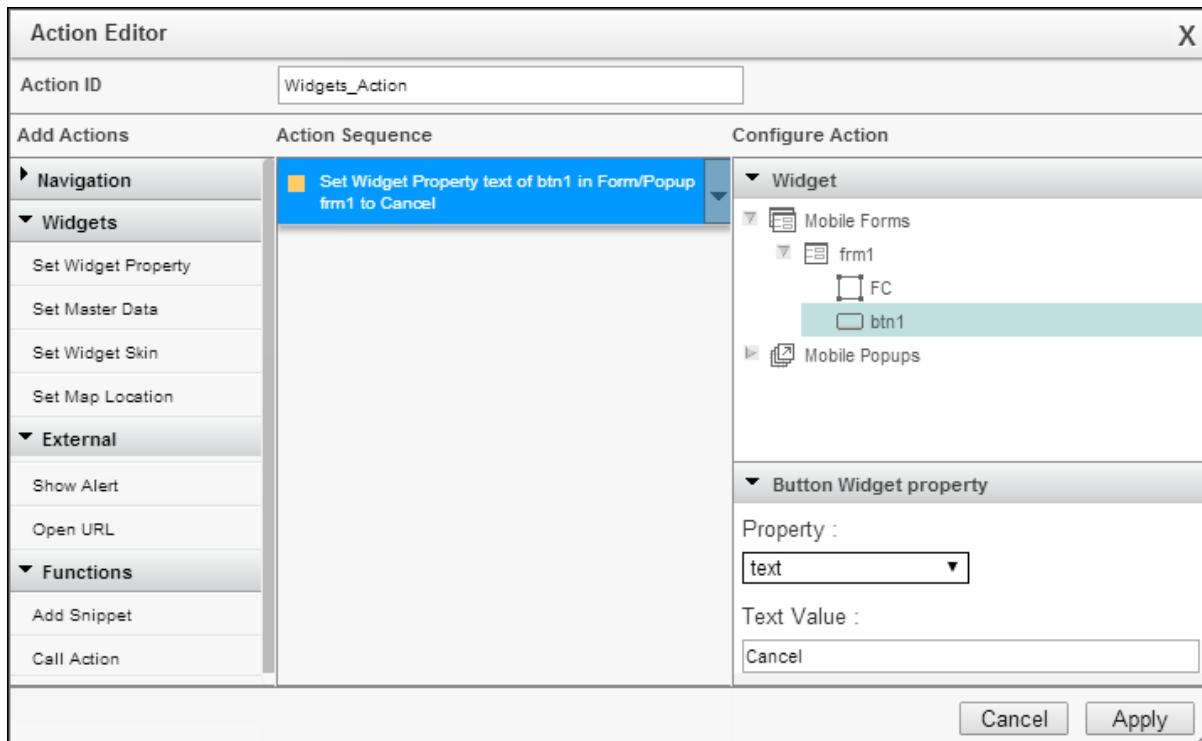
This section describes how to add the following actions:

1. [Set a Widget Property](#)
2. [Set Master Data](#)
3. [Set Widget Skin](#)
4. [Set Map Location](#)

### 11.4.2.1 Set a Widget Property

Allows you to update a widget's property.

To update a widget's property, follow these steps:



1. Click Set Widget Property from Add Actions > Widgets.
2. Click Set Widget Property from Action Sequence.
3. Select a widget from the Widget tab.
4. In the Button Widget Property tab,
  - a. Select a property from the Property list. For example, **text** property for a Button widget.
  - b. Based on the select property, you receive an option to modify the widget's property. For example, you can rename the Button widget.
5. Click Apply.

#### 11.4.2.2 Set Master Data

Allows for setting the master data for widgets such as CheckBoxGroup, ComboBox, DataGrid, RadioButtonGroup, PickerView, and Segment2.

To update master data, follow these steps:

The screenshot shows the Kony Visualizer Action Editor interface. The main window is divided into three tabs: 'Add Actions', 'Action Sequence', and 'Configure Action'. The 'Configure Action' tab is active.

- Add Actions:** A tree view showing 'Navigation' and 'Widgets' categories. Under 'Widgets', 'Set Widget Property', 'Set Master Data', and 'Set Widget Skin' are listed.
- Action Sequence:** A list containing a single item: 'Set MasterData of Countries of Form'.
- Configure Action:**
  - Widget:** A list of widget types: Mobile Forms, Dashboard, CheckBoxGroup (which is selected and highlighted in blue), and Mobile Popups.
  - Set Master Data:** A button labeled 'Set Master Data'.

A modal dialog titled 'Master Data : CheckBoxGroup' is open at the bottom. It contains a table with three rows, each representing a country with a key and display value:

| Key | Display Value | Selected Key                        |
|-----|---------------|-------------------------------------|
| 1   | India         | <input checked="" type="checkbox"/> |
| 2   | USA           | <input type="checkbox"/>            |
| 3   | France        | <input type="checkbox"/>            |

At the bottom of the dialog are 'Add', 'Delete', 'Apply', and 'Cancel' buttons. Below the dialog are global 'Apply' and 'Cancel' buttons.

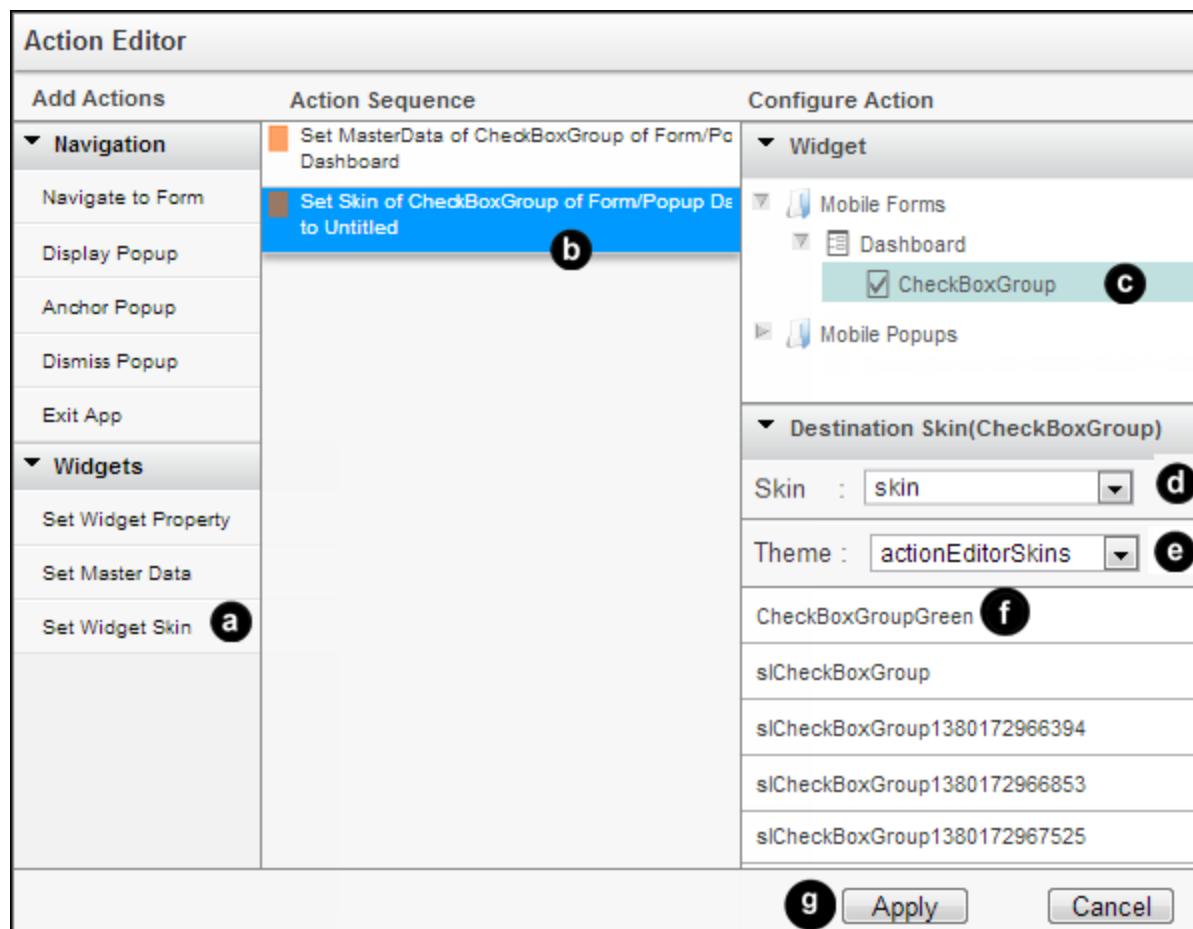
1. Click **Set Master Data** from Add Actions > Widgets.
2. Click **Set Master Data** from Action Sequence.
3. Select a widget from the Widget tab.
4. Click **Set Master Data**.

5. Update the master data of the widget.
6. Click Apply.

#### 11.4.2.3 Set Widget Skin

Allows for changing the skin of the widget based on the action selected.

To change the skin of a widget, follow these steps:



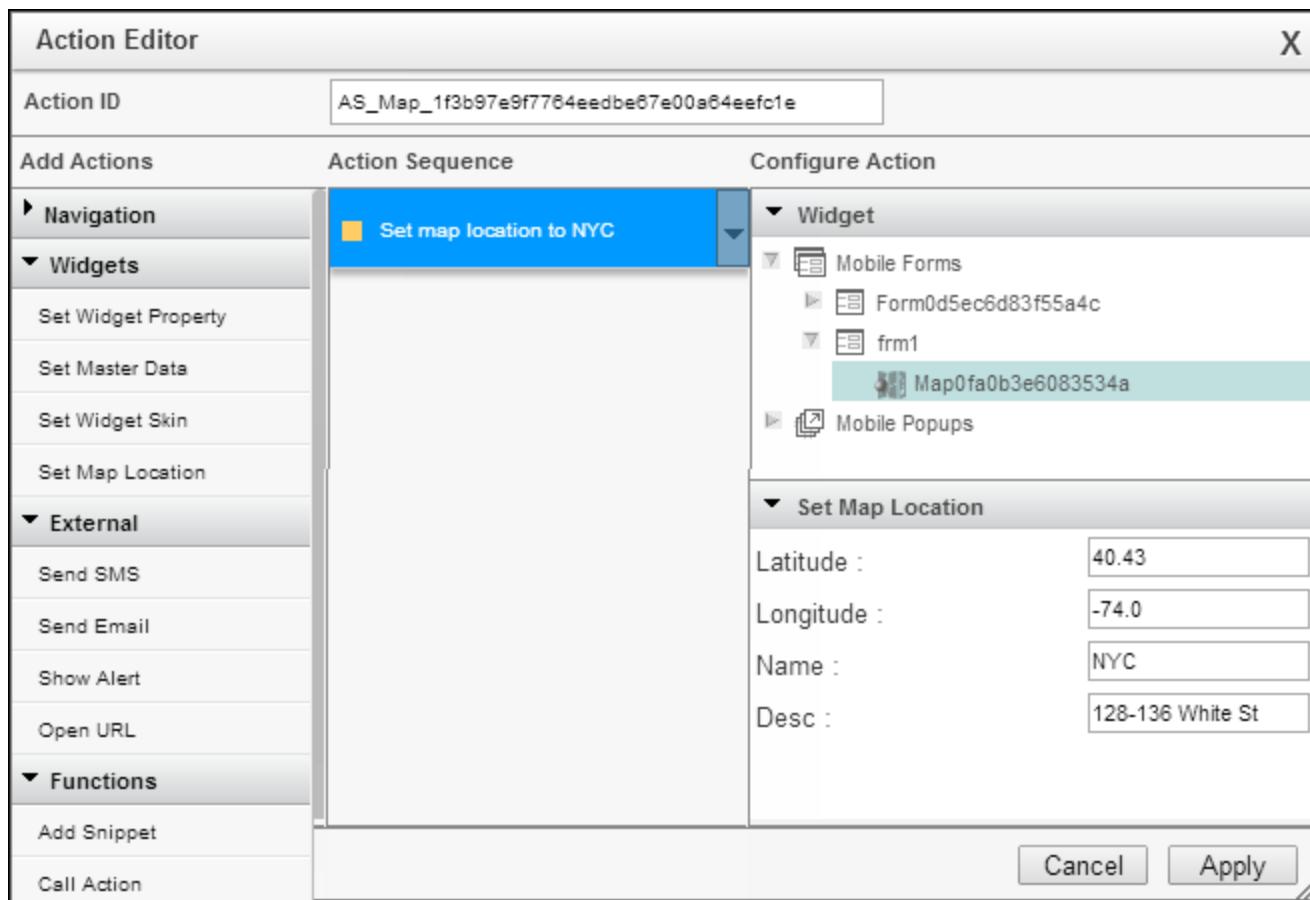
1. Click Set Widget Skin from Add Actions > Widgets.
2. Click Set Widget Skin from Action Sequence.
3. Select a widget from the Widget tab.

4. Click **Set Master Data**.
5. On the Destination Skin (widget),
  - a. From the **Skin** list, select the state of widget skin.
  - b. Select a theme from the **Theme** list. All the available skins from the chosen theme for the widget are displayed
  - c. Select a skin.
6. Click Apply.

#### 11.4.2.4 Set Map Location

Allows you to assign map coordinates to a Map widget.

To update Map coordinates, follow these steps:



1. Click **Set Map Location** from **Add Actions > Widgets**.
2. Click **Set Map Location** from **Action Sequence**.
3. Select a Map widget from the **Widget** tab.
4. In the **Set Map Location** tab,
  - a. Type the latitude of the location in the **Latitude** box.
  - b. Type the longitude of the location in the **Longitude** box.
  - c. Type the name of the location in the **Name** box.
  - d. Type the description of the location in the **Desc** box.
5. Click **Apply**.

### 11.4.3 External

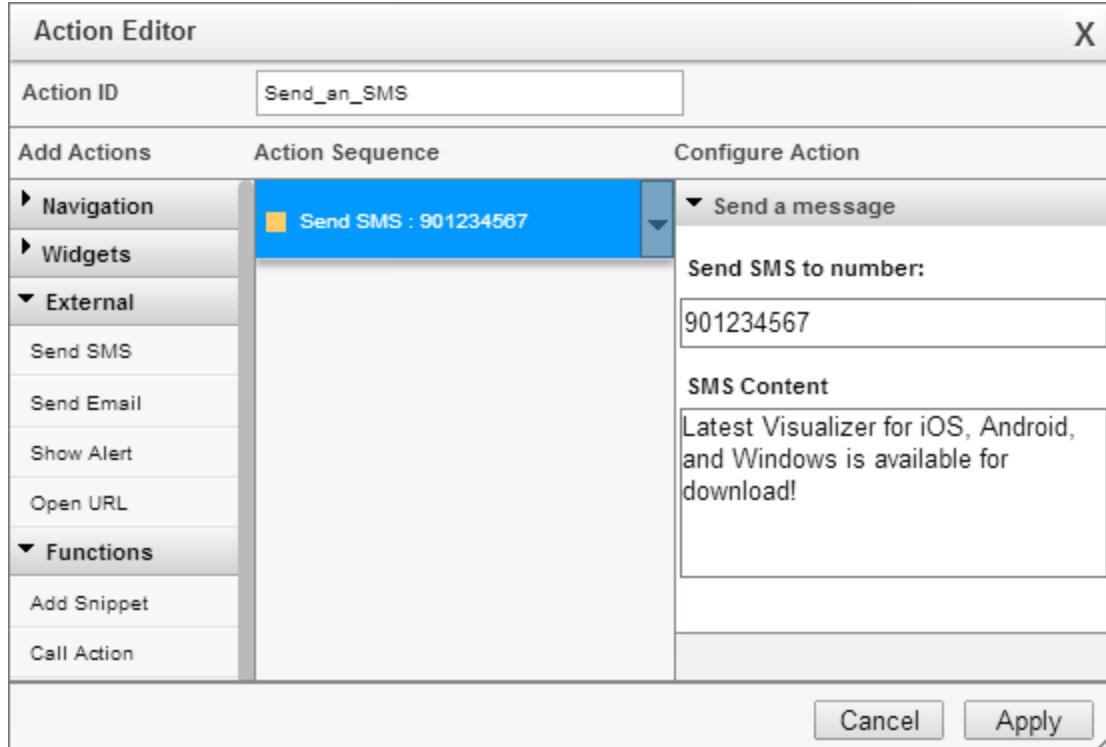
This section describes how to add the following actions:

1. [Send SMS](#)
2. [Send Email](#)
3. [Show Alert](#)
4. [Open URL](#)

#### 11.4.3.1 Send SMS

Allows you to send a text message.

To send a text message, follow these steps:

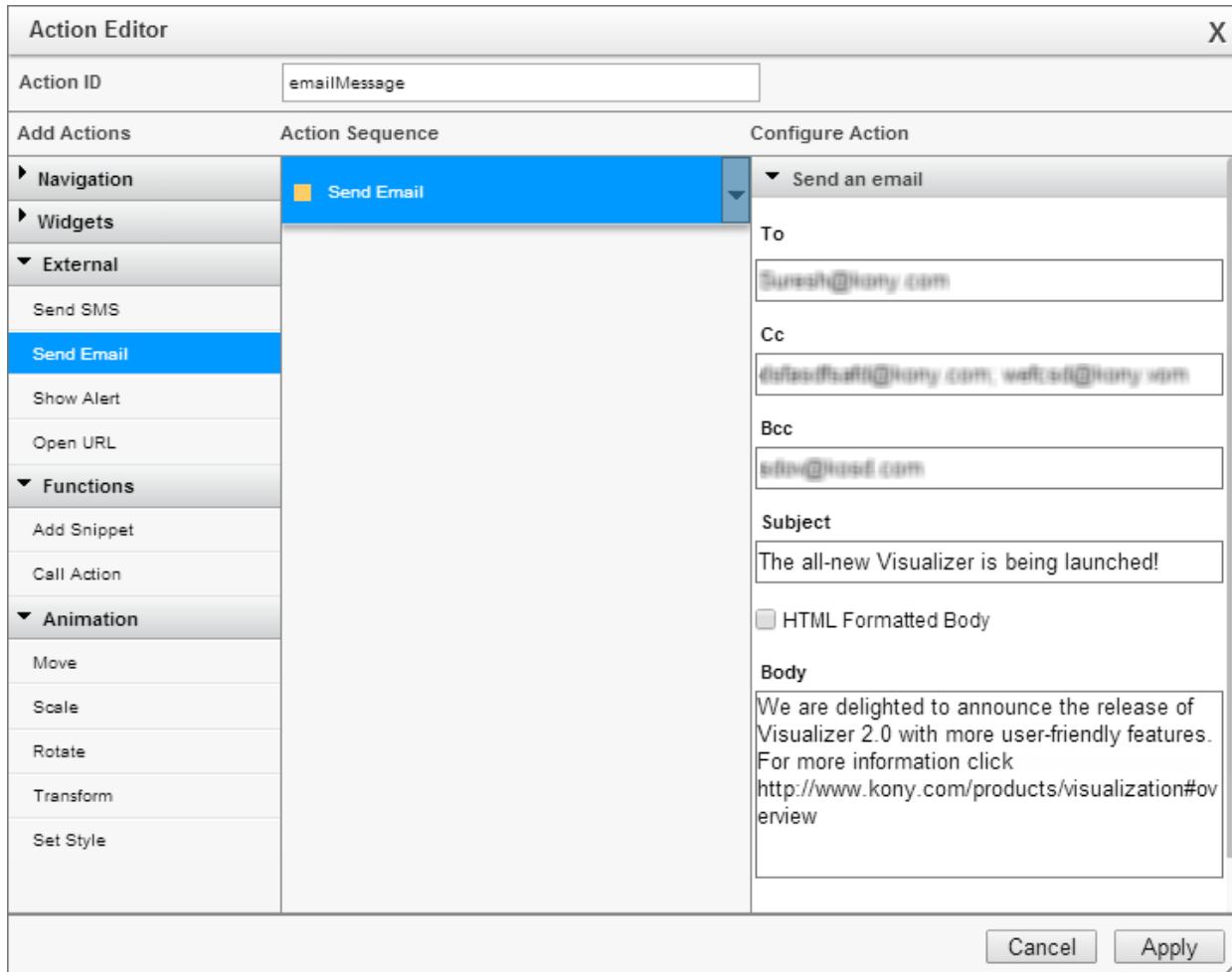


1. Click **Send SMS** from **Add Actions > External**.
2. Click **Send SMS** from **Action Sequence**.
3. In **Configure Action**, under **Send a message**, type the following details:
  - a. Type the mobile number in the **Send SMS to number** box.
  - b. Type the message in the **SMS Content** box.
4. Click **Apply**.

#### 11.4.3.2 Send Email

Allows you to send an email messages.

To send an email messages, follow these steps:



1. Click **Send Email** from **Add Actions > External**.
2. Click **Send Email** from **Action Sequence**.
3. In **Configure Action**, under **Send an email**, type the following details:
  - a. Type the receiver's email address in the **To** box.
  - b. To copy the message, type their email address in the **Cc** box.
  - c. To blind copy the message, type their email address in the **Bcc** box.
  - d. Type the subject of the mail in the **Subject** box.

- e. Select the HTML Formatted Body to enable HTML formatting of the mail.
  - f. Type the message in the **Body** box.
4. Click **Apply**.

#### 11.4.3.3 Show Alert

Displays an alert message to the user.

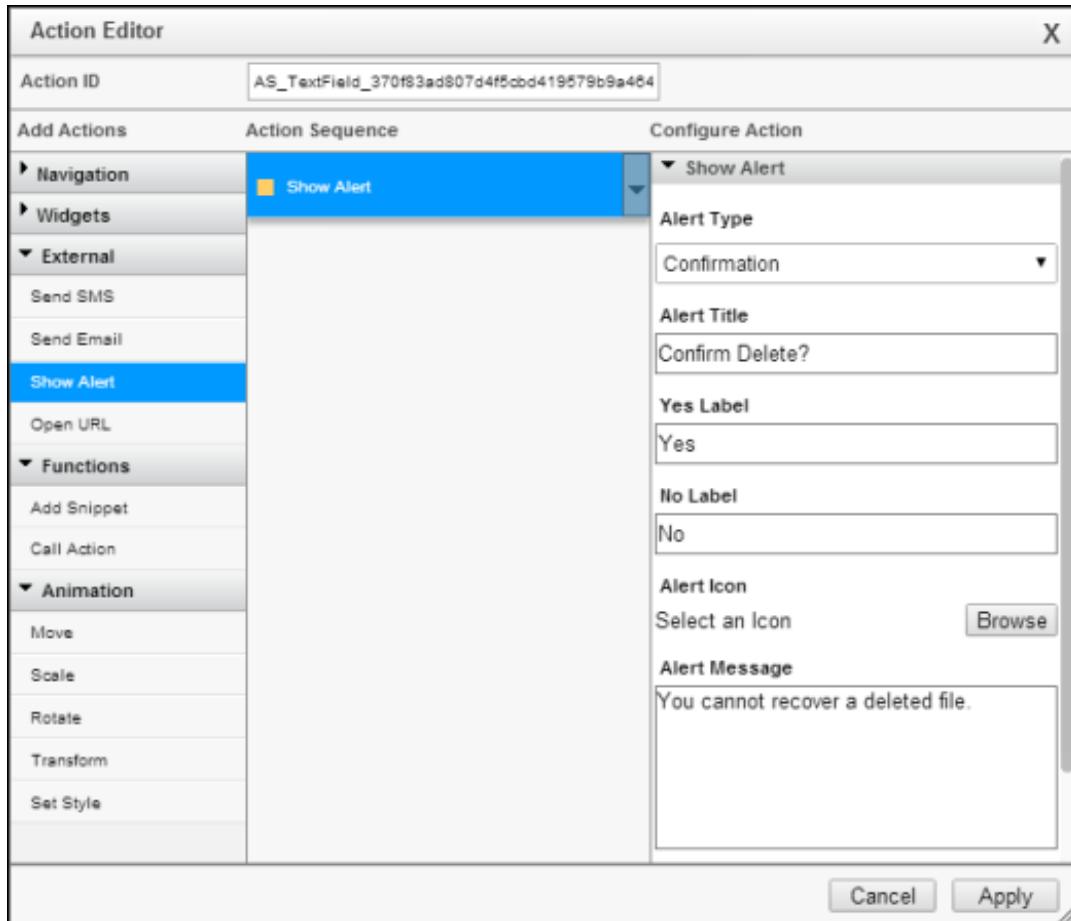
Alerts are of three types:

- **Error:** An error message is displayed on the screen.
- **Information:** An informative message is displayed on the screen. This message can be in turn a warning or a success message.
- **Confirmation:** A confirmation message with Yes and No options is displayed on the screen.

All the alerts are modal in nature. That is, the user cannot proceed with other UI operations unless the alert is dismissed.

**Note:** Do not use Show Alert multiple times in the same action sequence as it could lead to erroneous behavior of the application.

To show an alert, follow these steps:



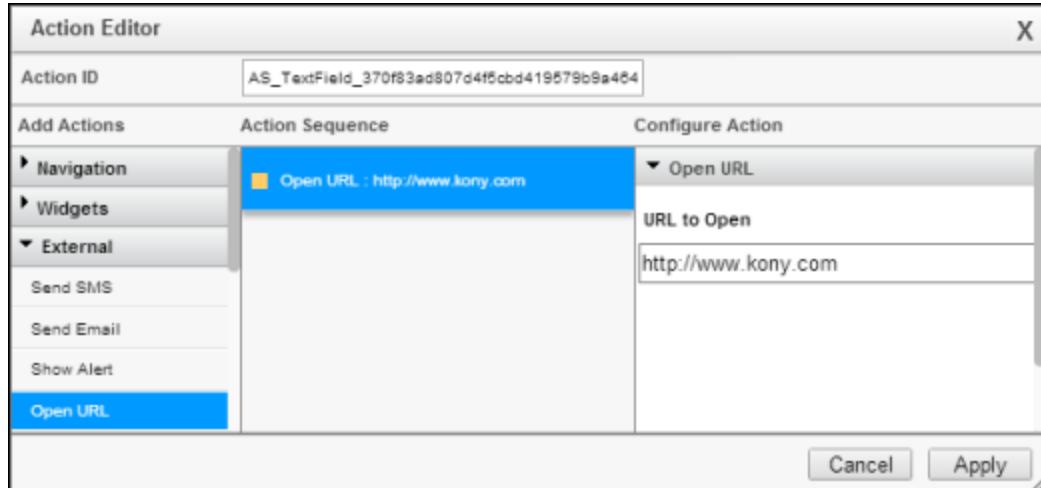
1. Click **Show Alert** from Add Actions > External.
2. Click **Show alert** from Action Sequence.
3. In **Configure Action**, under **Show Alert**, type the following details:
  - a. Select an alert type for the **Alert Type** list.
  - b. Type the alert title in the **Alert Title** box.
  - c. Type a value for **Yes label**.
  - d. Type a value for **No Label**.

- e. Select an icon from the list of existing images by clicking **Browse**.
- f. Type the alert message in the **Alert Message** box.
- g. Type a function's name that should be called when an alert is dismissed either through **Yes Label** or through **No Label** button.

#### 11.4.3.4 Open URL

Displays a webpage to the user.

To display a webpage, follow these steps:



1. Click **Open URL** from **Add Actions > External**.
2. Click **Open URL** from **Action Sequence**.
3. In **Configure Action**, under **Open URL**, type the webpage address in the **URL to Open** box.
4. Click **Apply**.

#### 11.4.4 Functions

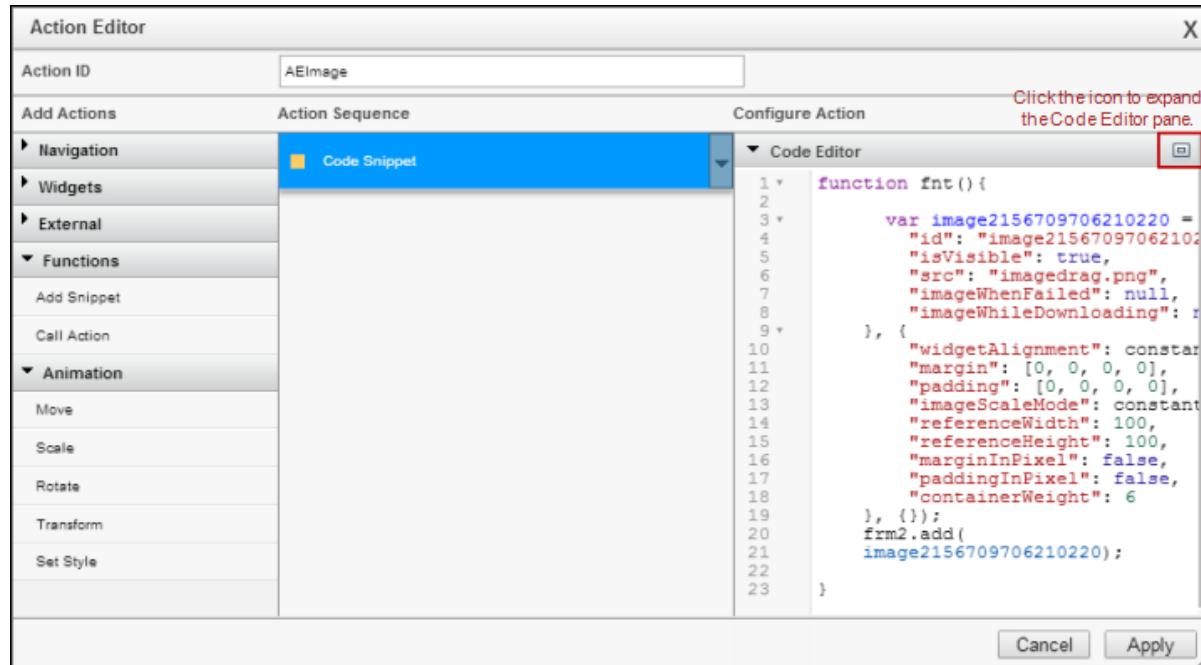
This section describes how to add the following actions:

- [Add Snippet](#)
- [Call Action](#)

### 11.4.5 Add Snippet

Allows you to add a snippet of code to run an action.

