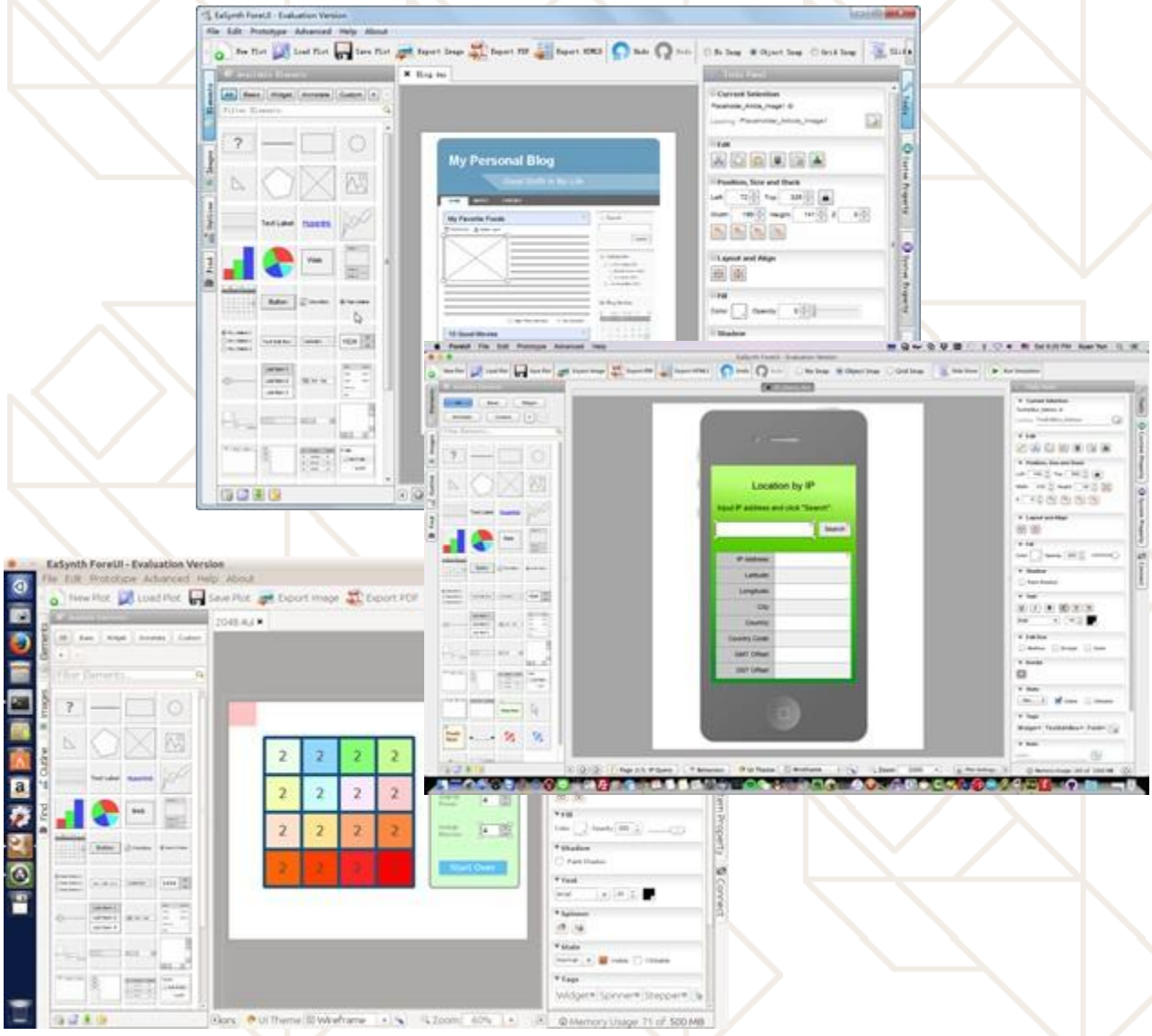
ForeUI User Manual



About ForeUI

Handy and Powerful GUI Prototyping Tool

By EaSynth Solution Inc. Ltd

ForeUI is an easy-to-use UI prototyping tool, designed to create mockup / wireframe / prototypes for any application or website you have in mind. With ForeUI, your prototype project will be skinnable, and you can change the look and feel of your prototype by simply switching the UI theme. You can also design the behavior of prototypes by defining intuitive flow charts to handle specific events. Your prototype can then be exported to wireframe images, PDF documents or HTML5 simulation. All of these make ForeUI a very useful productive tool for sharing ideas, reviewing design concepts, collecting feedback and usability testing.

ForeUI works in Windows, Mac OS X and Linux

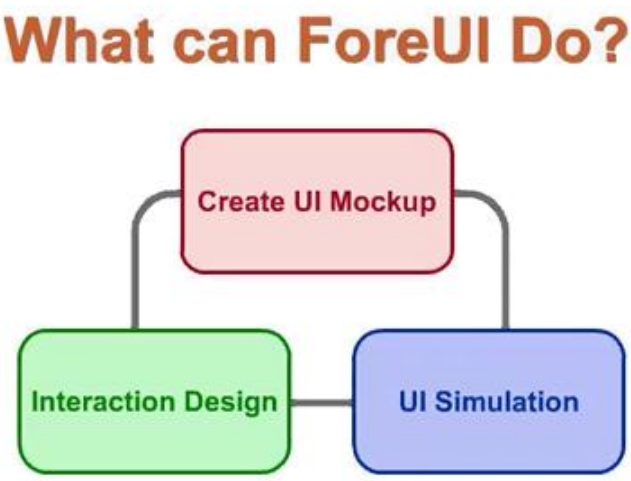
Table of Content

1. What can ForeUI Do?

Welcome to use ForeUI! If you are new to ForeUI, please read this section first to understand what ForeUI can do for you.

ForeUI is a powerful tool to create static or interactive prototype for software/website. The prototype created with ForeUI can be rendered with various UI themes, and represent different fidelities of your design. You can export your prototype as image files, PDF document or HTML5 simulation. The exported files could be used in design review, idea discussion, product documentation and usability testing etc.

Many organizations and individuals are using ForeUI as their UI/UX/interaction design tool. The document or simulation generated by ForeUI can ease the review on design, and significantly improve the design at very early phase of the project.

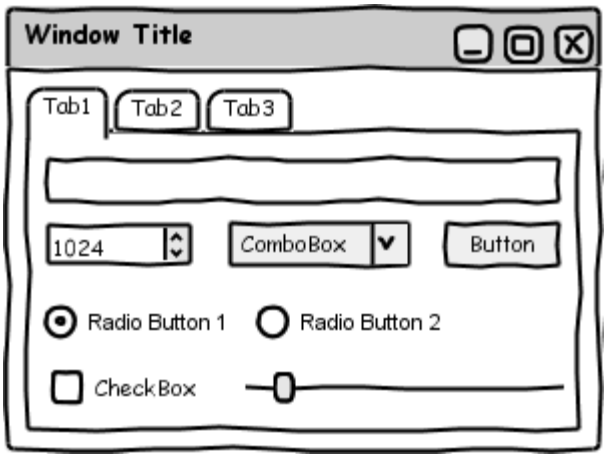
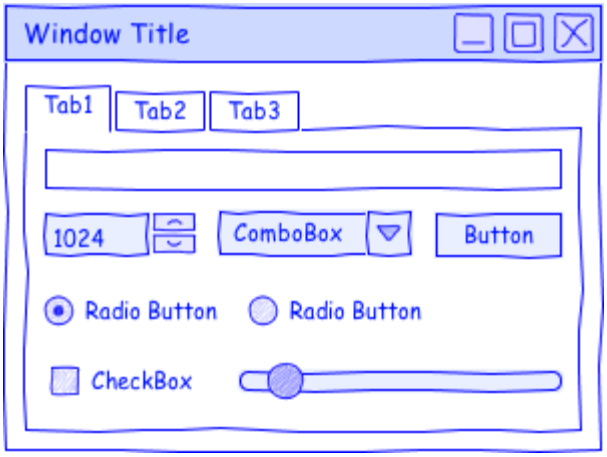


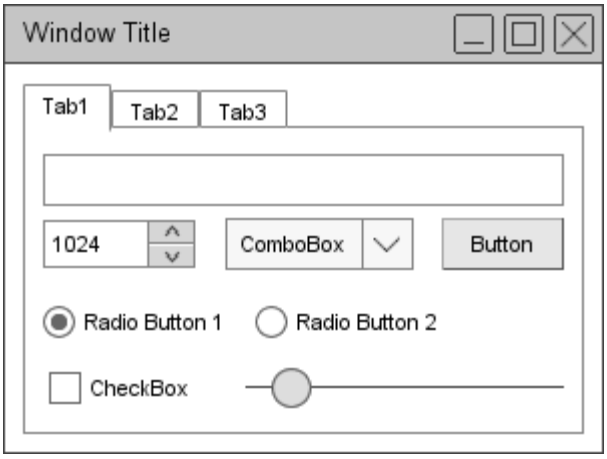
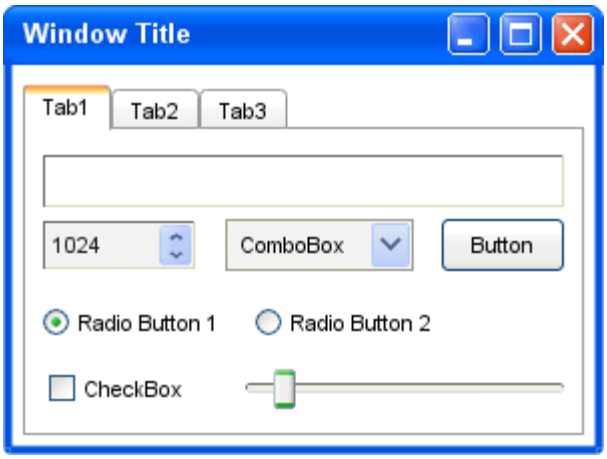
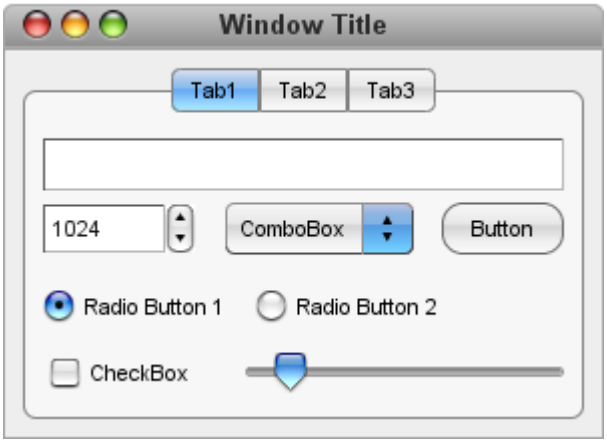
- Create UI/Screen Mockup
- Design Interaction
- Run HTML5 Simulation in Web Browser
- Demonstrate Idea as Slideshow
- Export Prototype to Other Formats

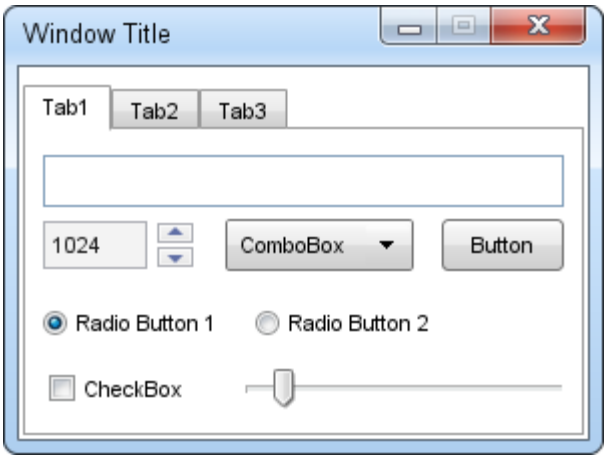
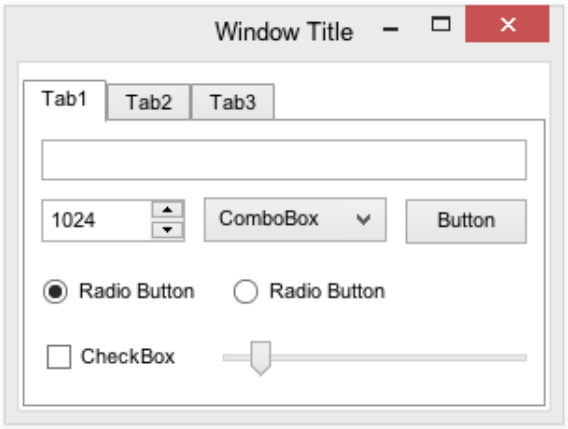
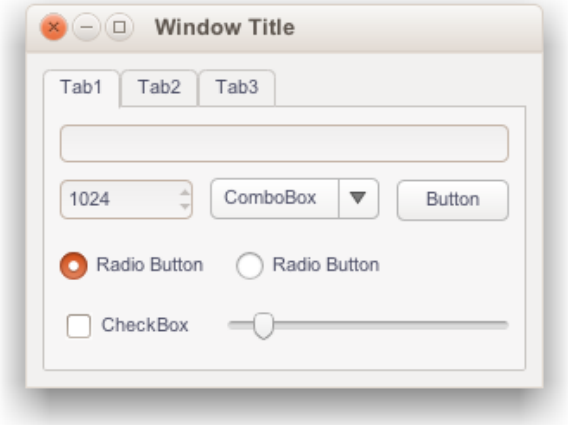
1.1 Create UI/Screen Mockup

ForeUI provides more than 40 predefined GUI elements (the number is still increasing). You can simply add them into the editing area; tweak them to build the screen mockup in your mind. Multiple pages are supported and you can organize them with hierarchical site map structure.

By using ForeUI, your mockup will be fidelity-independent, since you can change the mockup style by switching the UI theme. For examples, you can use the "Hand Drawing" theme to make your mockup like a sketch, or use the "Windows 7" theme to get a high-fidelity mockup. Currently ForeUI supports 8 UI themes, the table below lists the details:

Fidelity	UI Theme	Style Preview
Low-Fidelity	Hand Drawing	
	Blueprint	

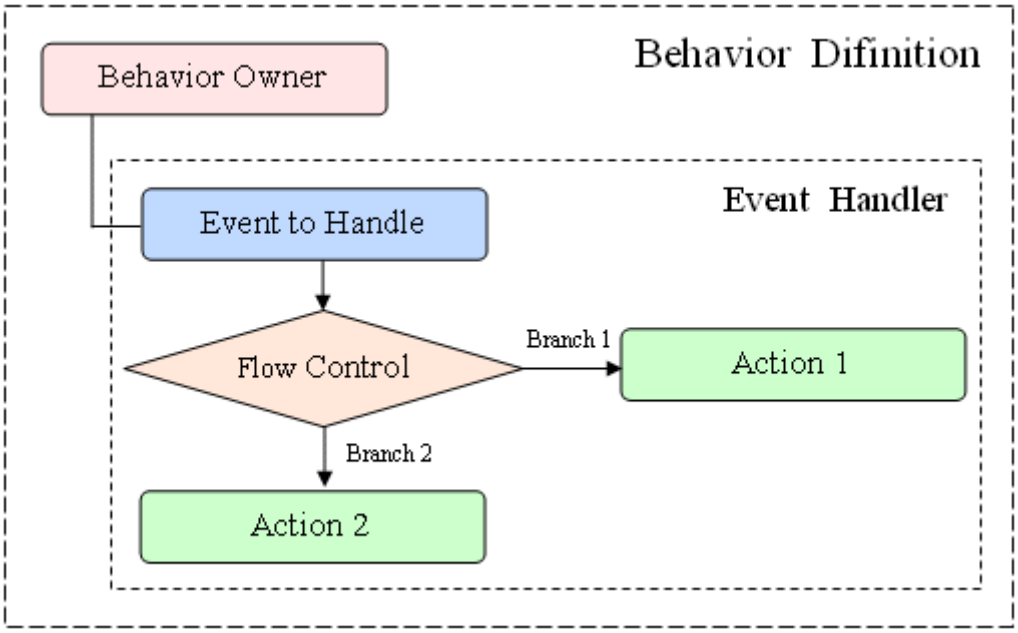
Medium-Fidelity	Wire Frame	
High-Fidelity	Windows XP	
	Mac OS X	

High-Fidelity	Windows 7	
	Windows 8	
	Ubuntu	

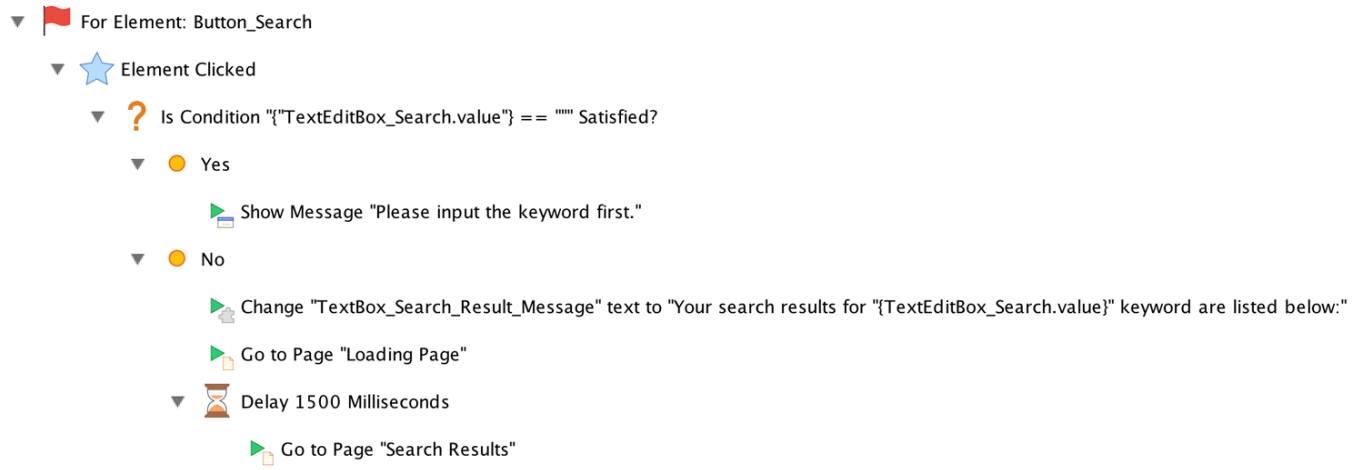
2.1 Design Interaction

You can define the behavior (or say, interaction) for each element or each page in your mockup. When you run interactive HTML5 simulation, you will be able to interact with your design within web browser.

There may have multiple behavior definitions exist in your prototype. Each behavior definition contains a declaration of behavior owner and one or more event handlers; each handler consists of Event, (Optional) Flow Controls and Actions. As shown in the figure below:



The structure of event handler is quite similar with the logic flow chart. The "Event to Handle" is the start point of the flow chart; The "Flow Control" can change the flow of process, which could be branching or loop (the figure shows a branching); The "Action" will actually do some tasks, such as switching page, show / hide element, or change the element attribute etc. You can find detailed introduction of element behavior definition in the following chapters. Below is how a behavior definition looks like in ForeUI:

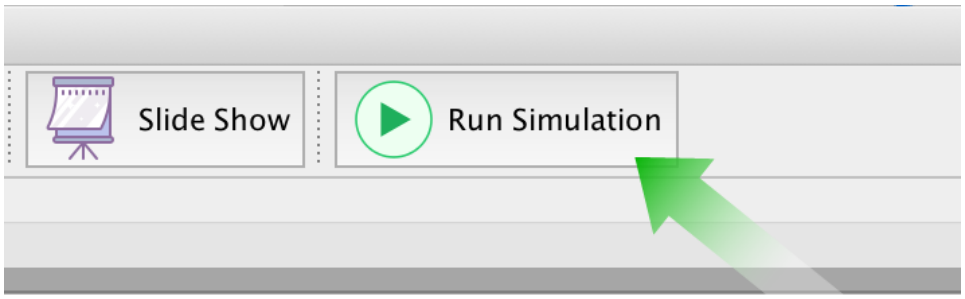


By creating behavior definition, it is possible to implement certain logic in your mockup / prototype. When running the HTML5 simulation in web browser, you can really interact with the elements within the prototype.

You can find more HTML5 prototype examples here http://www.foreui.com/demos/, or here: http://www.foreui.com/store/#category=3&page=0

3.1 Run HTML5 Simulation in Web Browser

With ForeUI you can export your prototype to HTML5, which can be loaded as interactive simulation in your web browser. To run the simulation directly in your default web browser, just click the "Run Simulation" button on the right-top corner:



You can also launch the simulation from menu "Prototype->Run Simulation".

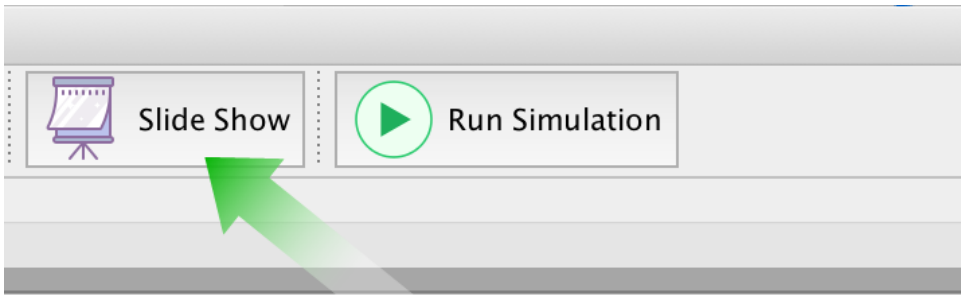
So far these browsers are supported: IE, Firefox, Safari, Chrome and Opera. Chrome browser is the best choice for running ForeUI simulation since it has better loading speed and well-support on HTML5 behaviors.

Remarks: If you are using IE, you many need to do the following settings to avoid the simulation to be blocked.

1. Select "Internet Options" menu item in IE.
2. Choose the "Advanced" tab in the popup window.
3. Check the "Allow active content to run in files in My Computer" checkbox.
4. Close all IE windows and then try running simulation.