To add a snippet of code follow these steps:

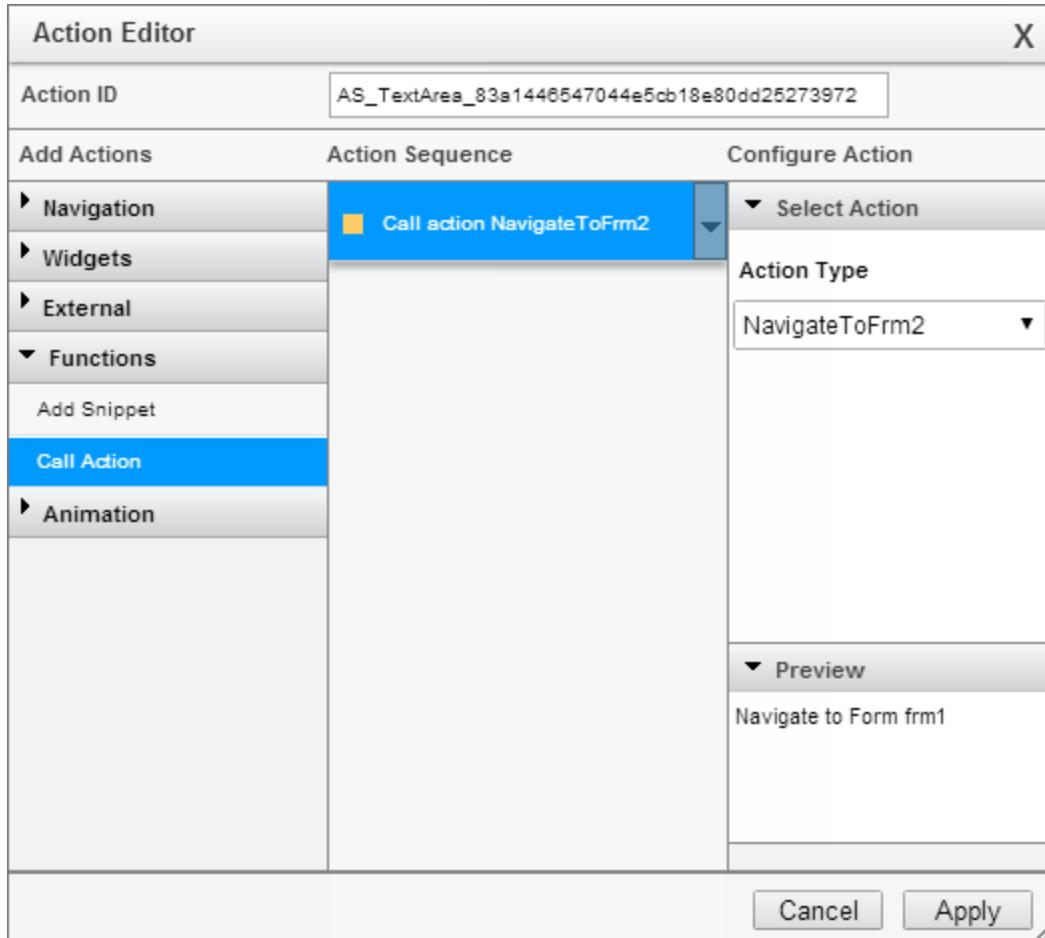


1. Click Add Snippet from Add Actions > Functions.
2. Click Code Snippet from Action Sequence.
3. In Configure Action, under Code Editor, type the code snippet.
4. Click Apply.

#### 11.4.5.1 Call Action

List all the available actions under the [Actions](#) tab.

To run Call Action function, follow these steps:



1. Click Call Action from Add Actions > Functions.
2. Click Call action from Action Sequence.
3. In Configure Action, under Select Action, from the Action Type drop-down, select an action.

**Note:** Preview of the action appears in the Preview tab.

4. Click Apply.

### 11.4.6 Animation

**Note:** Animation actions are available only for flex widgets.

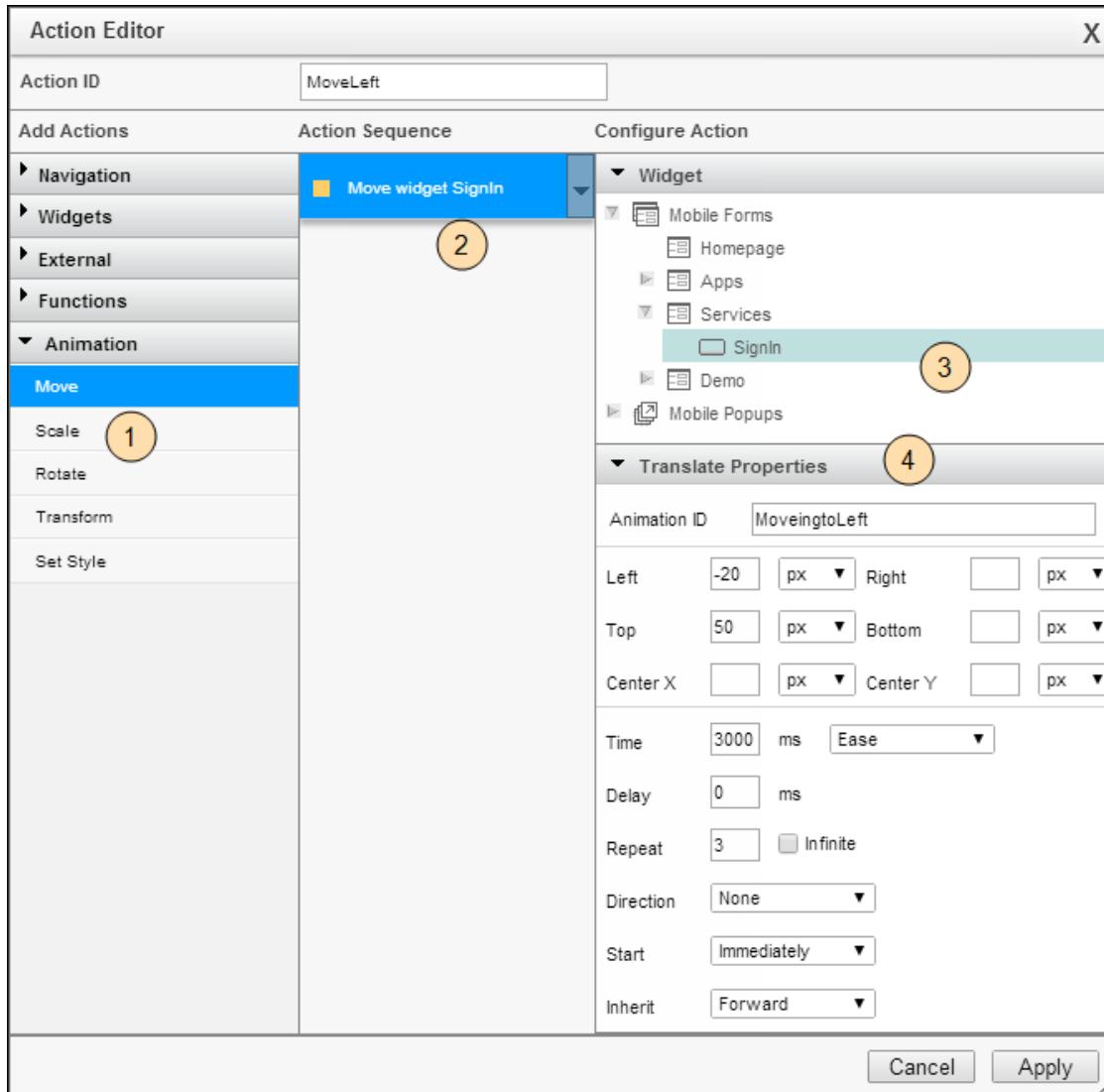
The following are the animation actions:

- [Move](#)
- [Scale](#)
- [Rotate](#)
- [Transform](#)
- [Set Style](#)

#### 11.4.6.1 Move

Moves a widget within its parent widget or a form.

For moving a widget, follow these steps:



1. Click **Move** from the Add Actions > Animation.
2. Click **Move Widget** from the Action Sequence.

3. Select a widget from the **Widget** tab.
4. Update the properties from the **Translate Properties** tab with the following details:

| Property     | Description   | Examples            |
|--------------|---|---------------------|
| Animation ID | Animation ID.   | Move_left           |
| Left         | Specifies the distance a widget moves to the left from its original position.           | -20 px, 100 dp, 15% |
| Right        | Specifies the distance a widget moves to the right from its original position.          | 10 px, 175 dp, 25%  |
| Top          | Specifies the distance a widget moves to the top from its original position.            | -15 px, 200 dp, 7%  |
| Bottom       | Specifies the distance a widget moves to the bottom from its original position.         | 23 px, 225dp, 6%    |
| Center X     | Specifies the distance a widget's center moves horizontally from its original position. | -21 px, 263 dp, 11% |
| Center Y     | Specifies the distance a widget's center moves vertically from its original position.   | -15 px, 200 dp, 7%  |
| Time         | Specifies the number of milliseconds (ms) in which an animation is completed.           | 1000                |

| Property      | Description   | Examples           |
|---------------|---|--------------------|
| Time Function | <p>The following time functions are available:</p> <ul style="list-style-type: none"> <li>• Ease: Specifies a transition effect with a slow start, then fast, then end slowly.</li> <li>• Linear: Specifies a transition effect with the same speed from start to end.</li> <li>• Ease-In: Specifies a transition effect with a slow start.</li> <li>• Ease-Out: Specifies a transition effect with a slow end.</li> <li>• Ease-In-Out: Specifies a transition effect with a slow start and end.</li> </ul> |                    |
| Delay         | Specifies the number of milliseconds (ms) to wait before starting an animation.   | 2000               |
| Repeat        | Specifies the number of times an action is repeated. To run the action indefinitely, select the <b>Infinite</b> check box.  | 5                  |
| Direction     | <p>Specifies the direction of an action. The following are the available options:</p> <ul style="list-style-type: none"> <li>• None: The animation action direction is always same.</li> <li>• Alternate: The animation action direction is same every odd time, and in reversed direction every even time.</li> </ul>  | None,<br>Alternate |

| Property | Description  | Examples    |
|----------|--|-------------|
| Start    | <p>Specifies how an animation action starts. The following options are available:</p> <ul style="list-style-type: none"><li>• Immediately: Start the animation immediately, with delay, if selected.</li><li>• With Animation: Action is run along with another action.</li><li>• After Animation: Action is run after completing another action.</li></ul>  | Immediately |
| Inherit  | <p>The following options are available:</p> <ul style="list-style-type: none"><li>• None: Upon completing this action, widget returns to its original size.</li><li>• Forward: Upon completing this action, widget stays at final position.</li><li>• Backward: Upon completing an action, widget will come return to the original position.</li><li>• Both: Applies Forward and Backward options.</li></ul> | Both        |

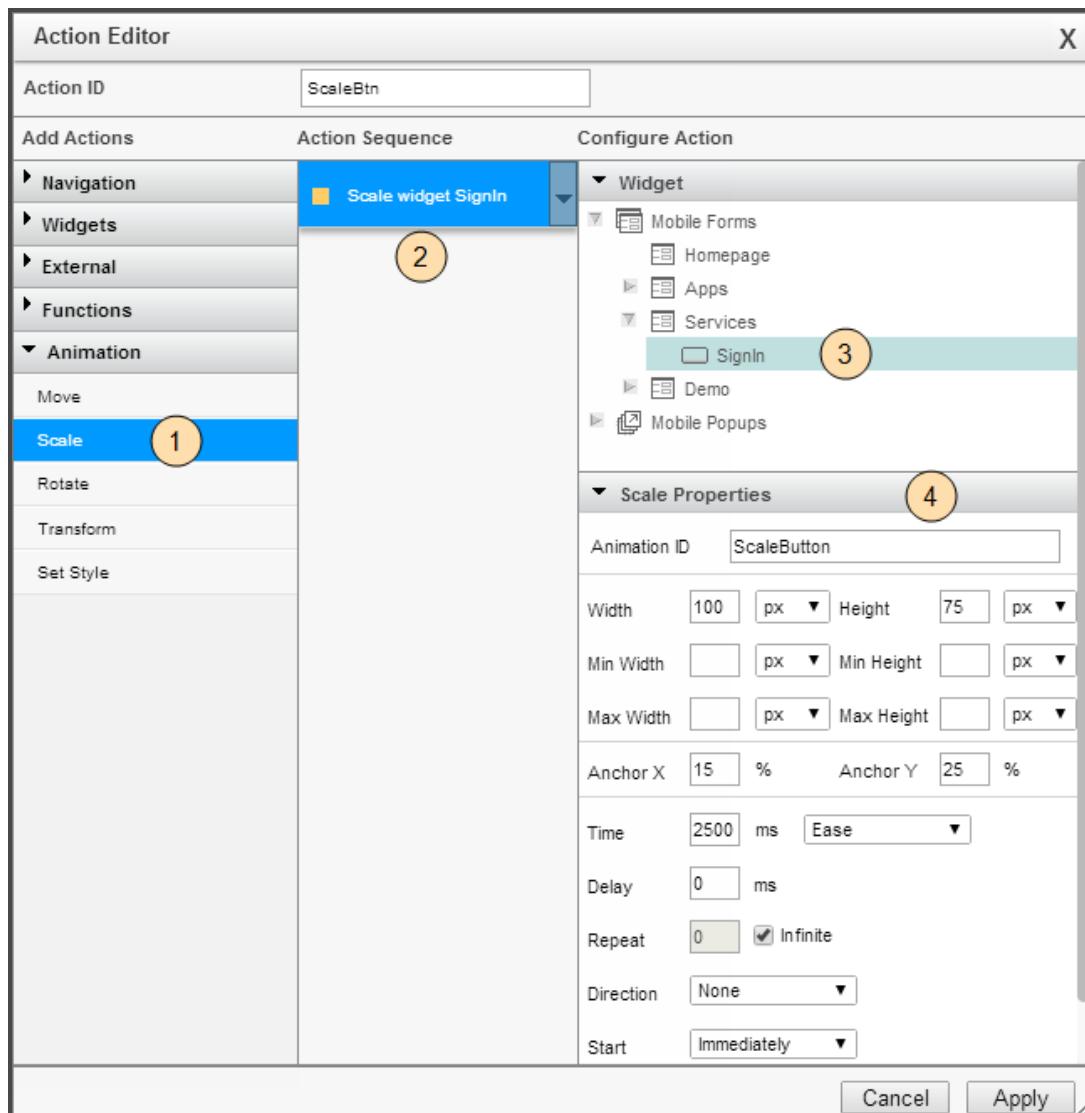
## 5. Click Apply.

**Note:** For the following properties Left, Right, Top, Bottom, Center X and Center Y, you can specify the values in Device Independent Pixels (dp), pixels (px), and percentage (%). You can also specify negative values for these properties.

### 11.4.6.2 Scale

Resizes a widget within its parent widget or a form.

For scaling a widget, follow these steps:



1. Click **Scale** from the **Add Actions > Animation**.

2. Click **Scale Widget** from the **Action Sequence**.

3. Select a widget from the **Widget** tab.
4. Update the properties from the **Translate Properties** tab with the following details:

| Property     | Description   | Examples     |
|--------------|---|--------------|
| Animation ID | Animation ID.   | Scale_widget |
| Width        | Specifies the width of a widget.  | 100 px       |
| Height       | Specifies the height of a widget.   | 250 dp       |
| Min Width    | Specifies the minimum width of a widget.                                      | 20%          |
| Min Height   | Specifies the minimum height of a widget.                                     | 15%          |
| Max Width    | Specifies the maximum width of a widget.                                      | 540px        |
| Max Height   | Specifies the maximum width of a widget.                                      | 300px        |
| Anchor X     | Specifies the horizontal anchor point from where widget rotation begins.      | 15%          |
| Anchor Y     | Specifies the vertical anchor point from where widget rotation begins.        | 25%          |
| Time         | Specifies the number of milliseconds (ms) in which an animation is completed. | 1000         |

| Property      | Description   | Examples        |
|---------------|---|-----------------|
| Time Function | <p>The following time functions are available:</p> <ul style="list-style-type: none"><li>• Ease: Specifies a transition effect with a slow start, then fast, then end slowly.</li><li>• Linear: Specifies a transition effect with the same speed from start to end.</li><li>• Ease-In: Specifies a transition effect with a slow start.</li><li>• Ease-Out: Specifies a transition effect with a slow end.</li><li>• Ease-In-Out: Specifies a transition effect with a slow start and end.</li></ul> |                 |
| Delay         | Specifies the number of milliseconds (ms) to wait before starting an animation.   | 2000            |
| Repeat        | Specifies the number of times an action is repeated. To run the action indefinitely, select the <b>Infinite</b> check box.  | 5               |
| Direction     | <p>Specifies the direction of an action. The following are the available options:</p> <ul style="list-style-type: none"><li>• None: The animation action direction is always same.</li><li>• Alternate: The animation action direction is same every odd time and in reverse direction every even time.</li></ul>   | None, Alternate |

| Property | Description   | Examples    |
|----------|---|-------------|
| Start    | <p>Specifies how an animation action starts. The following options are available:</p> <ul style="list-style-type: none"> <li>• Immediately: Start the animation immediately, with delay, if selected.</li> <li>• With Animation: Action is run along with another action.</li> <li>• After Animation: Action is run after completing another action.</li> </ul> <div style="background-color: #e0f2f1; padding: 10px; margin-top: 10px;"> <p><b>Note:</b> With Animation and After Animation properties are ignored if only single action is available in the <b>Action Sequence</b> pane.</p> </div> | Immediately |
| Inherit  | <p>The following options are available:</p> <ul style="list-style-type: none"> <li>• None: Upon completing this action, widget returns to its original size.</li> <li>• Forward: Upon completing this action, widget stays at final position.</li> <li>• Backward: Upon completing an action, widget will come return to the original position.</li> <li>• Both: Apply Forward and Backward options.</li> </ul>   | Both        |

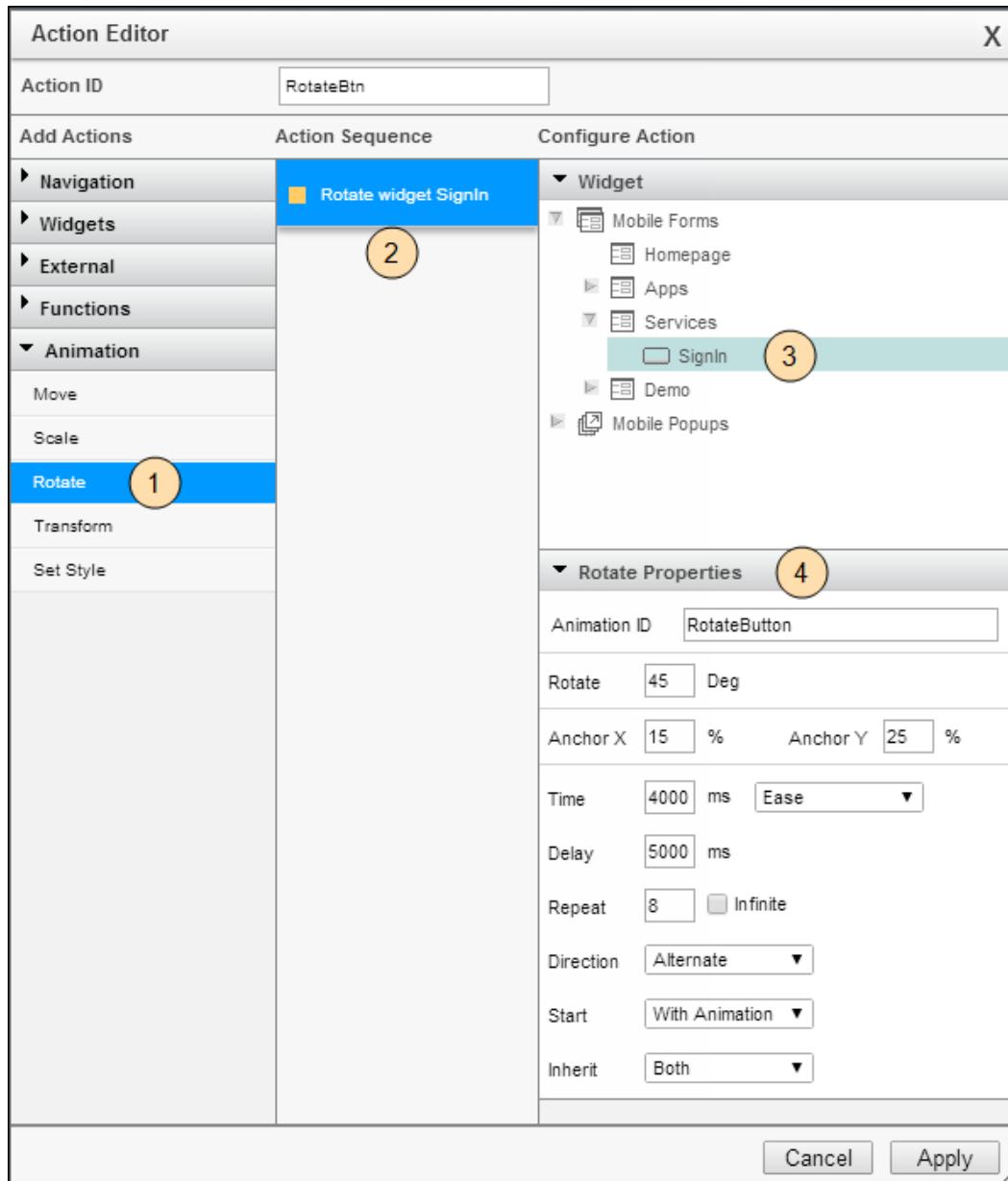
## 5. Click **Apply**.

**Note:** For the following properties Width, Height, Min Width, Min Height, Max Width, Max Height, you can specify the values in Device Independent Pixels (dp), pixels (px), and percentage (%). You can also specify negative values for these properties.

### 11.4.6.3 Rotate

Rotates a widget within its parent widget or a form.

For rotating a widget, follow these steps:



1. Click **Rotate** from the **Add Actions > Animation**.
2. Click **Rotate Widget** from the **Action Sequence**.
3. Select a widget from the **Widget** tab.
4. Update the properties from the **Translate Properties** tab with the following details:

| Property      | Description   | Examples     |
|---------------|---|--------------|
| Animation ID  | Animation ID.   | Rotate_angle |
| Rotate        | Specifies the angle at which a widget is to be rotated.   | 45           |
| Anchor X      | Specifies the horizontal anchor point from where widget rotation begins.  | 15%          |
| Anchor Y      | Specifies the vertical anchor point from where widget rotation begins.  | 15%          |
| Time          | Specifies the number of milliseconds (ms) in which an animation is completed.   | 1000         |
| Time Function | <p>The following time functions are available:</p> <ul style="list-style-type: none"><li>• Ease: Specifies a transition effect with a slow start, then fast, then end slowly.</li><li>• Linear: Specifies a transition effect with the same speed from start to end.</li><li>• Ease-In: Specifies a transition effect with a slow start.</li><li>• Ease-Out: Specifies a transition effect with a slow end.</li><li>• Ease-In-Out: Specifies a transition effect with a slow start and end.</li></ul> |              |

| Property  | Description  | Examples        |
|-----------|--|-----------------|
| Delay     | Specifies the number of milliseconds (ms) to wait before starting an animation.  | 2000            |
| Repeat    | Specifies the number of times an action is repeated. To run the action indefinitely, select the <b>Infinite</b> check box.   | 5               |
| Direction | Specifies the direction of an action. The following are the available options: <ul style="list-style-type: none"><li>• None: The animation action direction is always same.</li><li>• Alternate: The animation action direction is same every odd time and in reverse direction every even time.</li></ul>   | None, Alternate |
| Start     | Specifies how an animation action starts. The following options are available: <ul style="list-style-type: none"><li>• Immediately: Start the animation immediately, with delay, if selected.</li><li>• With Animation: Action is run along with another action.</li><li>• After Animation: Action is run after completing another action.</li></ul> | Immediately     |

**Note:** With Animation and After Animation properties are ignored if only single action is available in the **Action Sequence** pane.

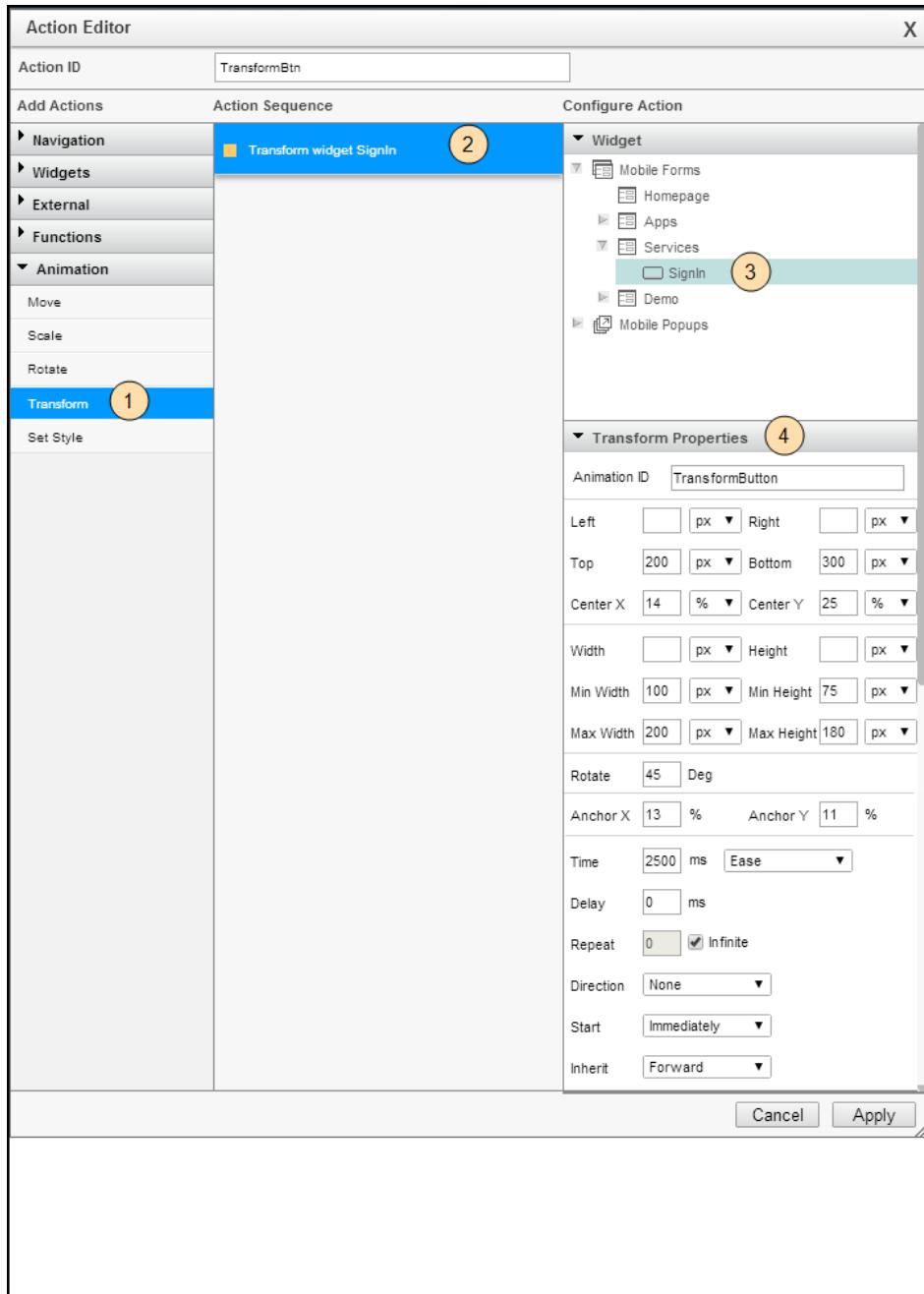
| Property | Description   | Examples |
|----------|---|----------|
| Inherit  | The following options are available: <ul style="list-style-type: none"><li>• None: Upon completing this action, widget returns to its original size.</li><li>• Forward: Upon completing this action, widget stays at final position.</li><li>• Backward: Upon completing an action, widget will come return to the original position.</li><li>• Both: Apply Forward and Backward options.</li></ul> | Both     |

5. Click **Apply**.

#### 11.4.6.4 Transform

Using this single action, you can move, scale, and rotate a widget.