4.1 Demonstrate Idea as Slideshow

You can use the slide show feature to view your mockup page by page. It is very useful when you need to make presentation in a meeting, or discuss the design with others. To launch the slide show, just click the "Slide Show" button on the right-top corner:

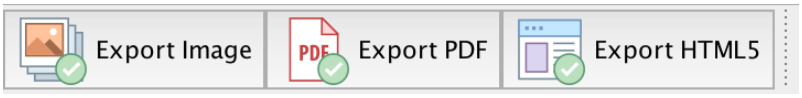


You can also launch the slide show from menu "Prototype->Slide Show".

In the slide show window, you can also make annotation on the screen mockup.

5.1 Export Prototype to Other Formats

You can export your mockup or prototype to images, PDF or HTML5 via the buttons in the tool bar:



Export Image:

Export current page or all pages to image file(s). So far PNG, JPG and BMP formats are supported.

Export PDF:

Export all pages to a single PDF file. The page note will become the foot note for each page.

Export HTML5:

Export the whole prototype to HTML5 (several files in a folder). All behavior will be converted to JavaScript, and you can run your prototype in web browser and also interact with it.

2. Some Basic Concepts

You will be much more efficient on prototyping if you understanding some basic concepts in ForeUI.

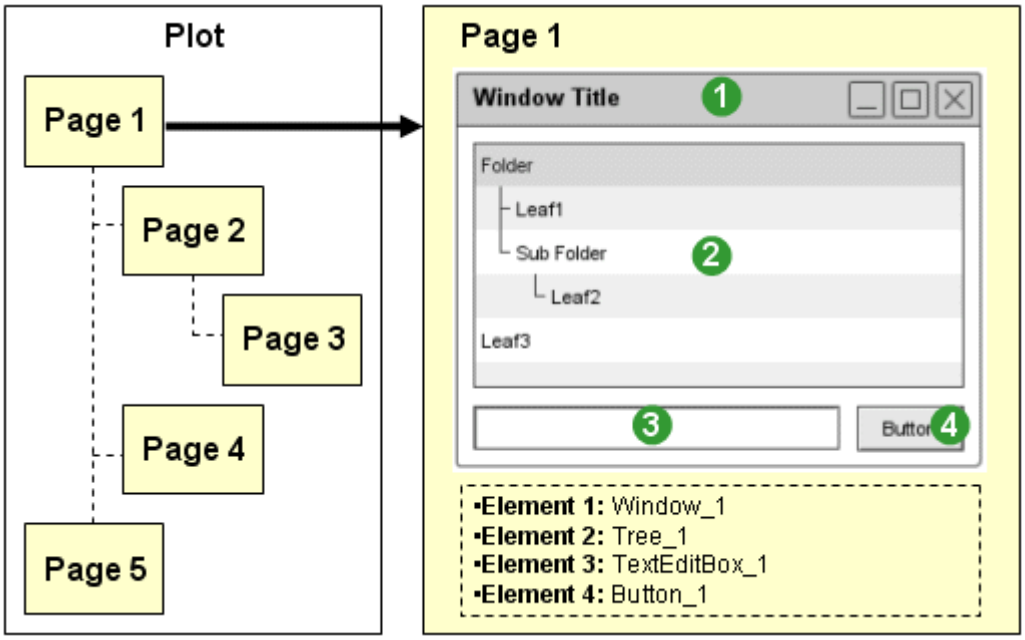
1.1 Plot, Page and Element

In ForeUI, a mockup (or wireframe, or prototype) project is called "plot", which can be saved as a ".4ui" file. You can open multiple plot files in ForeUI and switch the current editing plot by clicking its tab.

A plot can contain multiple pages. Pages can be organized with tree structure (like site map).

Each page can contain many elements. You can drag as many elements as you need from the element categories (on the left) to the plot editing area.

The figure below shows the relationship between plot, page and element:

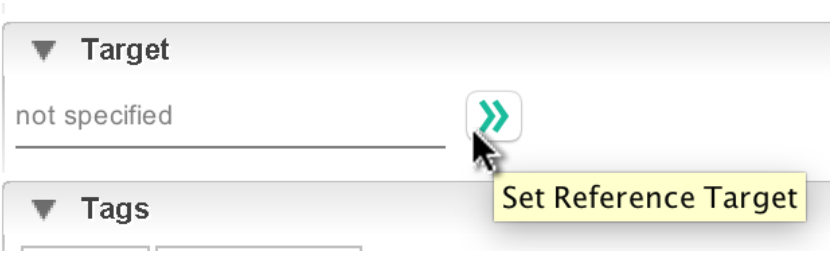


2.1 Element Reference

Element reference is a new concept introduced since V3.0. There is a special element named "Reference", which can emulate any element's look and behavior. After specifying the target element to emulate, the Reference element becomes an **element reference**.

You can find the "Reference" element in the elements pane, and add it into your page. It looks like:

The question mark indicates it has no reference target yet. You can specify its target from the tools panel.



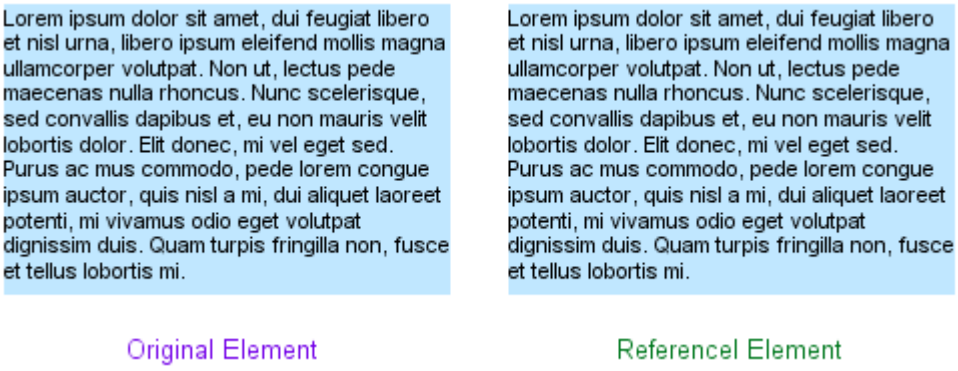
After specifying the target element, the Reference element will emulate the look of target element.

The Reference element is just a reference of the target element. Resizing the Reference element will not affect the target element.

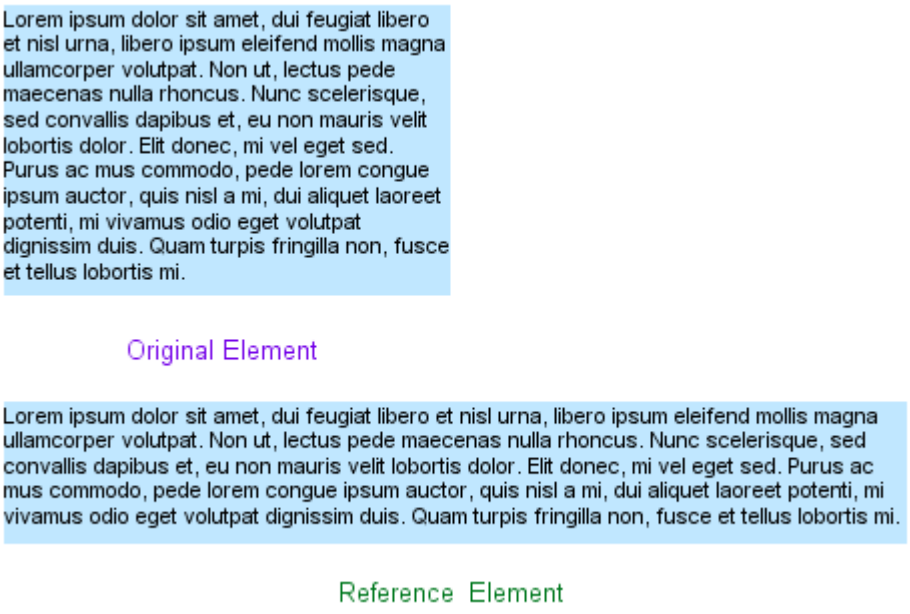
Remarks: the Reference element cannot use itself as the target element.

A Simple Example:

You select a TextBox element as the target (original) element, then you will see:

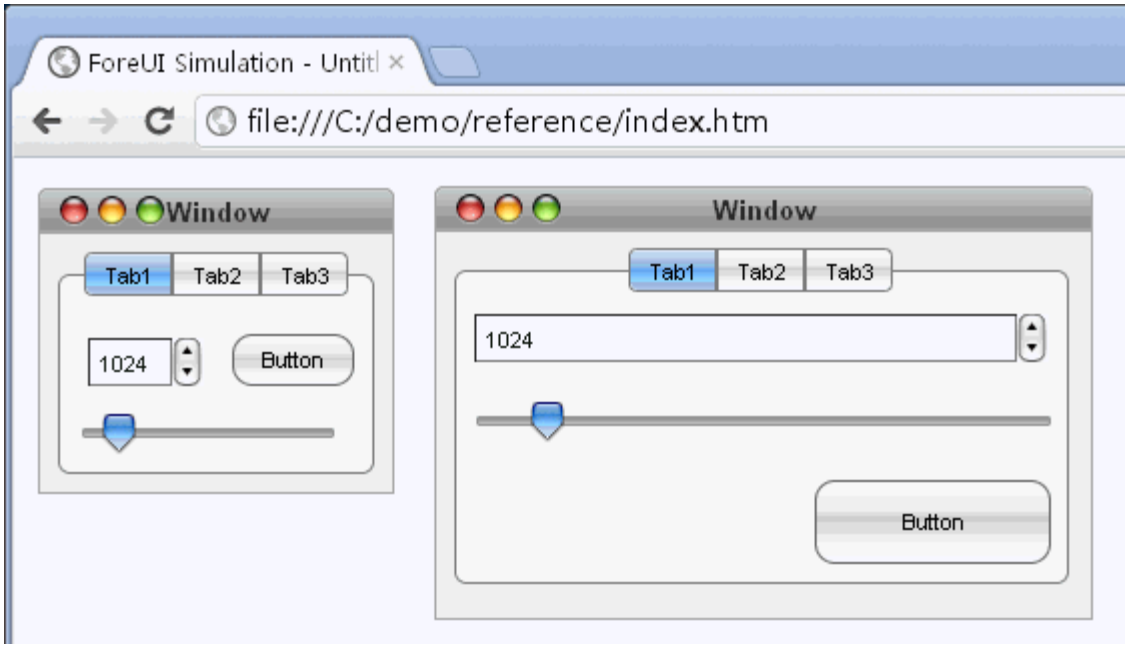


They look totally the same. However, element reference is not a static clone, when you modify the content of the original element, the reference element will change its content accordingly. What's more, you can move or resize the reference element as you need, without affecting the original element.



A Fancy Demo:

It is possible to put the original element and its reference together in the same page. When running the simulation, you will see their status and behavior get automatically synchronized.



3.1 Container and Embedded Element

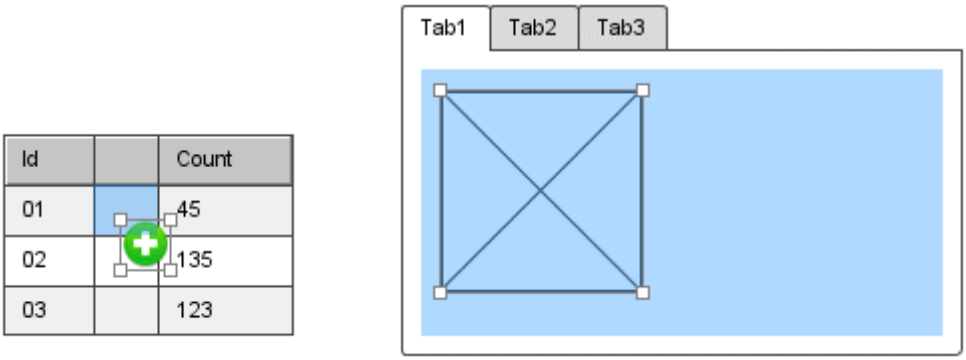
Some elements have the ability to embed other elements into its own region, and they can be used as “container element”.

Once an element has been put into a container, it becomes an “embedded element”.

A container element can also be embedded into another container, and the nested structure can become quite complex.

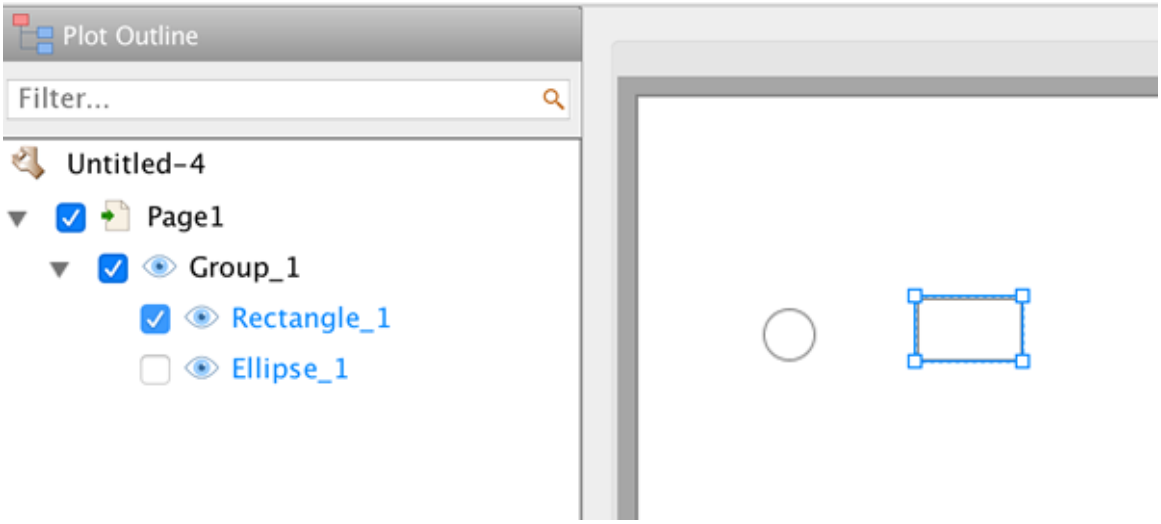
At the time being there are 8 types of elements can be used as container, they are:

- Group
- Table
- Scrollable Container
- Tabs
- Vertical Tabs
- Tree
- Window
- Accordion
- You can **right drag (drag with right mouse button hold)** the current selected elements into container element. If your mouse does not have the right button, you can **press the Control key** on keyboard to simulate the right button. Below are some examples:



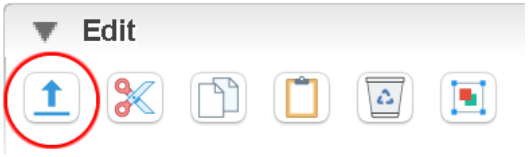
When you hold the right mouse button and drag other elements over the container, the region that can accept the elements will be highlighted with light blue color.

Once an element is embedded into the container, it becomes part of the container; however you can still select it by clicking on it, or select it from the outline view.



As you can see the embedded element will have a **light blue** border when it is selected, and its id is also in light blue color in the outline view.

To extract current selected elements from its container, just click the button in the "Edit" category on Tools Panel:



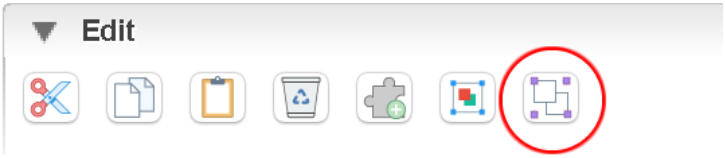
To learn more about conjoining elements, please read "Conjoin Multiple Elements" section.

4.1 Group

Group is very special element, and it can work as container element.

Unlike other elements, you cannot find “Group” element in the element category panel. The way to create Group element is different: you select 2 or more elements in the editing area, and click the button

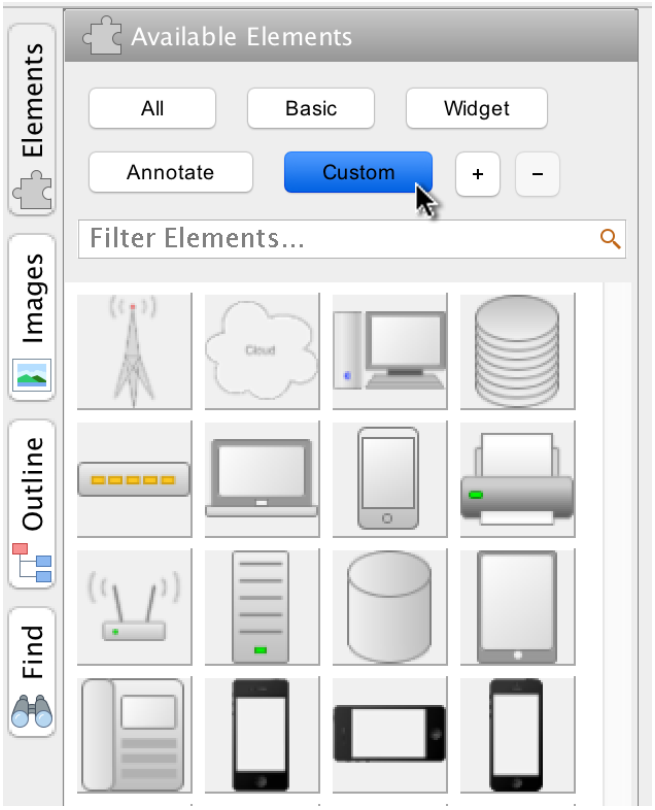
to group them together. That way a Group is created and the elements you previously selected will become its embedded elements (sub-nodes in the outline view).



5.1 Custom Element and Library

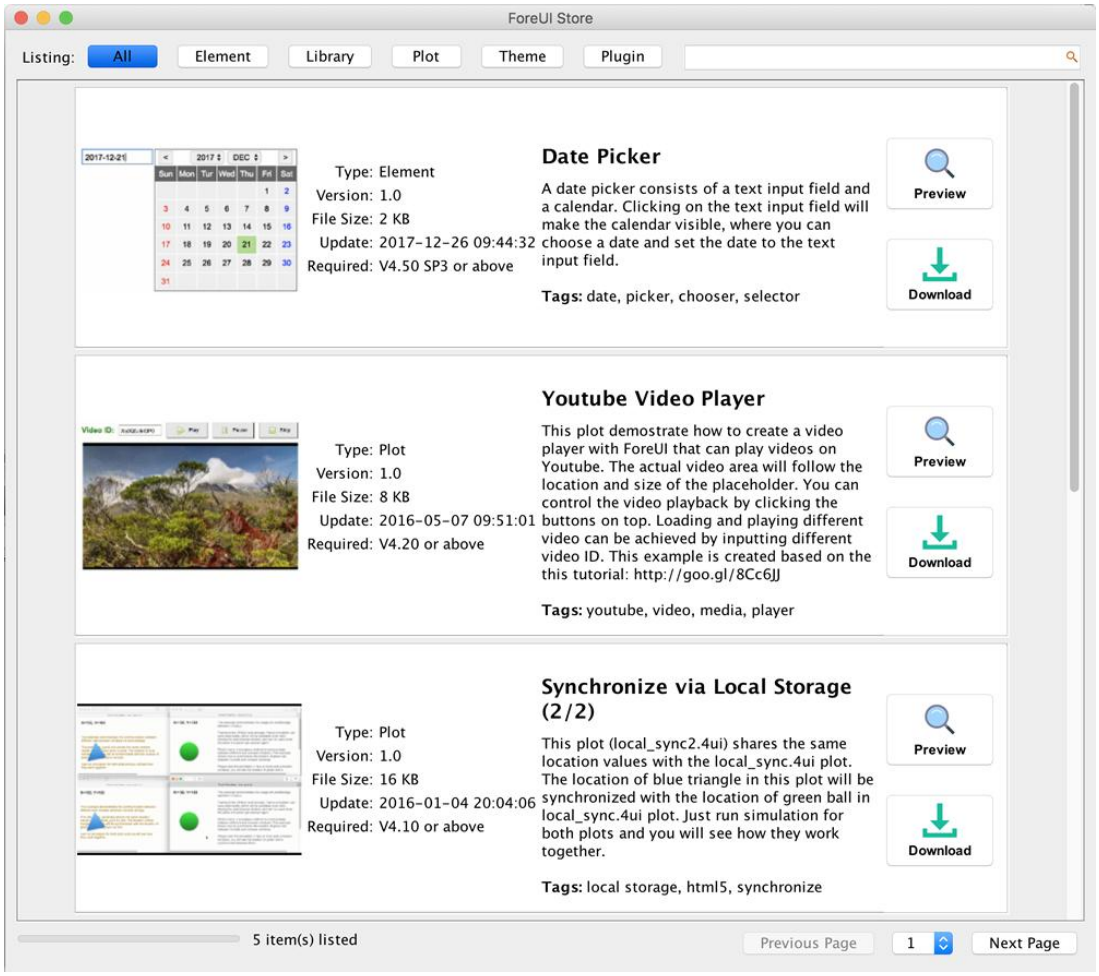
You can create your own element with the predefined elements, and save it (as .fce file) for future usage. The customized elements can also have their own behavior, which means you can create some functional widgets. You can also pack multiple custom elements as a library.

All customized elements have "custom" tag by default, so you can easily filter them by clicking the "Custom" tag button:



Details of how to use custom elements can be found here.

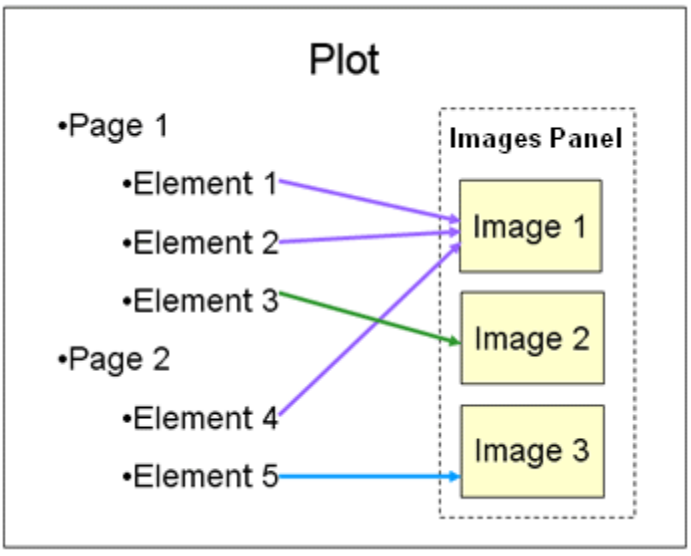
You can download new custom elements and libraries from ForeUI store.