For transforming a widget, follow these steps:



1. Click **Transform** from the **Add Actions > Animation**.
2. Click **Transform Widget** from the **Action Sequence**.

3. Select a widget from the **Widget** tab.
4. Update the properties from the **Translate Properties** tab with the following details:

| Property     | Description  | Examples            |
|--------------|--|---------------------|
| Animation ID | Animation ID.  | Transform_widget    |
| Left         | Specifies by how much a widget moves to the left from its original position.           | -20 px, 100 dp, 15% |
| Right        | Specifies by how much a widget moves to the right from its original position.          | 10 px, 175 dp, 25%  |
| Top          | Specifies by how much a widget moves to the top from its original position.            | -15 px, 200 dp, 7%  |
| Bottom       | Specifies by how much a widget moves to the bottom from its original position.         | 23 px, 225dp, 6%    |
| Center X     | Specifies by how much a widget's center moves horizontally from its original position. | -21 px, 263 dp, 11% |
| Center Y     | Specifies by how much a widget's center moves vertically from its original position.   | -21 px, 263 dp, 11% |
| Width        | Specifies the width of a widget.   | 100 px              |
| Height       | Specifies the height of a widget.  | 250 dp              |
| Min Width    | Specifies the minimum width of a widget.   | 20%                 |
| Min Height   | Specifies the minimum height of a widget.  | 15%                 |
| Max Width    | Specifies the maximum width of a widget.   | 540px               |

| Property      | Description   | Examples |
|---------------|---|----------|
| Max Height    | Specifies the maximum width of a widget.  | 300px    |
| Rotate        | Specifies the angle at which a widget is rotated.   | 45       |
| Anchor X      | Specifies the horizontal anchor point from where widget rotation begins.  | 15%      |
| Anchor Y      | Specifies the vertical anchor point from where widget rotation begins.  | 15%      |
| Time Function | <p>The following time functions are available:</p> <ul style="list-style-type: none"> <li>• Ease: Specifies a transition effect with a slow start, then fast, then end slowly.</li> <li>• Linear: Specifies a transition effect with the same speed from start to end.</li> <li>• Ease-In: Specifies a transition effect with a slow start.</li> <li>• Ease-Out: Specifies a transition effect with a slow end.</li> <li>• Ease-In-Out: Specifies a transition effect with a slow start and end.</li> </ul> | 1000     |
| Delay         | Specifies the number of milliseconds (ms) to wait before starting an animation.   | 2000     |
| Repeat        | Specifies the number of times an action is repeated. To run the action indefinitely, select the <b>Infinite</b> check box.  | 5        |

| Property  | Description  | Examples        |
|-----------|--|-----------------|
| Direction | <p>Specifies the direction of an action. The following are the available options:</p> <ul style="list-style-type: none"> <li>• None: The animation action direction is always same.</li> <li>• Alternate: The animation action direction is same every odd time and in reverse direction every even time.</li> </ul>   | None, Alternate |
| Start     | <p>Specifies how an animation action starts. The following options are available:</p> <ul style="list-style-type: none"> <li>• Immediately: Start the animation immediately, with delay, if selected.</li> <li>• With Animation: Action is run along with another action.</li> <li>• After Animation: Action is run after completing another action.</li> </ul> <p><b>Note:</b> With Animation and After Animation properties are ignored if only single action is available in the <b>Action Sequence</b> pane.</p> | Immediately     |
| Inherit   | <p>The following options are available:</p> <ul style="list-style-type: none"> <li>• None: Upon completing this action, widget returns to its original size.</li> <li>• Forward: Upon completing this action, widget stays at final position.</li> <li>• Backward: Upon completing an action, widget will come return to the original position.</li> <li>• Both: Apply Forward and Backward options.</li> </ul>  | Both            |

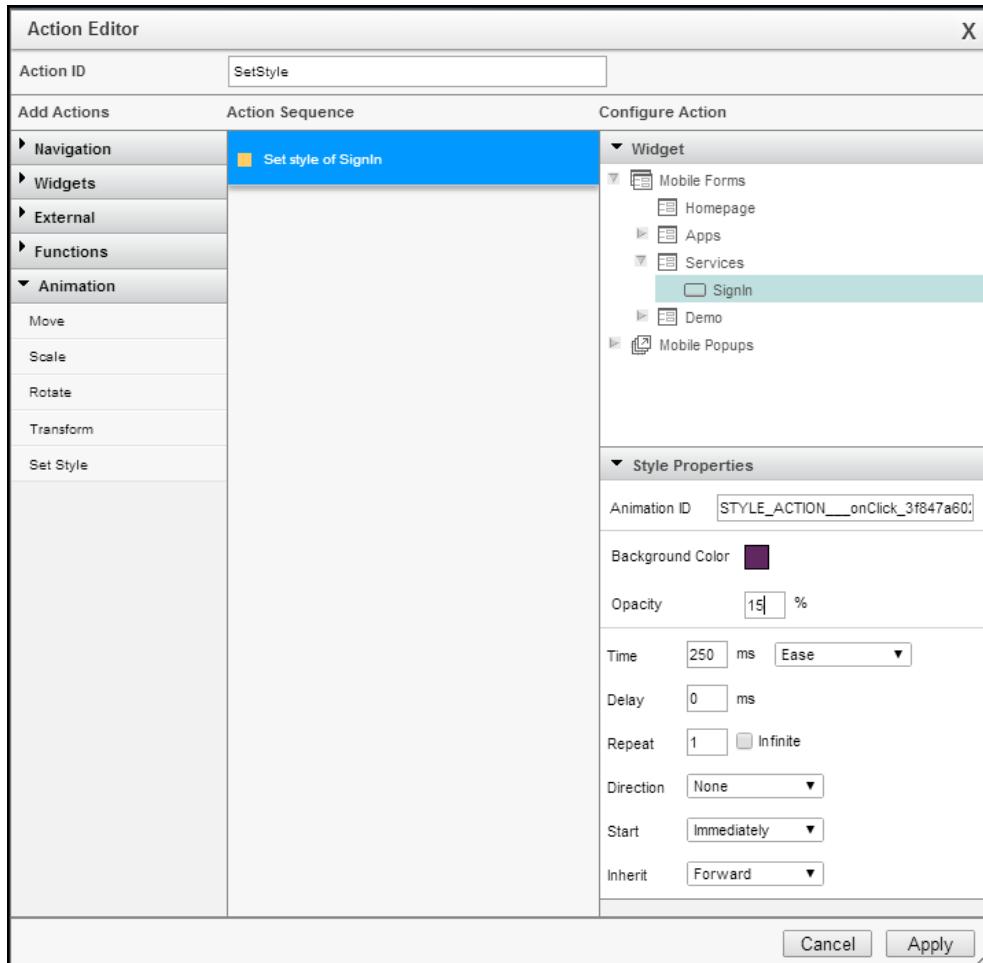
## 5. Click Apply.

**Note:** For the following properties Left, Right, Top, Bottom, Center X and Center Y, Width, Height, Min Width, Min Height, Max Width, Max Height, you can specify the values in Device Independent Pixels (dp), pixels (px), and percentage (%). You can also specify negative values for these properties.

### 11.4.6.5 Set Style

Allows you to change the background color of a widget.

For changing the background color of a widget, follow these steps:



1. Click **Set Style** from the **Add Actions > Animation**.
2. Click **Set Style** from the **Action Sequence**.
3. Select a widget from the **Widget** tab.
4. Update the properties from the **Translate Properties** tab with the following details:

| Property         | Description   | Examples       |
|------------------|---|----------------|
| Animation ID     | Animation ID.   | SetStyle_Color |
| Background Color | Select a color from the color picker. Only Single color ...   |                |
| Opacity          | Specifies the opacity of the background color.  | 15%            |
| Time             | Specifies the number of milliseconds (ms) in which an animation is completed.   | 1000           |
| Function         | <p>The following time functions are available:</p> <ul style="list-style-type: none"><li>• Ease: Specifies a transition effect with a slow start, then fast, then end slowly.</li><li>• Linear: Specifies a transition effect with the same speed from start to end.</li><li>• Ease-In: Specifies a transition effect with a slow start.</li><li>• Ease-Out: Specifies a transition effect with a slow end.</li><li>• Ease-In-Out: Specifies a transition effect with a slow start and end.</li></ul> |                |
| Delay            | Specifies the number of milliseconds (ms) to wait before starting an animation.   | 2000           |

| Property  | Description  | Examples        |
|-----------|--|-----------------|
| Repeat    | Specifies the number of times an action is repeated. To run the action indefinitely, select the <b>Infinite</b> check box.   | 5               |
| Direction | Specifies the direction of an action. The following are the available options: <ul style="list-style-type: none"><li>• None: The animation action direction is always same.</li><li>• Alternate: The animation action direction is same every odd time and in reverse direction every even time.</li></ul>   | None, Alternate |
| Start     | Specifies how an animation action starts. The following options are available: <ul style="list-style-type: none"><li>• Immediately: Start the animation immediately, with delay, if selected.</li><li>• With Animation: Action is run along with another action.</li><li>• After Animation: Action is run after completing another action.</li></ul> | Immediately     |

**Note:** With Animation and After Animation properties are ignored if only single action is available in the Action Sequence pane.

| Property | Description  | Examples |
|----------|--|----------|
| Inherit  | <p>The following options are available:</p> <ul style="list-style-type: none"><li>• None: Upon completing this action, widget returns to its original size.</li><li>• Forward: Upon completing this action, widget stays at final position.</li><li>• Backward: Upon completing an action, widget will come return to the original position.</li><li>• Both: Apply Forward and Backward options.</li></ul> | Both     |

5. Click **Apply**.

## 12. Importing Photoshop Styles

In Visualizer, creating a skin for a widget is easy and effective. Assigning widget skin properties such as background color, border size, or text shadow are accomplished with minimal effort. However, a good number of designers are adept with creating the required style for a designing element using design tools such as Photoshop. For these designers, Visualizer provides an option of creating the designing styles in Photoshop, and then importing these styles into Visualizer.

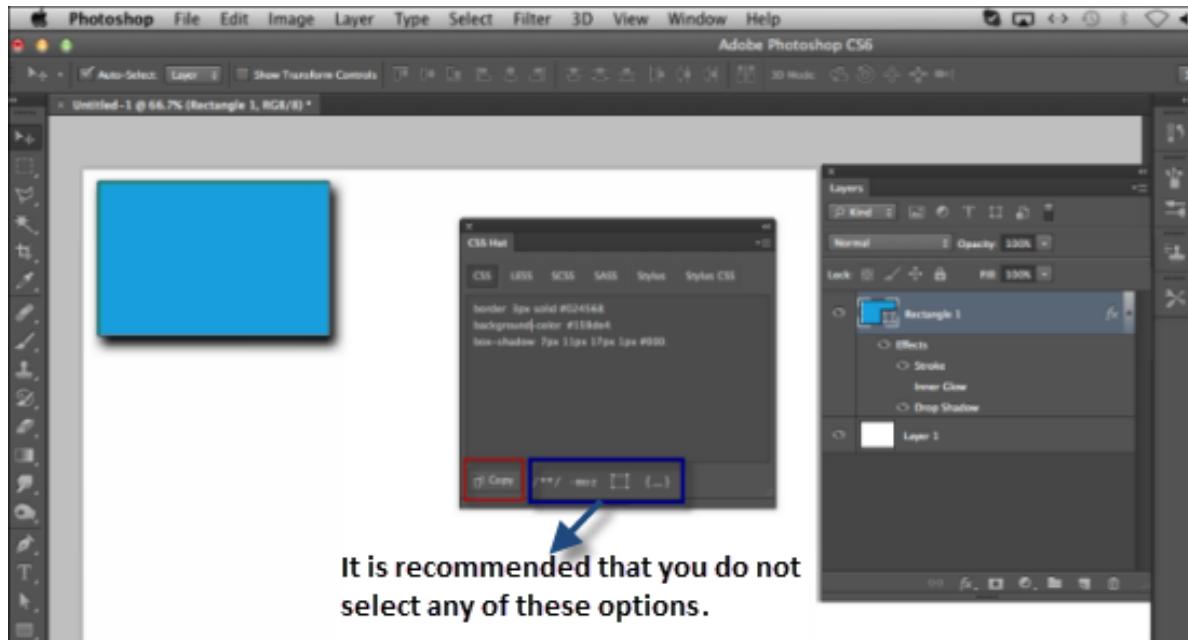
### 12.1 Prerequisites

Following software are required for importing the Photoshop styles into Visualizer:

1. Adobe Photoshop installed in your system.
2. CSS Hat plugin for Photoshop.

### 12.2 Importing Photoshop Styles into Visualizer

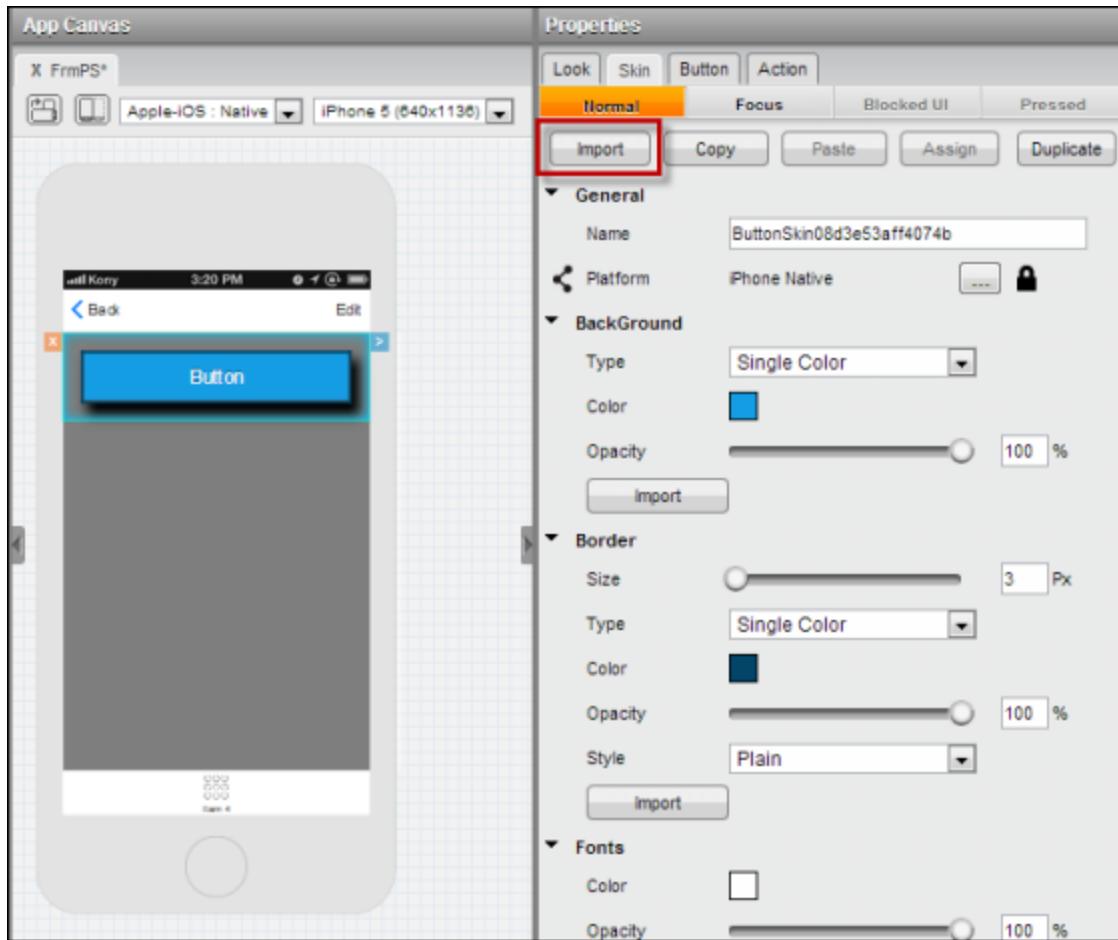
To import Photoshop styles into Visualizer, follow these steps:



1. In Photoshop, create a design. In this example, we have created a rectangle element and assigned values to background color, stroke, and drop shadow.
2. Select the layer.
3. Open CSS Hat Extension, and ensure that the equivalent CSS properties are populated. Click **Copy**. The CSS values are copied to theClipboard.

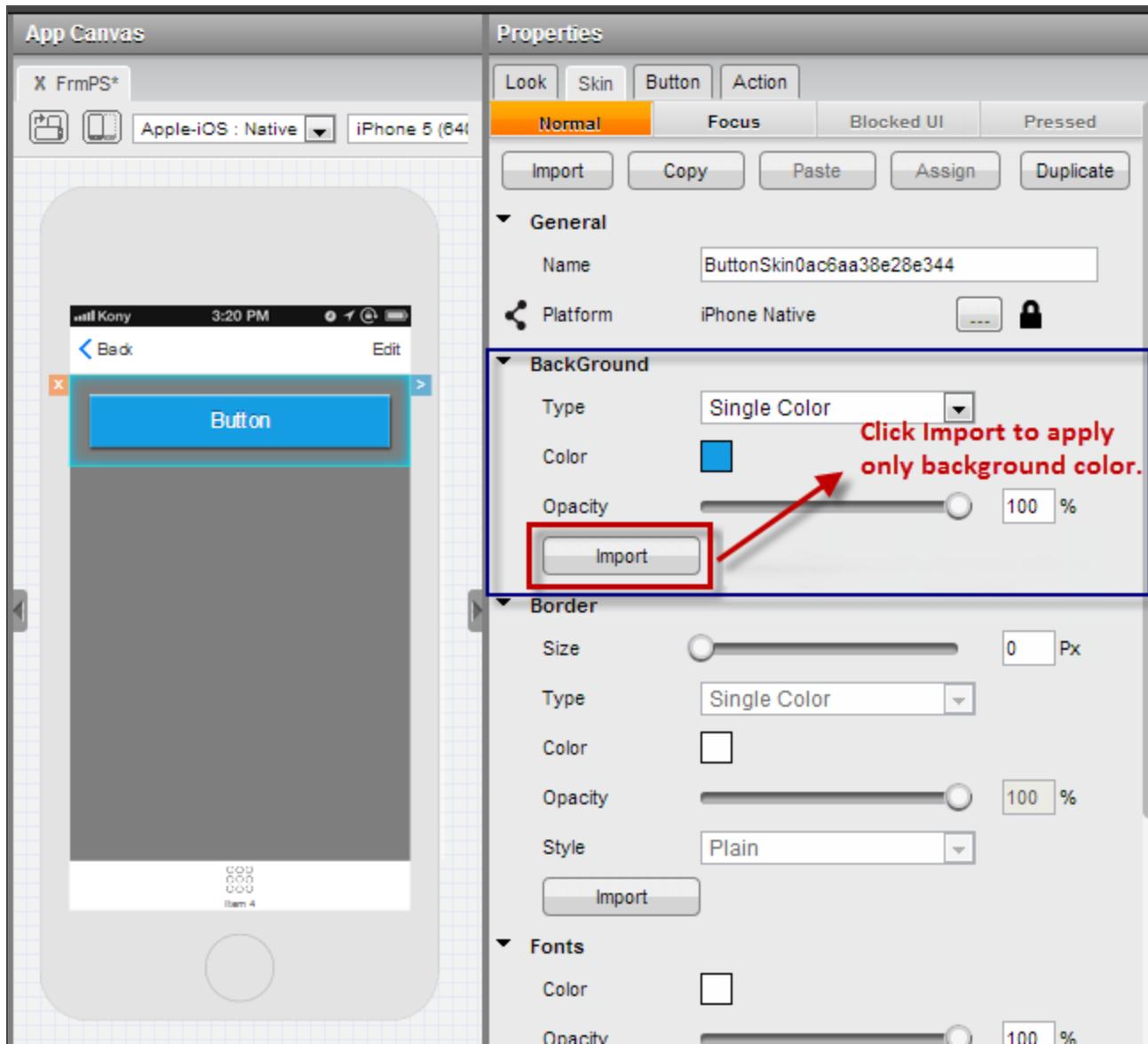
**Note:** Do not select any of these options  in the CSS Hat palette.

4. In Visualizer, select a widget on which you want to implement Photoshop styles.
5. In the **Properties** tab, click **Skin**.
6. Select a skin state such as Normal, Focus, or Blocked UI.
7. To import all the Photoshop styles (such as background, border and shadow), click **Import**. The widget skins are updated based on the Photoshop CSS.



**Selective Photoshop CSS import:** To import a selective Photoshop style (such as a background, a border, or a shadow style), click **Import** in the individual property tabs. Based on the Photoshop style, the individual property is updated.

For example, in our Photoshop design, we provided background color, stroke, and drop-shadow properties. Instead of importing all these values, we chose to import only background style.



### 12.3 Naming Conventions

The following table displays the Visualizer and Photoshop naming conventions:

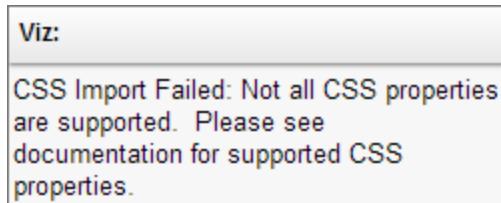
| Kony Visualizer | Photoshop |
|-----------------|-----------|
| Border          | Stroke    |

| Kony Visualizer                 | Photoshop        |
|---------------------------------|------------------|
| Background: Multi-Step Gradient | Gradient Overlay |
| Background: Two Step Gradient   |                  |
| Background: Single Color        | Color Overlay    |
| Shadow                          | Drop Shadow      |
| Shadow                          | Outer Glow       |
| Shadow (Inset)                  | Inner Shadow     |
| Shadow (Inset)                  | Inner Glow       |

## 12.4 Limitations

Following are the limitations on importing Photoshop styles:

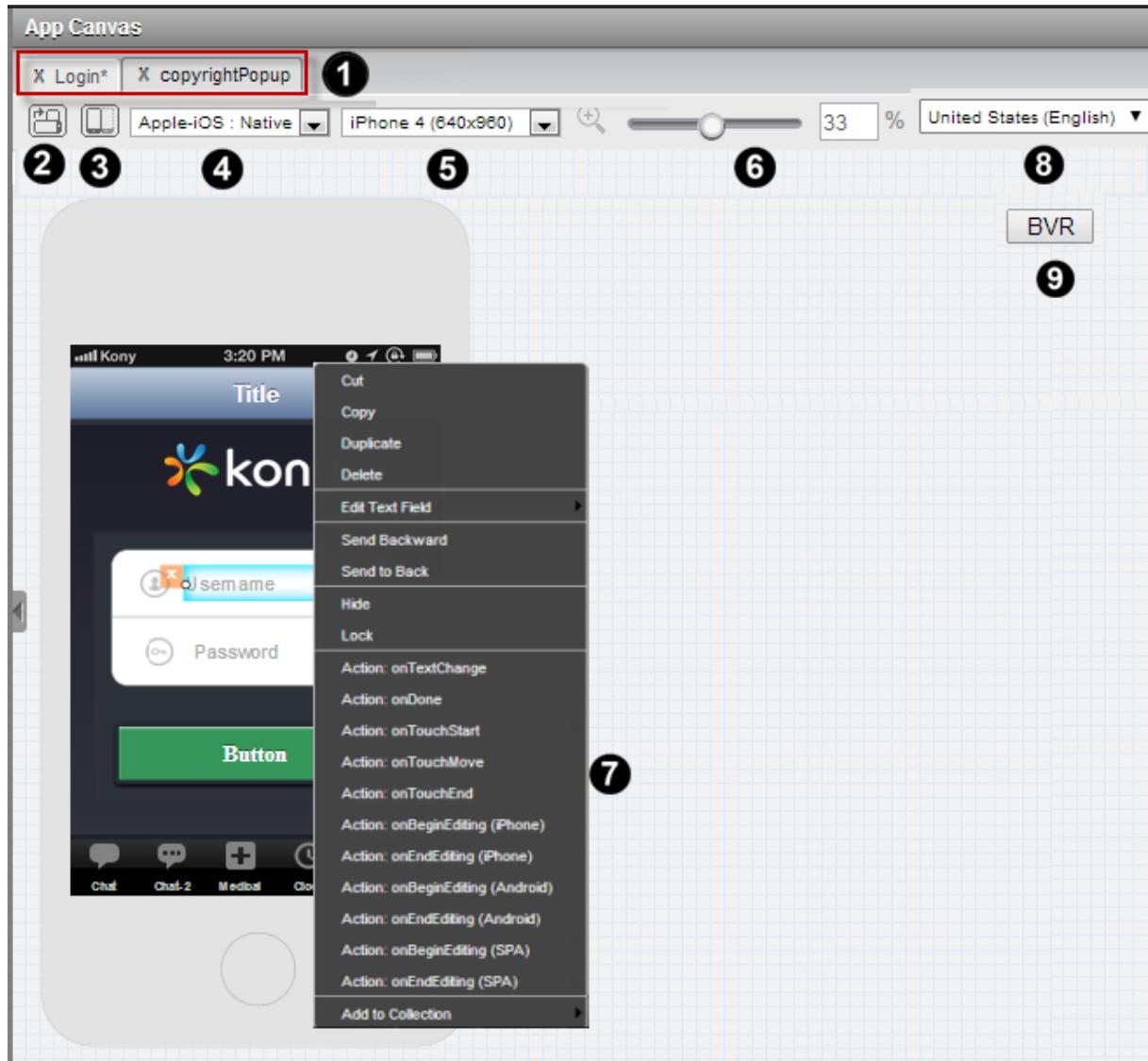
- Not all the CSS values are imported. Visualizer ignores the values that are unsupported. If such unsupported values exist in CSS, the following message is displayed:



- Font families are not imported from Photoshop. It is recommended that you provide the font family for a widget in Visualizer.
- Borders with Multi-Step Gradient (MSG) are not properly imported from Photoshop. It is recommended that you provide the border MSG in Visualizer.
- If you create nonuniform borders, you need to import the CSS styles in the Native and Web channels.

## 13. App Canvas

The central pane of the Kony Visualizer is called App Canvas. This is the core area of "What You See Is What You Get" (WYSIWYG) environment of Visualizer. Here, you can view your design and make the necessary changes.



App Canvas has the following features:

1. **Open Windows:** Lists all the open windows (such as forms, popups, and templates) . A window:

- containing any unsaved changes will have an asterisk(\*) .
- can be opened by clicking it.
- can be closed by clicking on the close button X.

2. **Toggle Orientation:** Click to toggle your design between landscape and portrait view.

**Note:** This option is applicable only when you select a form's [Orientation](#) as Both.

3. **Toggle Shell:** Click to view your design with or without the device shell.
4. **Platforms:** The list contains all the supported platforms. Choose a platform from the list. Your project design is displayed on the selected platform. Default: Apple-iOS : Native.
5. **Device:** Based on the selected platform, the device list contains all the supported devices.
6. **Zoom:** The slider allows you change the zoom level for better design viewing.
7. **Context menu:** Click the widget to display its context menu items. These menu items will vary depending upon the selected widget, and includes:
  - a. **Cut:** Allows you to move a widget.
  - b. **Copy:** Allows you to copy a widget. The copied widgets can be pasted inside a form or pop-up.
  - c. **Duplicate:** Allows you to duplicate a widget within the same parent widget.
  - d. **Delete:** Allows you to delete a widget.
  - e. **Edit Text Field:** If two or more text fields are available for a widget, click the > to display the list of editable text fields. This option varies based on the selected widget.
  - f. **Bring Forward:** Moves the selected widget one step forward.

**Note:** This options is available only for a Flex form.

- g. **Bring to Front:** Moves the selected widget to the front.

**Note:** This options is available only for a Flex form.

- h. **Send Backward:** Sends the selected widget one step backward.

**Note:** This options is available only for a Flex form.

- i. **Send to Back:** Sends the selected widget is moved to the last position in the order.

**Note:** This options is available only for a Flex form.

- j. **Hide:** Allows you to hide a selected widget from view on App Canvas, but will not affect the **Visibility** property of the widget.

**Note:** This options is available only for a Flex form.

- k. **Unhide:** Allows you to unhide a widget. This option is enabled only when the widget is hidden.

**Note:** This options is available only for a Flex form.

- l. **Lock:** Allows you to lock a widget. This action prevents you from inadvertently moving or scaling the widget within App Canvas. You will still be able to modify the widget properties from the property editor.

**Note:** This options is available only for a Flex form.

- m. **Unlock:** Allows you unlock a widget. This option is available only when the widget is locked.

**Note:** This options is available only for a Flex form.

- n. **Move Left and Move Right:** Allows you to move the widget either to the left or right of immediate widgets. These options are available only when:

- Multiple widgets are added to a container widget.
- For a leftmost widget within a container widget, **Move Left** option is unavailable. Similarly, for a rightmost widget within a container widget , **Move Right** option is unavailable.

**Note:** These options are available only for a VBox form

- o. **Move Up and Move Down:** Allows you move the widget either to the top or bottom of its immediate widgets. These options are available only when:

- Multiple widgets are added to a container widget.
- When the widget is docked at the bottom of a container widget, only **Move Up** option is available. Similarly, when a widget is docked at the top a container widget, only **Move Down** option is available.

**Note:** These options are available only for a VBox form.

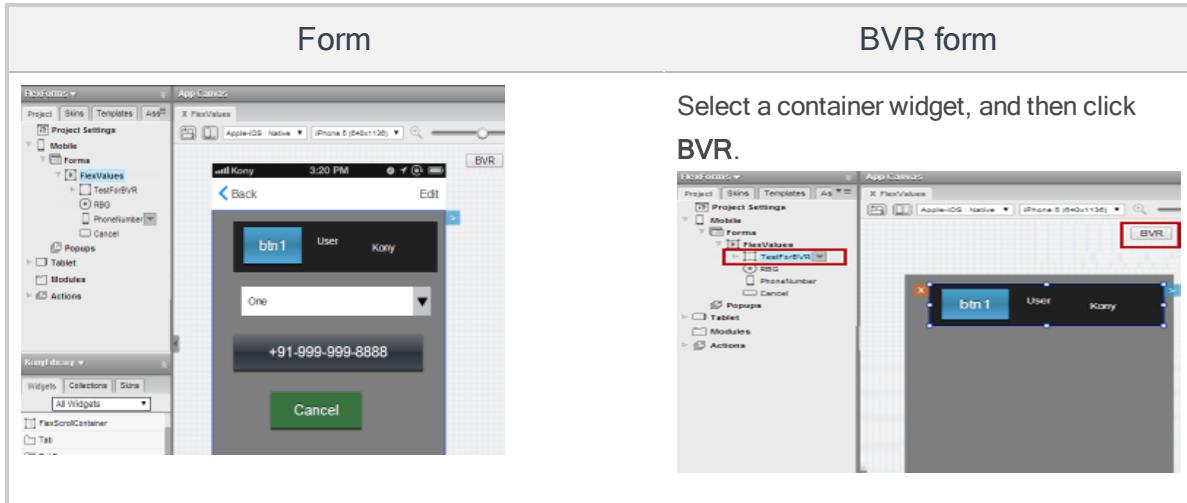
- p. **Actions:** Allows you to assign actions to a widget. For more information, refer section Action.