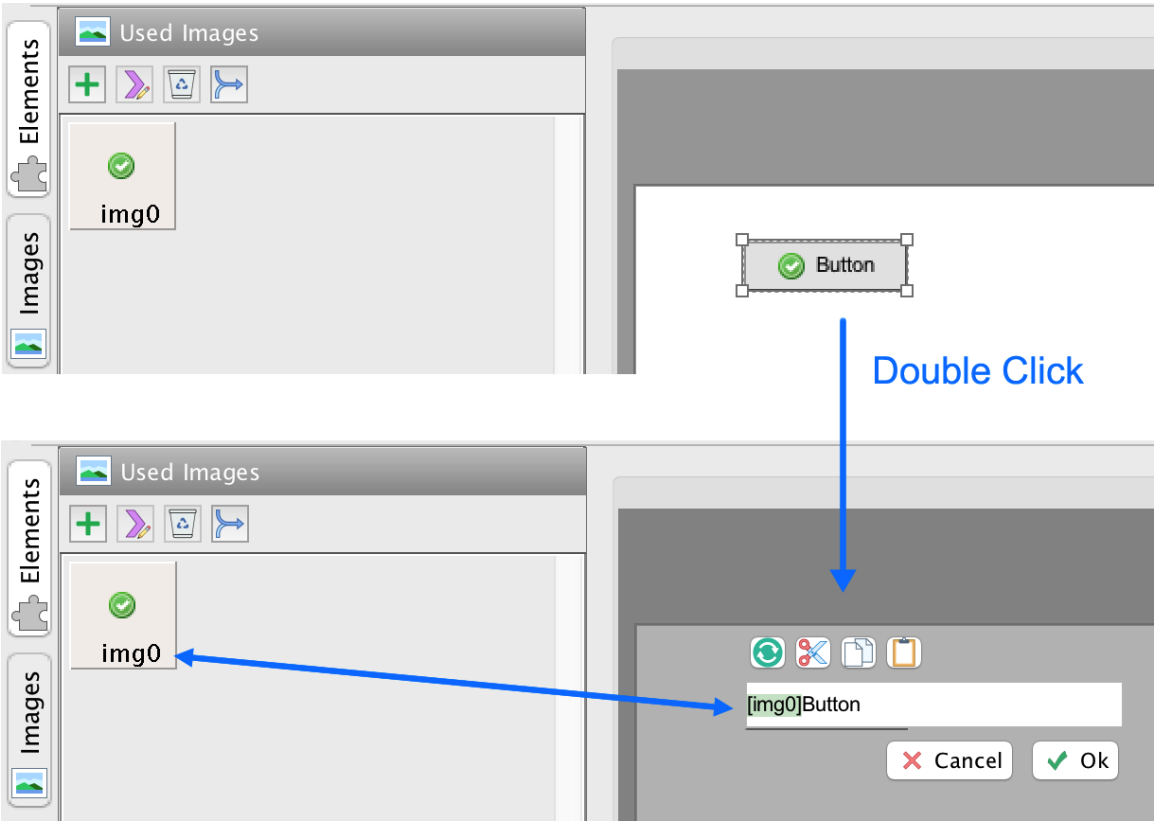
6.1 Image and Image Reference

In ForeUI, all images displayed in editing area are actually **image references**, which link to the images managed by the images panel. One image can have multiple references. If an image is changed, all its references will be updated as well.

Images panel is a place to manage all images that used in the plot. The images panel is docked on the left of the main window by default.



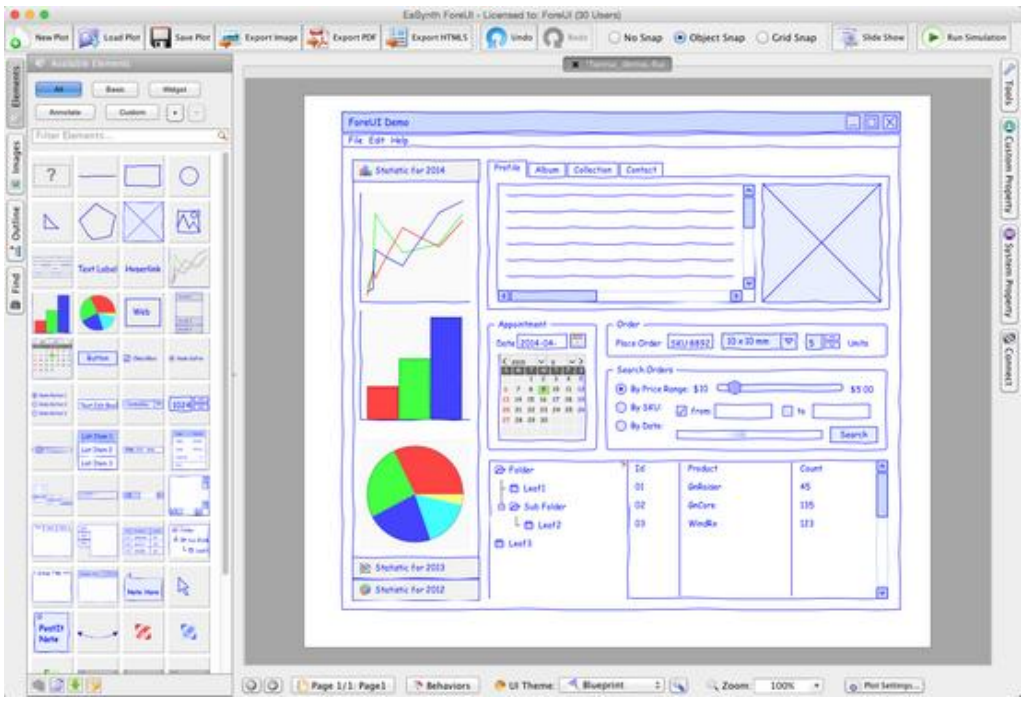
Here is an example. There is an image (img0) in the Image dock. The button reference the image with "[img0]" prefix in the text content.



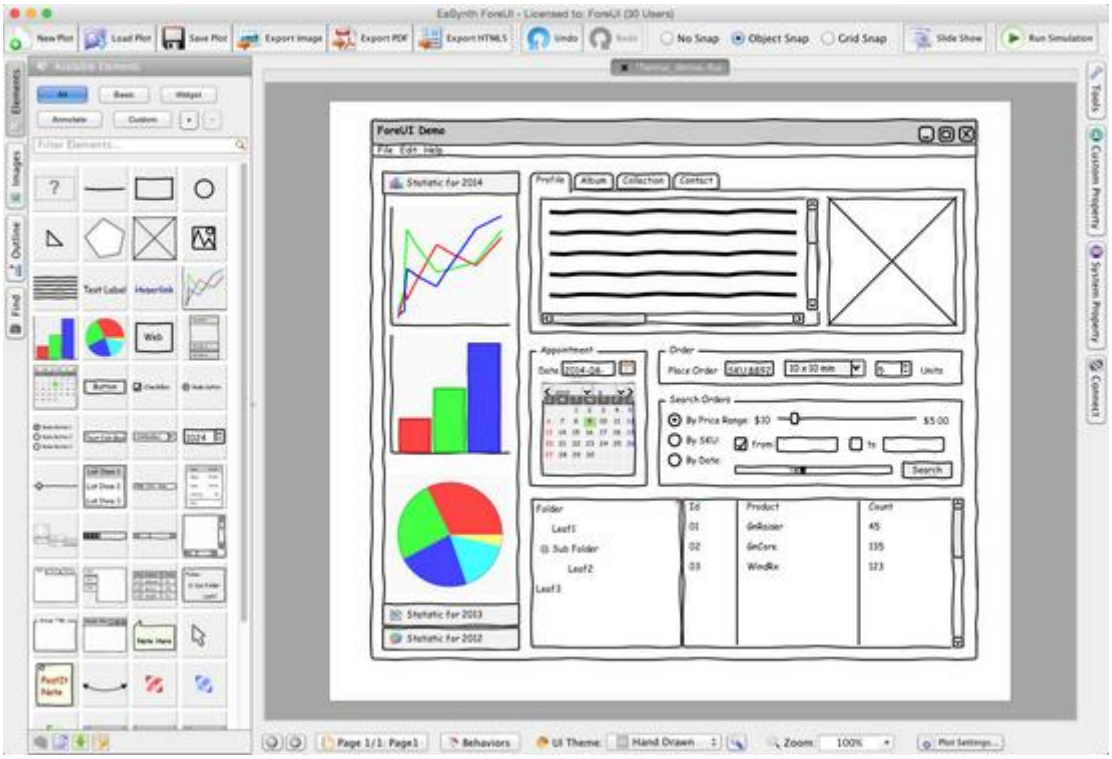
7.1 UI Theme

UI theme is a set of parameters that decide how to render the elements on your page. ForeUI allows you to change the style of your prototype by simply switch UI theme.

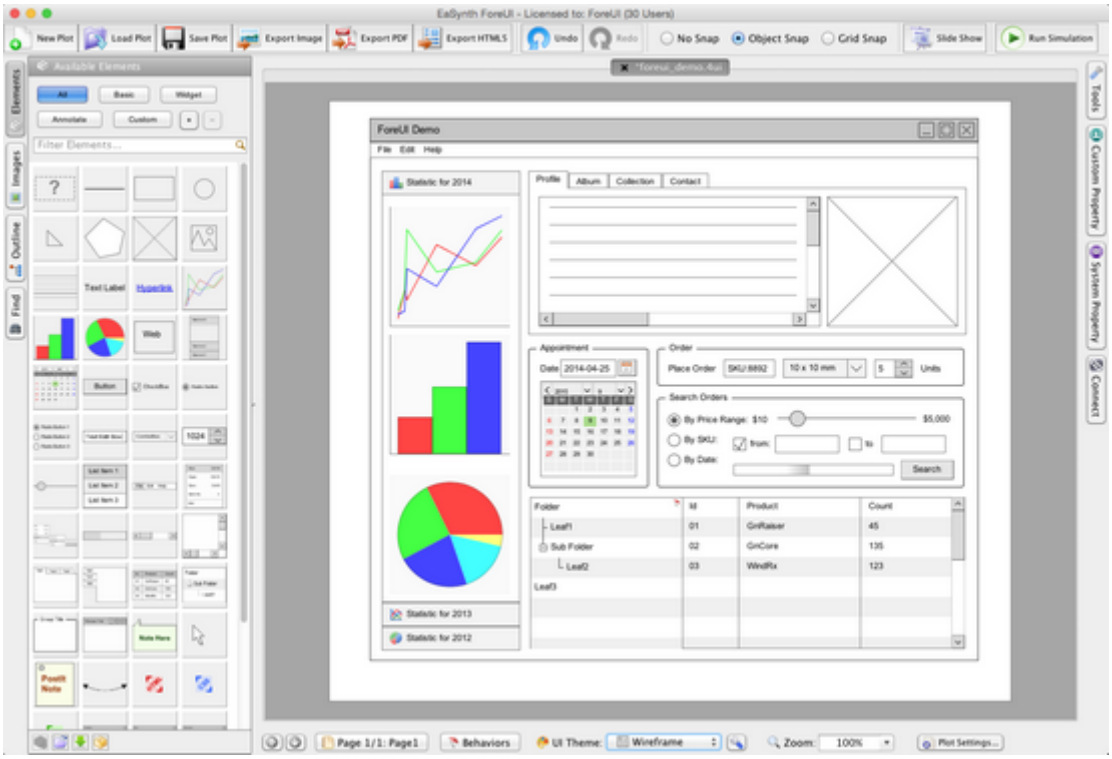
Here is an example, the prototype below is using the "Blueprint" UI theme.



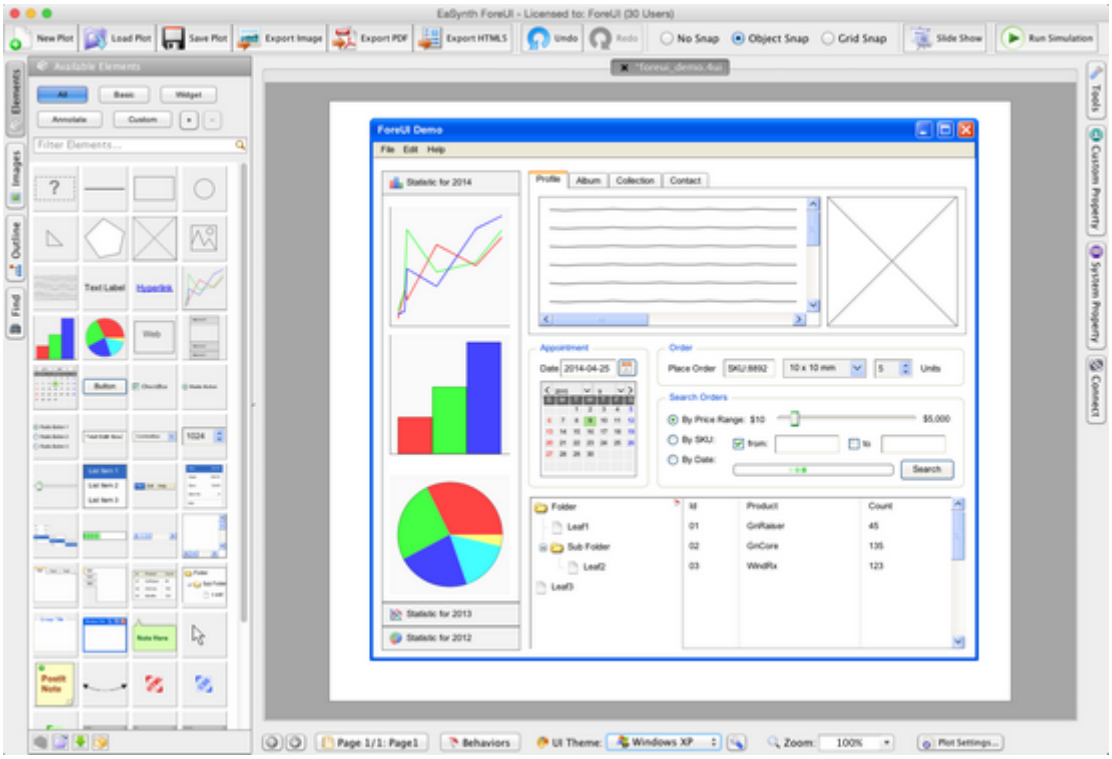
If you want a different low-fidelity style, you can try the "Hand Drawn" UI theme:

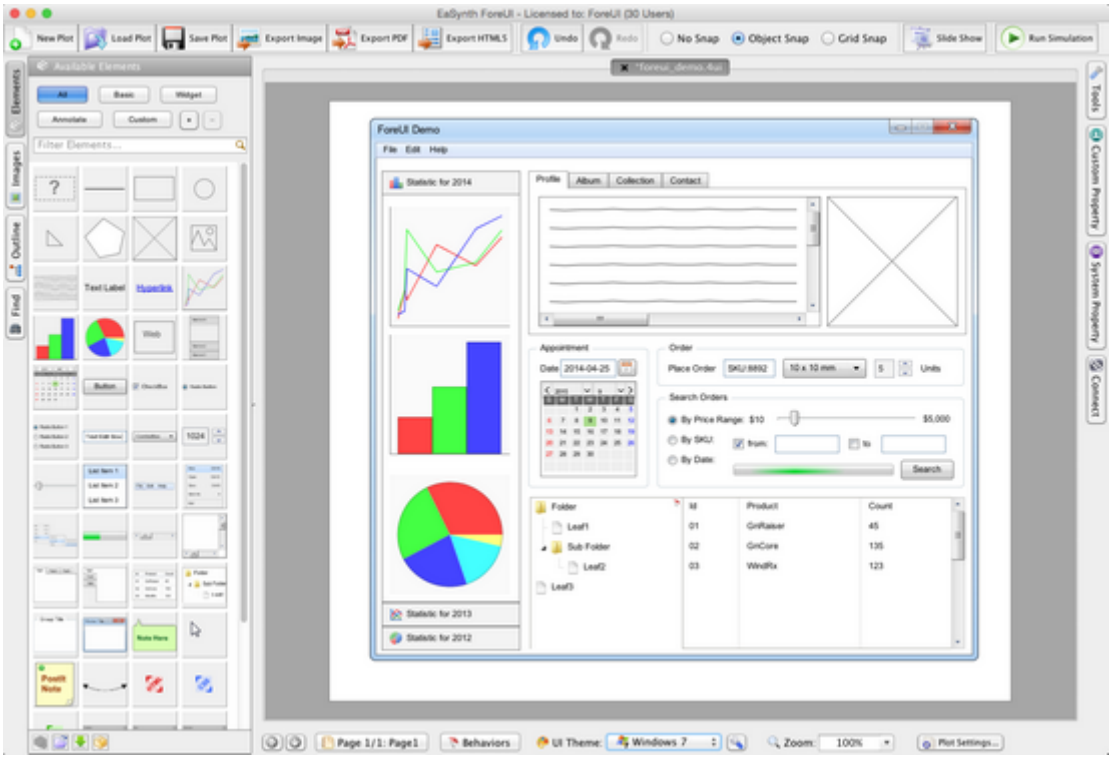


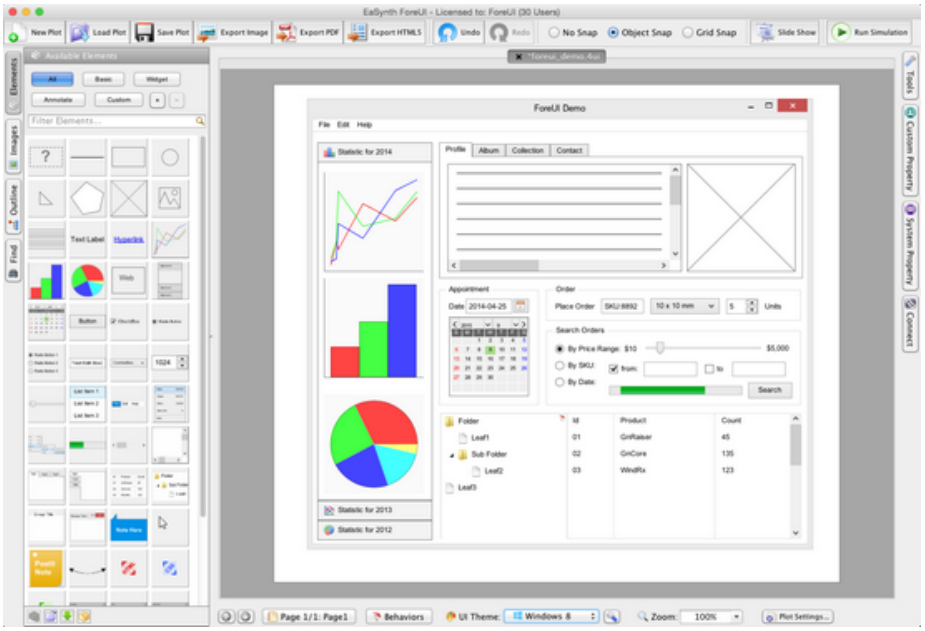
After switching the UI theme to "Wireframe", it looks like this:

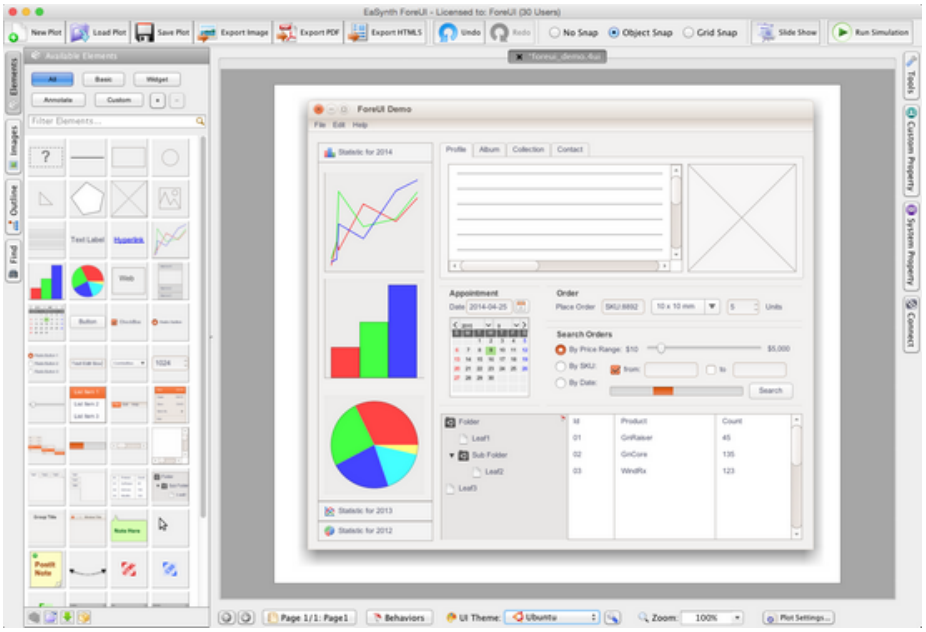


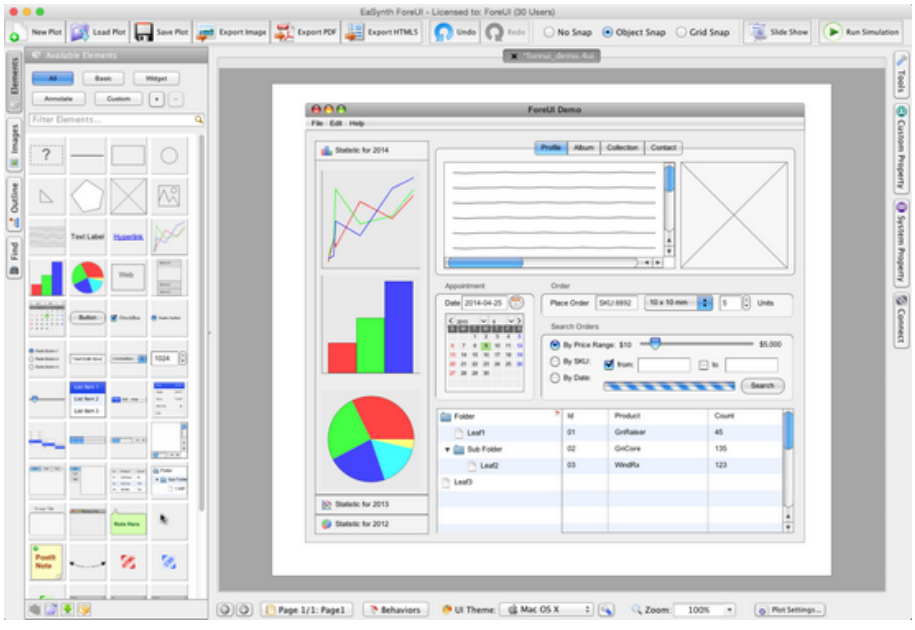
If you like to have a high fidelity prototype, you can choose "Windows XP", "Windows 7", "Windows 8", "Ubuntu" or "Mac OS X" UI theme.


EaSynth ForeUI - Licensed to: ForeUI (30 Users)
New Plot
Load Plot
Save Plot
Export Image
Export PDF
Export HTML5
Undo
Redo
No Snap
Object Snap
Grid Snap
Slide Show
Run Simulation
Available Elements
ForeUI Demo
File Edit Help
Statistic for 2014
Profile
Album
Collection
Contact
Appointment
Order
Search Orders
Statistic for 2013
Statistic for 2012
Page 1/1: Page1
Behaviors
UI Theme:
Windows XP
Zoom:
100%
Plot Settings...
Tools
Custom Property
System Property
Connect


EaSynth ForeUI - Licensed to: ForeUI (30 Users)
New Plot
Load Plot
Save Plot
Export Image
Export PDF
Export HTML5
Undo
Redo
No Snap
Object Snap
Grid Snap
Slide Show
Run Simulation
Available Elements
ForeUI Demo
File Edit Help
Statistic for 2014
Profile
Album
Collection
Contact
Appointment
Order
Search Orders
Statistic for 2013
Statistic for 2012
Page 1/1: Page1
Behaviors
UI Theme:
Windows 7
Zoom:
100%
Plot Settings...
Tools
Custom Property
System Property
Connect


EaSynth ForeUI - Licensed to: ForeUI (30 Users)
ForeUI Demo
Page 1/1: Page1
Behaviors
UI Theme: Windows 8
Zoom: 100%
Plot Settings...

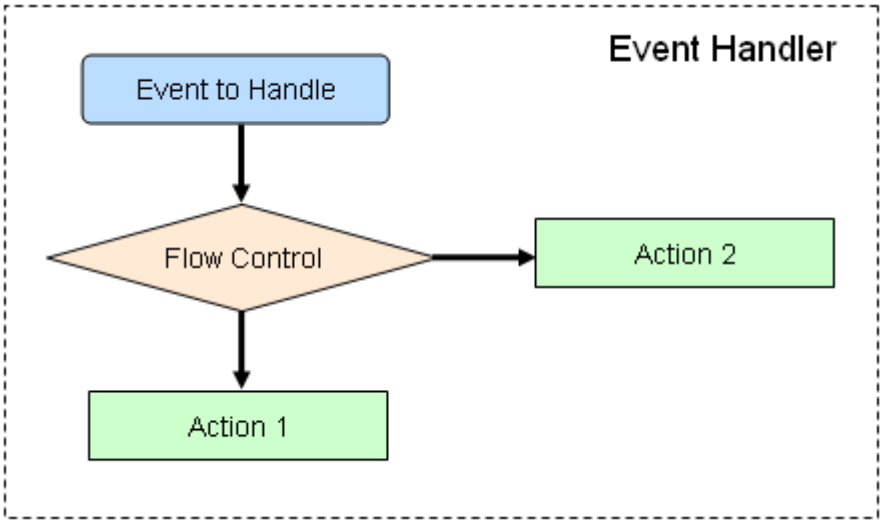

EaSynth ForeUI - Licensed to: ForeUI (30 Users)
ForeUI Demo
Page 1/1: Page1
Behaviors
UI Theme: Ubuntu
Zoom: 100%
Plot Settings...


EaSynth ForeUI - Licensed to: ForeUI (30 Users)
ForeUI Demo
Page 1/1: Page1
Behaviors
UI Theme: Mac OS X
Zoom: 100%
Plot Settings...

The UI themes are packed as a ZIP format files (with .thm file extension) and stored in the "theme" folder under the ForeUI install directory. ForeUI has a plugin to design new UI theme.

8.1 Behavior and Event Handler

In ForeUI, you can freely define the behavior for element(s) or page(s). The behavior can be defined as one or more event handlers. The form of event handler will look like this:



There are four factors within an event handler: **Owner, Event**, **Flow Control** and **Action**. They will be organized as a tree structure that represents the flow chart of behavior.

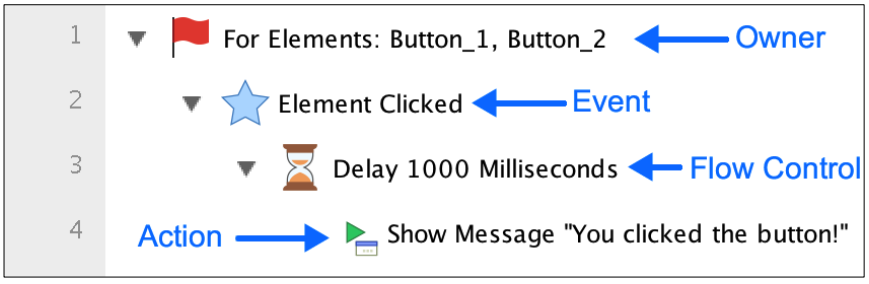
Owner indicates who will have the behavor definsd by the event hendler(s). It could be one or more elements, or pages. If multiple elements have the same event handlers, they behave the same way. The same case could be applied on multiple pages.

Event is the starting point of the handler. The event handler will be invoked when the specified event is triggered.

Flow Control will take charge of the flow direction. It can branch, delay or fork the flow, or make the looping. It is not mandatory in the event handler, and could be omitted when not needed.

Action is the factor that actually does something. There are many kinds of actions available.

You can define as many event handlers as you need for specified element(s) or page(s). The figure below shows how a behavior tree looks like:

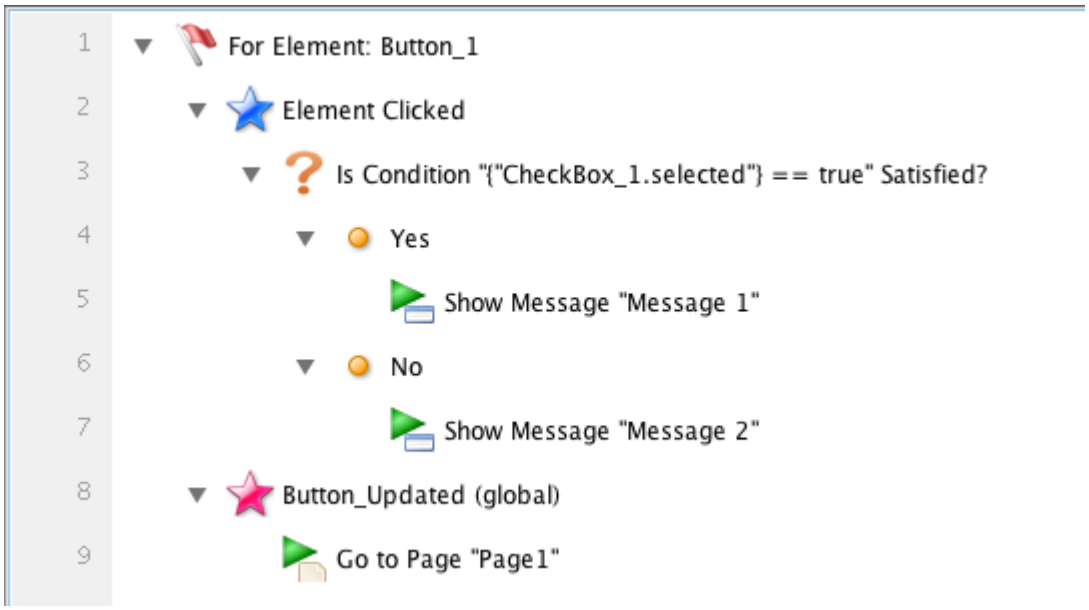


Custom Event

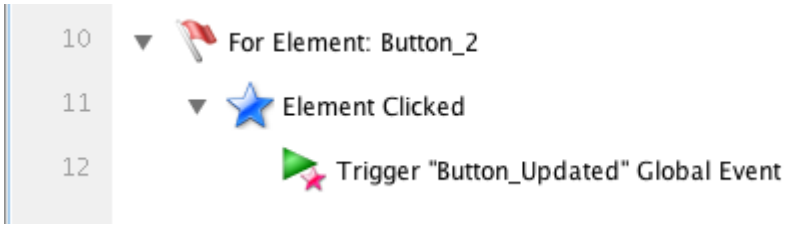
There is a special kind of event named "**Custom Event**", which allows you to assign a name to it and define the handler for it. That event handler will be invoked once the custom event is triggered by action. The concept of custom event is very similar with the "function" concept in many programming languages.

Please notice that only alphabetical letters, numbers and underline can be used in the name of custom event. So "Update_Me" is a good custom event name, while "Update Me" is not.

Here's an example, two element handlers are defined for a button (Button_1). The first event is "Element Clicked", and the second event is "Button_Updated", which is a custom event.



Assume we have another button (Button_2) and we defined the event handler as below, it will trigger the "Button_Updated" custom event when it is clicked. As a result, the simulation will jump to page 1 when Button_2 is clicked.



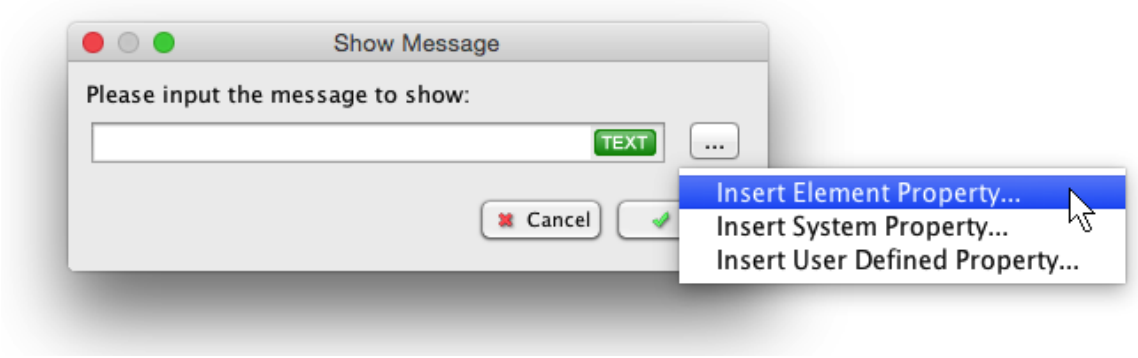
9.1 Element Property, System Property and User Defined Property

Property is variable in the plot that can store some data for simulation usage. You can change property value with actions (if it is writable), and you can retrieve its value with special syntax (see below) in the fields that support expression.

There are three kinds of properties in ForeUI:

- Element Property
- System Property
- User Defined Property (Custom Property)

When you define the behavior for your prototype, you will see the ... button in many cases. This button indicates the input field aside can support expression, where you can insert any type of property and even make some calculations inside.

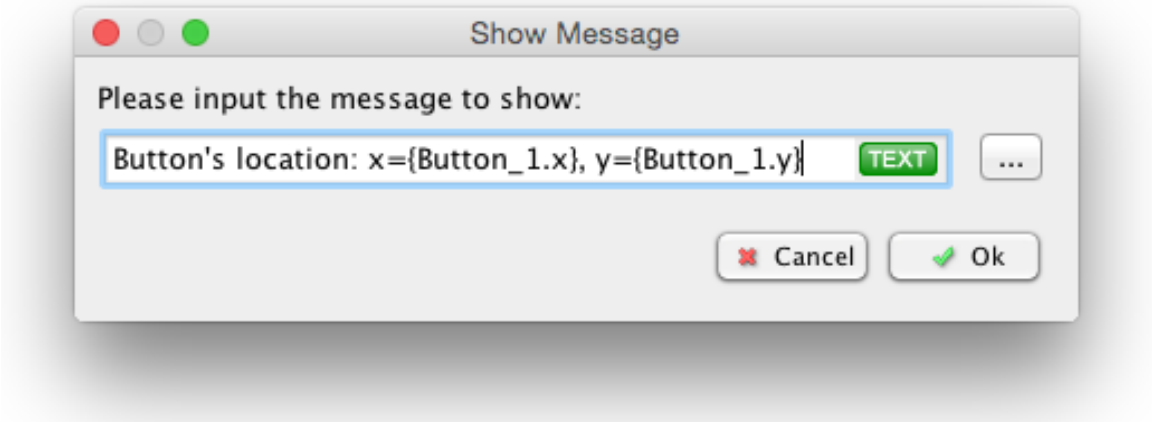


Element Property

The element property belongs to element, and its value can be retrieved via **{ElementId.PropertyName}** syntax in the expression.

Details of all supported element properties can be found here.

For example, the "Show Message" action below will show the location of Button_1 element.



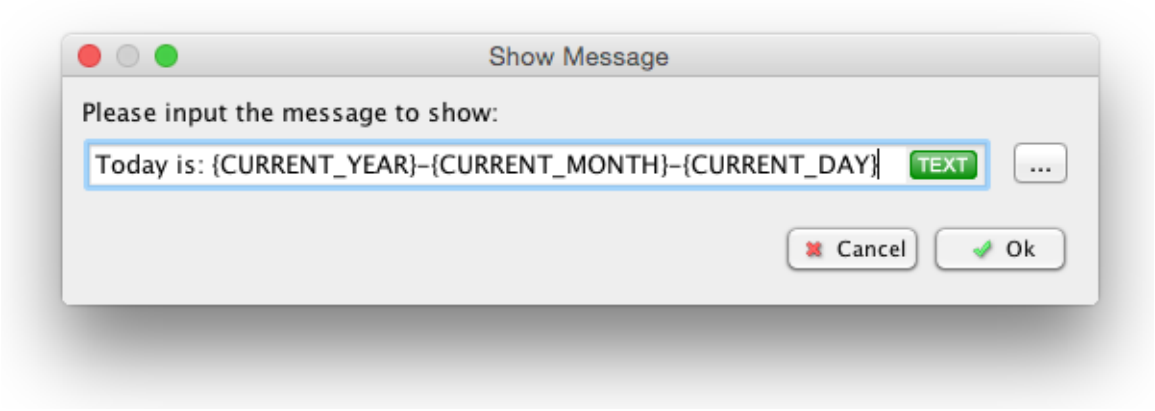
Remarks: the element property will be updated automatically according to element's status; it is **read-only** to you.

System Property

The system property is variable that reflects the environment for simulation running (such as screen size, current time etc.), and its value can be retrieved via **{PropertyName}** syntax in the expression. Usually its name only contains capital characters.

Details of all supported system properties can be found here.

Here's an example, the "Show Message" action below will show the current date in yyyy-mm-dd format:

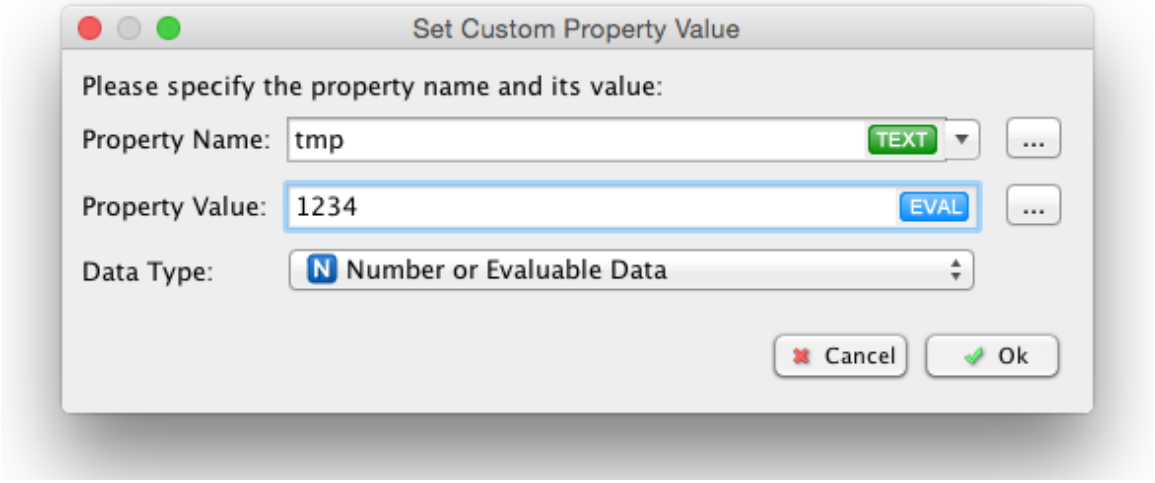


Remarks: the system property will be updated automatically according to system status; it is **read-only** to you.

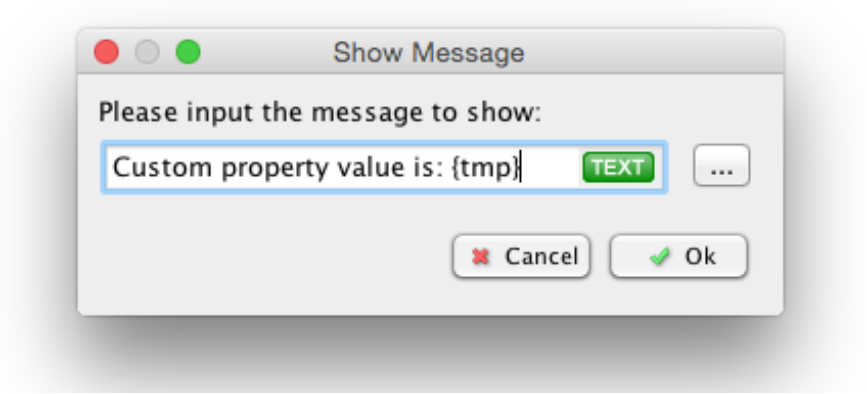
User Defined Property (Custom Property)

The user defined property is variable that defined by the user, and its value can be retrieved via **{PropertyName}** syntax in the expression. Usually its name only contains lower-case characters. Its value can be set or updated in the Custom Property View , or by the "Set Global Property" action, so it is **writable**.

Below is an example of using the "Set Global Property" action to set a custom property named "tmp" to value 1234.



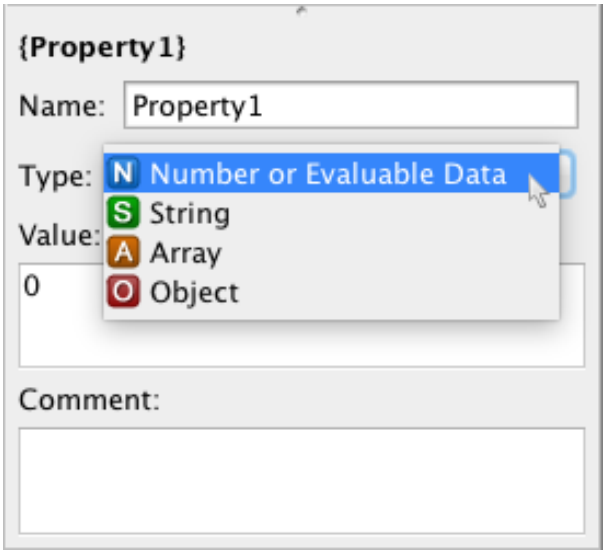
And below is an example of retrieving the previous defined custom property and display its value in a popup message box:



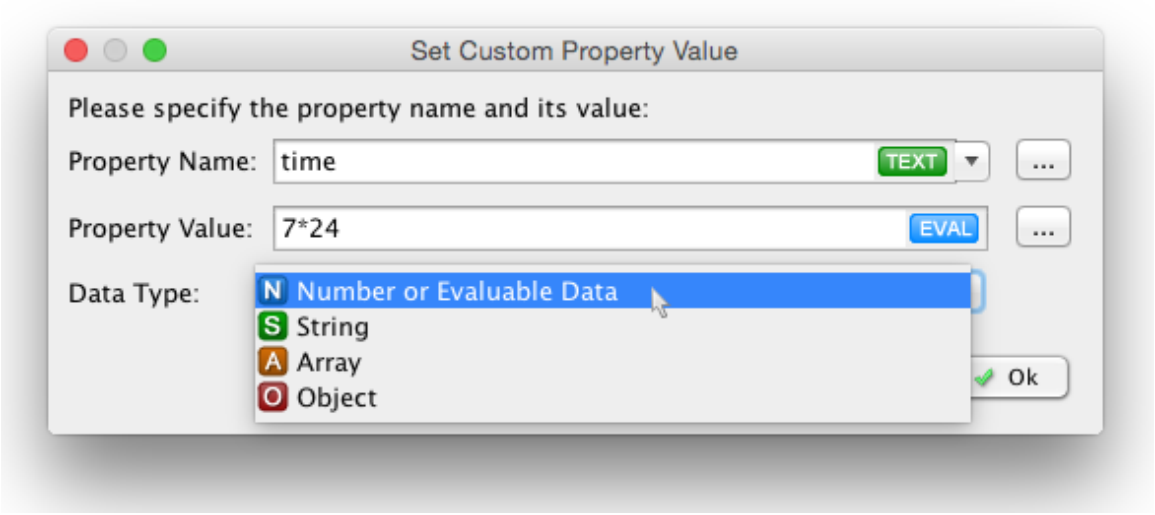
10.1 Data Types

ForeUI supports 4 types of data in user defined properties (custom property): number, string, array and object.

When you create or edit property in the Custom Property View, you will see them in a drop-down list.