- q. **Add to Collection:** Allows you to add a widget to a collection. You can either store a widget to an existing collection and library or create a new collection and library.

8. **i18n Locale:** The default locale language is displayed. Click the drop-down arrow to view all the locales available for the project. For more information, refer [i18n Application Content](#).

9. Beyond-Visual-Range (BVR): This option allows you to view only the selected container widget and its child (including those widgets that are beyond visible view.) The remaining elements on a form are hidden.

Example of BVR:

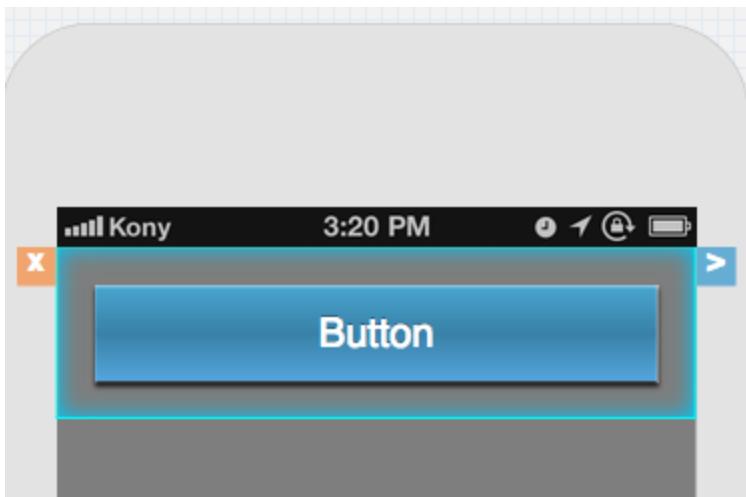


**Note:** The BVR option is available only for flex forms.

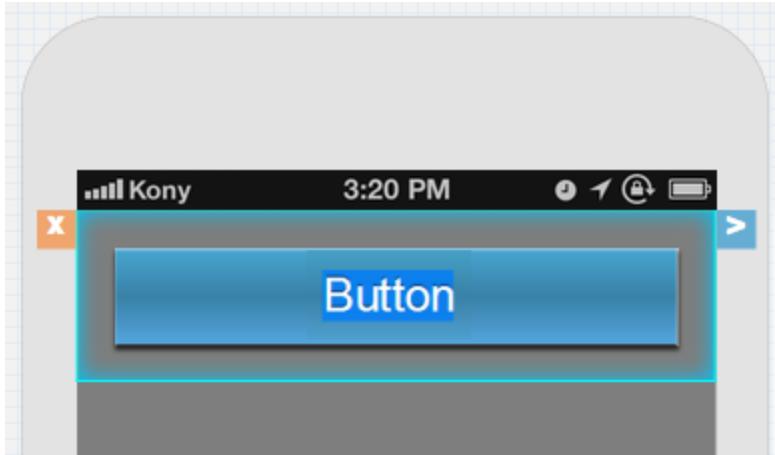
## 13.1 Inline Editing

Inline editing allows you to edit a widget's text property by double-clicking the widget on the App Canvas. This option enables the user to quickly make the edits to a widget's text. Inline editing is available for the widgets such as Button, TextBox2, Label and Phone.

### 13.1.1 An example of inline editing of widget that has a single text field:



Double-click the Button widget on the App Canvas.



Button widget text is selected and is in editable mode.

### 13.1.2 An example of Inline Editing of widget that has two or more text fields:

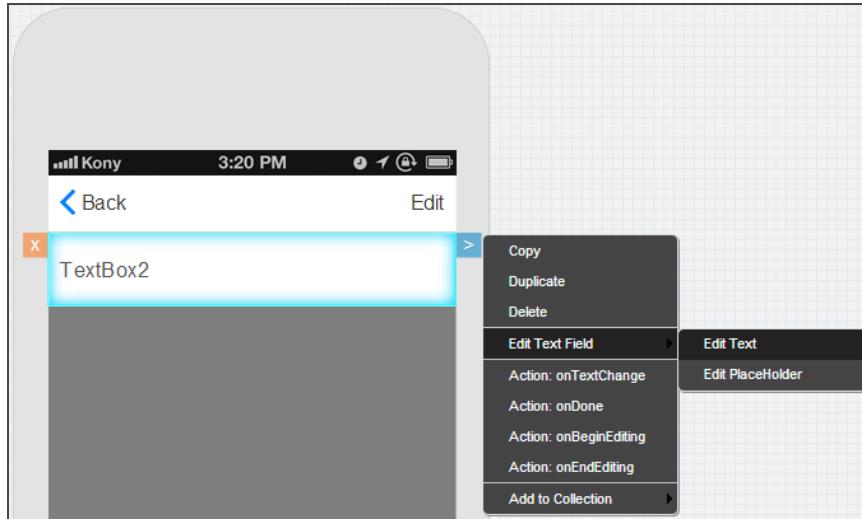
TextBox2 widget has following text fields: **Name** and **Placeholder**.

You can rename the TextBox2 by:

- Double-clicking the TextBox2 widget on the App Canvas

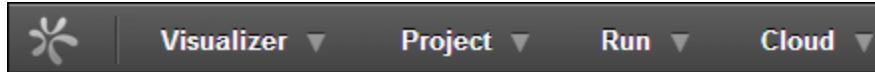
or

- From the TextBox2 menu, click **Edit Text Field > Edit Text**.



## 14. Top Menu

Top menu items consist of the following items:



- **Visualizer**: Allows you to view Quick preview and Functional Preview preferences along with the links to Visualizer documentation.
- **Project**: Allows you to create, open, import and export a project.
- **Run**: Allows you to run a project with only Visualizer action events or with Kony Studio actions events (If available for a project).
- **Cloud**: Allows you to publish and share your project.

## 14.1 Visualizer

Visualizer menu provides the following details:

1. [About Visualizer](#)
2. [Preferences](#)
3. [Documentation](#)

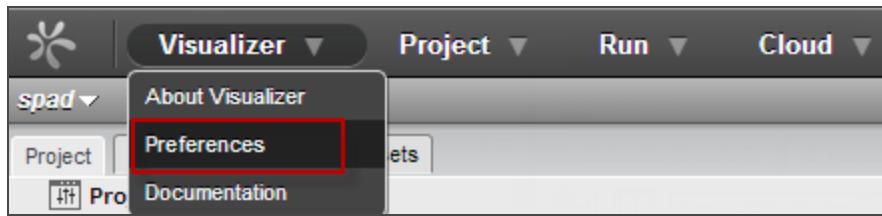
### 14.1.1 About Visualizer

Click **About Visualizer** and from the pop-up, click **View Credits** to view the list of all Kony, Inc. third-party licenses that are used in developing this software.

## 14.1.2 Preferences

Visualizer Preferences enables you to configure the **Quick Preview** and the **Functional Preview** process for a project.

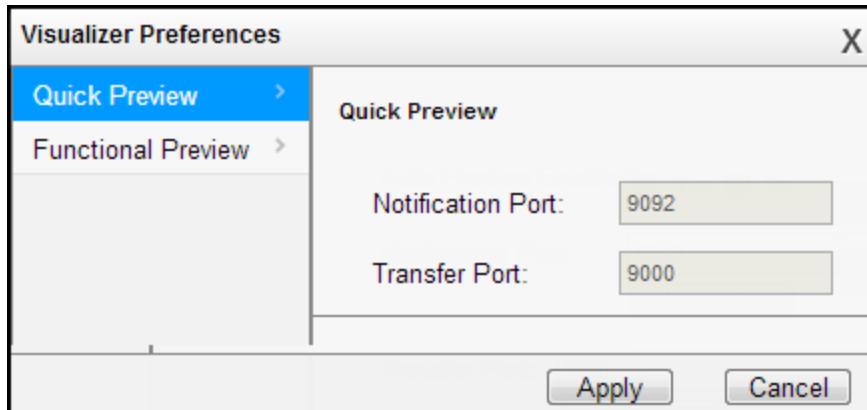
For accessing the Visualizer preferences, click **Visualizer > Preference** from the top menu bar.



### 14.1.2.1 Quick Preview

Quick Preview provides you the ability to preview the current form. For more details on quick preview, refer to the section [Quick Preview](#).

The default tab of **Visualizer Preference** dialog box is **Quick Preview**.

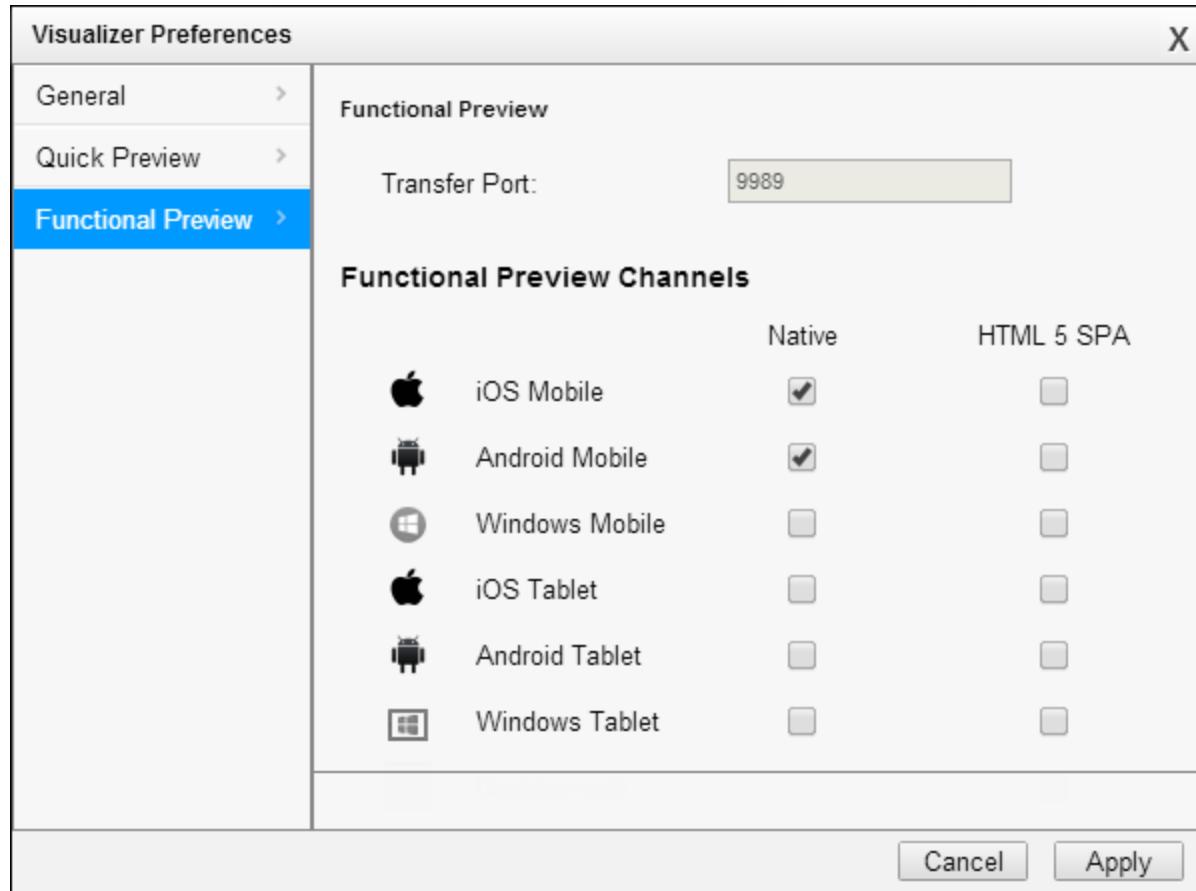


Here, you can view the port numbers assigned to the **Notification Port** and the **Transfer Port**.

### 14.1.2.2 Functional Preview

Functional preview allows you to view the prototype of a project. For more details on functional preview, refer to the section [Functional Preview](#).

Click the **Functional Preview** tab in the **Visualizer Preference** dialog box to access the **Functional Preview** configurable items.



In this tab, you can:

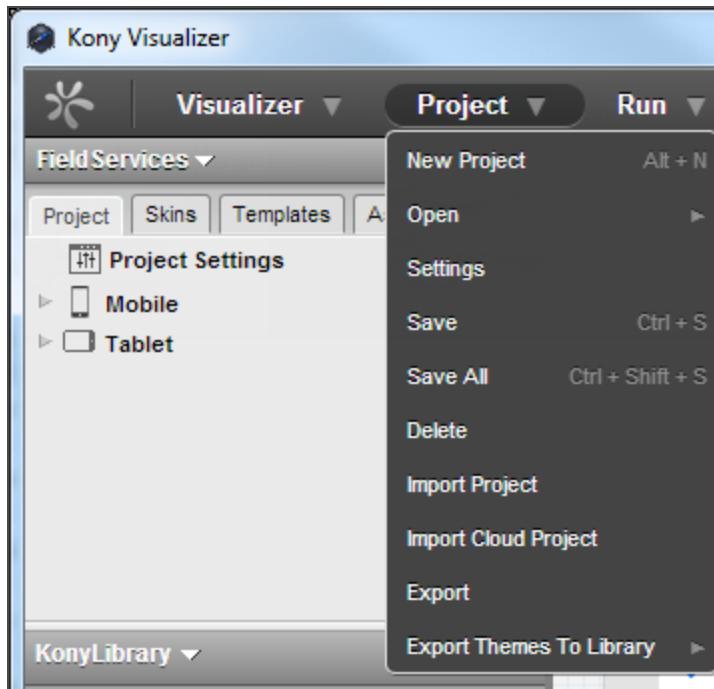
- View the Transfer Port number that is required for the functional preview.
- Select the channels on which you wanted to view the functional preview of a project. Before you select the channels, ensure that a project is available for the device. For example, if you have created a project for the Mobile devices, than you should select **Native** and / or **HTML 5 SPA** options for the **iOS Mobile** and the **Android Mobile**.

### 14.1.3 Documentation

Click this link to view all the documentation available for Visualizer such as Video Tutorials, Installation Guides, User Guide and Release Notes.

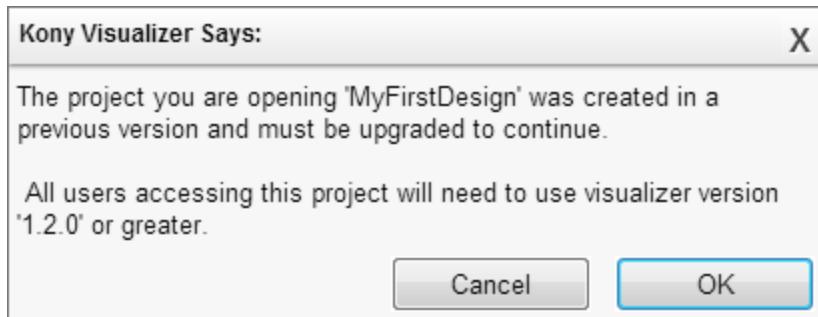
## 14.2 Project

From **Project**, you can perform following actions:



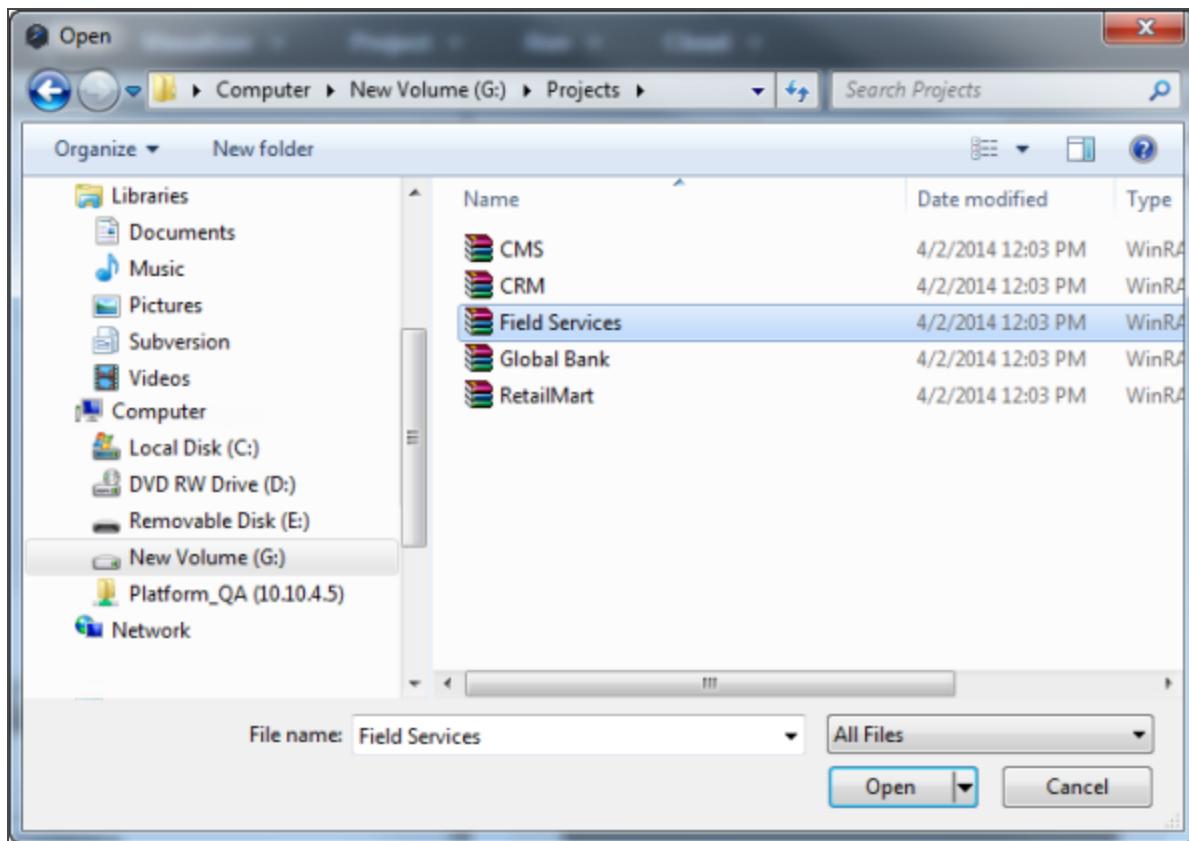
- **New Project:** A new project with a unique name is created.
- **Open:** Displays a flyout menu with a list of all the available projects within the local drive. Click a project to open it.

If you attempt to open a project that was created in an earlier version of Visualizer, you receive the following message:

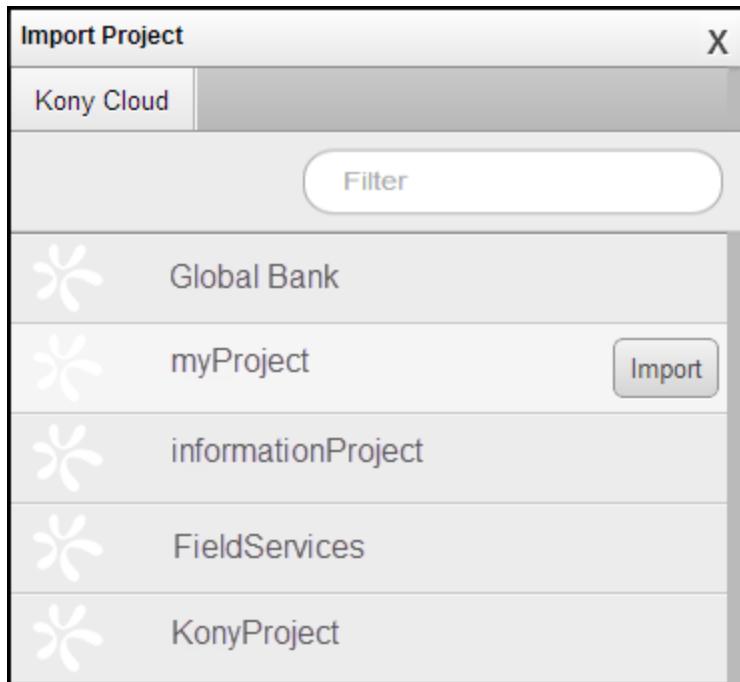


Click **OK** to continue. The project is automatically ported to the latest version of Visualizer, and then the project is opened.

- **Settings:** Refer to [Project Settings](#).
- **Save:** Saves the active form or pop-up. This option is unavailable when all the work in the active form or pop-up is saved.
- **Save All:** Saves all the changes made to the project. This option is unavailable when there are no pending changes are found for a project.
- **Delete:** Allows you to delete the current project.
- **Import Project:** Allows you to import a project from your local drive. When you click this option, an **Open** dialog box is displayed. You can specify the drive, directory, and the name of a project to be imported.



- **Import Cloud Project:** Allows you to import projects from your cloud account. When you click this option, the **Import Project** dialog box, containing a list of projects is displayed. Rest the pointer on a project to display **Import** button. Click **Import** to download the project into your local drive.



**Note:** If a project with the same name exists in your local drive, system prompts you to choose either to overwrite the existing project or rename the project being imported.

- **Export:** Allows you to save the project on your desktop as a compressed file. When you click this option, the **Save As** dialog box appears. From here, you can navigate to a location on your desktop to store the project. Rename the file (if required), and click **Save**.
- **Export Themes To Library:** Allows you to store the themes on your desktop. These themes are available across the projects within your desktop.

## 14.3 Run

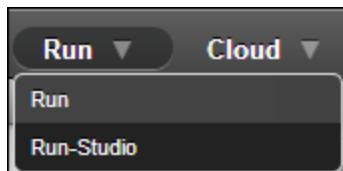
Run allows you to view the [Function Preview](#) of a project locally by generating the "functional preview" build.

### 14.3.1 Prerequisites

Select the desired channels on which you would like to view the functional preview. Refer to the section [Functional Preview Preferences](#) for steps to select channels.

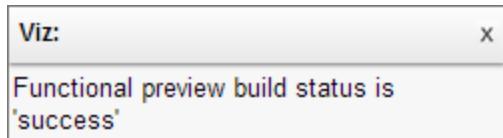
### 14.3.2 Run Menu

From the **Run** menu, following actions are available:



- **Run:** Generates the functional preview build for a project with Visualizer actions.
- **Run-Studio:** Generates the functional preview build for a project with Kony Studio actions.

After you select one of the options, and if the functional preview build is successfully, you receive the following confirmation message:



### 14.3.3 Viewing the Functional Preview

For viewing the functional preview of a project locally, refer to the following section [Functional Preview \(Local\)](#).

## 14.4 Cloud

Cloud provides an easy way of sharing designs with your stakeholders in a very secure environment.

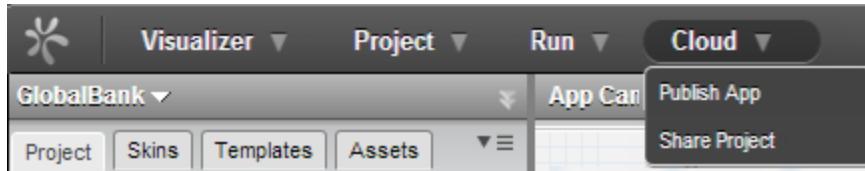
You can collaborate with your stakeholders by using below options, :

- [Publish App](#): Allows your stakeholders to view the functional preview of the project, and execute any action events attached to the widgets.
- [Share Project](#): Allows your stakeholders to download the complete project. They can open the project, make changes (if required), and then upload the modified project back to the cloud.

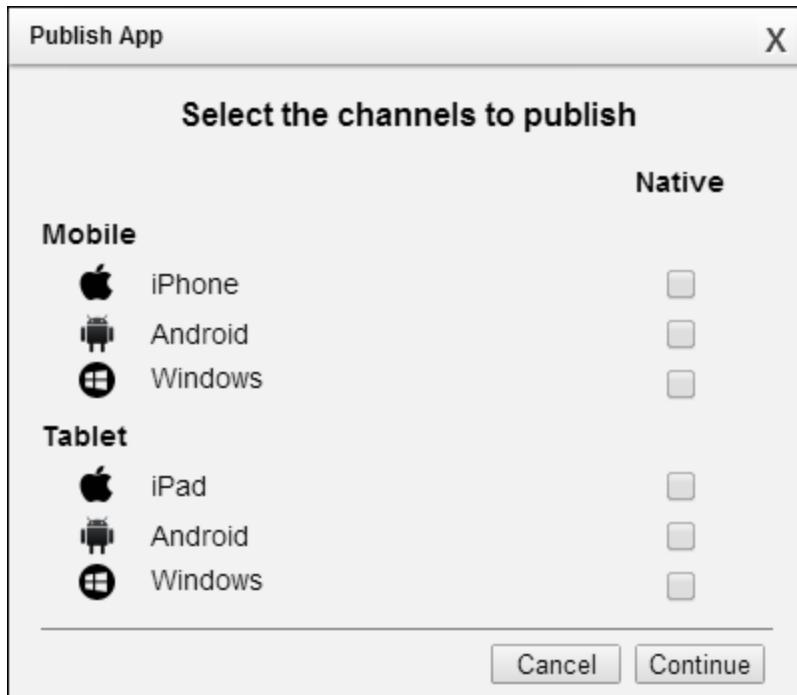
### 14.4.1 Publish App

Following are the steps for publishing an app to the cloud.

1. Open your project and from the top-menu bar, click **Cloud > Publish App**.



2. In the **Publish App** window, select the channels on which you desire to publish your application, and then click **Continue**.



**Note:** The Publish App dialog box displays only those channels for which the project is created. For example, if you have created a project only for the Mobile channels, then the Tablet channels are not displayed.

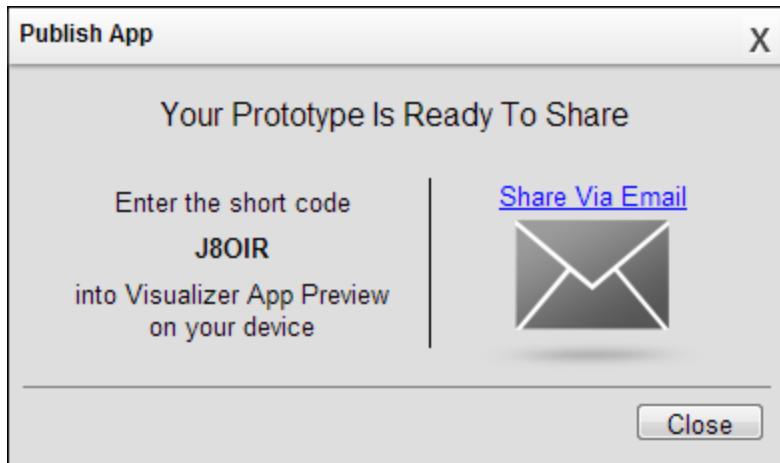
3. Provide the necessary details and click **Apply**.

The screenshot shows a 'Publish App' dialog box. At the top left is the title 'Publish App' and at the top right is a close button ('X'). Below the title, there is a radio button group labeled 'Create New' which is selected. The form contains three input fields: 'Account' with a dropdown menu showing 'Kony, Inc.', 'Name' with the value 'GlobalBank', and 'Description' with the value 'New UI approved by the SMEs.'. At the bottom of the dialog are two buttons: 'Apply' and 'Cancel'.

- **Name:** (optional) Enter a name for your project. This name is updated in the Visualizer Services under the **Resource Name** of a project.
- **Description:** (optional) Enter a description for your project. This description is updated in the Visualizer Services under the **Description**.

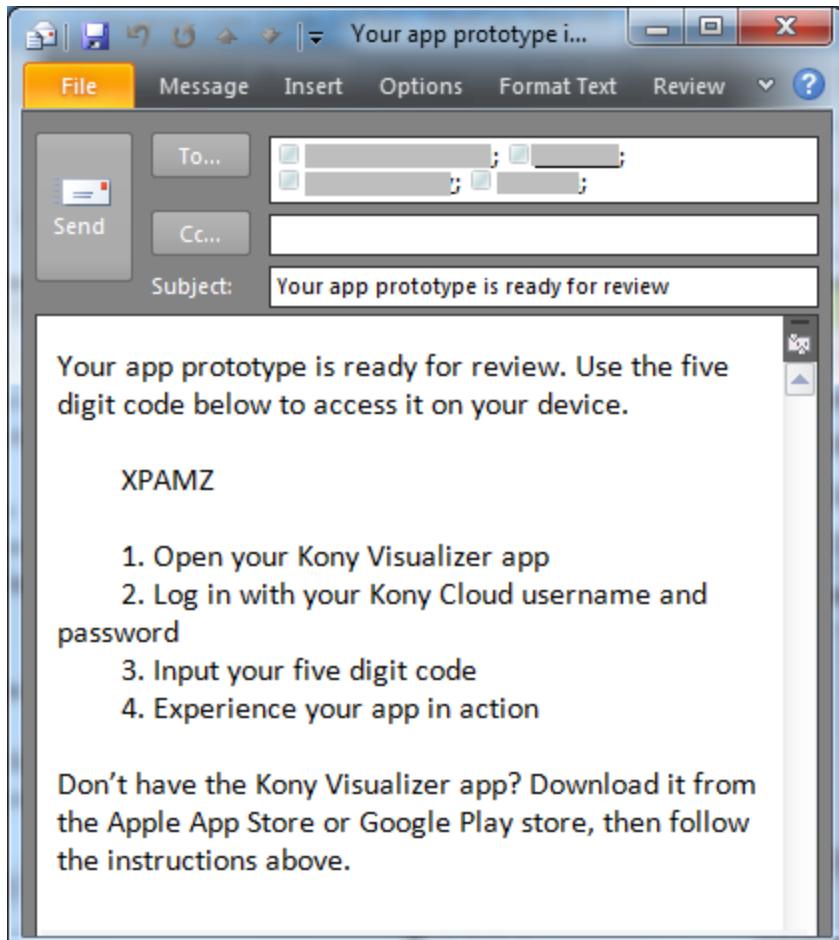
**Note:** If you profile is linked to a single account, the above screen will not contain **Account** list.

4. Depending on the size of your application and the speed of your internet connection, uploading an application may take time. Once the application is uploaded to your cloud account, a code is generated as show below.