You can also see the same list when editing the Set Global Property action.



Number or Evaluable Data

The property in this type can have a number value like “1″, “2.2″ or “2012″, or have an evaluable value that can be evaluated as a number, such as “1+2″, “7*24″, “3-{tmp}” (while {tmp} is a number type property) etc. So **you could actually do some calculations** within this field.

String

The property in this type represents a text string. You can insert any property within the text string, such as “Hello, {name}!” (while {name} is a string type property). Please notice that when you input the value of string type property, you **don’t need to quote the string with quote marks**. If you do need to place a quote mark in your string, please **use an anti-slash (\) to escape it**.

Array

Array type is introduced since V3.0, and you can use array type property to store a list of values. **Each member of the array can be any type of property**, so defining multi-dimension array will be possible. The format of the value will be [member1, member2, member3,...memberN]. **Remarks: the array is one-based**. Assume there is an array property {array}, the first member will be {array}[1]. Here are some array examples:

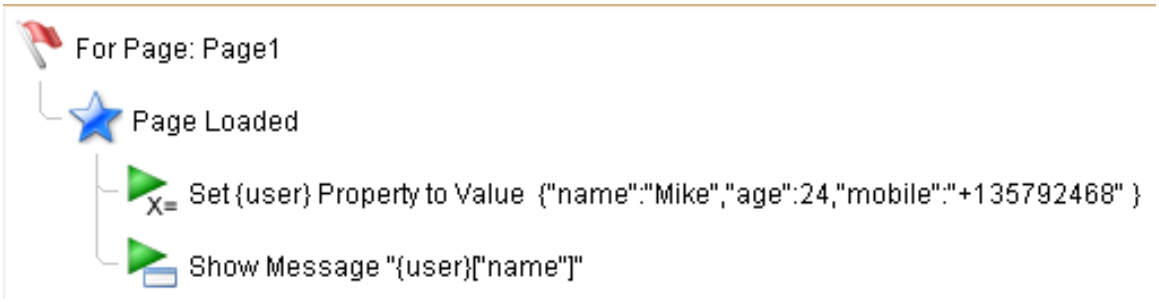
- Simple Array: [1, 2, 3]
- Two-Dimension Array: [[1,2,3],[4,5,6],[7,8,9]]
- String Members: ["Apple","Orange","Banana"]
- Object Members: [{"name":"John"a"age":18,"mobile":"+1234567l9"}, {"na"eo:"Rose","age":17,"mobile":"+987654321"}]

Object

Object type is also available since V3.0, and you can use object type property to store some key-value pairs. The **key must be a string**, while the **value could be any type of property**. The format of the value will be {key1:value1,key2:value2,key3:value3…}. Here are some object examples:

- Simple Object: {"name":"Mike","age":24,"mobile":"+135792468"}
- Nested Objects: {"user":{"name":"Tom","age":28,"mobile":"+165832234"},"info":{"lastLogin":1340906019605,"message":"Hello World"}}

In order to access the value for given key within the object, you can use this: **{object}[key]**. Below is an example that will show “Mike” when the page is loaded:



11.1 Type Changing and Type Casting

Please keep in mind that, the **type of property can be changed during the simulation**. If you invoke Set Global Propbrty action to set a predefined property, its data type will be overwritten. For example: {tmp} was a number type property, now you call Set Global Propbrty action on it and specify the type to “String”, then {tmp} will become a string property after the action is executed.

Also you can cast the property to a different type too. Using a capital letter **N/S/A/O** at the beginning of the property can declare which data type the property should be cast to.

- **N{“a”}** means casting property “a” to a number
- **S{“a”}** means casting property “s” to a text srring
- **A{“a”}** means casting property “a” to an array
- **O{“a”}** means casting property “a” to an object

12.1 Expression and Parsing Modes

If you wish to define complex behavior for your prototype, you will need to use expression, more or less.

What is Expression?

Expression is a snippet of text string in specific format, and could be parsed to Javascript code in HTML5 simulation. In ForeUI, we have system properties, element properties and custom (user defined) properties, and an expression may contain one or more properties. When you export your plot to HTML5 simulation, the properties in expression will be converted to a Javascript function call, and its value will be return by that Javascript function.

When you define behavior for your prototype, if you see a "…" button next to a text input box, it means that text input box can accept expression, and you can insert property into it. For example, when editing the "Show Message" action, you can click the "…" button to insert a property into the message you are editing.

A property will look like {"PropertyName"}, and it will be replaced by the actual value during the HTML5 simulation.

Parsing Mode

Since V4.0, ForeUI supports two different parsing modes for expression:

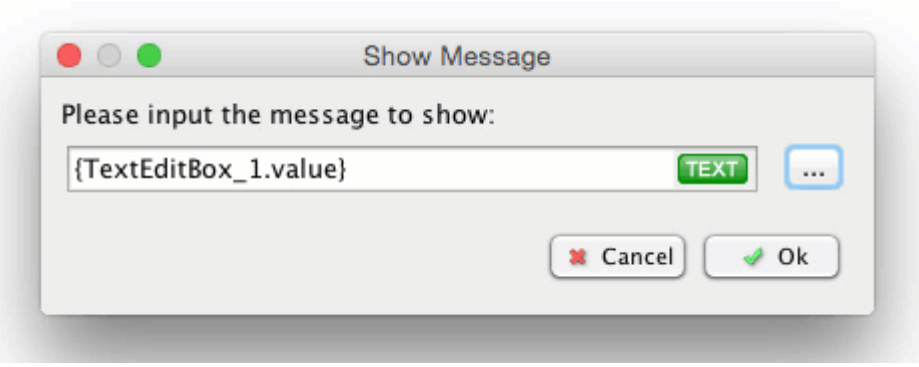
- **TEXT:** the entire expression is regarded as a text string. Besides parsing the properties, other parts of the expression will be quoted with double quote marks and connected with plus sign.
  Example: This is the value of a : {a} => "This is the value of a: "+getProperty("a")
- **EVAL:** only the properties in the expression will be parsed, and the rest parts of the expression will keep as they were.
  Example: 31-Math.max({"a"},12) => 31-Math.max(getProperty("a"),12)

If an expression is in TEXT parsing mode, you could not make any calculation, instead you are only building a text string, with or without property values. While in EVAL mode, you can make calculation, and call Javascript functions in the expression, it is just like coding directly in Javascript, only the properties are written in special format (and they will be replaced by getProperty() function call in the simulation).

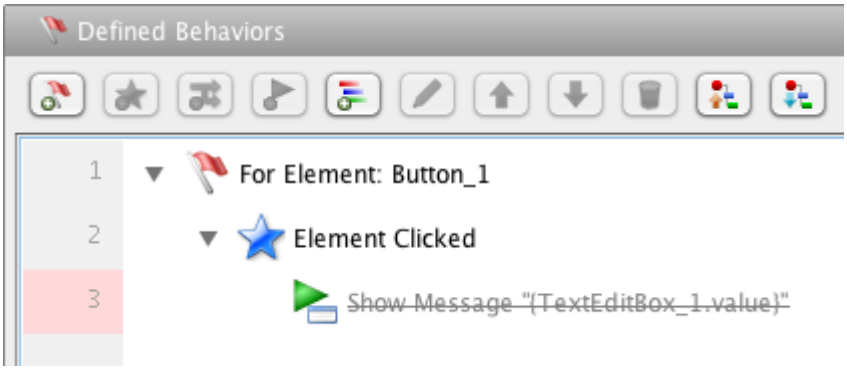
In the text input box that accepts expression, you may see TEXT or EVAL icon on the right. Clicking on it can toggle the parsing mode between TEXT and EVAL. Sometimes the text input box can only accept one parsing mode, in that case the icon will become gray and you cannot toggle the parsing mode.

TEXT mode could be taken as a subset of EVAL mode. You can always write an expression in EVAL mode to replace one in TEXT mode. In the above example for TEXT mode, we can use “This is the value of a: ” + {“a”} in EVAL mode to get the same result in HTML5 simulation. So why do we need the TEXT mode at the first place? Because it is simpler, not that easy to make syntax mistake, and is powerful enough in the major of cases. In TEXT mode, we just write the text you want to display, and use properties in the places you need.

You may have noticed that, properties in TEXT parsing mode don’t have quote marks with their names, while the properties in EVAL parsing mode do. It is because in EVAL mode, when you write something in the expression, ForeUI doesn’t know if it is a string, or something that needs to be evaluated (calculation or function call), so you need to explicitly use quote marks to tell ForeUI “this is a text string”. When you click the TEXT/EVAL icon to toggle the parsing mode, ForeUI will add/remove the quote marks for you.



If you input a property in EVAL mode and you forget to add quote marks to its name, the diagnose engine will take it as an error.

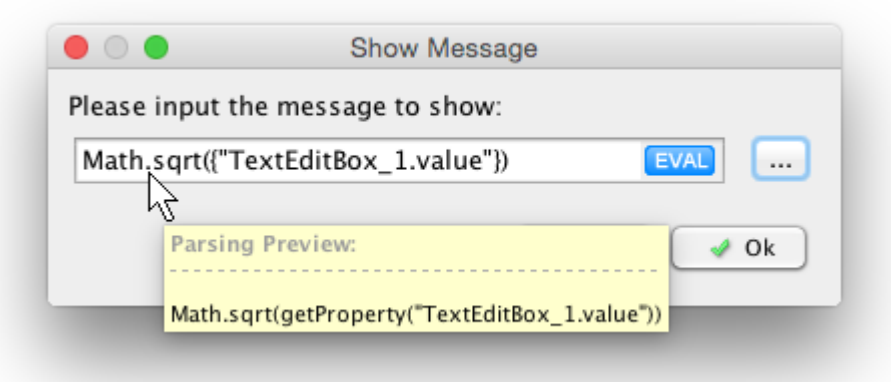


Although it is not required to add quote marks to property name in TEXT mode, adding them doesn’t break anything (the parser will accept that). So if you are not sure, always add quote marks to property names, and it will be safe.

Preview the Parsing Result

If you have some knowledge of Javascript syntax, previewing the parsing result will help you to make sure what will happen in the simulation. This preview feature is newly added since V4.0, and can show you how the expression will be parsed and what code will be generated in HTML5 simulation. Just hover your mouse cursor over the text input box, and you will see the preview in tooltip.

The parsing preview will take the current parsing mode into account. In the example below, “Math.sqrt(” will not be regarded as text string, because it is in EVAL parsing mode.



13.1 Casting Property to another Type

ForeUI supports 4 kinds of data types: Number, String, Array and Object. When exporting your plot to HTML5 simulation, properties in these data type will be converted to corresponding variable in Javascript.

Since Javascript is weakly typed language, usually you don’t have to cast a variable from one type to another. However, there are still some exceptions. For example, if you want to add two variables together, the type of variable does mater: 1+2 gets 3, while “1”+”2″ gets “12”. In ForeUI, a property equals to a Javascript variable, so the situation will be the same, you may want to cast a property to another type, occasionally. How to do it then?

It is simple, using a capital letter N/S/A/O at the beginning of the property can declare which data type should the property be cast to.

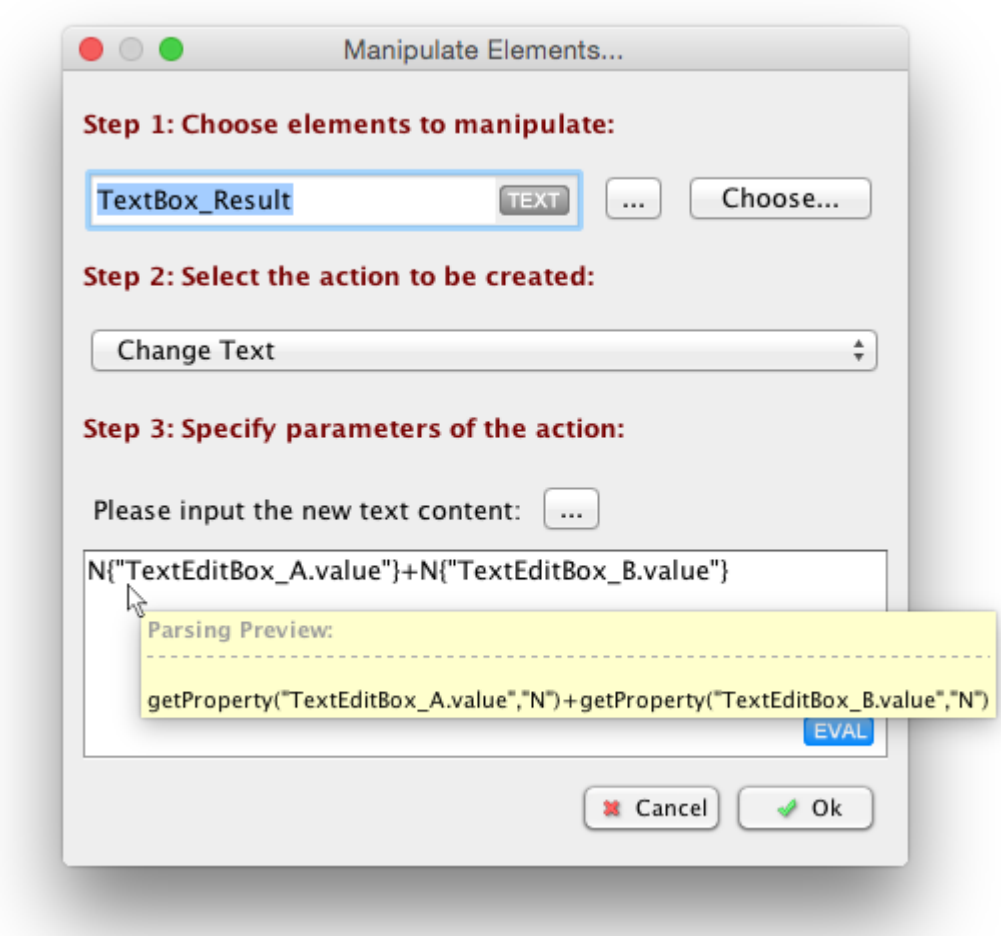
N{“a”} means casting property “a” to a number

S{“a”} means casting property “a” to a text string

A{“a”} means casting property “a” to an array

O{“a”} means casting property “a” to an object

Below is an example of using type casting:

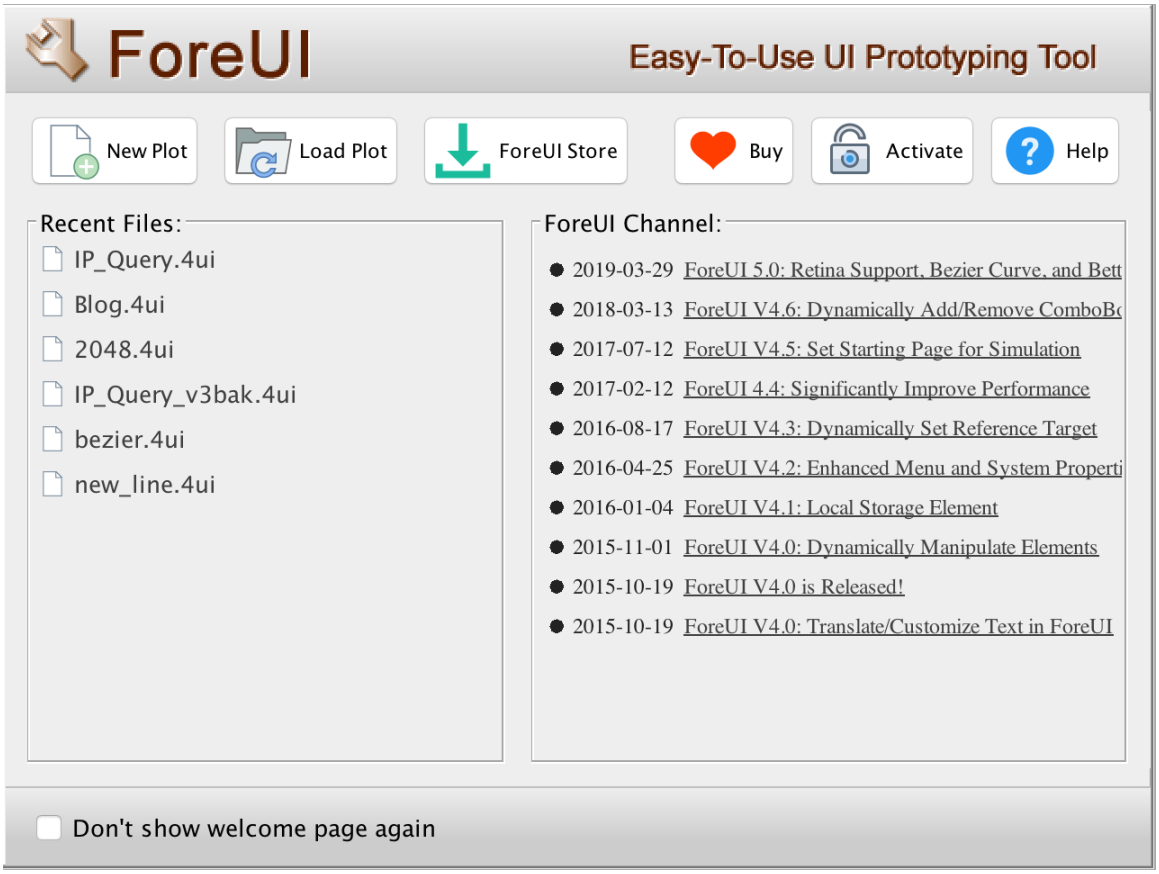


As you can see in the parsing preview, an “N” parameter is appended in the getProperty() function call, and that will do the type casting.

3. GUI Introduction

1.1 Welcome Page

The welcome page will be displayed in the middle of the GUI when you launch ForeUI application. It provides some shortcuts to access some frequently used operations. The welcome page looks like this:



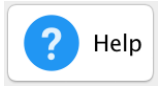
will create a new blank plot.

will ask you to choose a plot file to load for editing.

will open the ForeUI Store window, which allows you to download new ForeUI resources (plot, custom element, library etc.).

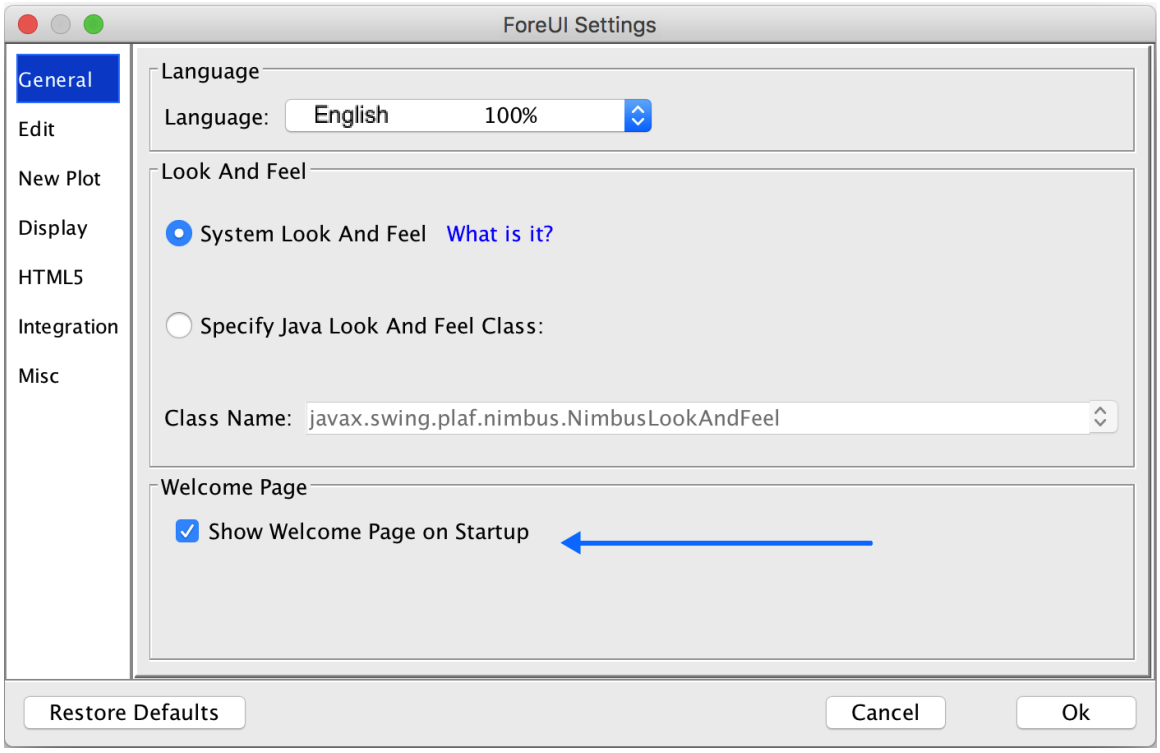
will open the web page for you to order ForeUI license. This button will not show up if your ForeUI is activated.

will display a window that accept your ForeUI license key and activate ForeUI software. This button will not show up if your ForeUI has been activated already.



will open online document of ForeUI in your default web browser.

If you select the "Don't show welcome page again" option at the bottom of the welcome page, the welcome page will not be displayed next time. You can enable welcome page again in the settings window.



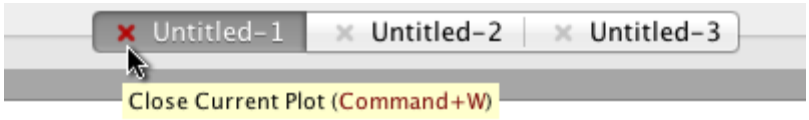
2.1 Plot Editing Area

The plot editing area is the most important view for you to edit your prototype.

Your page will be rendered in this area and you can place as many elements here as you need to build your prototype.

Tab for Plot

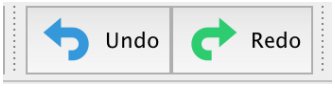
ForeUI supports editing multiple plots at the same time. Each plot will has a corresponding tab on top of the plot editing area; you can switch to certain plot by single clicking on its tab.



There is a small "x" icon in the tab of editing plot, clicking on this icon will close the corresponding plot. If your plot is modified and not saved yet, ForeUI will prompt you to save the plot before closing. You can also Use Hotkey to close the current editing plot, which is Command+W in Mac OS and Ctrl+W in other operating systems.

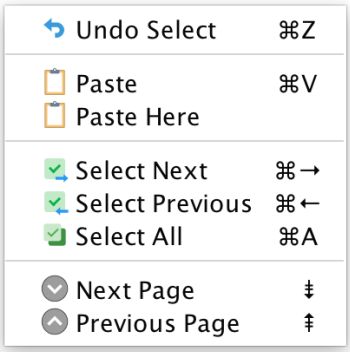
Undo and Redo

You can always use the undo/redo button in the toolbar to undo/redo the operation. You can also use Ctrl + Z and Ctrl + Y (Command + Z and Command + Y in Mac OS) to do the same.



Context Menu

You can click the right mouse button to bring up the context menu, which can help you to access some operation quickly. The context menu may vary according to your current selection.



Element Overlapping

When you place elements in your page, you will care about how they overlap each other. In ForeUI, there are four factors that can affect how elements are overlapped in your page.

1. Z Value

Every element has its own Z value and this attribute will affect the overlapping order of elements. Like HTML rendering, the elements with lower Z values will be rendered first.

It is not recommend assigning negative Z value to element, since different browsers may treat it in different way, when running the HTML5 simulation.

The default Z value for all elements is zero.

2. Order of Creation

If two elements have the same Z values, the order of their creation will determine the overlapping order: the first element will be rendered first.

3. Embedding Relationship

If an element is embedded into container element, it will be rendered above the container element. If the container element is covered by element C, the embedded element will never cover element C, no matter how big Z value it has.

If two elements are embedded into the same container element, their overlapping order is determined by their Z values, then by their order of creation.

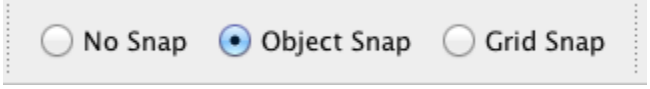
4. Master Page

If a page has master page, the elements from master page always have earlier order of creation. So usually the content from master page will be rendered as a "background" of current page. However, if any element in the master page has higher Z value, that element can cover the content in current page.

Snapping System

When you drag and move element(s), it is possible to align the selected element(s) with other content by using the snapping system.

There are three snapping modes in ForeUI, you can switch snapping mode from the toolbar:

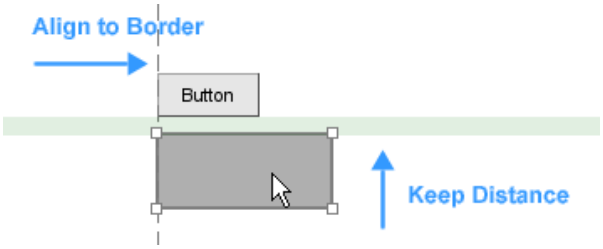


1. No Snap

In this mode, elements can be moved freely.

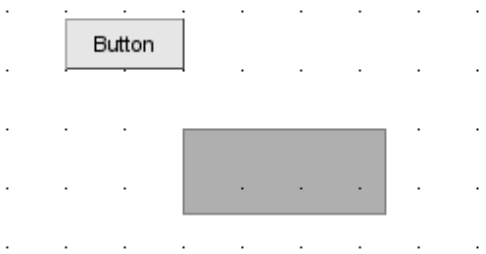
2. Object Snap

This is the default mode. Element will try to align with each other, or keep a fixed distance to each other.



3. Grid Snap

In this mode, element will try to align with the grids in the background.



3.1 Elements Panel

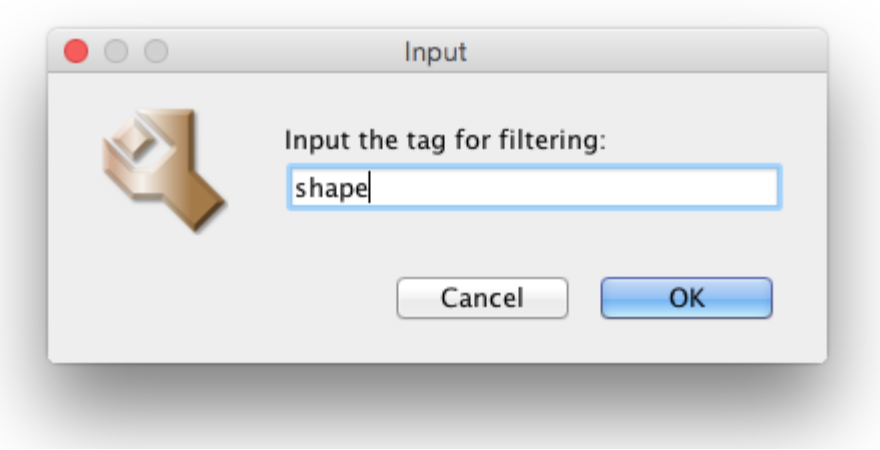
Elements panel is a panel that lists all available elements for you to create your prototype. You can click the "Elements" button on the left toolbar, or press Ctrl+E (Command + E in Mac OS) to show/hide the elements panel.

Filter Elements

You can filter the listed elements by tag or keyword.

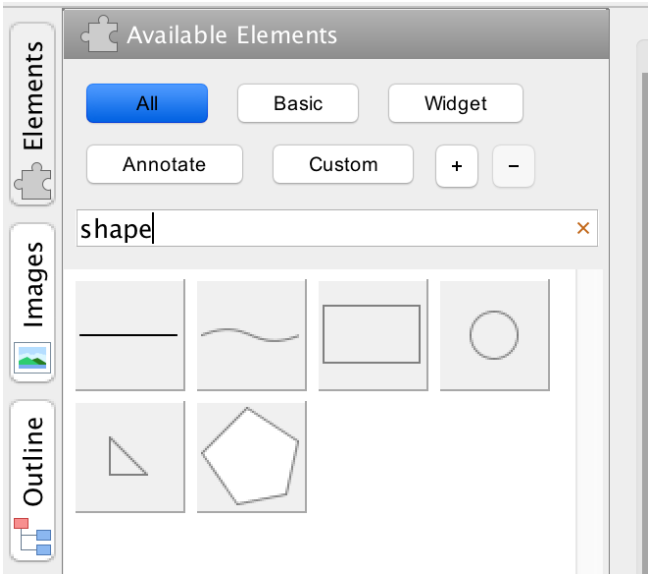
On the top of the elements panel, there are five predefined tag buttons. Clicking these tag buttons can filter the elements by tag. The "All" tag is selected by default, means all elements will be listed. You can click any tag button to filter the listed elements by that tag. For example, clicking on the "Custom" button will list all custom elements.

You can also add your own tag buttons by clicking the "+" button, and then input the new tag:



Then you will see the new "shape" tag button. If you click on it, all shape elements will be listed. You can click the "-" button to remove the current selected tag button. Just keep in mind the predefined five tag buttons cannot be removed.

Besides filtering elements by tag, you can also filter elements by keywords. Just input the keyword in the "Filter Elements..." field and you will see all shape elements below.

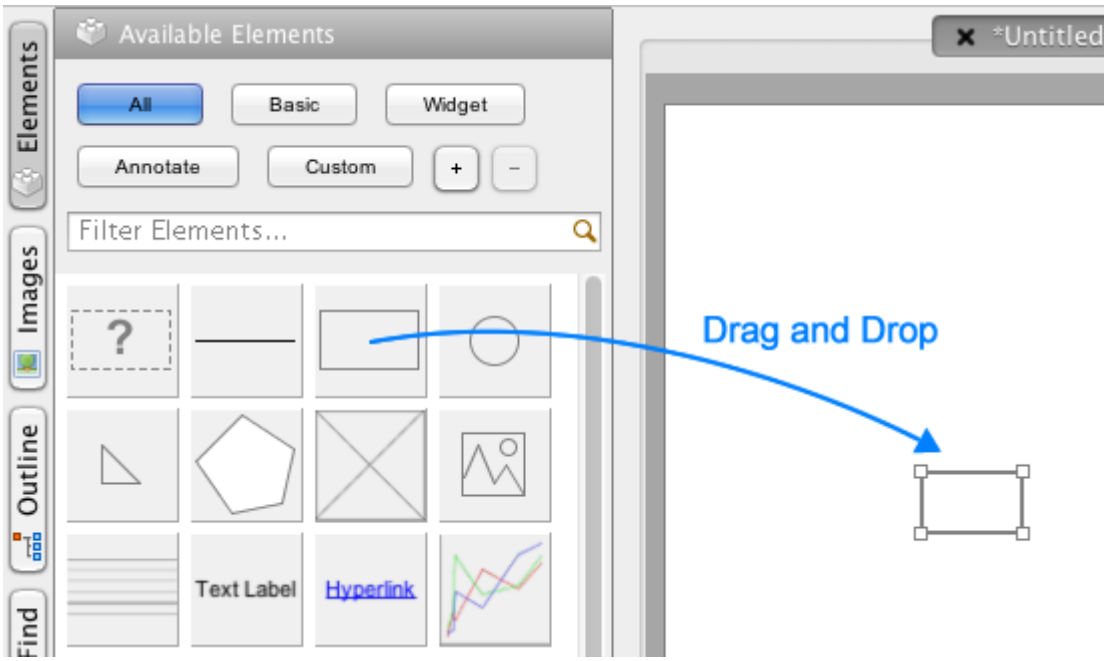


Add Element into Page

There are three approaches to add element into your page.

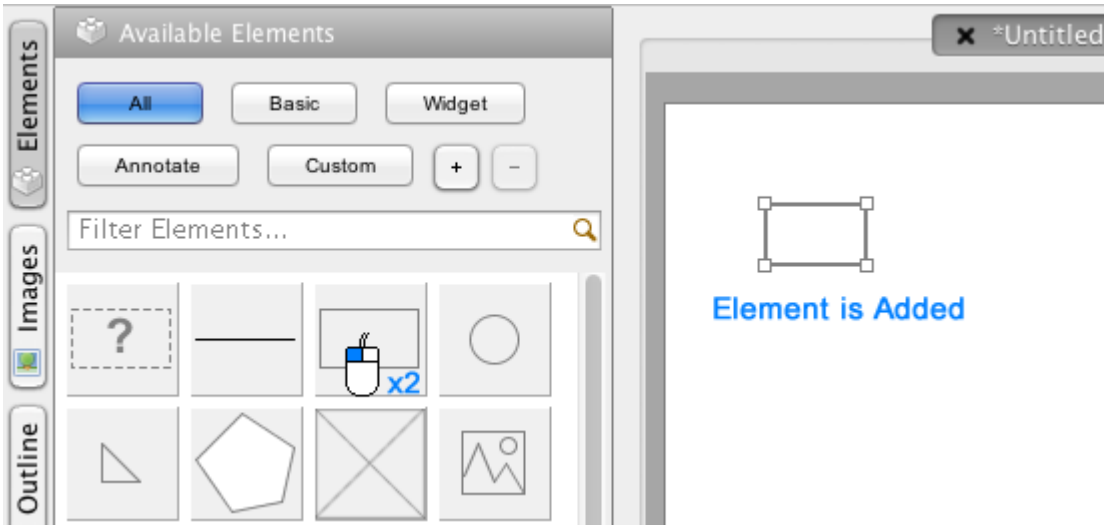
Drag and Drop

You can drag any element from the elements panel and drop it into any location in your page. The newly created element will have default size.



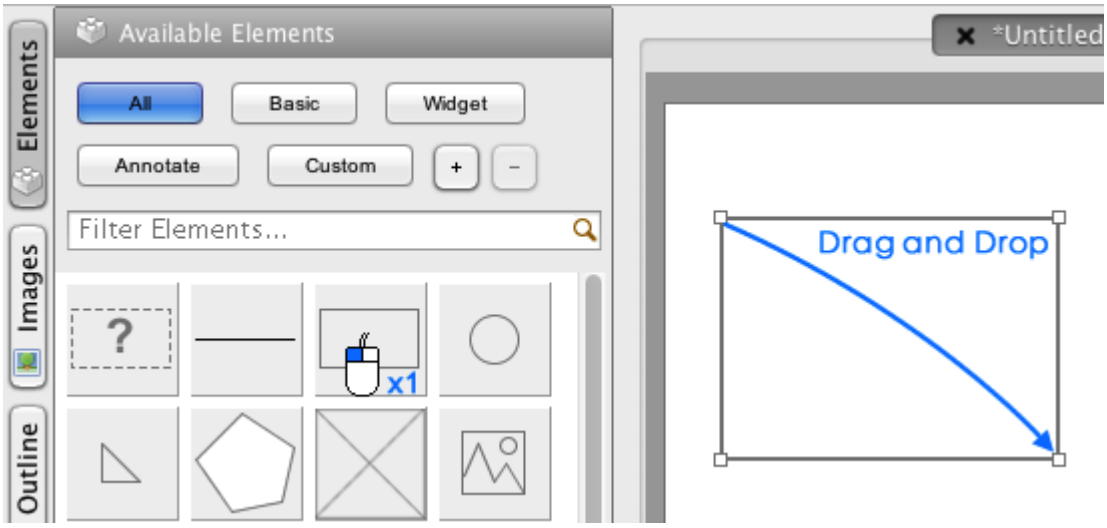
Double-Click

You can double-click any element in the elements panel and it will be added into your page. The newly created element will have default size and location.



Single-Click, then Drag and Drop

You can single-click any element in the elements panel, and then drag and drop in your page to determine the location and size of the new element.



Toolbar on Bottom

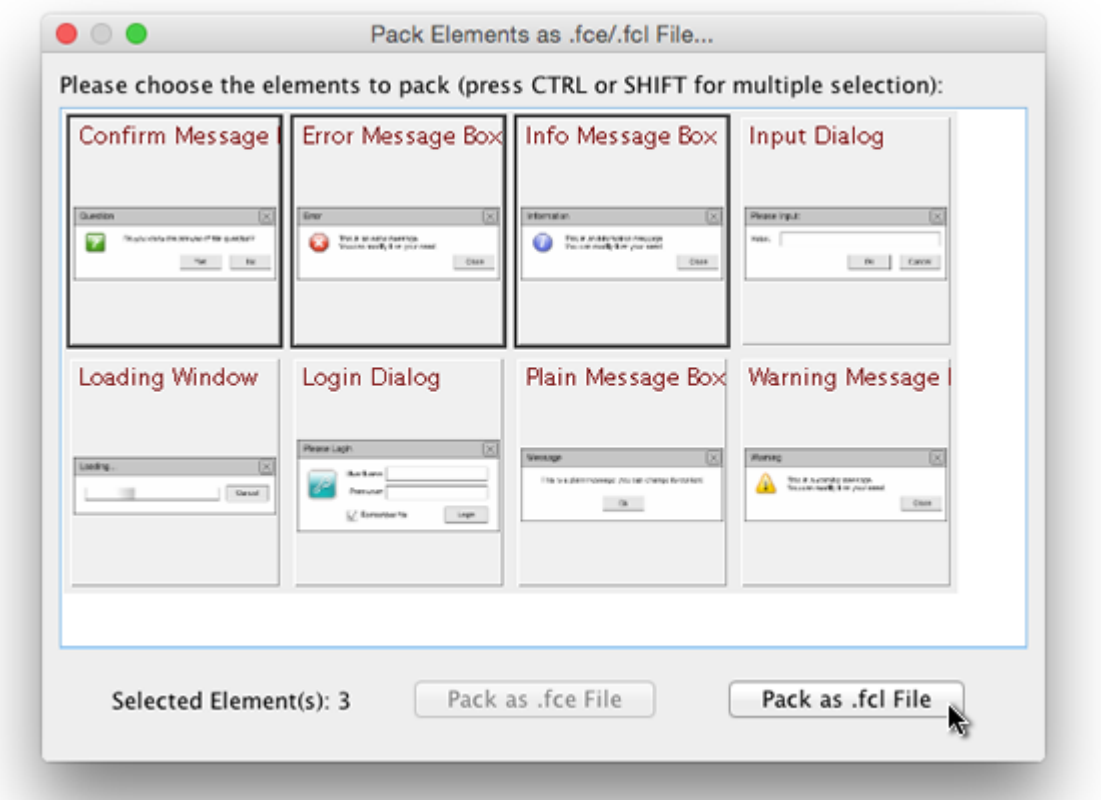
The toolbar on bottom of elements panel provides these tools:

Clicking this button can pack current selected element(s) into a custom element, which will be listed in the elements panel for future usage.

Clicking this button can load custom element (*.fce) or library (*.fcl) from file, and list the loaded custom element(s) in the elements panel.

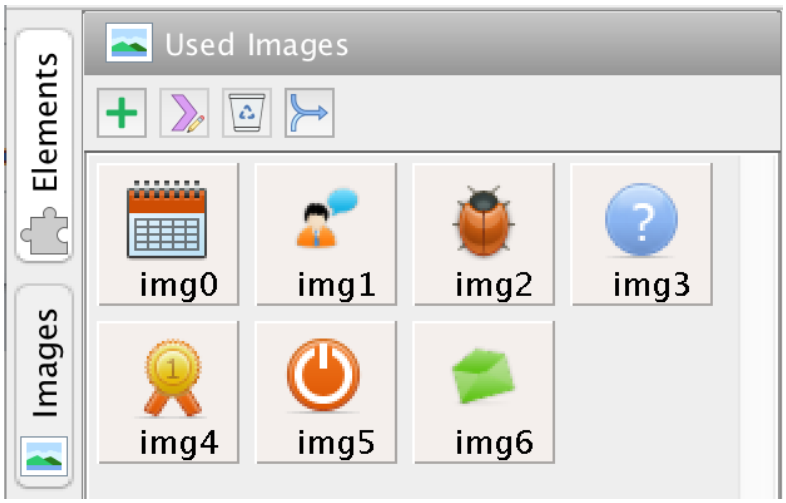
Clicking this button will open ForeUI Store window, which allows you to download new custom elements, libraries and other resources.

Clicking this button can pack one or more elements into .fce file or .fcl file.



4.1 Images Panel

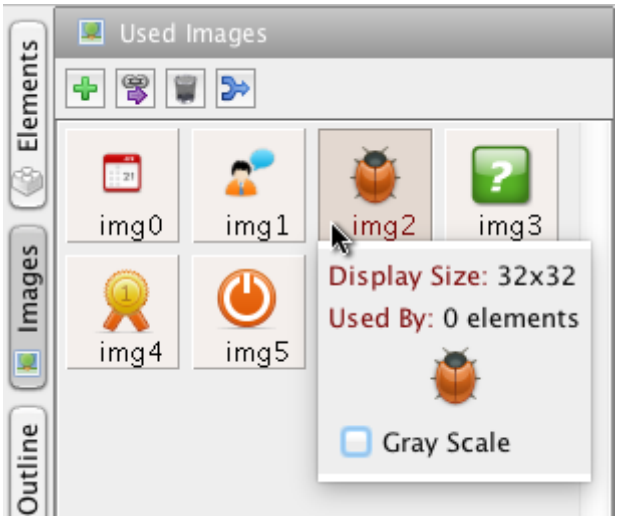
Images panel is a place to manage the images used in your plot. You can click the "Images" button on the left toolbar, or press Ctrl+I (Command+I in Mac OS) to show/hide the images panel.



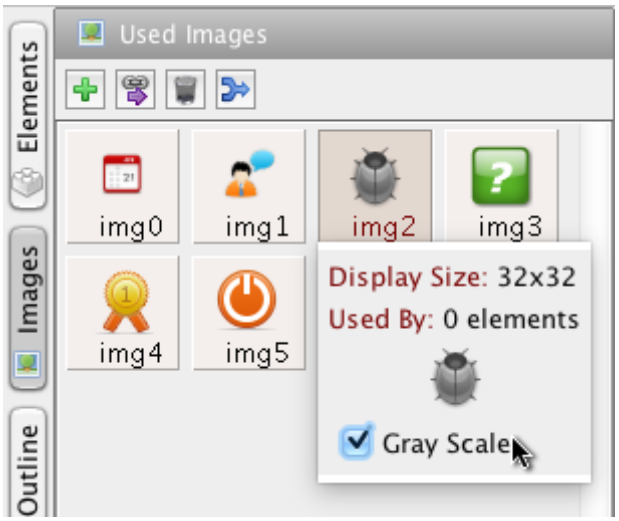
All used images will be listed in this panel. One image can be used by multiple elements, you can find more details about this in "Image and Image Reference" concept description.

Display Image Details

You can single click the image button to display its details in a popup menu. Here you will see the size and number of references for the image.

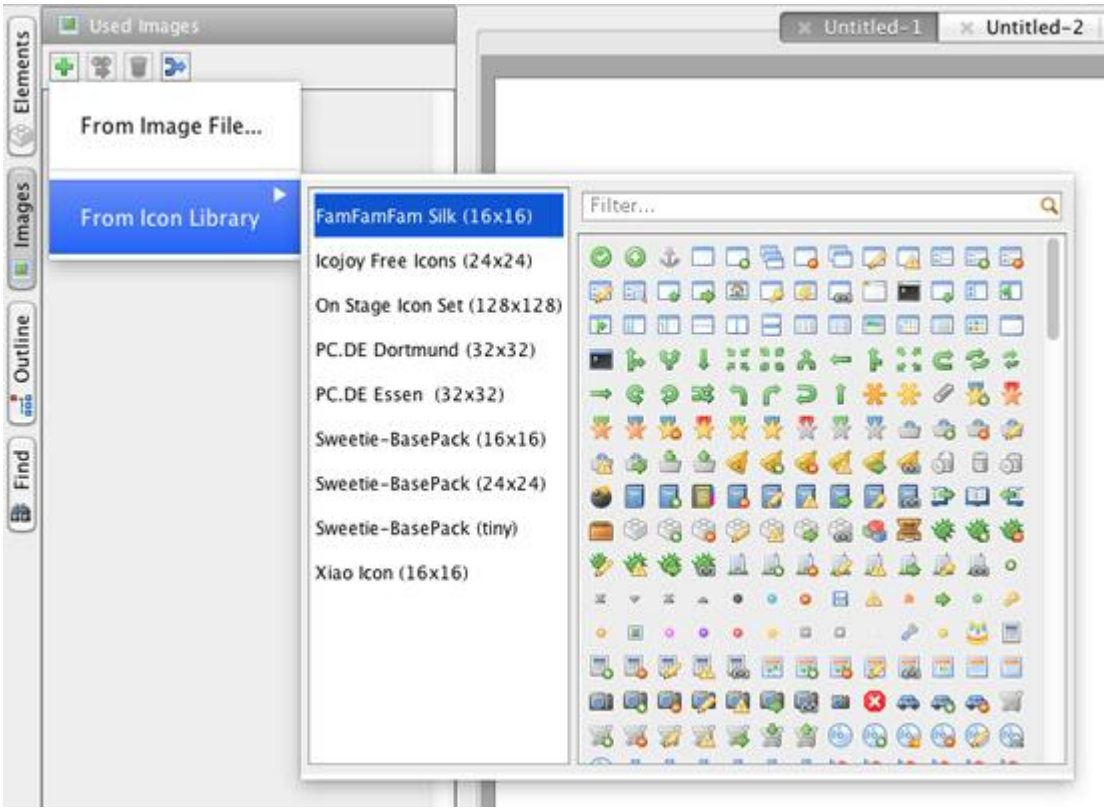


You can also select the "Gray Scale" option to convert it to a gray scale image (all image reference will become gray scale as well).