**Note:** This code is vital for viewing the functional preview of the application. This code contains (uppercase) alphabets and numbers.

5. You can easily share this code by clicking **Share Via Email**. A **New Message** window from your default mail client (for example, Outlook 2010) appears. The code along with the steps for viewing the functional preview of the app is auto-updated in the body of your new email message as shown below.



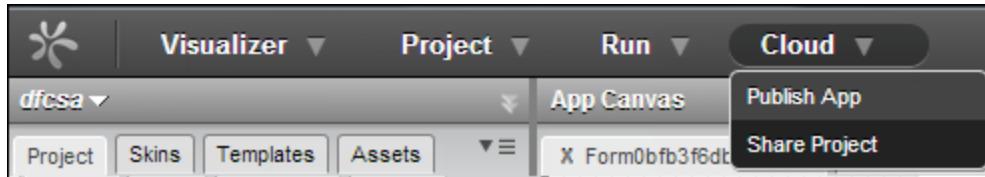
Add the email address of all the stakeholders to whom you would like to provide the functional preview of the application, and then send the email.

**Note:** Steps for Functional Preview of an application can be viewed in the section [Functional Review](#).

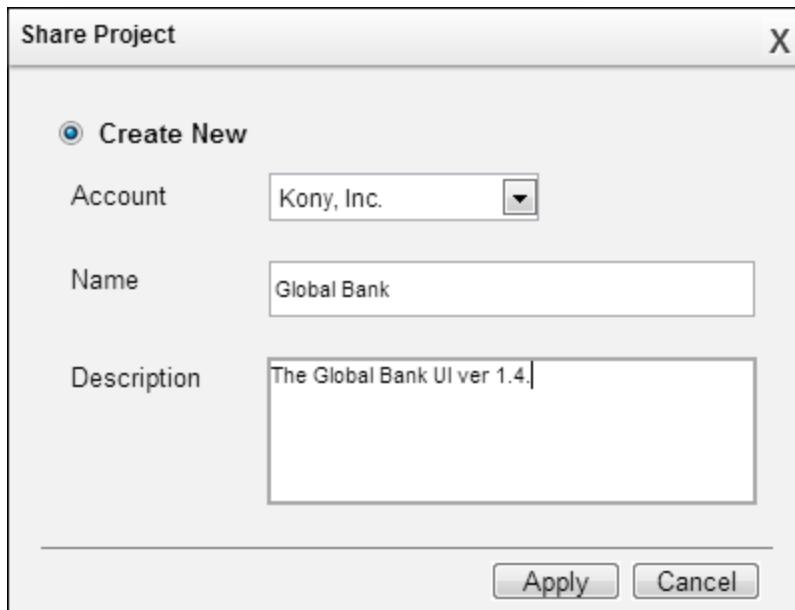
#### 14.4.2 Share Project

Following are the steps for sharing a project on the cloud.

1. When your design is ready to be shared with others, from the top-menu bar click **Cloud > Share Project**.



2. In the **Share Project** window, provide the necessary details and click **Apply**.

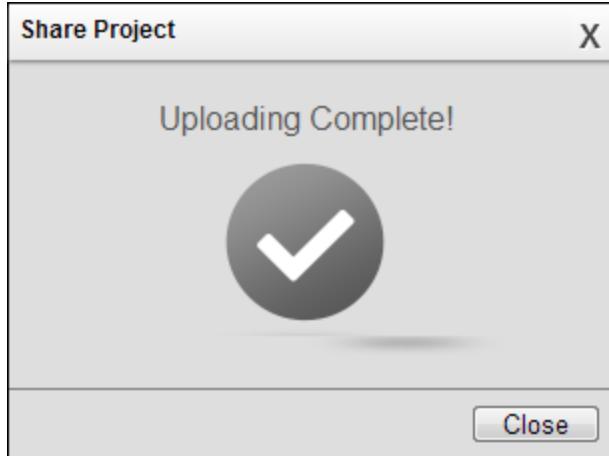


- **Name:** (optional) Enter a name for your project. This name is updated in the Visualizer Services under the **Resource Name**.
- **Description:** (optional) Enter a description for your project. This description is updated in the Visualizer Services under the **Description**.

**Note:** If you profile is linked to a single account, the above screen will not contain **Account** list.

3. Depending on the size of your project and the speed of your internet connection, uploading a project may take time. Once the project is successfully uploaded, you receive a confirmation

message as show below.



Click Close to complete the process of sharing the project.

Now the project is uploaded to the cloud, users can download and view the project using the option [Import Cloud Studio Project](#).

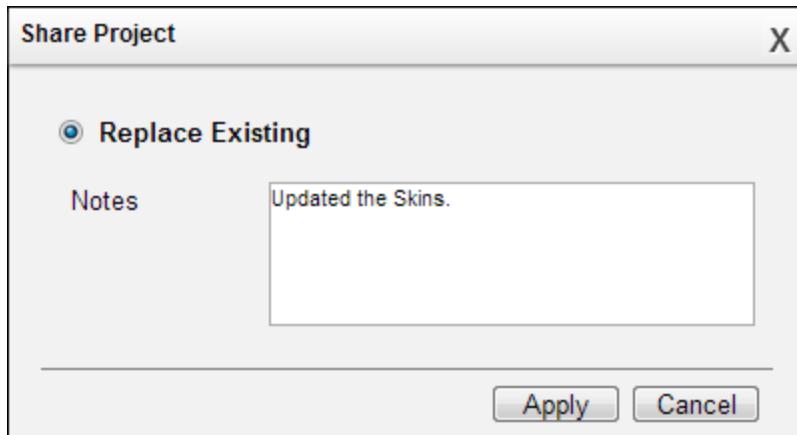
#### 14.4.2.1 Modifying a Shared Project

A project can be imported and modified by following the below steps:

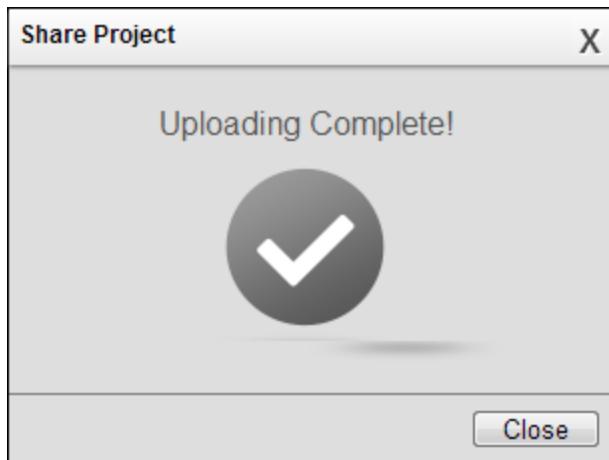
1. Import a project using the option: [Import Cloud Studio](#). The project is imported and opened in your Kony Visualizer.
2. Make the necessary changes to the project and save it.

**Note:** In the Notes (Form Properties >Notes), enter all the modification made to the form. This would allow others to know the changes you have made to the form.

3. Click Cloud > Share Project.
4. System prompts you to confirm replacing the existing project on the cloud. Click **Apply** to upload the modified project.



5. Depending on the size of your project and the speed of your internet connection, uploading the project may take time. Once the project is successfully uploaded, you will receive a confirmation message as show below.



Click Close to complete the process of sharing the project.

#### 14.4.2.2 Shared Project Limitation

A project that is shared on a cloud account can only be reuploaded to the same cloud account. For example, a user shares a project on a cloud account A. You can import this project (if you have necessary permissions) and make changes to the project (if necessary) and then re-export it back to the same cloud account A. You will not be allowed to share this project on any other cloud account even if you have access to multiple accounts (like Cloud account B, C and D).

Hence, it is always advisable that an admin create a cloud account specific to a project and invite all the stakeholders of the project to this account. This would ensure that project is available for all to share and modify.

#### **Steps for inviting users to a cloud account**

As an admin of a cloud, you can invite users by following the below steps:

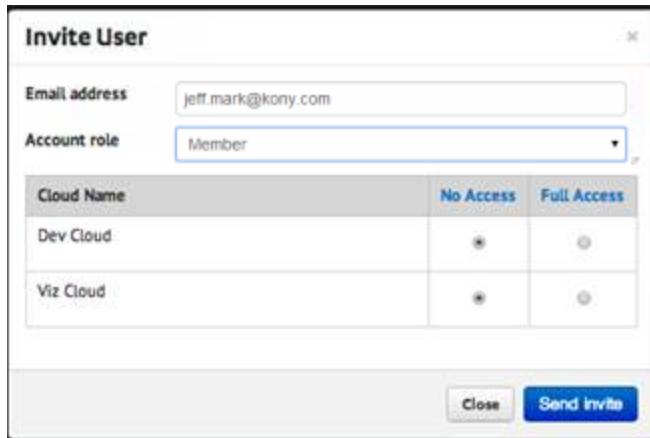
1. Visit <http://manage.kony.com> and enter your Kony Cloud login credentials.
2. After successfully validation login credentials, you are redirected to the **Manage Clouds** page.

All your cloud accounts are displayed. From the left panel, click **Invite Users**.

The screenshot shows the 'Manage Clouds' page at <https://manage.kony.com/#/manage-clouds/>. The left sidebar has sections for CLOUDS (Manage Clouds, + Add a new Cloud), USERS (Manage Users, + Invite user), ACCOUNT INFORMATION (Profile, Manage Payments, Invoices), and REPORTING (Standard Reports). The main area displays two clouds:

| Cloud Name  | Current Monthly Usage | Administration   |
|---|-----------------------|--|
| <b>Dev Cloud</b><br>Development Cloud - Enterprise Apps<br>Free Usage Cloud |                       | Management Services  App Services  |
| <b>Viz Cloud</b><br>Visualization Cloud<br>1 1 active users                 |                       | Messaging Services<br>Kony Visualizer for Windows  Kony Visualizer for Mac<br>Visualization Services |

3. In the **Invite User** window,



- Enter the email address of user being invited.
  - Select an account role to be assigned to the user. Available roles include: Owner, Admin, Billing, and Member.
  - From the existing cloud accounts, select the type of access to be provided to the user. Selecting **Full Access** option, results in adding the user to the cloud account.
4. User receives an email invite to join the cloud account and follows the process outlined in the mail to register for specified cloud account.

## 14.5 Previewing an App

You can preview an app at any stage of designing to ensure that it renders exactly the same as viewed on the Kony Visualizer.

You can preview using:

- [Quick Preview](#): Allows you to preview the form being viewed on your Kony Visualizer.
- [Functional Preview](#): Allows you to preview the entire application and also, navigate from one form to other, while performing any actions that have been attached to the widgets.

### 14.5.1 Quick Preview

Quick Preview allows you to view the current form being designed in Kony Visualizer on an emulator. Currently, you can perform quick preview from Kony Studio only.

Prerequisite for quick preview:

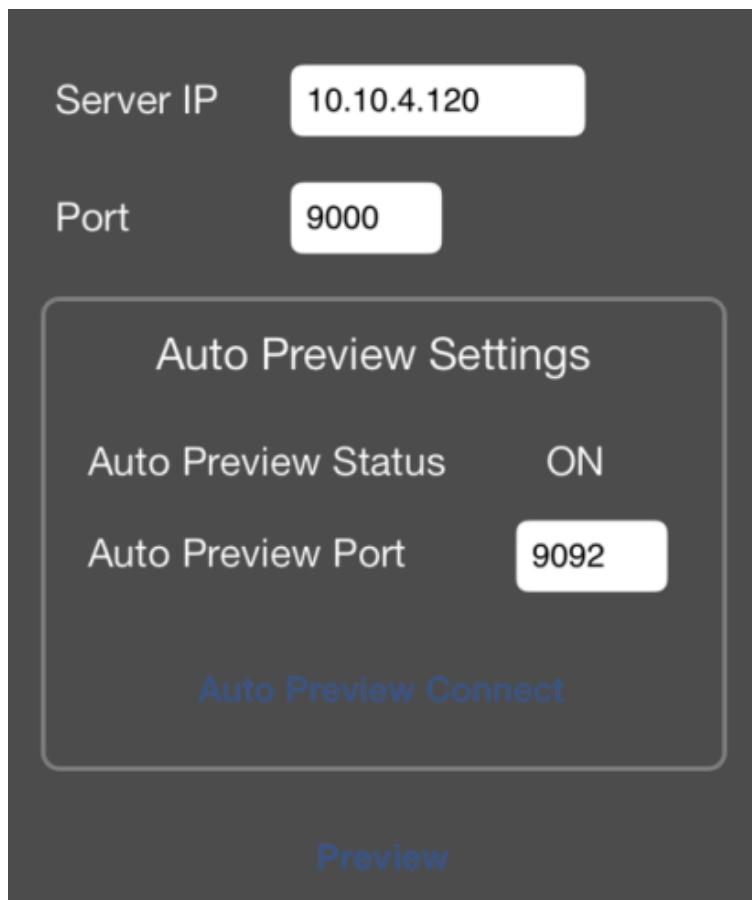
- Kony Studio installed in your system.
- **Automatic Preview** installed and configured.

**Note:** Refer to Kony Studio User Guide for steps to install and configure Automatic Preview.

**Note:** In Kony Studio, Quick Preview is referred as Automatic Preview.

Steps for configuring and viewing quick preview

1. Launch Automatic Preview emulator from your Kony Studio.
2. The emulator displays **Preview Settings** window.



Enter the following details:

1. Server or IP address: Provide the IP address of your computer.
2. Port: Provide the port number as **9000**.
3. In **Auto Preview Settings**:
  - a. Auto Preview Status: If you enable this option, changes made to a form reflect immediately on you Quick Preview emulator.

- b. Auto Preview Port: Provide the port number as **9092**.
3. Click **Preview**.The app interacts with your Visualizer design, and displays a preview of the current form that exist in Visualizer.

## 14.5.2 Functional Preview

Functional preview allows you to view the front-end prototype of a project on a device. During functional preview, you can run any action attached to a widget such as navigating from one form to other, applying skin to a widget, and opening a pop-up window.

To view functional preview, the following are the prerequisites:

- [Kony Account](#)
- [Kony Visualizer App Preview](#)

You can perform functional preview using one of the below methods:

- [Functional Preview \(Cloud\)](#): Allows you to view the functional preview of a prototype that is published to the cloud.
- [Functional Preview \(Local\)](#): Allows you to view the functional preview of a prototype that is published within your intranet.

### 14.5.2.1 Kony Account

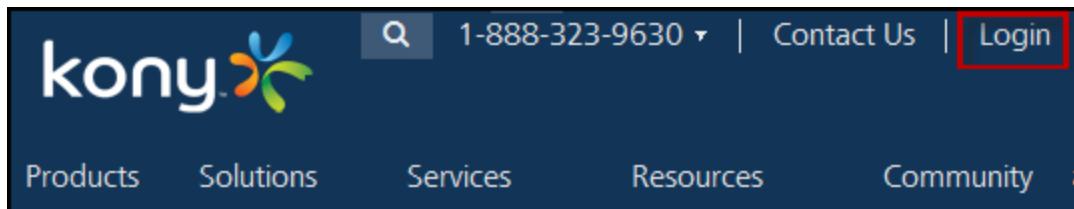
You must have a Kony account to view functional preview of a prototype. If you do not have a Kony account, you can create an account using one of the following methods:

- [Self-Registration](#): Go to Kony website and create a new account.
- [Receive an Invitation](#): You receive an invitation to register with Kony from an existing Kony user.

#### Self-Registration

To self-register, follow these steps:

1. Go to <http://www.kony.com/products/visualization/trial>.
2. Provide the required details, and then click **Sign Up**.



A message appears confirming that your request for registration is accepted.

3. You will receive an email from the **Kony Accounts** with an activation link. Click **Activate My Kony Account**.

**From:** Kony Accounts [mailto:support@kony.com]  
**Sent:** Monday, December 09, 2013 1:46 PM  
**To:** [REDACTED]  
**Subject:** Thank You for registering with Kony Cloud

## You're almost done!

Dear [REDACTED],

Thank you for registering for a Kony account. Please click below to activate your account:

[Activate My Kony Account](#)

Once you activate your account, you will have full access to the [Kony Cloud Portal](#) and the [Kony Cloud Management Console](#), where you can access information and downloads that will enable you to design, build, deploy and manage multi-channel applications.

If you are having trouble with the activation link above, please copy and paste the following link into your browser:

<https://manage.kony.com/registration/confirm/ff413032-1b93-4cbf-93be-a325d7d05ee2>

Sincerely,

Kony Accounts Team

The Activate Your Account page appears.

4. Provide the required details, and then click **Create Cloud**.

The screenshot shows the 'Create your Kony Cloud Account' form. At the top, the Kony logo is displayed next to the word 'Cloud'. Below the title, there is a descriptive text about signing up for a free account. The form contains four input fields: 'First name' (John), 'Last name' (Greg), 'Work email' (redacted), and 'Company' (Kony, Inc.). At the bottom left, there is a link 'Already registered? Log in >'. On the right, a blue button with the text 'Create your account >' is highlighted with a red border.

|            |            |
|------------|------------|
| First name | John       |
| Last name  | Greg       |
| Work email | [Redacted] |
| Company    | Kony, Inc. |

Your account is activated, and the dashboard of Kony Cloud appears.

#### Receive an invitation

The owner of a Kony cloud account sends an invite to allow you access the cloud. You receive an email with a Kony account registration link.

**To create a Kony account after receiving an invitation, follow these steps:**

1. In the invitation mail, click Accept Invitation.

[REDACTED] has invited you to a Kony Cloud Account:

Company: [REDACTED]  
Account Name: [REDACTED]

Please click below to accept this invitation and become a user on the Account:

**Accept Invitation **

If you do not want to be added to this account, you can ignore this email. The Invitation will expire after 24 hours.

Sincerely,

Kony Accounts Team

**Connect with US:**



**Contact Info:**

**Kony, Inc.**  
**Empowering Everywhere™**  
7380 West Sand Lake Road, Suite 390  
Orlando, FL 32819  
[www.kony.com](http://www.kony.com)

The Accept Invitation page appears.

- Provide the required details, and then click **Accept invitation**.

## Accept Invitation

Please provide the following information to access the Kony Cloud Portal, the Kony Community and the Kony Cloud Management Console. You'll have immediate access to product information, downloads, support and the ability to deploy apps to the Kony Cloud.

|                  |   |
|------------------|---|
| Email address    | <input type="text"/>  |
| First name       | <input type="text"/>  |
| Last name        | <input type="text"/>  |
| Job Title        | <input type="text" value="Sr. Technical Writer"/>   |
| Industry         | <input type="text" value="Telecom"/> <span style="border: 1px solid #ccc; padding: 2px;">▼</span>       |
| Country          | <input type="text" value="United States"/> <span style="border: 1px solid #ccc; padding: 2px;">▼</span> |
| State for US     | <input type="text" value="California"/> <span style="border: 1px solid #ccc; padding: 2px;">▼</span>    |
| Phone Number     | <input type="text"/>  |
| Password         | <input type="password" value="....."/>  |
| Confirm password | <input type="password" value="....."/>  |

**Accept invitation**

Your account is activated, and the dashboard of Kony Cloud appears.

#### 14.5.2.2 Kony Visualizer App Preview Application

Kony Visualizer App Preview allows you to view the complete prototype designed in Kony Visualizer on your mobile or tablet device.

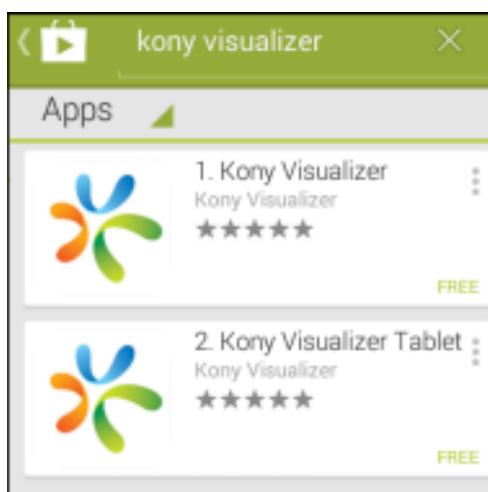
You can download the Kony Visualizer App Preview on your

- Android devices from Google Play Store.
- iOS devices from Apple Store.
- Windows devices from Windows Store.

##### Android

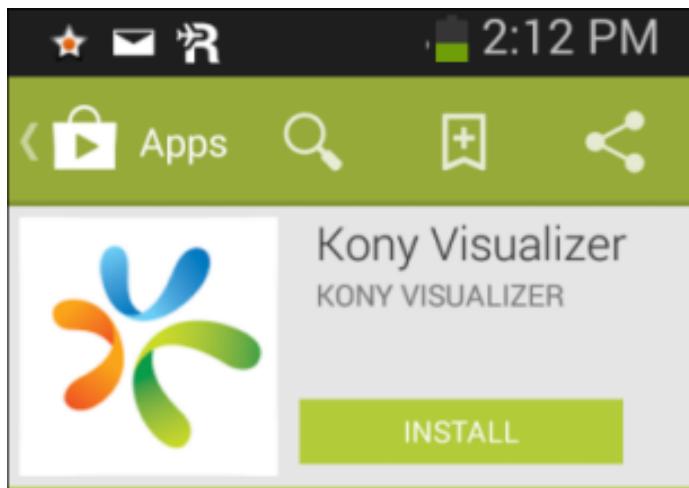
To install Kony Visualizer App Preview, follow these steps:

1. Tap Applications, and then tap Google Play. Google Play Store opens.
2. Search for **Kony Visualizer**.

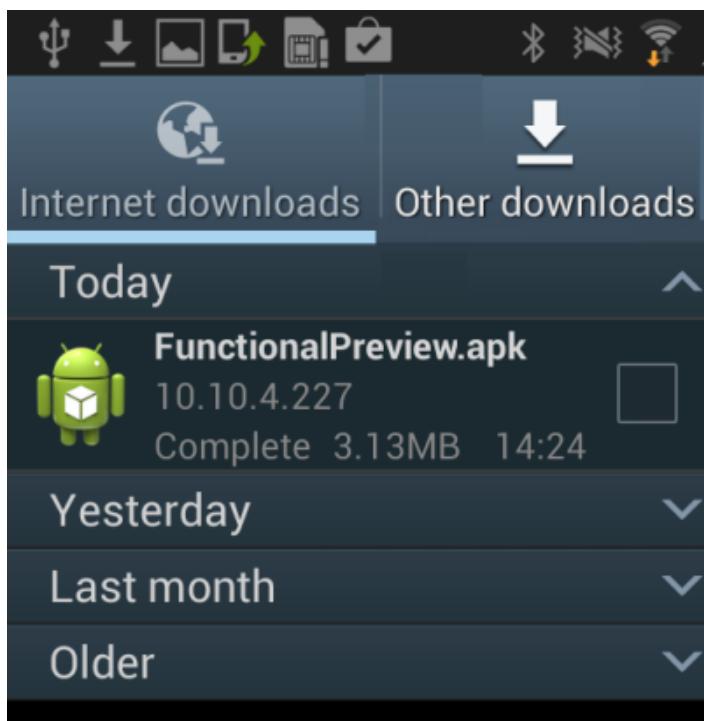


Search results display the Kony Visualizer Functional Preview app installer for mobile and tablet devices.

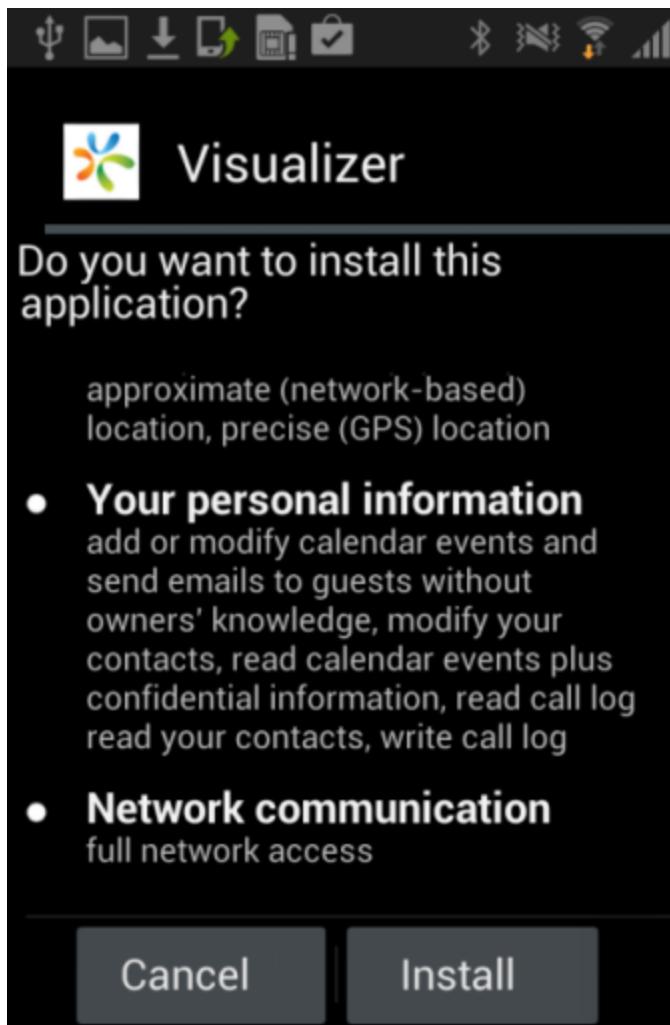
3. From the search results, tap **Kony Visualizer**.



4. Tap **Install**. Type your login credentials, if requested.
5. After completing the installation, navigate to the **Internet Downloads** folder.



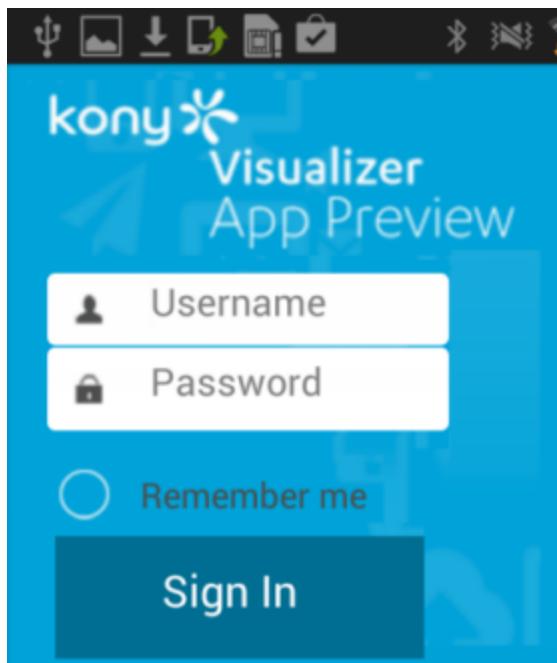
6. Tap **FunctionalPreview.apk** app.



7. Tap **Install**.

8. After installing, you will receive the **Application Completed** message. Tap **Done**.

9. Tap Applications folder, and then tap **Kony Visualizer App Preview** to open it.



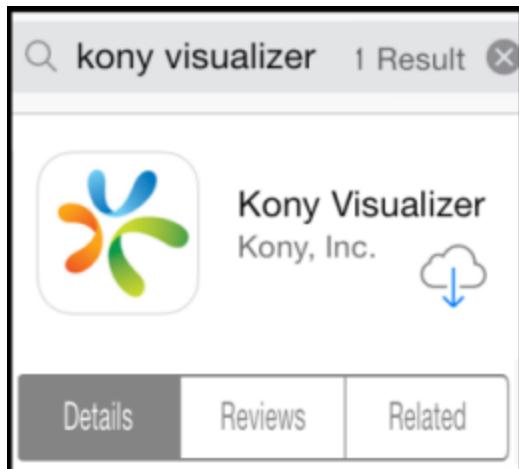
**Note:** Follow the same process for installing the Kony Visualizer App Preview on your Android tablet.

#### iPhone

To install Kony Visualizer App Preview, follow these steps:

1. Tap App Store.
2. Search for **Kony Visualizer**.

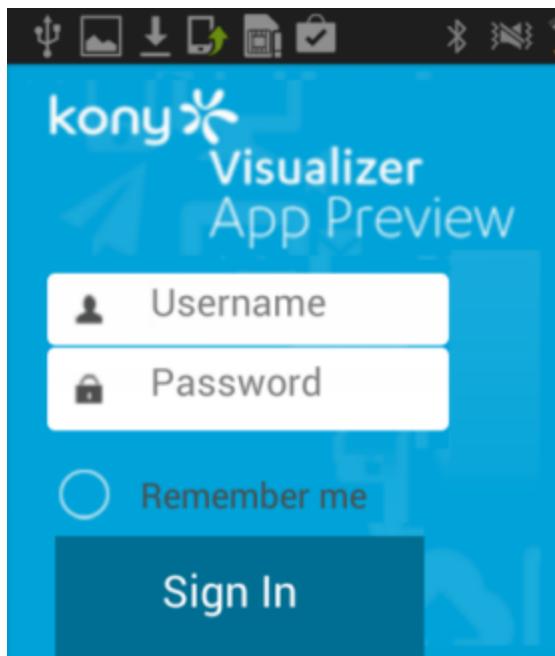
3. From the search result, tap **Kony Visualizer**.



4. Tap **Install**. Type your login credentials, if requested.

The app is downloaded and installed in your mobile device.