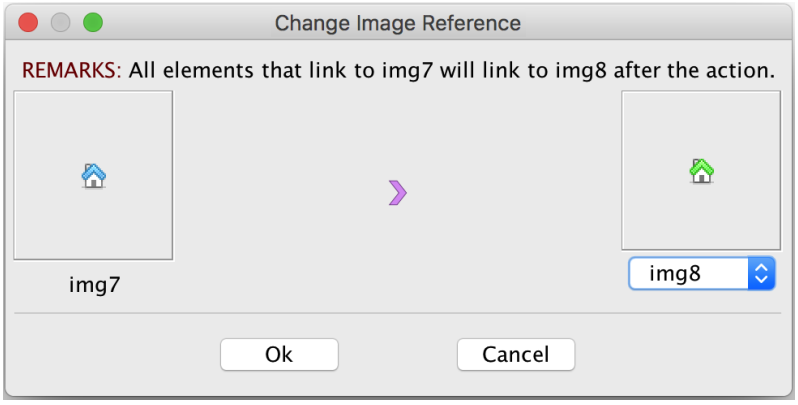
Add Image into Panel

If you specify the image in any element, the image will be added into this panel automatically. Meanwhile you can add an image before using it by clicking the  button. After clicking the button, a menu will pop up and you can choose image from file system or predefined icon library. As shown below:

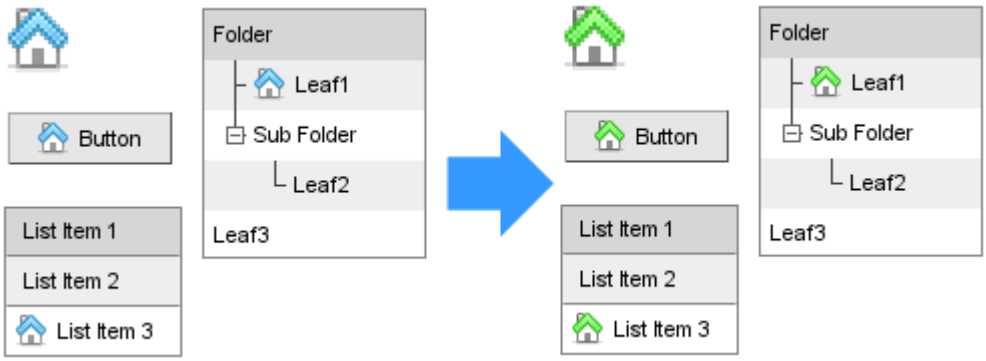


Change Image References

In the Images panel, you can change all references from one image to another via the  button. After clicking the button, you will see the window below:



You can choose the target image from the drop down list. After clicking the "Ok" button, you will see all image references for img0 (blue) are now reference to img1 (green).

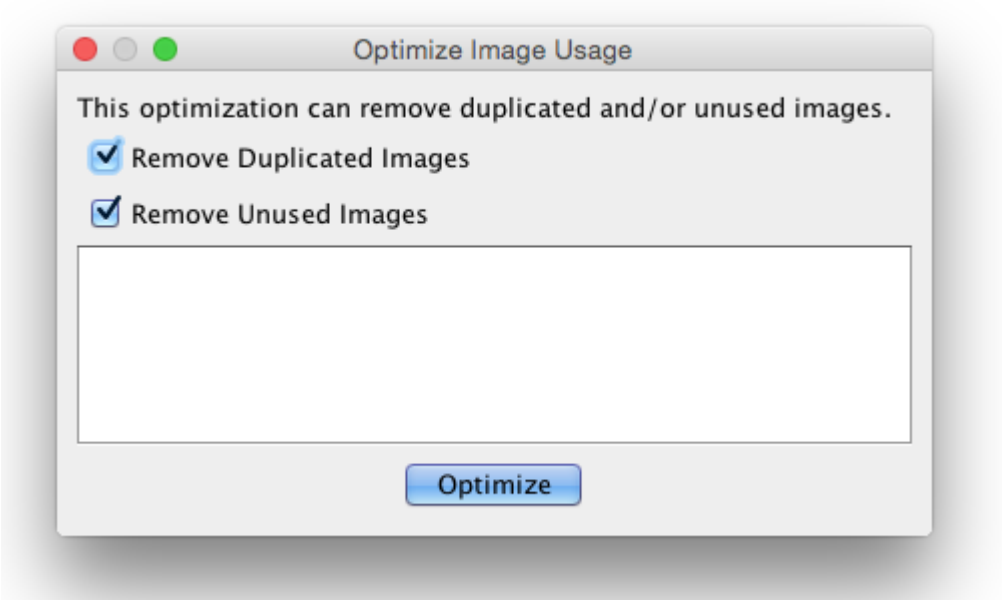


Delete Image

You can delete the image in the panel by clicking the button. Please keep in mind that it will affect all elements that use the image. If you delete an image by mistake, you can revert it by clicking undo button (or press Ctrl + Z).

Optimize Image Usage

If you have some duplicated images in the panel, or there are some unused images in the panel, you can optimize these by clicking the button. After clicking on the button, you will see the window below.

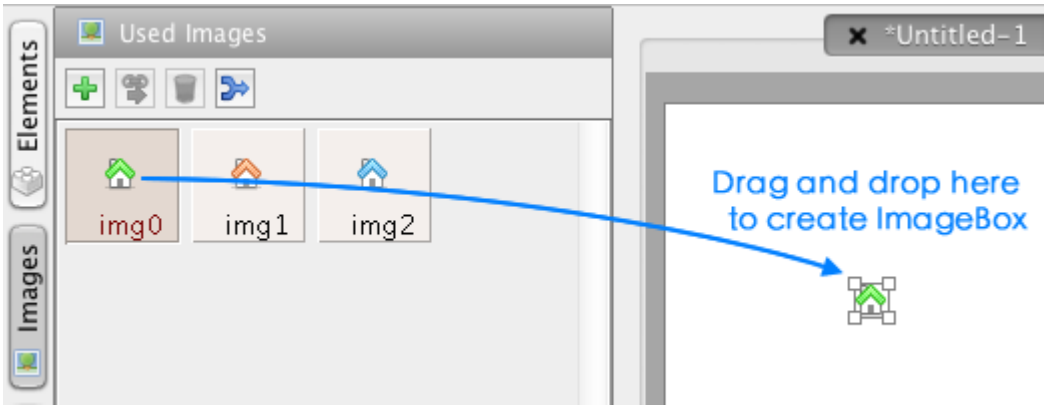


Here you can decide whether to remove the duplicated images and/or unused images. Clicking the "Optimize" button will start the optimization and you will see the logging information in the message view.

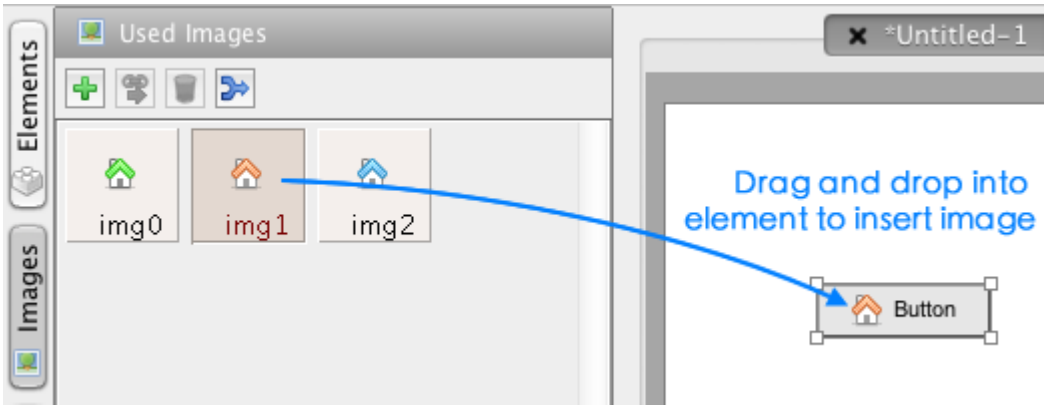
Put Image into Element

You can put image into element via the tools panel. For elements that support images, usually you will see the button in the tools panel, clicking the button will popup a menu for you to choose image from file, images panel, or icon library.

You can also drag image from the Images panel into the editing area directly. Dragging image to an empty area will create the ImageBox element automatically.

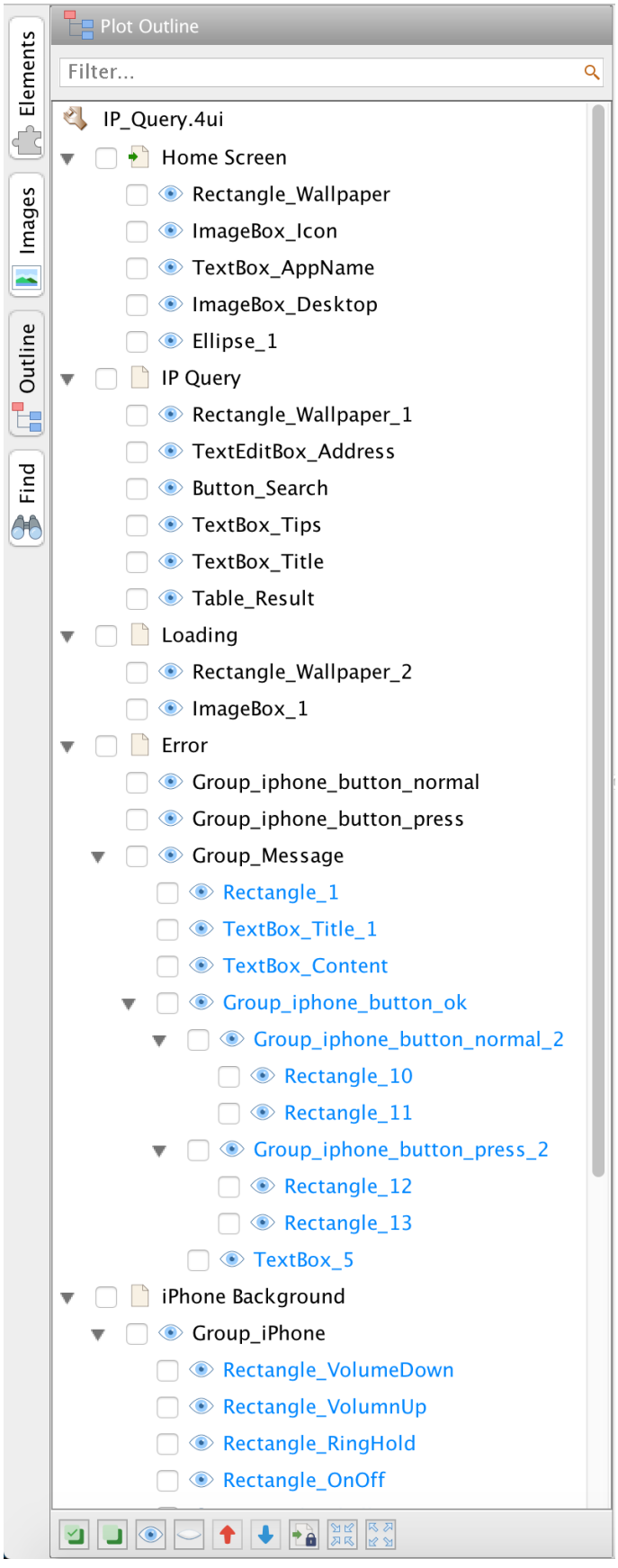


Dragging image into element that can accept image will insert the image into the element automatically.



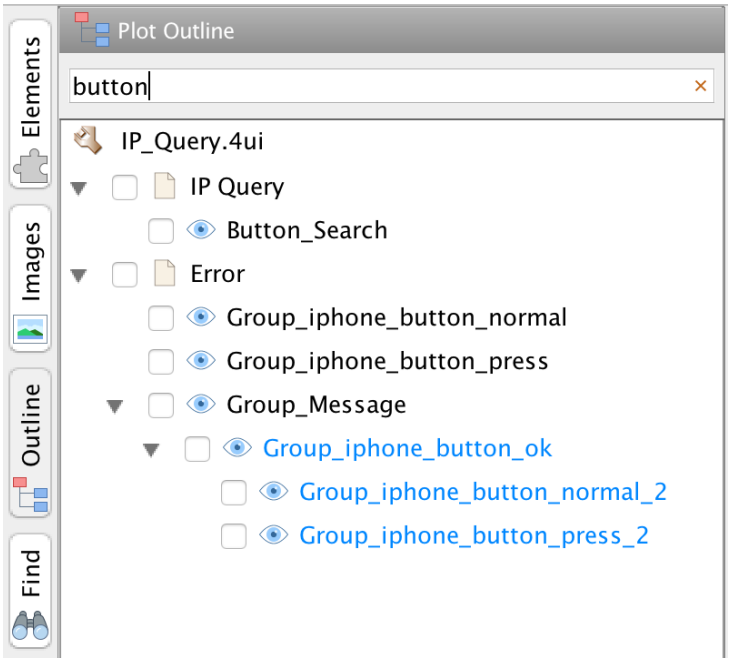
5.1 Outline View

The outline view gives you the overview of the plot content. The content of the plot will be listed as tree structure. All pages will be listed as top level tree node and the elements inside will be placed under the page node. If the element contains embedded element, the embedded elements will be listed as child nodes (with light blue text color). You can also use this view to manage the content of your plot.



Filter Content

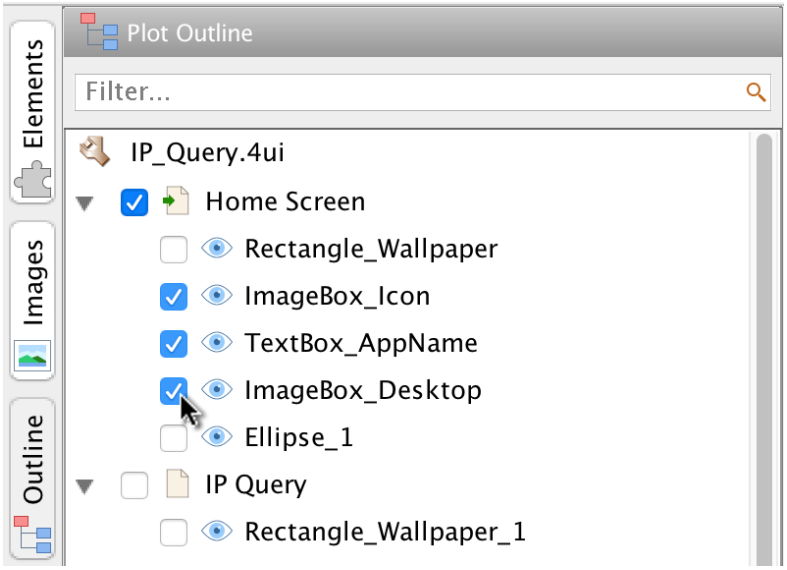
If you want to quickly locate the content you are interested, you can input the keyword in the "Filter..." box on top of the view. Below is an example that filters the content by "button" keyword:



Select Page or Element

There is a check box on the left of every node in the tree view, you can single click this check box to select the corresponding page or element. Selecting the parent node will automatically select its child nodes as well. Selecting elements here may be easier for you if you are dealing with a complex page with many overlapped elements.

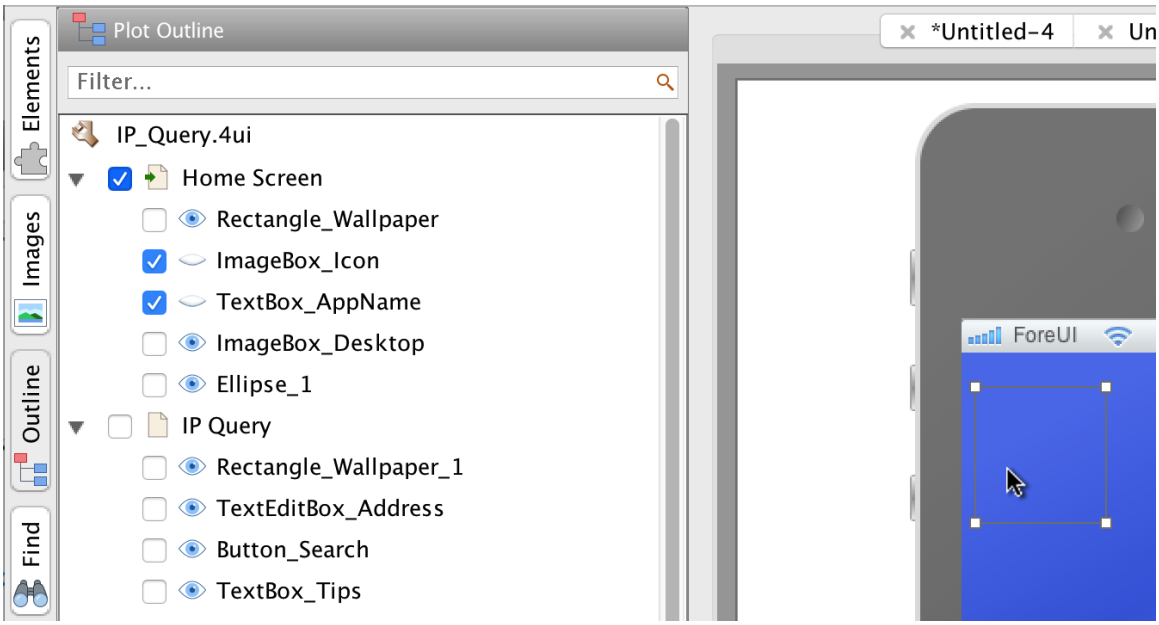
Please notice that, clicking on the checkbox has different behavior than clicking on the text label of the node. Clicking on checkbox will add/remove current node into/from selection, and will not clear the existing selection. While clicking on text label will clear existing selection and only select the node being clicked.



Clicking the  button in bottom toolbar can select all elements in the current page. Clicking the  button in bottom toolbar can unselect all elements in the current page.

Show/Hide Element

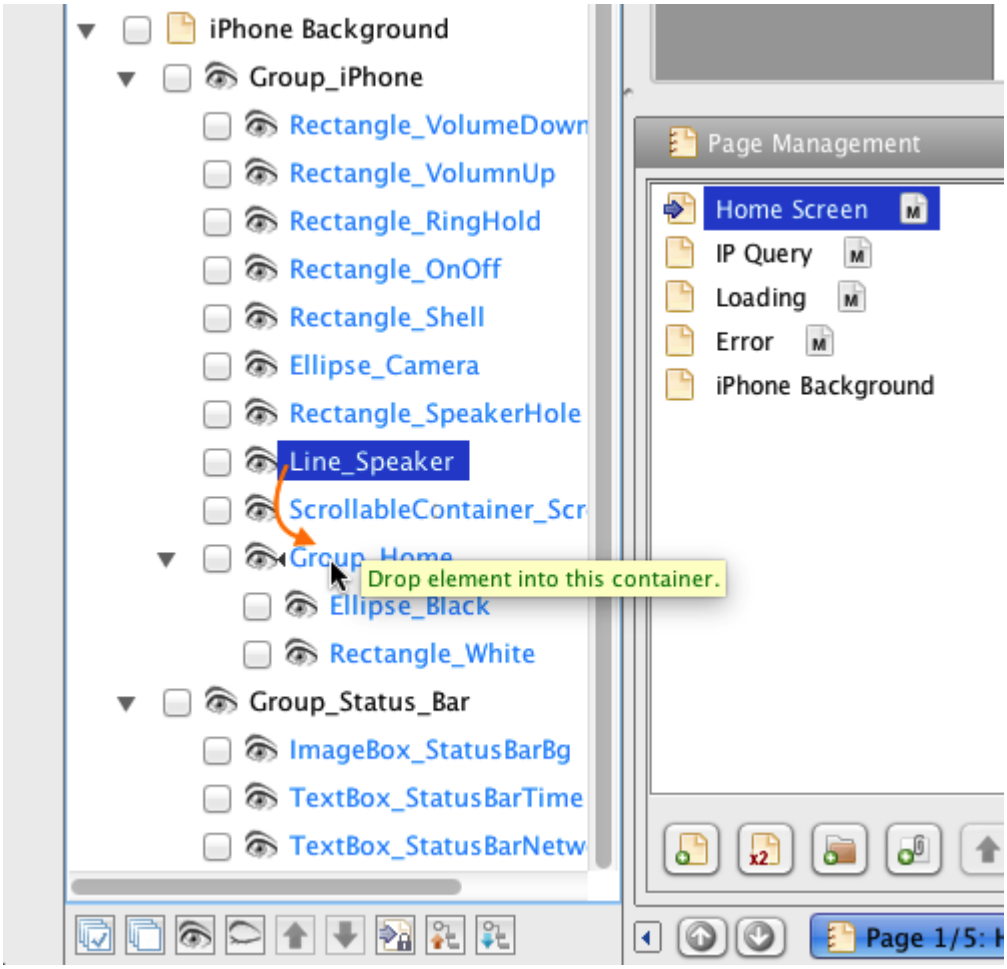
In outline view, you can show/hide the selected elements as you need. You can click the  button to show selected elements, and you can click the button to hide selected elements. Below is an example that hides two selected elements.