5. Tap the **Kony Visualizer App Preview** app icon to open it.

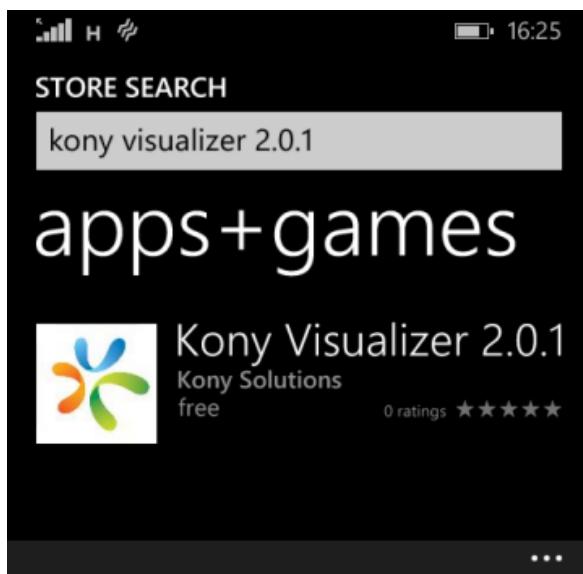


**Note:** Follow the same process for installing Kony Visualizer App Preview app on your iPad.

## Windows 8.1

To install Kony Visualizer App Preview, follow these steps:

1. Tap **Store** and search for **Kony Visualizer**.
2. From the search result, tap **Kony Visualizer**.

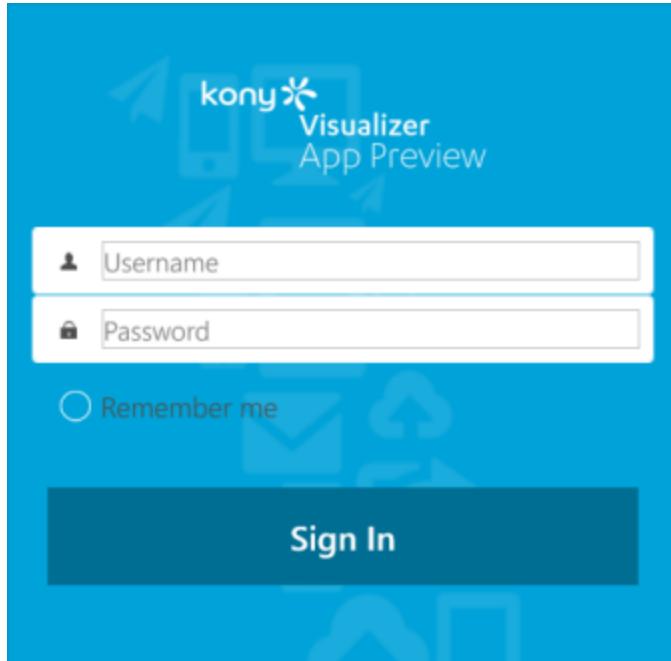


3. Tap **Install**.



The app is download and installed in your mobile device.

4. Tap the Kony Visualizer App Preview app icon to open it.

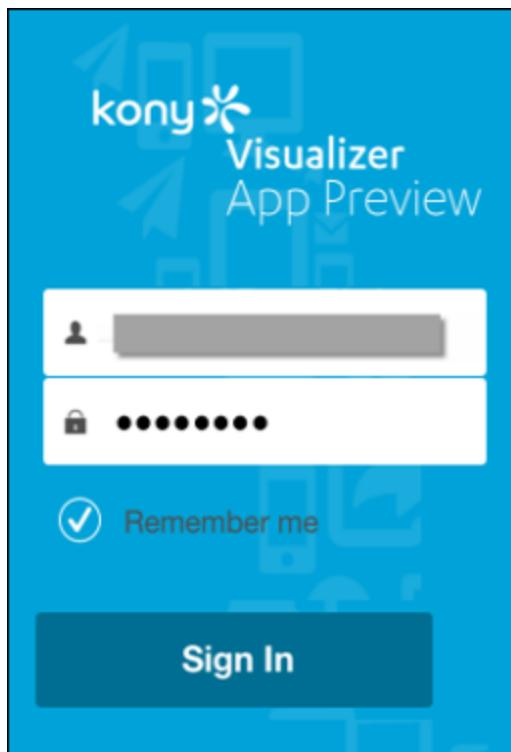


**Note:** Follow the same process for installing Kony Visualizer App Preview on your Windows tablet.

#### 14.5.2.3 Functional Preview (Cloud)

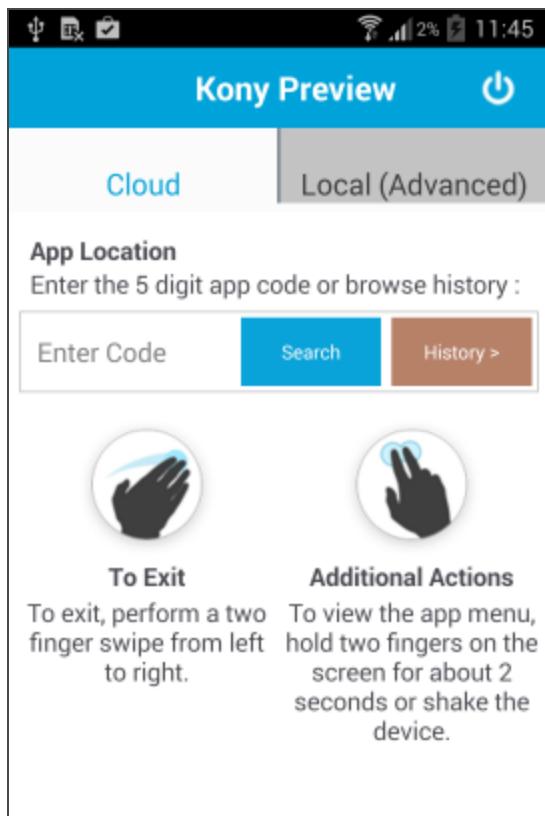
To view the functional preview of your prototype on cloud, follow these steps:

1. Tap Kony Visualizer app on your device. The application launches.
2. Provide your Kony Cloud account credentials on the sign-in page, and then tap Sign In.



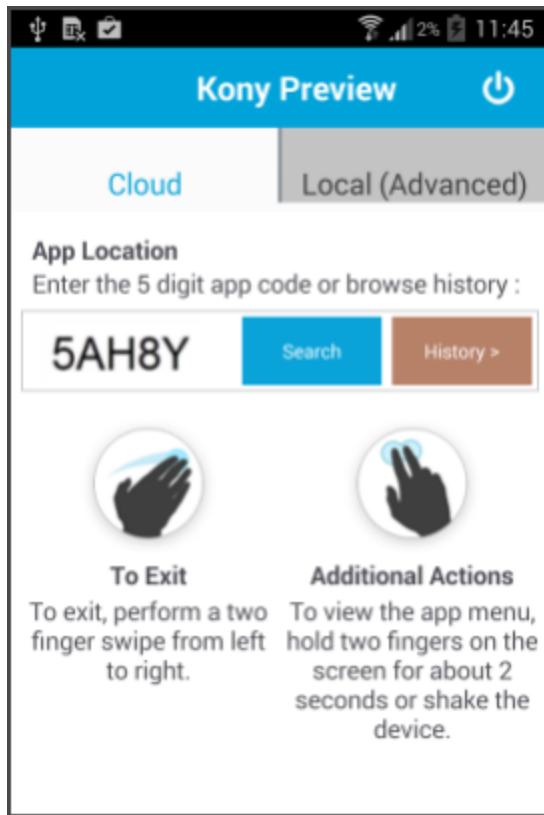
**Note:** To create a new Kony cloud account or to reset your Kony cloud account password, visit <http://manage.kony.com>.

3. The Kony Preview > Cloud page appears.



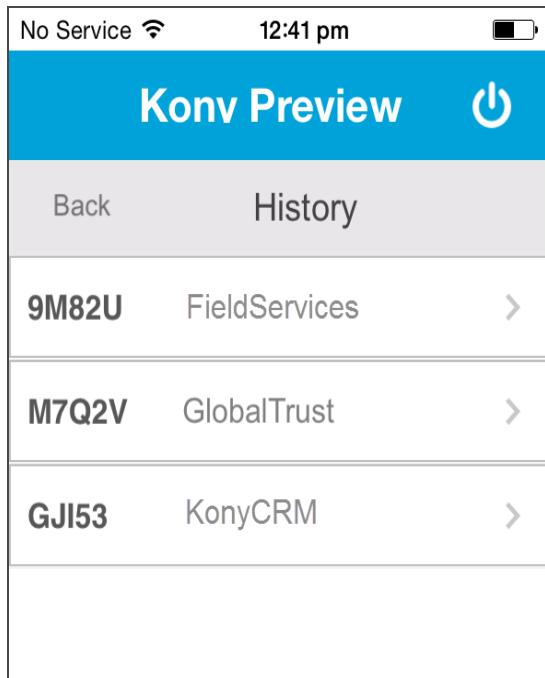
Here, you can perform functional preview using one of the below methods:

- Search using a prototype code: In the **Search** box, enter the prototype code, and then tap **Search**.

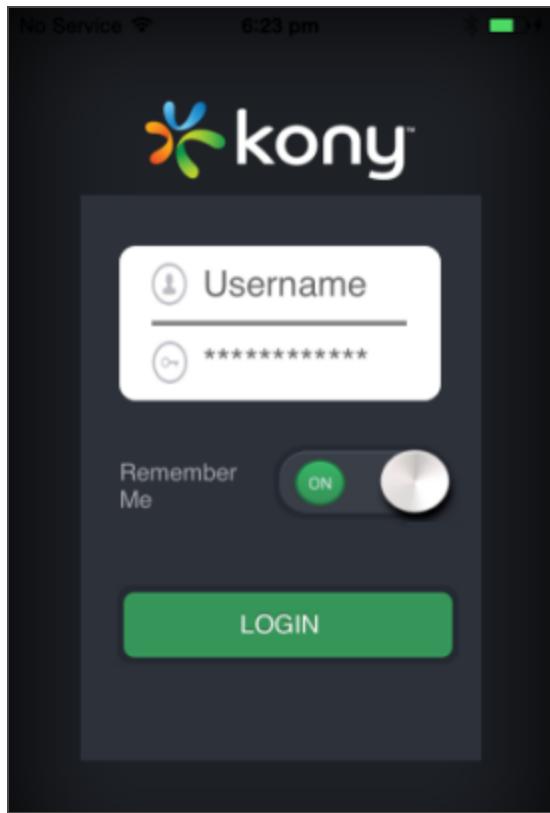


**Note:** The prototype code is generated when an app is published to the cloud. For more details, see [Publish App](#).

- Search for a prototype code in the **History** page. Tap **History** to open a list of prototypes, and then tap a prototype to view the functional preview.



4. The functional preview build downloads to your device. View the prototype of the project, and run any actions attached to the prototype.



View the notes by holding two fingers on the screen for about 2 seconds, or shake the device.

To exit the current prototype, swipe two fingers from left to right.

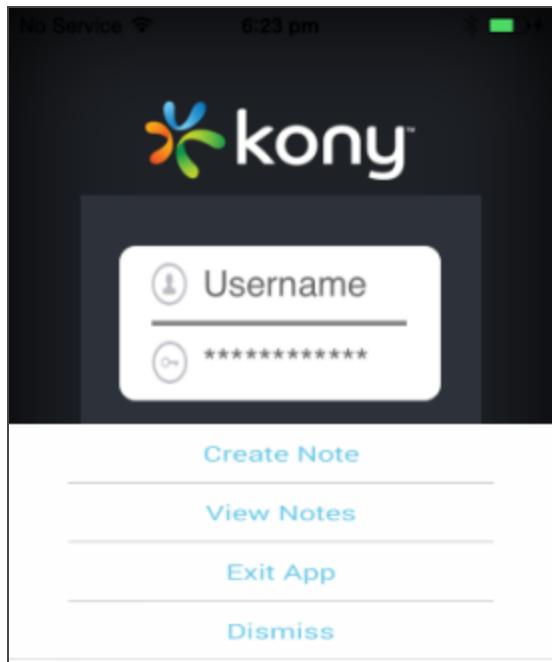
#### Notes

Notes allow your users to provide form-level comments, suggestions, and feedback. Team members interact through notes to build a better design and user experience. A note is visible to everyone, but it can only be modified by the user who posted the note.

The Notes tab includes the following posts:

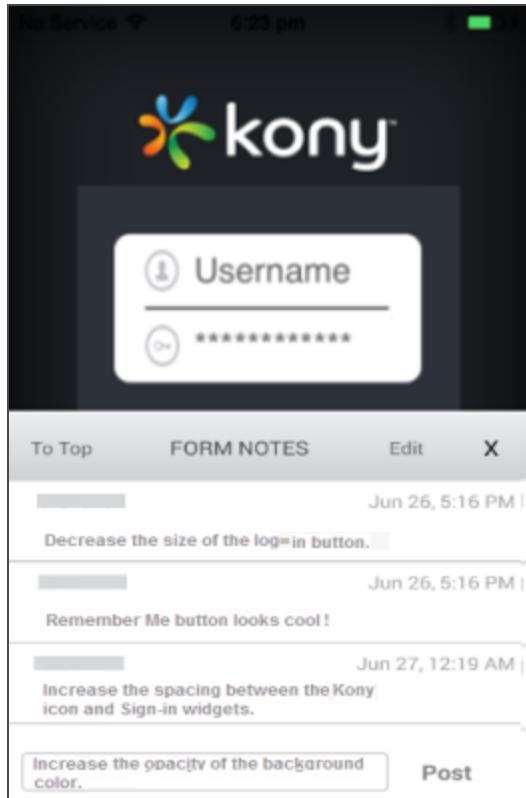
- [Notes](#) recorded before publishing a prototype.
- Notes posted by users while viewing a prototype.

To view the Notes menu during the functional preview, hold two fingers on the screen for about two seconds or shake the device.



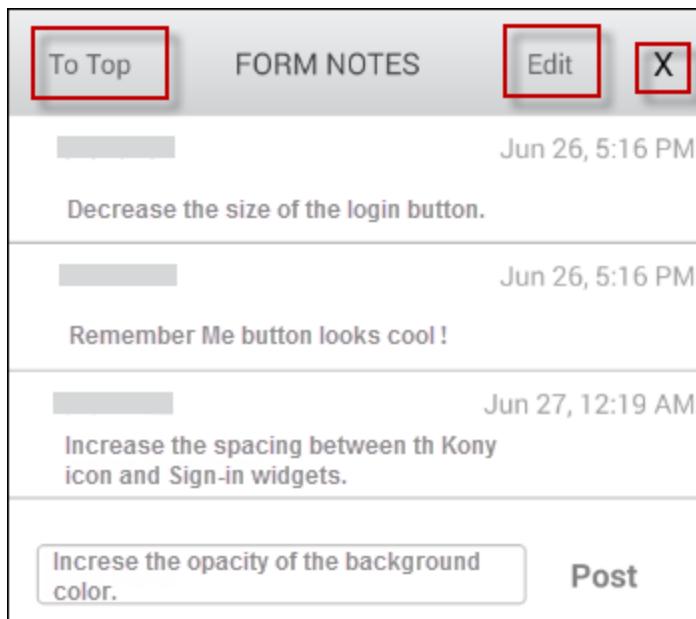
The Following menu items are displayed:

- **Create Notes:** Allows you to view the existing notes and add a new note. In the **Notes** text field, type your comments, suggestions or feedback, and then tap **Post**. Your post is immediately updated in the Notes tab and is visible to all the users. The same note is also updated within the project form and is a visible under **Notes** tab.



- **View Notes:** View the existing notes and also add a new note, if required.
- **Exit App:** Allows you to exit the current prototype. You are redirected to the [Kony Preview > Cloud Page](#).
- **Dismiss:** Allows you to close the **Notes** menu.

While Notes tab is displayed, you can perform the following actions:

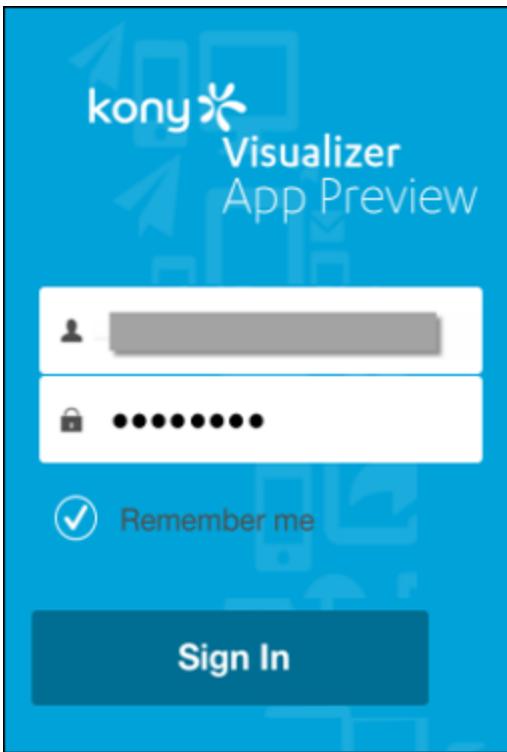


- **To Top:** Allows you to increase the size of the **Notes** tab by tapping **To Top**. This results in **Notes** tab occupying the complete device screen space. To return the **Notes** tab to the default size, click **To Bottom**.
- **Edit:** Allows you to modify or delete own post or add a new note. When you tap **Edit**, an **X** icon is displayed next the post. A post that had red color **X** icon indicates that you have posted the note. Tap the **X** icon to delete the post or to modify the post tap the post and make the necessary changes. Complete the editing process by tapping **Done**.
- **X:** Allows you to close the **Notes** menu list.

#### 14.5.2.4 Functional Preview (Local)

**Note:** Before you view the functional preview locally, you need to generate the Functional Preview build. For more details, refer to [Run](#).

1. Tap **Kony Visualizer** app on your device.
2. When the application is launched, in the sign-in page provide your Kony Cloud Account credentials, and then tap **Sign In**.

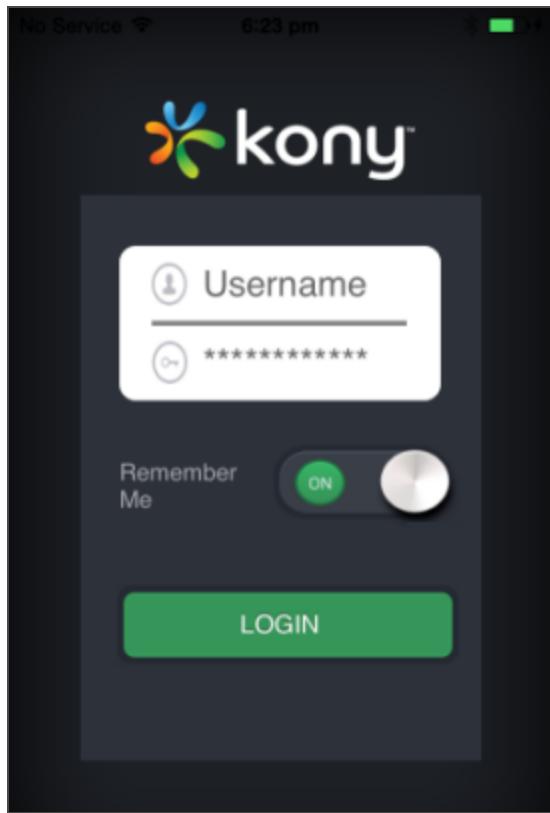


**Note:** To create a new Kony cloud account or to reset your Kony cloud account password, visit <http://manage.kony.com>.

3. After successfully logging to the cloud, you are redirected to **Kony Preview > Cloud** page. Tap **Local (Advance)** tab.



4. Input the required details.
  - **IP Address:** The IP address of the machine that where the prototype exist.
  - **Port:** Type the port address as **9989**.
  - **Application:** Type the name of the project.
5. Tap **Launch App**.
6. The functional preview build is downloaded to your device. You can view the prototype of the project, and execute any actions attached to the prototype.



At any time, to exit the current prototype, swipe two fingers from left to right.

## 15. Importing and Exporting a Project

Typically, a UI/UX design for a project is built in Kony Visualizer and then exported to Kony Studio, where developers code the back-end operations such as adding events to a widget, enabling business logic and integrating it with a database. The seamlessly integration of Kony Studio and Kony Visualizer results in building highly interactive applications for mobile and tablet devices.

This section details the steps for

- Exporting a project to Kony Studio
- Importing a project from Kony Studio

### 15.1 Exporting a Project to Kony Studio

You can export a Kony Visualizer project using one of the below methods:

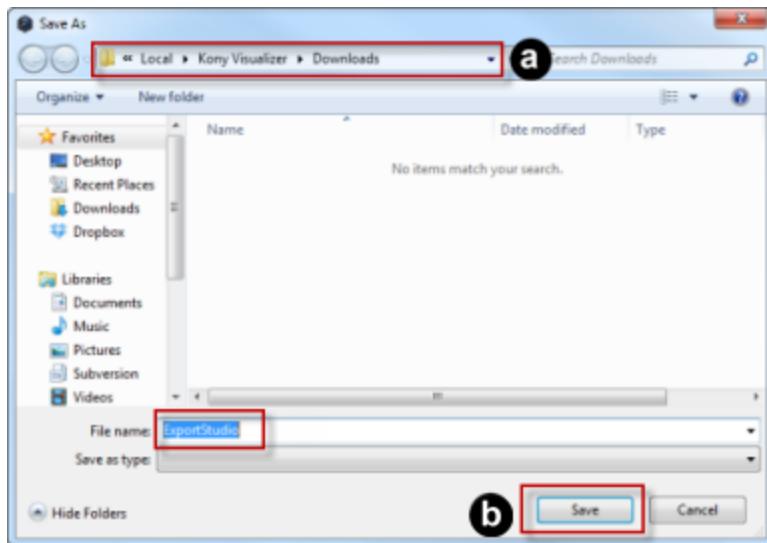
- **Traditional Export Method:** Save the project in your local machine and then share the project as an email attachment or by using any other means of sharing a file.
- **Export to Cloud:** Upload the project file directly to your cloud account and allow your users to import the project file into their Kony Studio.

#### 15.1.1 Traditional Export Method

For exporting a Kony Visualizer project, follow these steps:

**Note:** Ensure you save the project (ctrl+shift+s) before exporting.

1. In Kony Visualizer, click **Project > Export**
2. In the **Save As** dialog box,
  - a. Select a location in your local machine where you wanted the file to be saved.  
Rename the file (if required).
  - b. Click **Save** to save the Visualizer project.



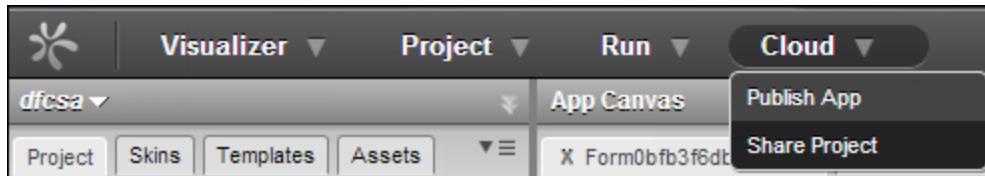
For the steps to import this file into Kony Studio, refer to Kony Studio User Guide.

### 15.1.2 Exporting to Cloud

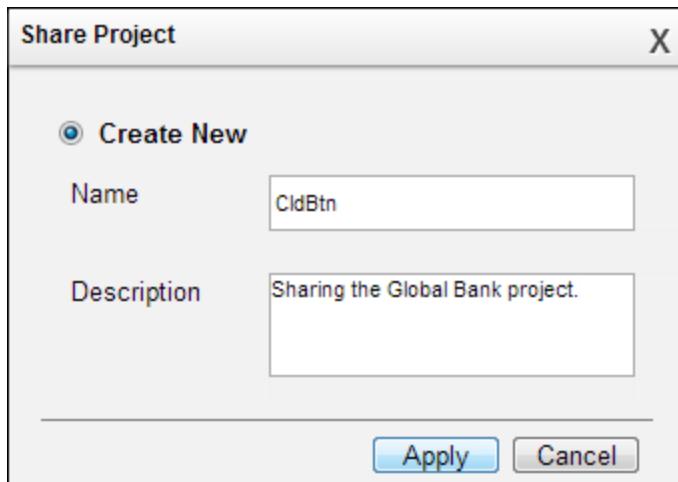
For exporting a Kony Visualizer project to a cloud account, follow these steps:

**Note:** Ensure you save the project (ctrl+shift+s) before exporting.

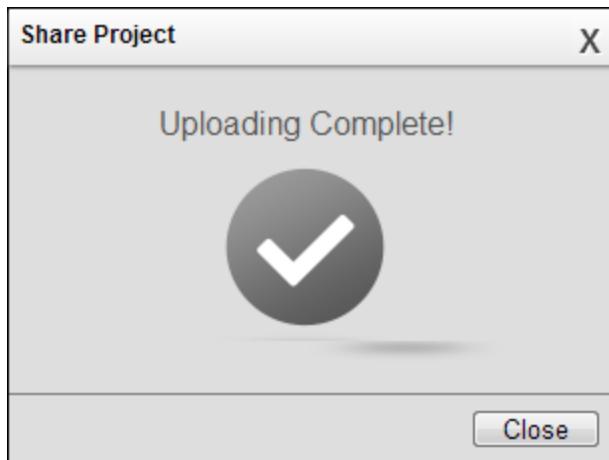
1. In Kony Visualizer, from the top-menu bar click **Cloud > Share Project**.



2. In the **Share Project** window, provide the necessary details and click **Apply**.



3. Depending on the size of your project and the speed of your internet connection, uploading a project may take time. Once the project is successfully uploaded, you receive a confirmation message as show below.



Click Close to complete the process of sharing the project.

For the steps to import this file into Kony Studio, refer to Kony Studio User Guide.

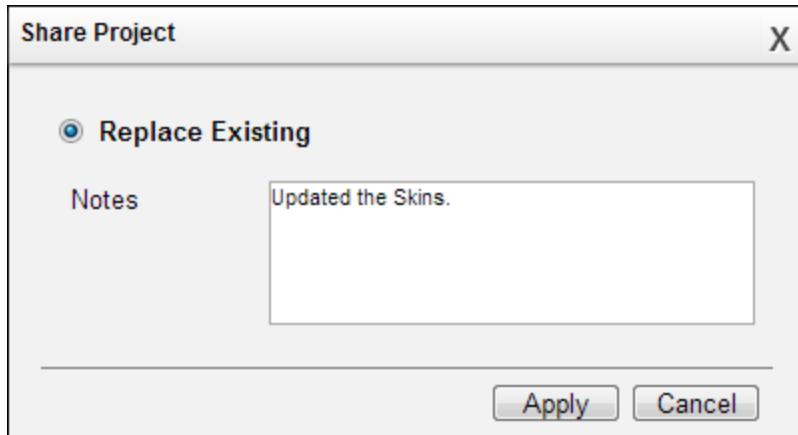
#### 15.1.2.1 Modifying a Project

A project can be modified and exported to the cloud by following these steps:

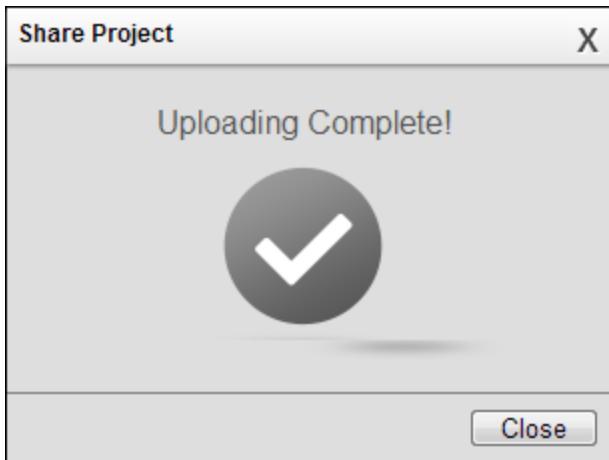
1. Import a project using the option: [Import Cloud Studio](#). The project is imported and opened in your Kony Visualizer.
2. Make the necessary changes to the project and save it.

**Note:** In the **Notes** (Form Properties > Notes), enter all the modification made to the form. This would allow others to know the changes you have made to the form.

3. Click Cloud > Share Project.
4. System prompts you to confirm replacing the existing project on the cloud. Click **Apply** to upload the modified project.



5. Depending on the size of your project and the speed of your internet connection, uploading the project may take time. Once the project is successfully uploaded, you will receive a confirmation message as show below.



Click Close to complete the process of sharing the project.

**Note:** A project imported from the cloud can only be exported to the same cloud account. That is, you cannot copy or move a project from one cloud account to other.

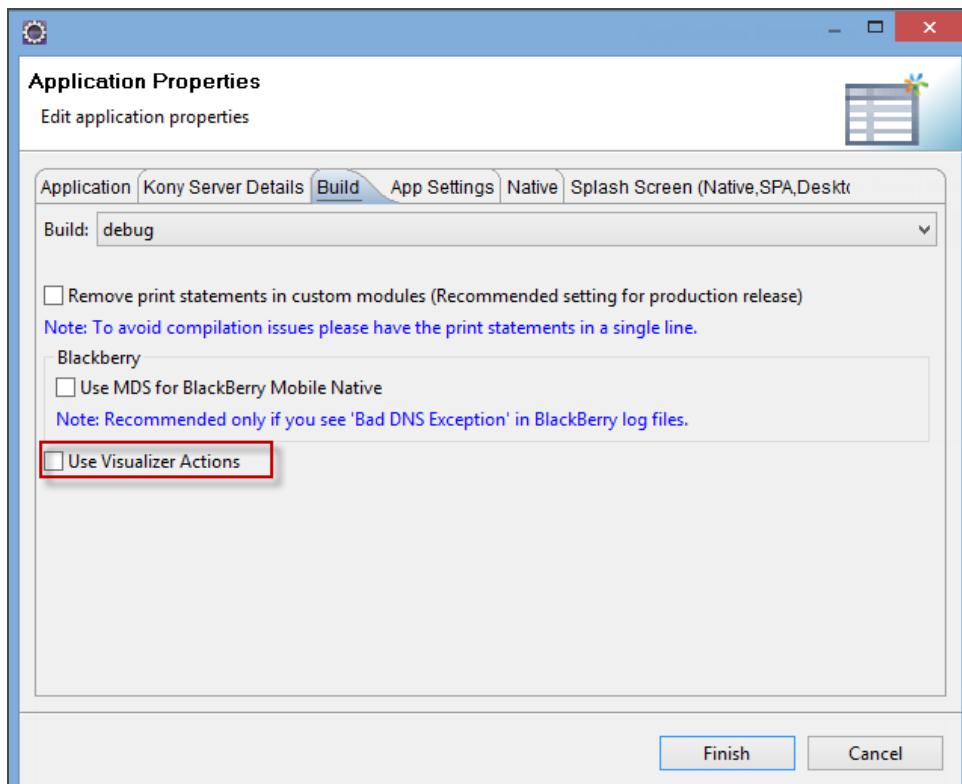
For limitations on Shared project refer [Shared Project Limitation](#).

### 15.1.3 Building a project with Visualizer actions in Kony Studio

In Visualizer, you can generate the functional preview of a Kony Studio project, and run a Kony Studio specific action-events using [Run-Studio](#). Similarly, in Kony Studio, you can generate the functional preview of a Visualizer project and run a Visualizer specific action-events using the following steps:

1. In Kony Studio, right-click a Kony Visualizer project and then click **Properties**. The **Application Properties** window appears.

2. On the Build tab, select the Use Visualizer Actions check box.



3. Click Finish.

#### 15.1.4 Limitations on Exporting to Kony Studio

Following are the limitations on exporting a project to Kony Studio:

- Slider background skins: There are intermittent issues with slider background skins when you import to Kony Studio. The slider background skins are not applied to the widget, once you import.
- Multiple map widgets on export: Export fails if multiple map widgets are present on the same form.

- Notes data lost when exported to studio, and then re-imported into Visualizer: When a project is exporting to Kony Studio, and then re-importing into Visualizer, there are intermittent instances when the Notes data is lost.
- Version 5.5 specific skin features are lost when exporting to version 5.0 of the Kony Studio: Skin features such as multi-step gradient, shadow, and border gradient are lost on import into version 5.0 of the Kony Studio as they are not supported.

## 15.2 Importing a Project from Kony Studio

You can import a Kony Studio project into Kony Visualizer using one of the below methods:

- **Traditional Import Method:** Save a Kony Studio project in your local machine and then import the project into Kony Visualizer.
- **Import from Cloud:** Import the project directly from your cloud account.

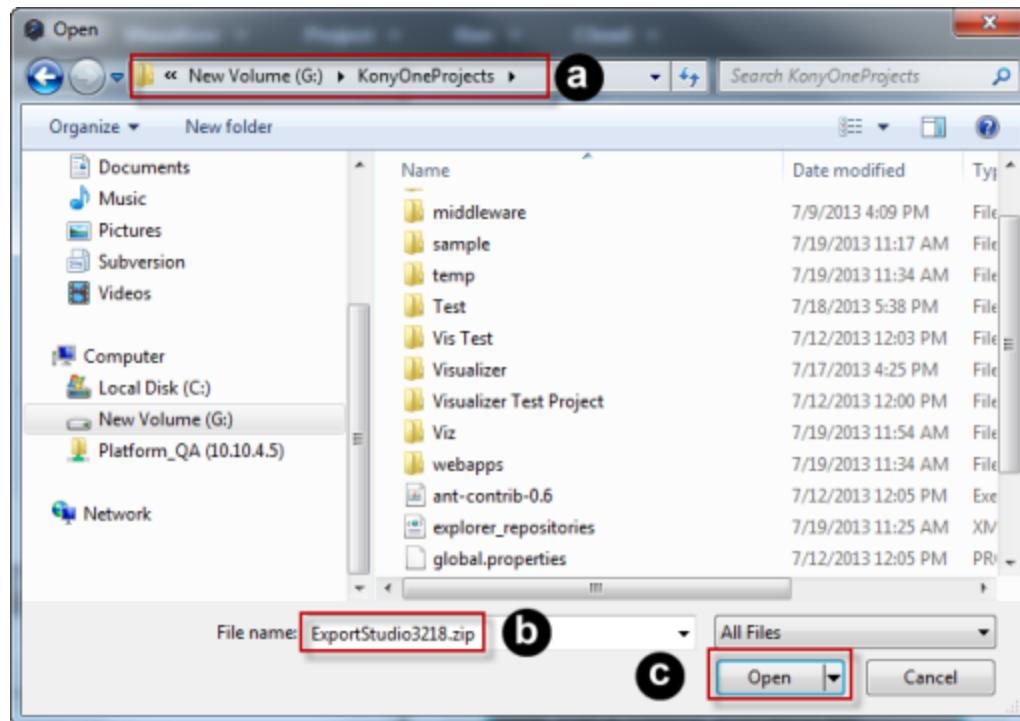
### 15.2.1 Traditional Import Method

For importing a Kony Studio project, follow these steps:

**Note:** Ensure you save the current project (ctrl+shift+s) before importing a project.

1. In Kony Visualizer, from the top-menu bar click **Project > Import Project**.
2. In the **Open** window,
  - a. Navigate to the location in your local machine where you have saved the Kony Studio project file.
  - b. Select the Project.

- c. Click **Open** to import and open the project into Kony Visualizer.



**Note:** While importing, if the project has the same name as that of an existing project, system displays warning message.

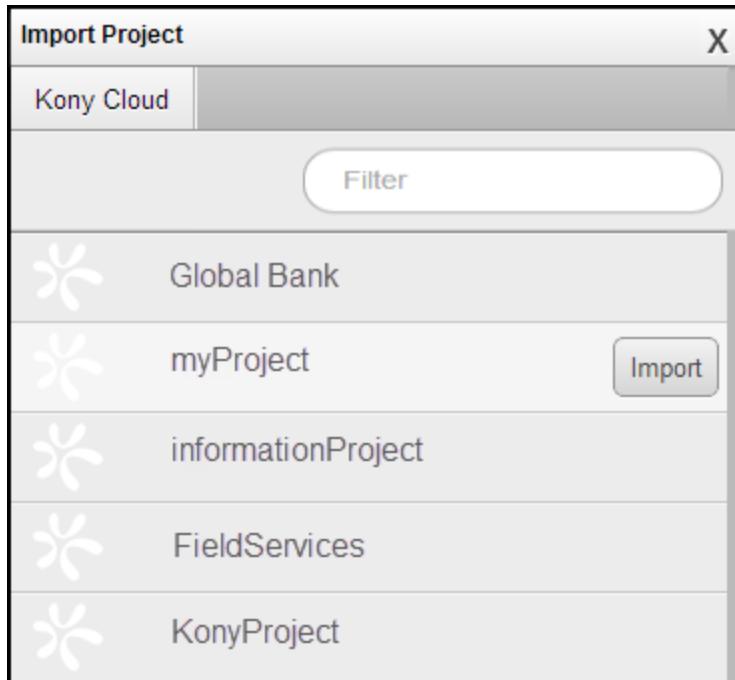
### 15.2.2 Import from Cloud

For importing a Kony Studio project from a cloud account, follow these steps:

**Note:** Ensure you save the current project (ctrl+shift+s) before importing a project.

1. From the top-menu bar, click **Project > Import Cloud Project** to open **Import Project** window.
2. All the projects that are shared by your team on the same cloud account are displayed. Rest the pointer on a project to display **Import** button.

3. Click **Import** to download the project into your local drive and open it in Kony Visualizer



**Note:** If a project with the same name exists in your local drive, system prompts you to choose either to overwrite the existing project or rename the project being imported.

### 15.2.3 Limitations on Importing from Kony Studio

Following are the limitations on importing a project form Kony Studio:

- Platform specific fonts not applied: Platform specific fonts are not always applied to a widget that uses them when exporting from Kony Studio to Visualizer.
- Platform specific images not applied: Platform specific images are not always applied to a widget that uses them when exporting from Kony Studio to Visualizer.
- App Menu icons set in Kony Studio are not showing up in App Canvas after importing: App Menu icons set in Kony Studio are not showing up in App Canvas after you import to Visualizer.

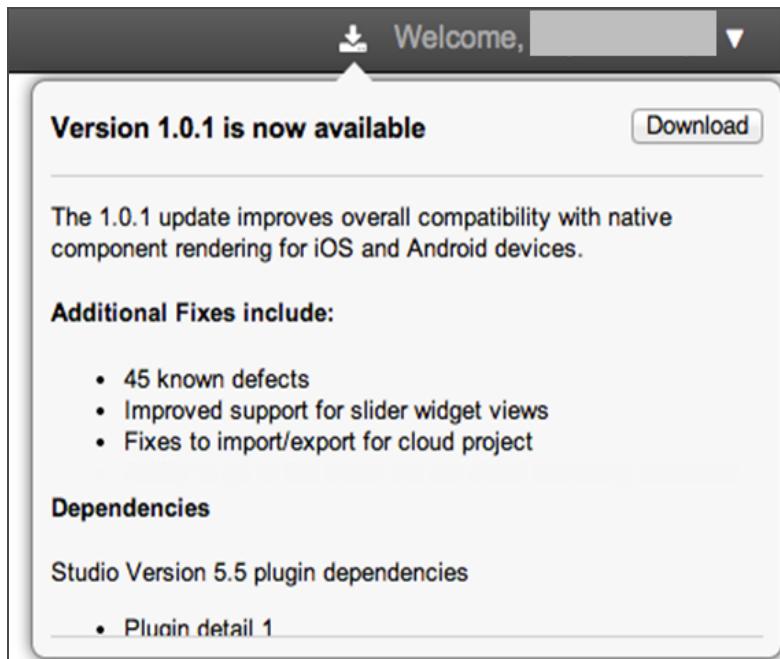
- Skin defaults not applied: Skin defaults specified in Kony Studio are not respected in Visualizer.
- Widget background image lost: For some widgets, skin background images set for various platforms are lost after importing to Visualizer.
- Segment : Unable to edit the segment structure: When you import a Segment widget from Kony Studio, there are instances when editing the existing widgets (inside the segment) or adding widgets is not possible.
- Browser widget in container not visible in canvas: Browser widget placed inside a container in the Kony Studio is not visible in App Canvas after you import into Visualizer.
- Platform specific skins not importing for Slider widget: When you import a Slider widget from Kony Studio, the slider specific skins are not applied.
- Platform specific skins not importing for Switch widget: When you import a Switch widget from Kony Studio, the slider specific skins are not applied.
- Width % import issue when total width is less than 100: When you import Kony Studio project containing widgets that are inside an HBox and where the total width percent of these widget is less than 100, certain issues are noticed in the App Canvas and Properties editor.
- Issues when renaming a Visualizer project prior to export: Renaming Visualizer project results in failure to import into Kony Studio.
- Unable to see app menu on iPhone in Quick Preview when project imported from Kony Studio.
- Import fails when Calendar widget cell template is present: If the project that you import contains a Calendar widget cell template, then import into Visualizer fails.
- Deprecated widgets are not supported.
- Widget and Skin IDs must be unique (inclusive of case-insensitivity.)

## 16. Upgrade Kony Visualizer

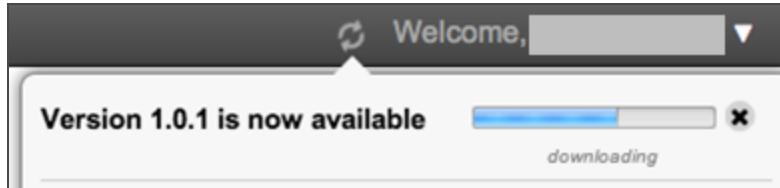
Kony Visualizer regularly checks for upgrades and notifies if any are available.

To download and install Kony Visualizer upgrade, follow these steps:

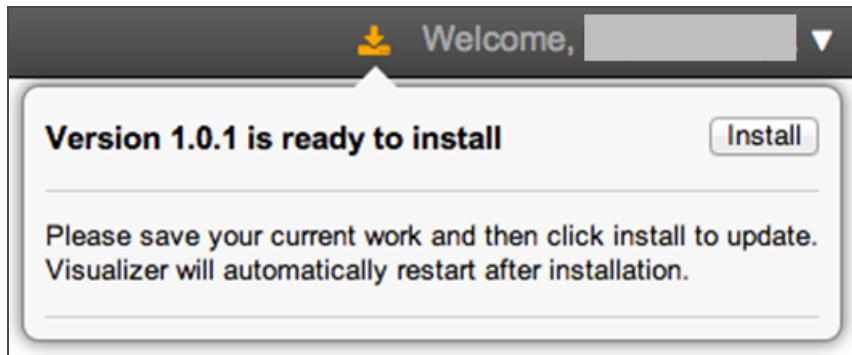
1. When you are notified about an upgrade in Visualizer, click the notification button to view the details.



2. Click Download.



When the download is complete, the following message appears:



3. Click **Install**.

## 17. Internationalizing (i18n) Application Content

The process of designing or developing an application in such a way that it supports various languages and regions is known as Internationalization. Internationalization is abbreviated as i18n. i18n support for an application has to be defined at the time of designing the application.

This section details the following process:

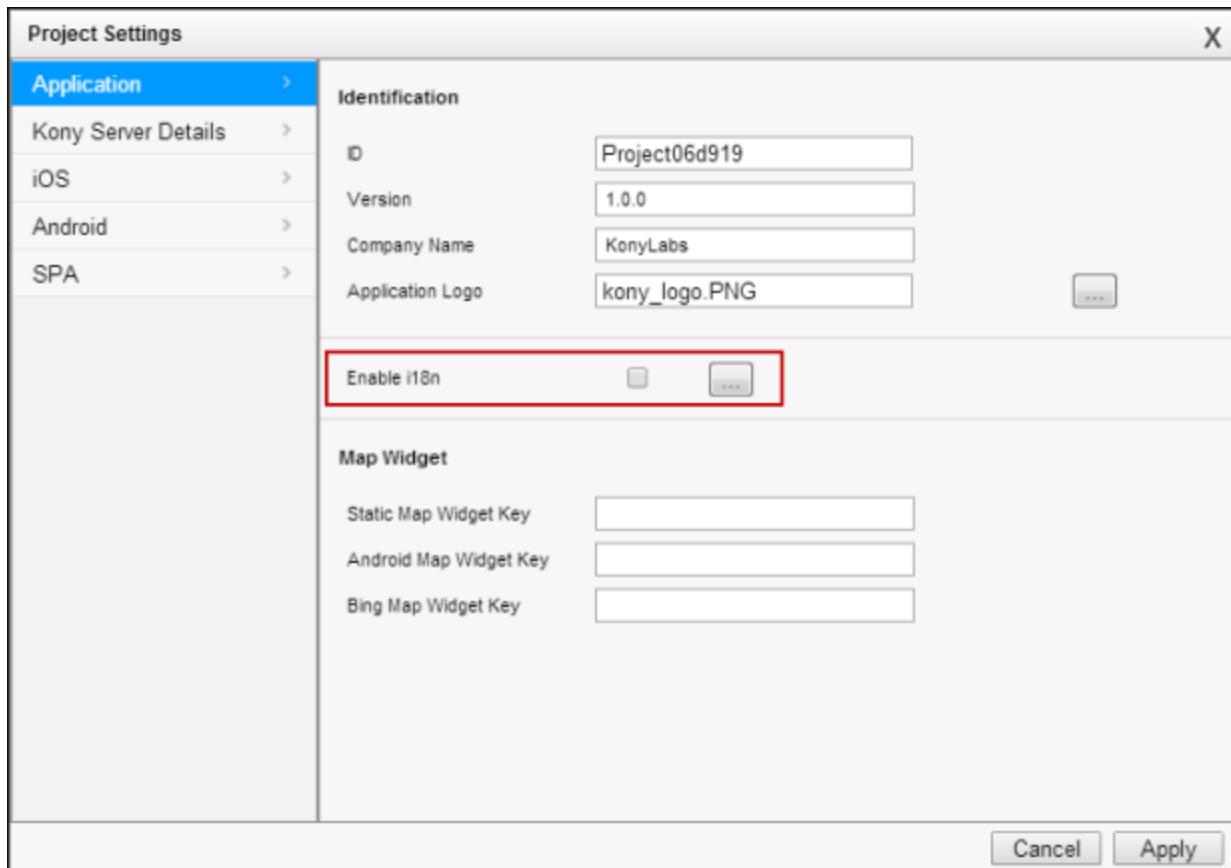
- [Configuring i18n Locales and Adding Keys](#)
- [Assigning an i18n Key to a widget](#)
- [Custom Locales](#)
- [Functional Preview](#)

### 17.1 Configuring i18n Locales and Adding Keys

Visualizer supports predefined and custom locales.

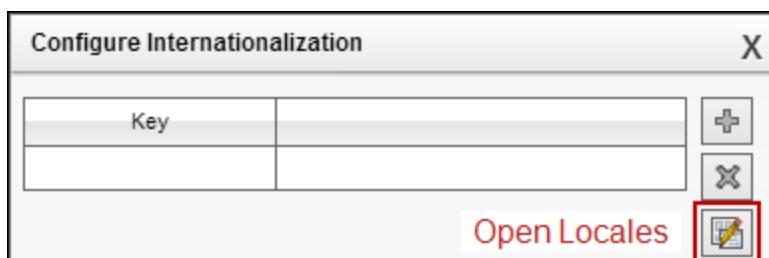
To configure the locales, follow these steps:

1. Click Project > Settings from the top-menu of Visualizer.



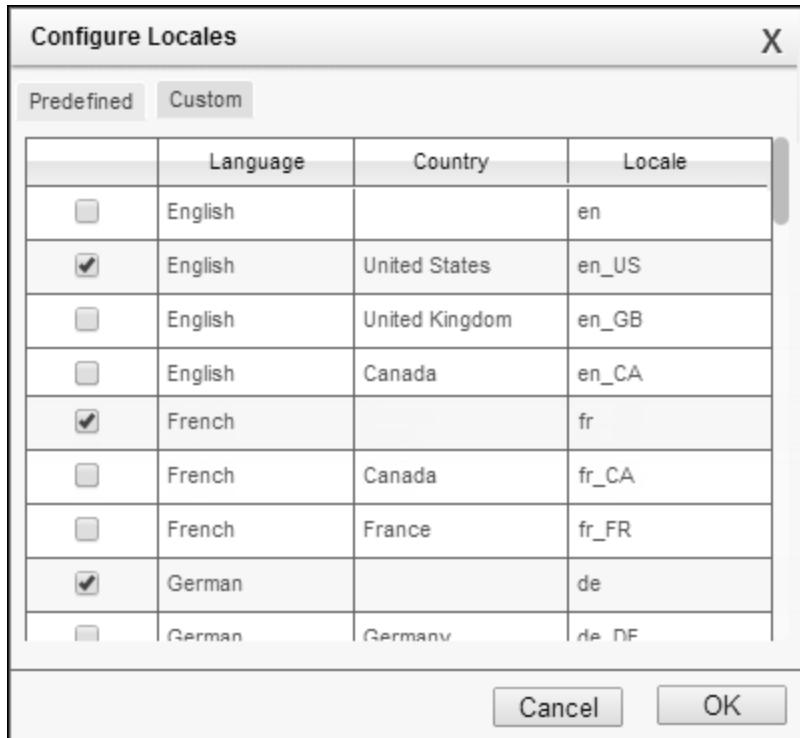
The Project Settings > Application page appears.

2. Select the **Enable i18N** check box.
3. Click ellipsis.



The Configure Internationalization dialog appears.

4. Click the Open Locales icon.

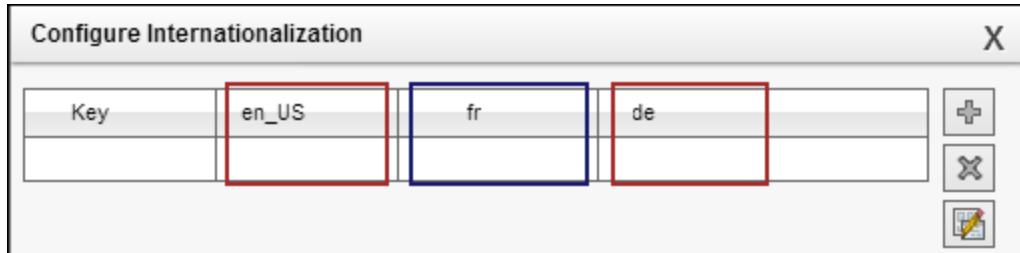


The Configure Locales page appears.

5. From the predefined and custom locales, select the required locales.

**Note:** For more information on custom locales, see [Custom Locales](#).

6. Click OK.



The **Configure Internationalization** page with the chosen locales appear. Here, the following locales have been selected: US English (**en\_US**), France (**fr**), and Germany (**de**).

7. Add the keys, and the corresponding values in different locales.

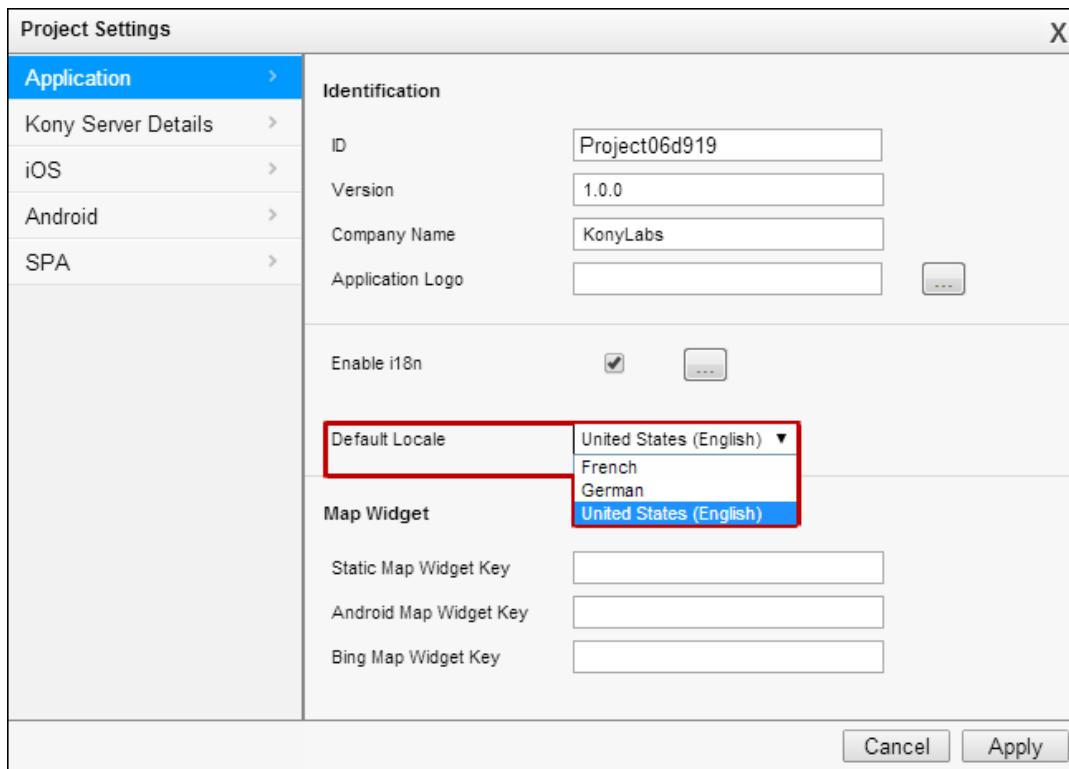
| Key  | en_US    | fr                | de           |  |
|------|----------|-------------------|--------------|--|
| btn1 | Accept   | Accepter          | Akzeptieren  |  |
| btn2 | Cancel   | Annuler           | Stornieren   |  |
| btn3 | Next     | Prochaine         | Nächste      |  |
| btn4 | Previous | Précédent         | Früher       |  |
| btn5 | Sign In  | Se Connecter      | Anmelden     |  |
| txt1 | Username | Nom d'utilisateur | Benutzername |  |
| txt2 | Password | Mot de Passe      | Passwort     |  |

Buttons at the bottom: Cancel, Finish

- Click the plus icon (+), to add a new row.
- Click delete icon (x), to delete a selected row.

8. Click **Finish**.

9. Select a default locale for your project.

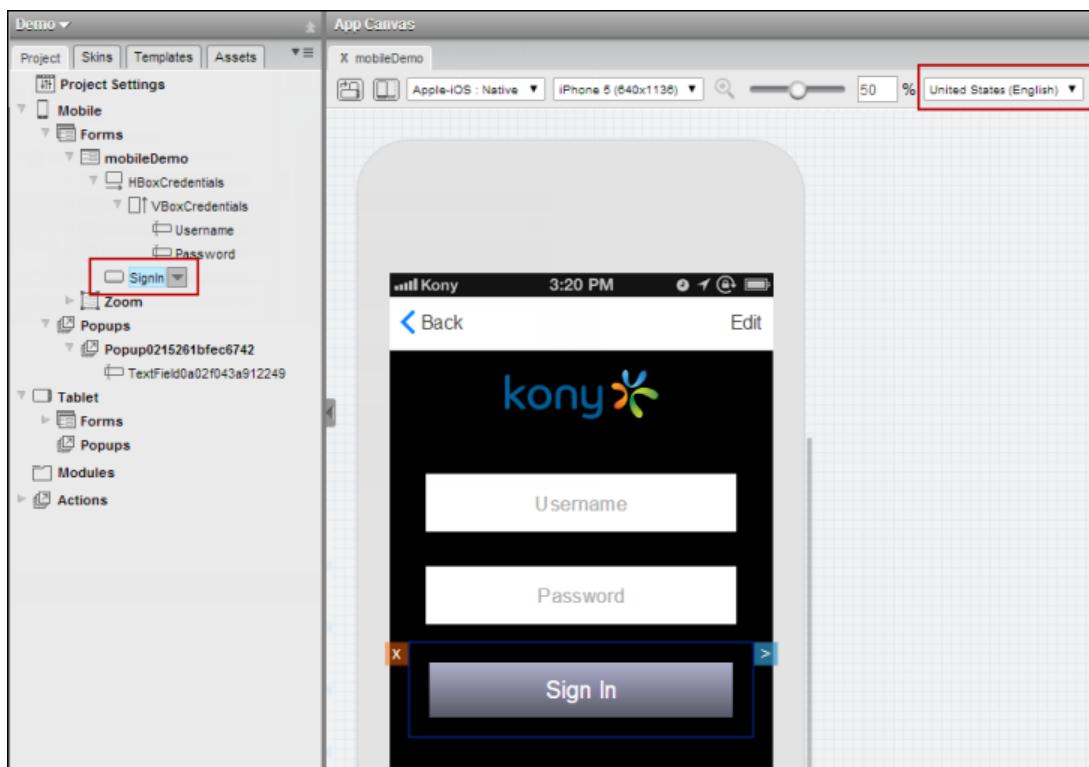


10. Click Apply.

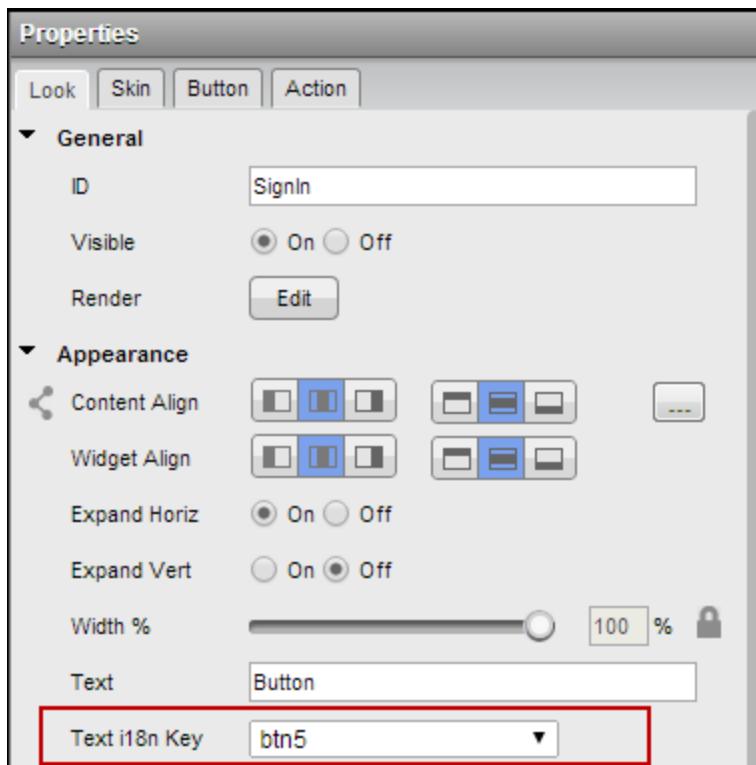
## 17.2 Assigning an i18n key to a widget

To assign an i18n key, follow these steps:

1. Select a widget.



2. Select a locale from the App Canvas.

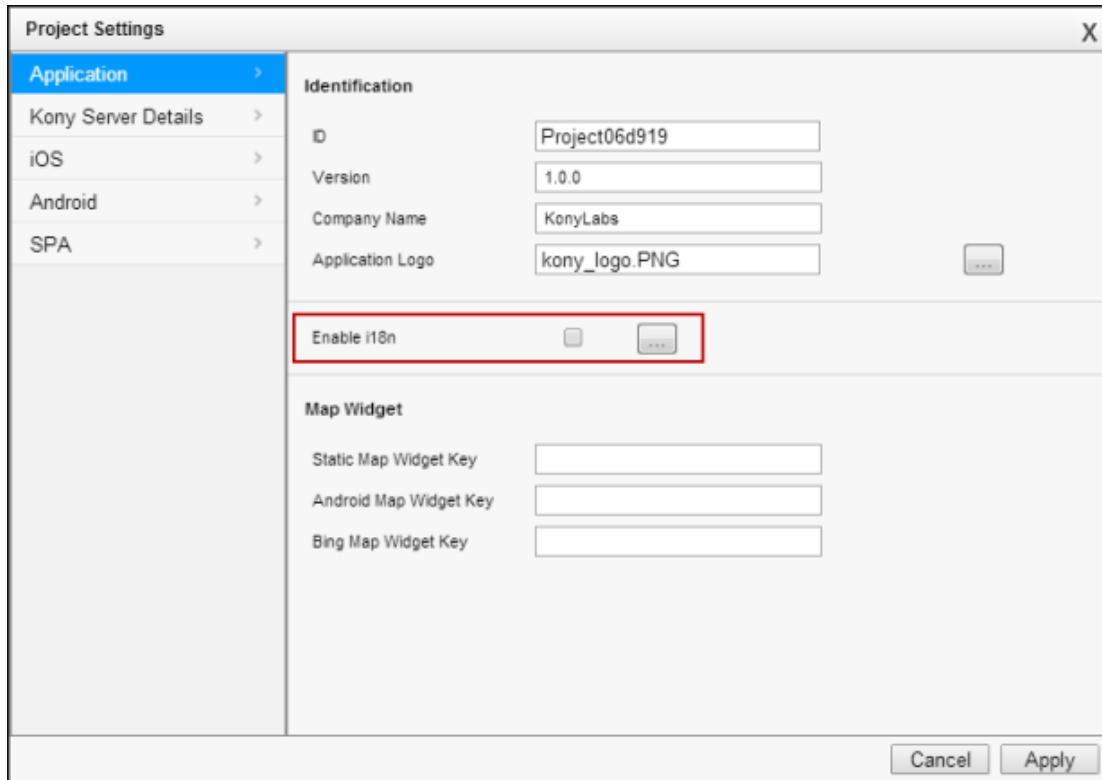


3. Select the key from the **Text i18n key** list from the widgets properties tab.
4. Repeat the preceding steps to assign i18n keys to required widgets.

### 17.3 Custom Locales

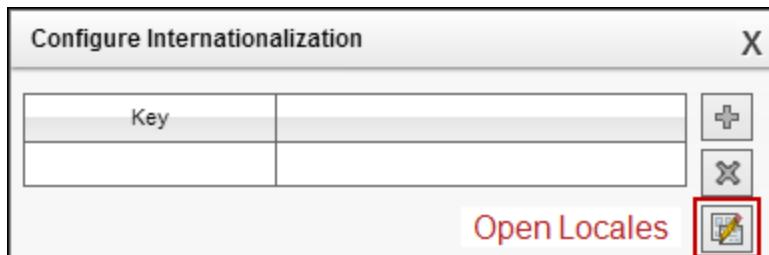
To add custom locales to your application, follow these steps:

1. Click Project > Settings from the top-menu of Visualizer.



The Project Settings > Application page appears.

2. Select the **Enable i18N** check box.
3. Click ellipsis.



The Configure Internationalization dialog appears.

4. Click Open Locales icon.



The Configure Locales page appears.

5. Click the Custom tab.
6. Click the check box in the first empty row.



- Provide the required details.

**Note:** 1. The Locale field must be of the format aa\_AA or aa. For example, jp\_JP, or jp would be a valid locale name.  
2. The Country field can only contain alphabets .  
3. The Language field must necessarily have its first character as an alphabet.

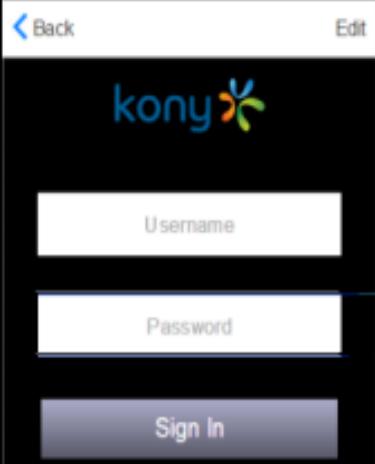
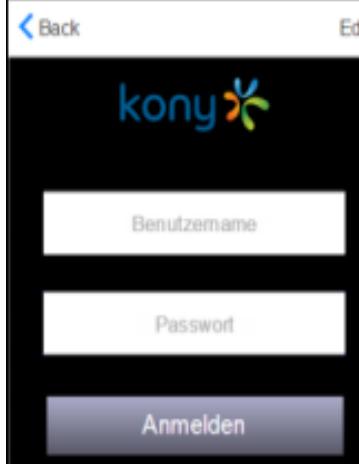
- Click OK.

## 17.4 Functional Preview

To view the functional preview of an i18n app, follow these steps:

- Select a default locale for your application. Refer [Configuring i18N Locales and Adding Keys](#) for assigning a default locale.
- Perform the functional preview of an app. For more information on Functional Preview, refer [Functional Preview](#).

Functional preview of the app on various locales is shown below

| US English (en_US)  | France (fr)   | German (de)   |
|---|---|---|
|  |  |  |

## 18. Visualization Services

Visualization Services provides you with the following details:

- Prototypes: List of the apps published to the cloud.
- Visualizer Projects: List of Visualizer projects uploaded to the cloud.
- Studio Projects: List of Kony Studio projects uploaded to the cloud.

### 18.1 Accessing Visualization Services

To view the Visualization services, follow these steps:

1. Go to <https://manage.kony.com>, provide your Kony account login credentials.

| Cloud Name  | Current Usage             | Administration  |
|---|---------------------------|---|
| <b>Vlz Cloud</b><br>Visualization Cloud -<br>Trial (Unlimited)<br>Available<br><br><a href="#">Modify</a><br><a href="#">Rename</a><br><a href="#">Delete</a> | <b>24</b><br>active users |  <a href="#">Kony Visualizer 2.0.1.1 for Windows</a><br> <a href="#">Kony Visualizer 2.0.1.1 for Mac</a><br> <a href="#">Visualization Services</a> |

After validating your credentials, the **Manage Clouds** page appears.

## 2. Click Visualizer Services.

| <input type="checkbox"/> | Name          | Description | Prototype ID | Platforms | Created By    | Created On       | Last Modified By | Last Modified On |
|--------------------------|---------------|-------------|--------------|-----------|---------------|------------------|------------------|------------------|
| <input type="checkbox"/> | RSImage       |             | EGGFS        | iPhone    | Project Admin | 2/10/15 10:28 AM | Project Admin    | 2/10/15 11:04 AM |
| <input type="checkbox"/> | MyFirstDesign |             | NZIT4        | iPhone    | Project Admin | 8/7/14 7:35 AM   |                  |                  |

The **Console > Prototypes** page appears.

## 18.2 Prototypes

The **Prototypes** page lists all the ( Visualizer and Studio) apps that are published to the cloud.

| <input type="checkbox"/> | Name          | Description | Prototype ID | Platforms | Created By    | Created On       | Last Modified By | Last Modified On |
|--------------------------|---------------|-------------|--------------|-----------|---------------|------------------|------------------|------------------|
| <input type="checkbox"/> | RSImage       |             | EGGFS        | iPhone    | Project Admin | 2/10/15 10:28 AM | Project Admin    | 2/10/15 11:04 AM |
| <input type="checkbox"/> | MyFirstDesign |             | NZIT4        | iPhone    | Project Admin | 8/7/14 7:35 AM   |                  |                  |

This page provides the following information:

- **Name:** Name of a Prototype. Click the name to view a list of viewers who have viewed the prototype.
- **Description:** Provides description of a prototype.
- **Resource Name:** Name assigned while uploading or sharing.

- **Resource Type:** Either **prototype** or **visualizer**. When you publish an app, the resource type is **prototype**. When you share a project, the resource type is **visualizer**.
- **Prototype ID:** The alphanumeric code required for viewing the functional preview of a prototype.
- **Platforms:** Lists the platforms on which you can view the functional preview.
- **Created By:** Displays the name of person who has published the prototype for the first time.
- **Created On:** Date when the prototype is first published.
- **Last Modified By:** Displays the name of the person who has last modified the prototype.
- **Last Modified On:** Displays the date when the prototype was last modified.

### 18.3 Visualizer Projects

The **Visualizer Projects** page lists all the Visualizer projects that are uploaded to the cloud.

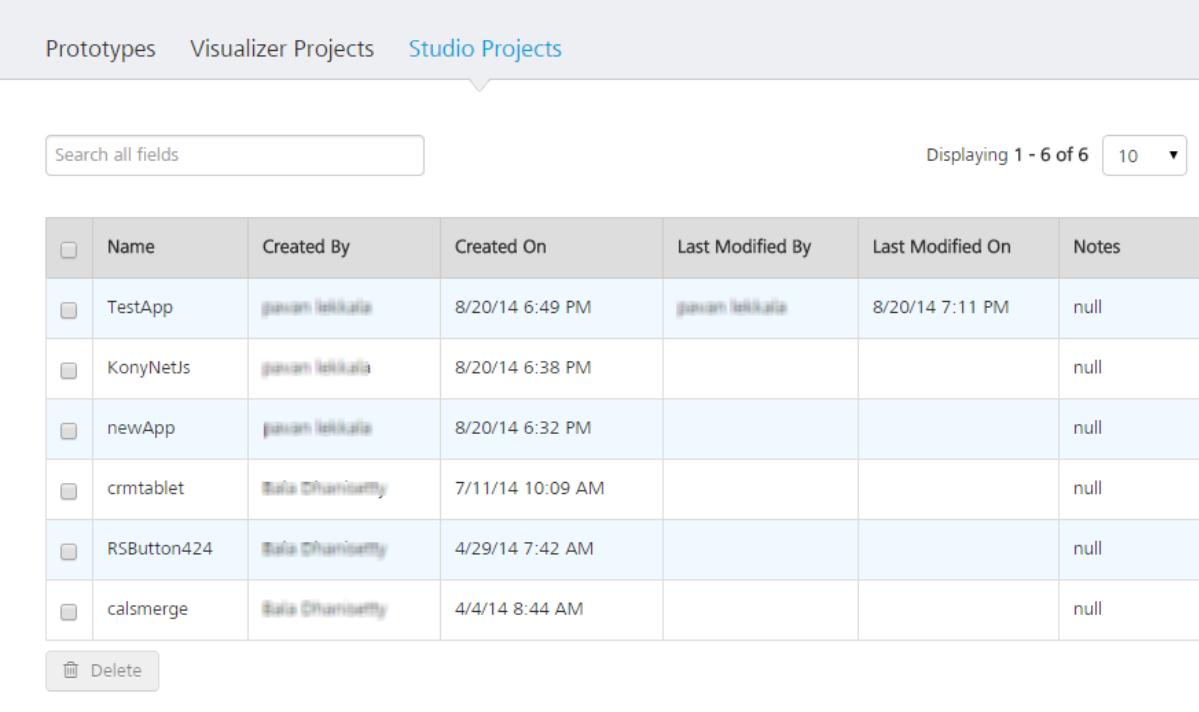
| Prototypes    Visualizer Projects    Studio Projects |           |               |                  |                  |                  |             |
|--|-----------|---------------|------------------|------------------|------------------|-------------|
| <input type="text" value="Search all fields"/>       |           |               |                  |                  |                  |             |
| Displaying 1 - 4 of 4 <span>10 ▾</span>              |           |               |                  |                  |                  |             |
|  | Name      | Created By    | Created On       | Last Modified By | Last Modified On | Notes       |
| <input type="checkbox"/>                             | RDBankSol | Project Admin | 2/12/15 7:44 AM  | Project Admin    | 2/12/15 7:48 AM  | Updated the |
| <input type="checkbox"/>                             | ide       | Project Admin | 6/11/14 11:42 AM |                  |                  |             |
| <input type="checkbox"/>                             | Test      | Project Admin | 3/18/14 10:49 AM |                  |                  |             |
| <input type="checkbox"/>                             | HBox      | Project Admin | 1/22/14 6:46 AM  |                  |                  |             |

This page provides the following information:

- **Name:** Displays the name of a Visualizer project uploaded to the cloud.
- **Created By:** Displays the name of person who has published the prototype for the first time.
- **Created On:** Displays the date when the prototype is first published.
- **Last Modified By:** Displays the name of the person who has last modified the prototype.
- **Last Modified On:** Displays the date when the prototype was last modified.
- **Notes:** Displays the information entered (if any) when the project is last modified.

## 18.4 Studio Projects

The **Studio Projects** page lists all the studio projects that are uploaded to the cloud.



The screenshot shows a web-based application interface for managing studio projects. At the top, there is a navigation bar with three tabs: 'Prototypes', 'Visualizer Projects', and 'Studio Projects'. The 'Studio Projects' tab is currently selected, indicated by a blue background. Below the navigation bar is a search bar labeled 'Search all fields'. To the right of the search bar is a status message 'Displaying 1 - 6 of 6' and a dropdown menu set to '10'. The main content area is a table with the following data:

| <input type="checkbox"/> | Name        | Created By       | Created On       | Last Modified By | Last Modified On | Notes |
|--------------------------|-------------|------------------|------------------|------------------|------------------|-------|
| <input type="checkbox"/> | TestApp     | gaurav telikala  | 8/20/14 6:49 PM  | gaurav telikala  | 8/20/14 7:11 PM  | null  |
| <input type="checkbox"/> | KonyNetJs   | gaurav telikala  | 8/20/14 6:38 PM  |                  |                  | null  |
| <input type="checkbox"/> | newApp      | gaurav telikala  | 8/20/14 6:32 PM  |                  |                  | null  |
| <input type="checkbox"/> | crmtablet   | Bala Channisetty | 7/11/14 10:09 AM |                  |                  | null  |
| <input type="checkbox"/> | RSButton424 | Bala Channisetty | 4/29/14 7:42 AM  |                  |                  | null  |
| <input type="checkbox"/> | calsmerge   | Bala Channisetty | 4/4/14 8:44 AM   |                  |                  | null  |

At the bottom left of the table area is a button labeled 'Delete' with a trash icon.

This page provides the following information:

- **Name:** Displays the name of a Visualizer project uploaded to the cloud.
- **Created By:** Displays the name of person who has published the prototype for the first time.

- **Created On:** Displays the date when the prototype is first published.
- **Last Modified By:** Displays the name of the person who has last modified the prototype.
- **Last Modified On:** Displays the date when the prototype was last modified.
- **Notes:** Displays the information entered (if any) when the project is last modified.

## 19. Support for Legacy Widgets and Old Projects

This section provides the details information:

- [How Visualizer supports legacy widgets.](#)
- [How a project created in earlier versions of Visualizer is upgraded.](#)

### 19.1 How Visualizer supports legacy widgets

Visualizer supports legacy Kony Studio widgets such as TextBox0, TextArea0, Image0, Segment0, HorizontalImageStrip0, ImageGallery0, and Form0. Before you import a Kony Studio project containing legacy widgets, the project should be upgraded to Kony Studio 6.0. For more details on upgrading a Kony Studio project, see [Kony Studio User Guide](#).

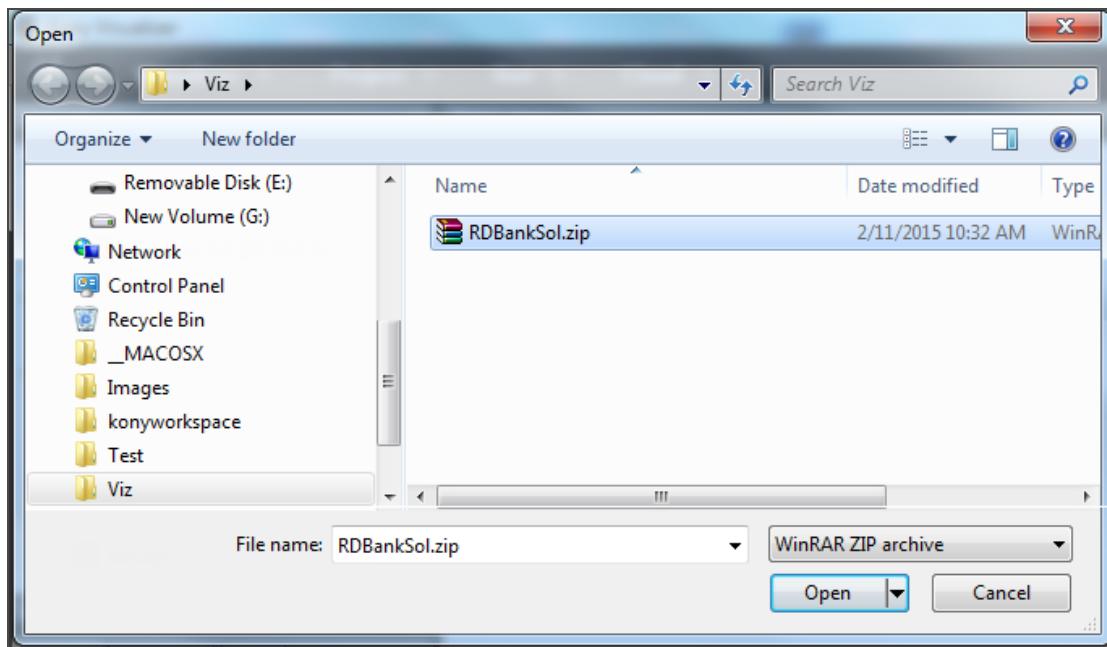
**When a Kony Studio project containing legacy widgets is imported to Visualizer, the following limitations are observed:**

1. The legacy widgets may appear distorted on App Canvas.
2. The legacy widgets are unavailable in the Library > Widgets tab.
3. Properties such as Cut, Copy, Paste, Duplicate, and Add to Collection are unavailable for the legacy widgets.
4. The following properties are unavailable for a container widget having legacy widgets: Cut, Copy, Paste, Duplicate, and Add to Collection.

### 19.2 How a project created in earlier versions of Visualizer is upgraded

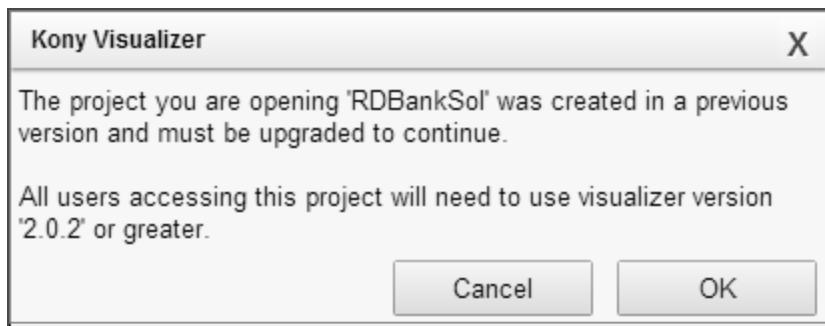
To upgrade a project created in earlier versions of Kony Visualizer, follow these steps:

1. Click Project > Import Project.



The Open window appears.

2. Click the project and then click **Open**.



3. Click **OK**.

## 20. Actions

Actions tab contain a list of actions created for the mobile and tablet devices. These actions can be reused across the project but within the same channel. For example, an action created under **Mobile** tab can only be used for mobile channel elements. The following are the ways to create an action:

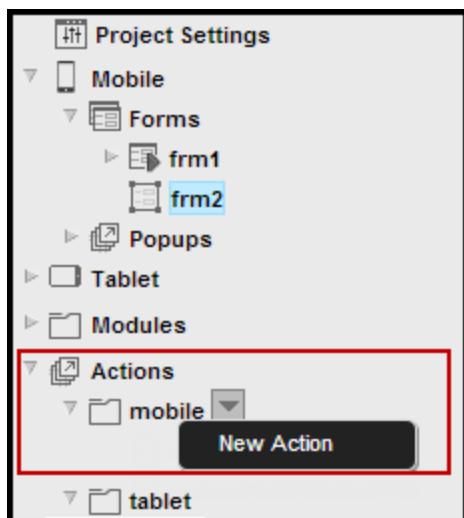
- [Creating an actions from the Action tab.](#)
- [Creating an action from the Action tab of a Widget \(or Pop-up or form\).](#)
- [Assigning an Action](#)

### 20.1 Important Considerations

- You cannot delete an action.
- Only one action can be assigned to an event. If you attempt to assign more than one action, the last action assigned will overwrite the previous actions.

### 20.2 Creating an actions from the Action tab

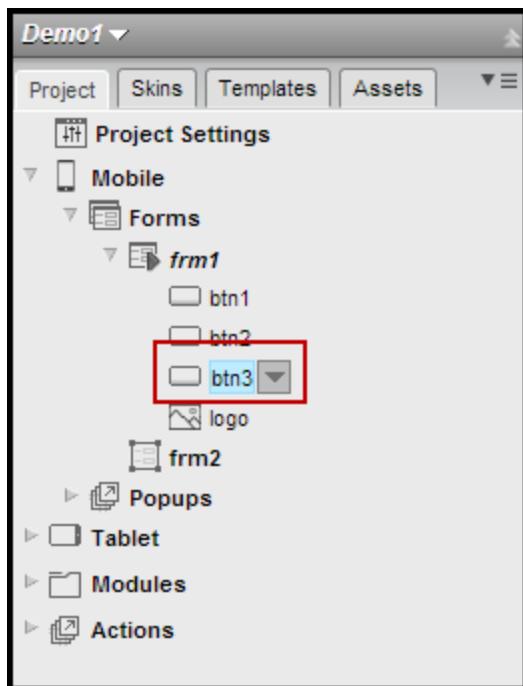
To create an action from the Action tab, follow these steps:



1. Expand the Action tab from the Projects tab.
2. Select a channel (Mobile or Tablet.)
3. Right-click the channel, and then click New Action.
4. Add the required action by referring to the following section: [Adding and configuring an action.](#)

### 20.3 Creating an action from the Action tab of a Widget (or Pop-up or form)

To create an action from the Action tab, follow these steps:



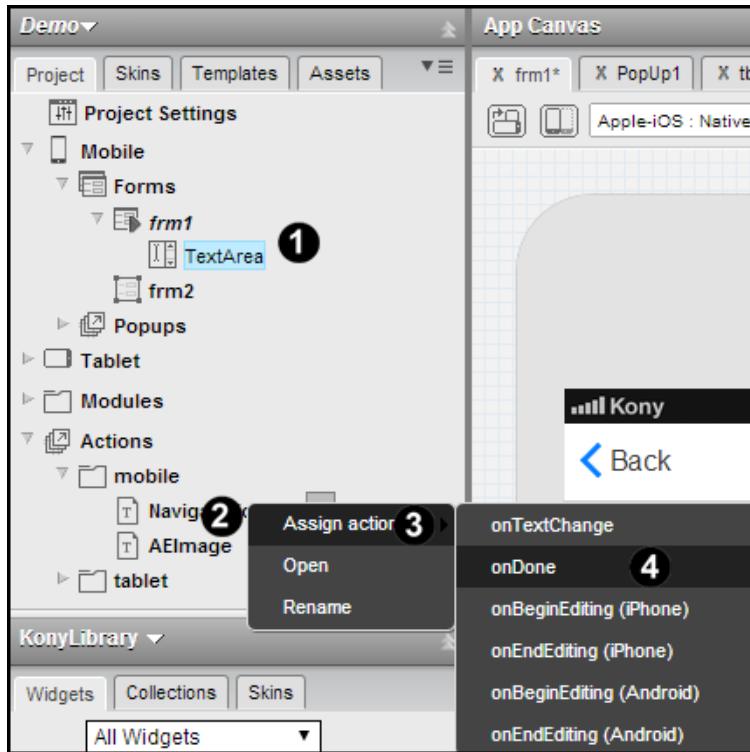
1. Select a widget (or a form or pop-up) from the Project tab,



2. Click the Action tab from the properties tab.
3. Select an event. The Action Editor dialog appears.
4. Add the required action by referring to the following section: [Adding and configuring an action](#).

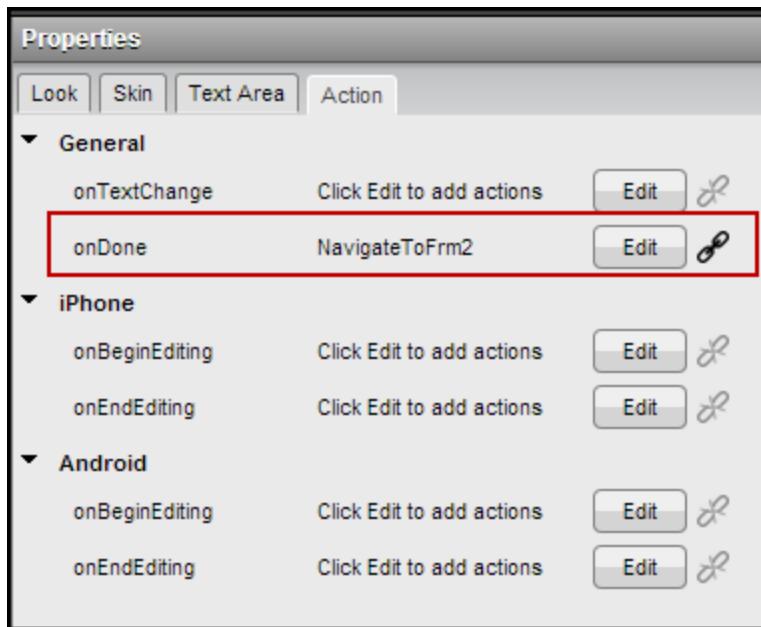
## 20.4 Assigning an Action

To assign an action, follow these steps:

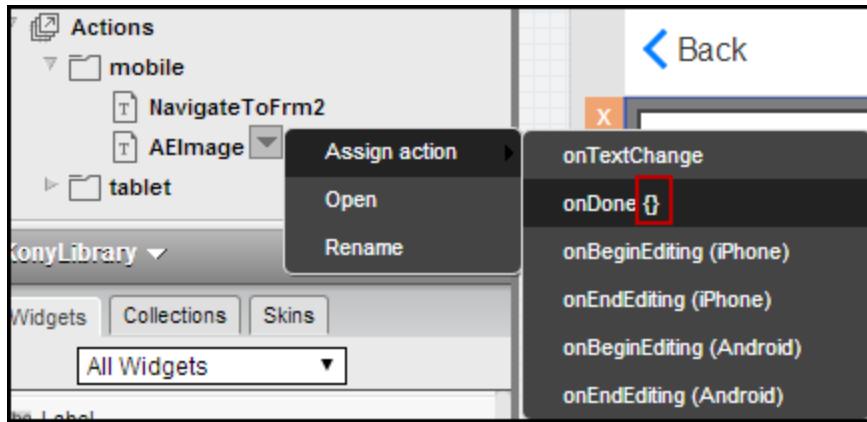


1. Select a widget from project tab or App Canvas.
2. Right-click an action from the **Action** tab > **Mobile** (or **Tablet**.)
3. Click **Assign Action**.
4. Select an event

The action is assigned to an event. You can confirm if an action is assigned to an event by viewing the **Properties** > **Action** tab of a widget.



When an action is assigned to an event, braces appear next to the event.



## 21. Keyboard Shortcuts

Following the keyboard shortcuts that are available for both Windows and Mac platforms.

For the below shortcuts, in Windows use **ctrl** and in Mac use Command **⌘**.

| Shortcut key                                  | definition         |
|---|--------------------|
| Ctrl + Shft + S                               | Saves All          |
| Ctrl + S                                      | Save active form   |
| Alt + N                                       | Create new project |
| Ctrl + C                                      | Copy               |
| Ctrl + D                                      | Duplicate          |
| Ctrl + V                                      | Paste              |
| Alt + Left                                    | Move Left          |
| Alt + Right                                   | Move Right         |
| Alt + Up                                      | Move Up            |
| Alt + Down                                    | Move Down          |
| Delete (Applicable only for Windows platform) | Delete Widget      |
| Alt + F10                                     | Select Previous    |
| F10   | Select Next        |
| Alt + F11                                     | Select Parent      |
| F11   | Select First Child |

## 22. Flex Layout FAQ

1. Is there an easy way to convert VBox forms to Flex forms without having to redesign them?

We currently do not have an automatic conversion between VBox and Flex forms, but the 2.0 release allows VBox and Flex form usage within the same project so that conversions can be done as time permits by the developers.

2. Is Flex forms the recommended way to do forms now?

Flex forms provide many new great features to enhance the UI of the application and is the recommended approach. You can see some of these features in the new Visualizer tutorials (<http://www.kony.com/resources/videos>). We will most likely deprecate the VBox forms in the coming releases. We strongly encourage new applications and application enhancements to begin utilizing the 2.0 Visualizer Flex forms.

3. Can we create an app with both VBox and Flex forms? If so, what are pitfalls in doing so?

Anything that they cannot share (themes, headers, footers, etc.)

Applications can have both VBox and Flex forms. Headers and footers must have common layout types (VBox or Flex) to be utilized on a form of the same layout type. Headers and footers are supported in Flex forms, but are not the recommended approach for creating these types of UI elements. With flex, containers can be **pinned** anywhere using the positioning properties.

4. Is there any overhead in using Flex forms in terms of app size (app is X% bigger, and so on?)

As part of the 2.0 release, the layout engine that Kony uses was redesigned for Flex layout. One of the major considerations in this besides flexibility in the UI was performance. We have generally seen a large performance increase when implementing a UI in Flex containers for Flex layouts. One use case in an iOS and Android tablet application resulted in a 150% increase in performance others have ranged from 50 to 100%. In regards to application size, there were modifications to framework which will have a slight increase in overall application size, but it is minimal (Kbs).

5. Is there an automated way that syncing of forms from Visualizer to KonyOne?

We will be releasing solutions in the coming months to address this issue. In the near term, a new import functionality will be exposed to expedite the import process and allow you to only import specific forms, themes, etc. In the longer term, we are working to unify the experiences between Kony Studio and Visualizer so that they can share a common project.

6. Do pop-ups, headers or footers differ or do they work the same way regardless of form type?

Headers and footers are supported in flex forms, but are not the recommended approach for creating these types of UI elements. With Flex, containers can be **pinned** anywhere using the positioning properties.

7. How does the Actions stay in sync between Visualizer and Kony Studio? As an example, if a page has been created in Visualizer and then imported into Kony Studio, a set of actions added (JavaScript functions) in Kony Studio and then the page is exported and re-imported into Visualizer are the JavaScript functions also imported? If so, are they also exported back?

Currently, actions created in Visualizer and Studio are mutually exclusive. Developers have options to use the Visualizer created events, but they are retained. In the forthcoming releases, we will be addressing the action synchronization.