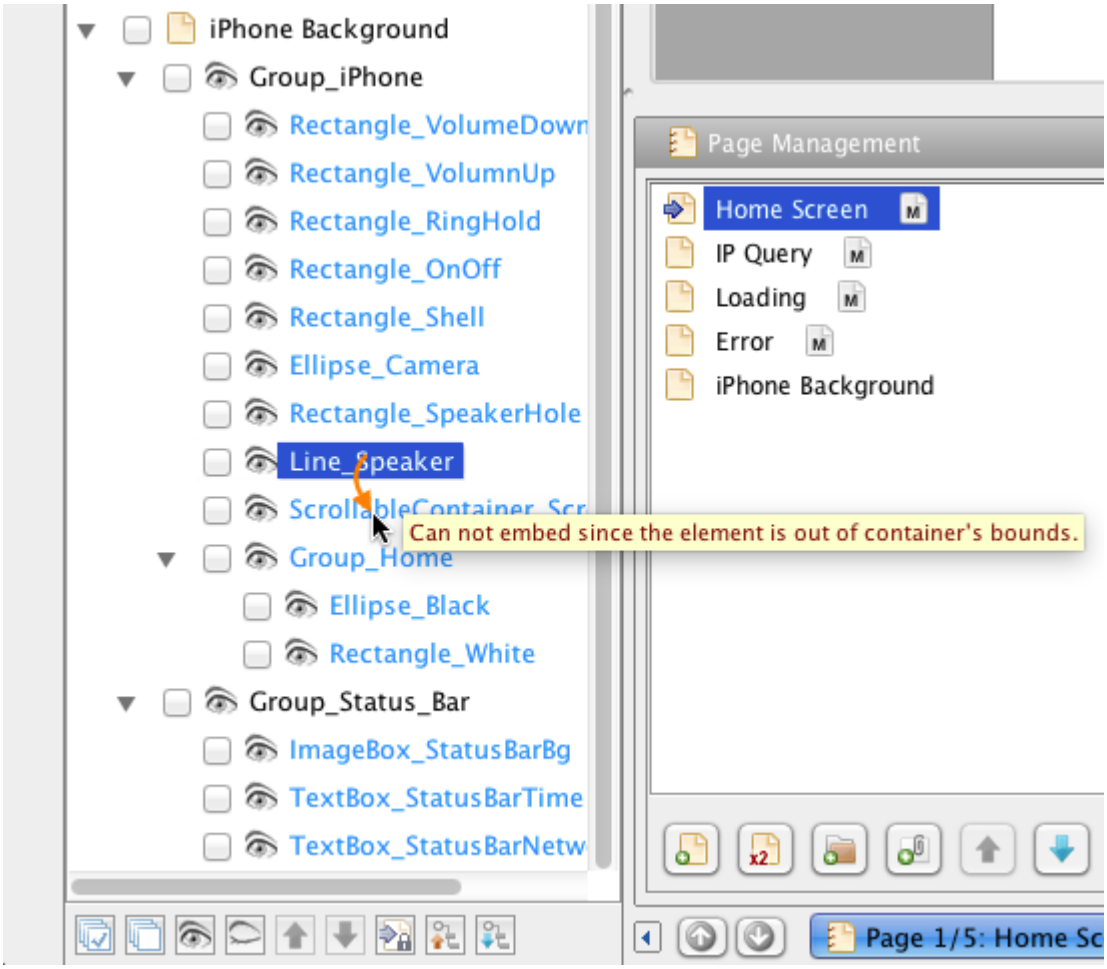
Remarks: the elements hidden by outline view are still present in your plot, they are just not rendered. Hiding elements here will not affect their actual status, they are just temporarily hidden. This feature is very useful to give you a clear view of content that covered by other content.

Embed Element by Drag and Drop

You can drag and drop the current selected elements into container in the outline view. This is a fast way to embed element into container.



Please notice the tool tip when hovering on the container element, sometimes the container refuse the incoming element for some reasons, here is an example:



The embedding is refused because the element is not within the container's area. In this case you move the element into the container's bounds and try again.

You can find more details about embedding element on Conjoin Multiple Elements section.

Change Element Order

In ForeUI, the way that elements overlap each other is decided by the the element's order and its Z value. If two elements have the same Z value, the one with smaller index will be rendered first (and be covered by the other element). So changing element's order is very useful to adjust the element overlap order.

You can change the element's order in the outline view by selecting it and click the  or  button at the bottom, you will see the rendering order is changed accordingly.

Show/Hide Non-Current Pages

Here you can toggle the  button to show/hide the non-current pages. When this button is pressed, only the content in current page will be listed in the outline view.

Collapse/Expand All Nodes

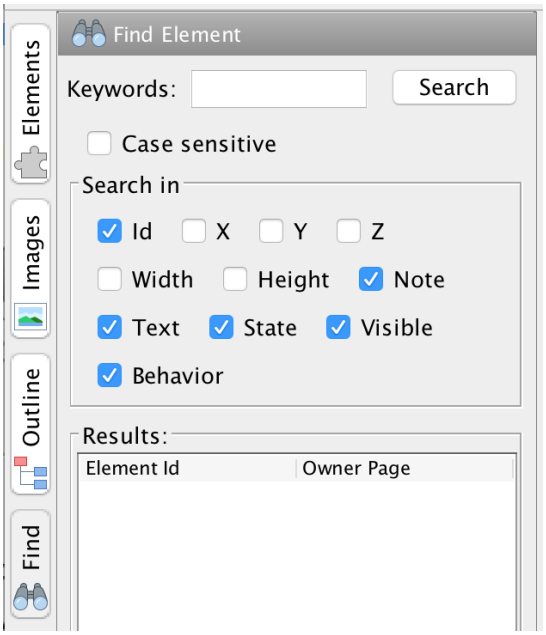
By clicking the  button, you can collapse all nodes in the outline view.

By clicking the  button, you can expand all nodes in the outline view.

6.1 Element Search Panel

Sometimes you may need to find an element in a complex project. If you remember some characters of that element, you can find it easily via element search.

You can open the element search panel by clicking the "Find" button in left toolbar, or press CTRL+F (Command + F in Mac OS).



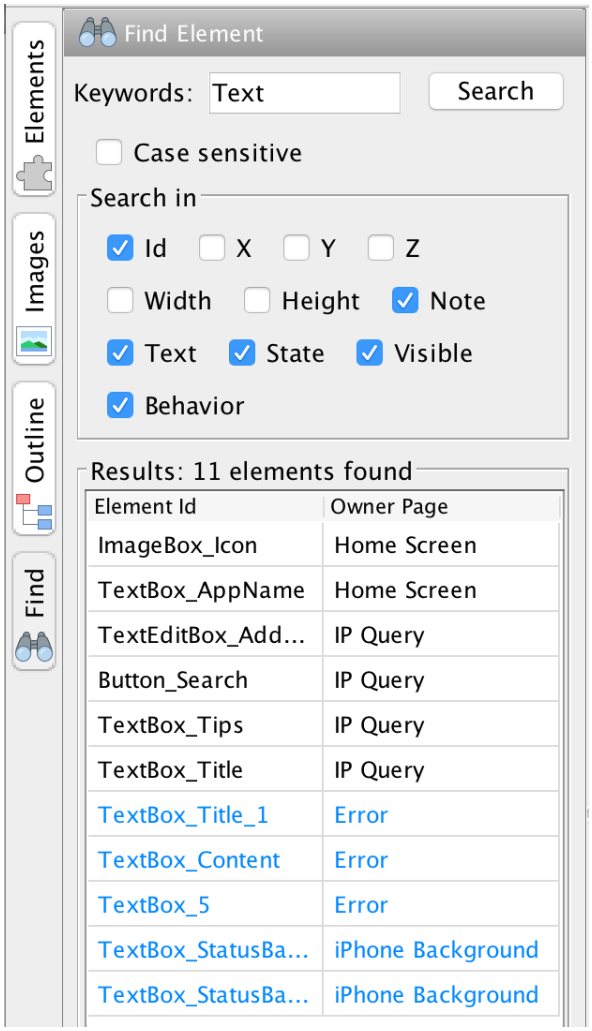
You can input keywords and press the "Search" button (or press ENTER on keyboard) to start the searching. If you have multiple keywords, you can separate them with spaces or commas. Please remember: if you input several keywords, the element will not be listed until it matches all keywords. For example, you use two keywords: "Text" and "Edit", then search in Id, the search results will contain all TextEditBox elements, but no TextBox element will be listed.

If you check the "Case sensitive" option, the search will be case sensitive.

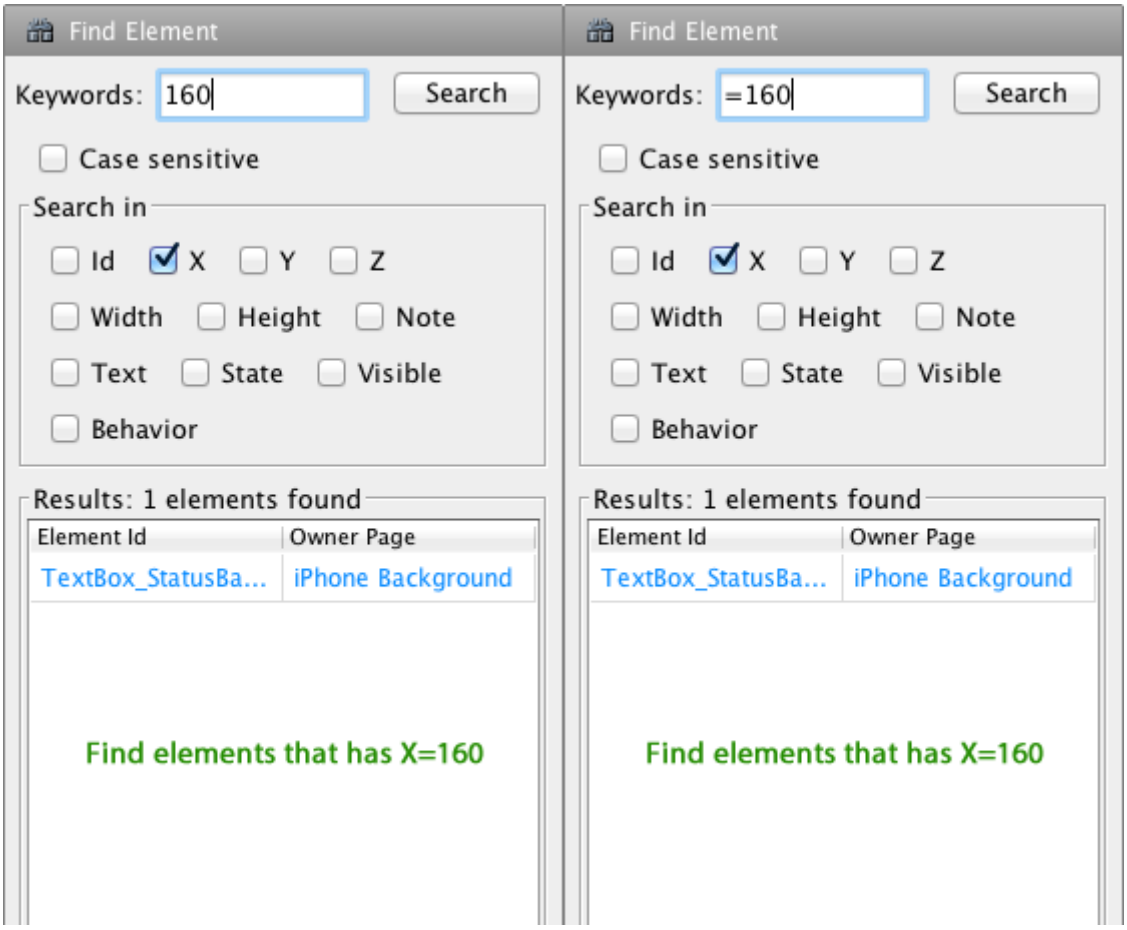
If you check the "Current page only" option, the search will be performed in current page only (Otherwise it will search in the entire plot).

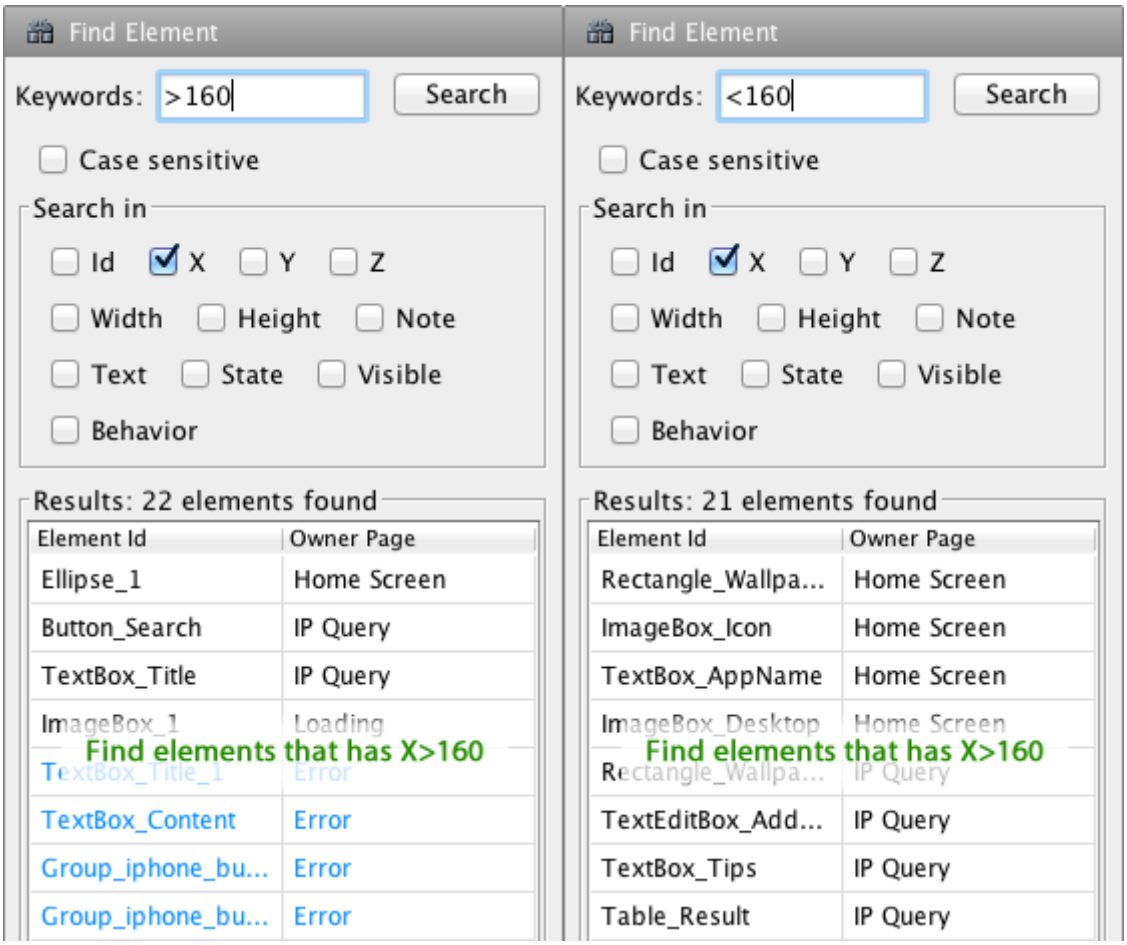
You can specify the fields for searching; just check the options you need in the "Search in" area. The fields belong to **three categories**:

1. The "Id", "Note", "Text" and "Behavior" fields are for common string comparison, the element will be added into results if any of the fields contains the given keywords.

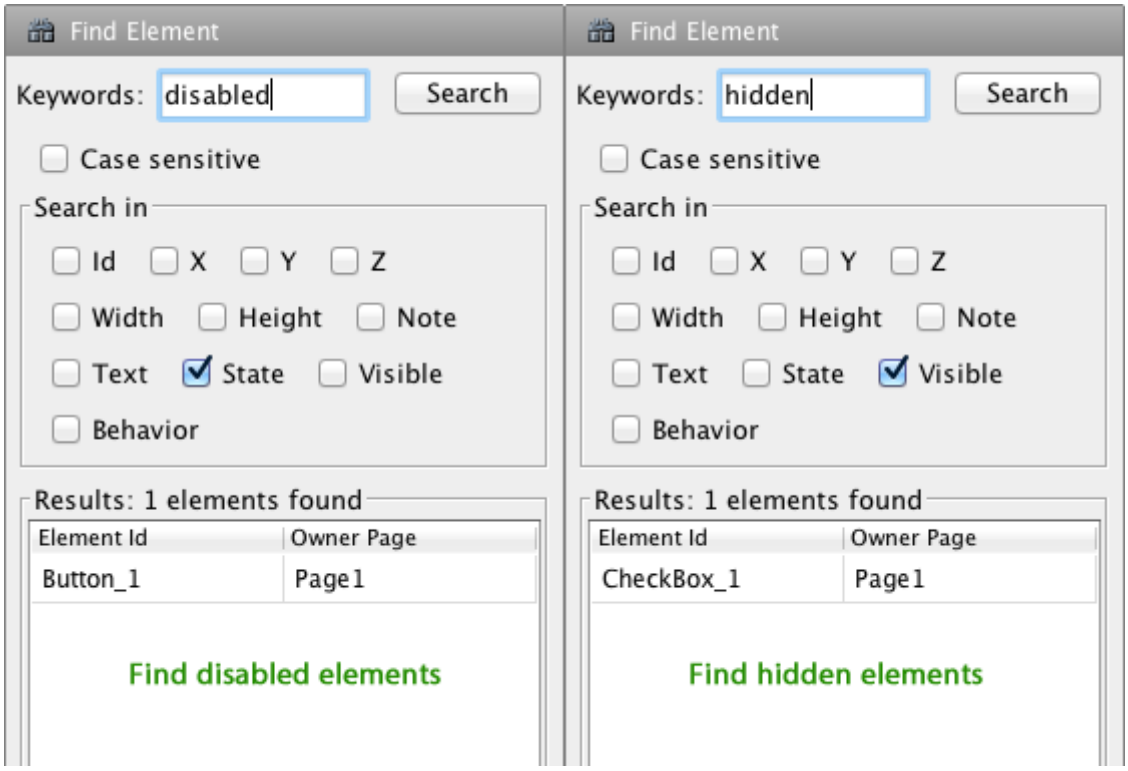


2. The "X", "Y", "Z", "Width" and "Height" fields can be used to find elements that have rough or certain numeric property. The cases below are all supported:


Find Element
Keywords: 160
Search
Case sensitive
Search in
Id X Y Z
Width Height Note
Text State Visible
Behavior
Results: 1 elements found
Element Id
Owner Page
TextBox_StatusBa...
iPhone Background
Find elements that has X=160
Find Element
Keywords: =160
Search
Case sensitive
Search in
Id X Y Z
Width Height Note
Text State Visible
Behavior
Results: 1 elements found
Element Id
Owner Page
TextBox_StatusBa...
iPhone Background
Find elements that has X=160


Find Element
Keywords: >160
Search
Case sensitive
Search in
Id X Y Z
Width Height Note
Text State Visible
Behavior
Results: 22 elements found
Element Id
Owner Page
Ellipse_1
Home Screen
Button_Search
IP Query
TextBox_Title
IP Query
ImageBox_1
Loading
Find elements that has X>160
TextBox_Content
Error
Group_iphone_bu...
Error
Group_iphone_bu...
Error
Find Element
Keywords: <160
Search
Case sensitive
Search in
Id X Y Z
Width Height Note
Text State Visible
Behavior
Results: 21 elements found
Element Id
Owner Page
Rectangle_Wallpa...
Home Screen
ImageBox_Icon
Home Screen
TextBox_AppName
Home Screen
ImageBox_Desktop
Home Screen
Find elements that has X>160
TextEditBox_Add...
IP Query
TextBox_Tips
IP Query
Table_Result
IP Query

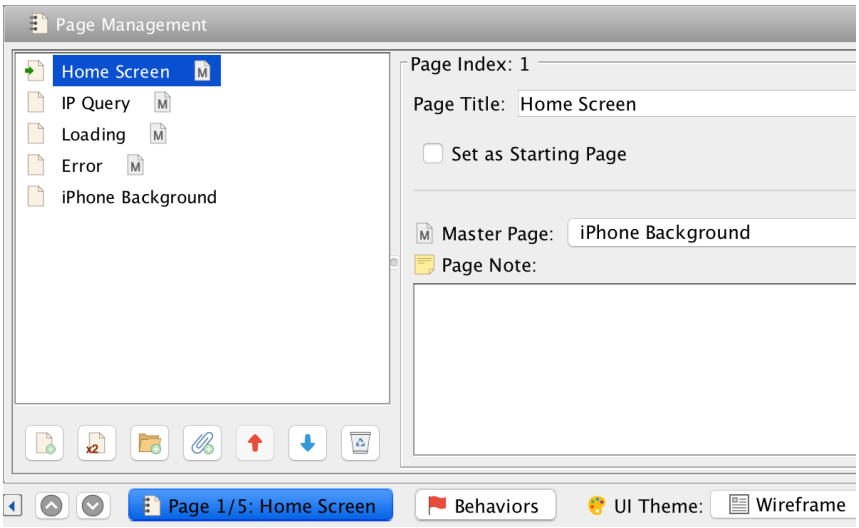
3. The "State" and "Visible" fields can match elements that in certain status, such as:



7.1 Page Manage View

ForeUI supports multipage prototyping, so each plot can have as many pages as you need. To open the page manage window, please click the corresponding button at the bottom toolbar, or press Ctrl + P (Command + P in Mac OS):

The page management window looks like this:



This view is separated to two panels. The tree view on the left allows you to manipulate the pages or folder with the buttons in toolbar:

 Create a new blank page.

 Duplicate the current selected page.

 Create new folder (to hold pages).

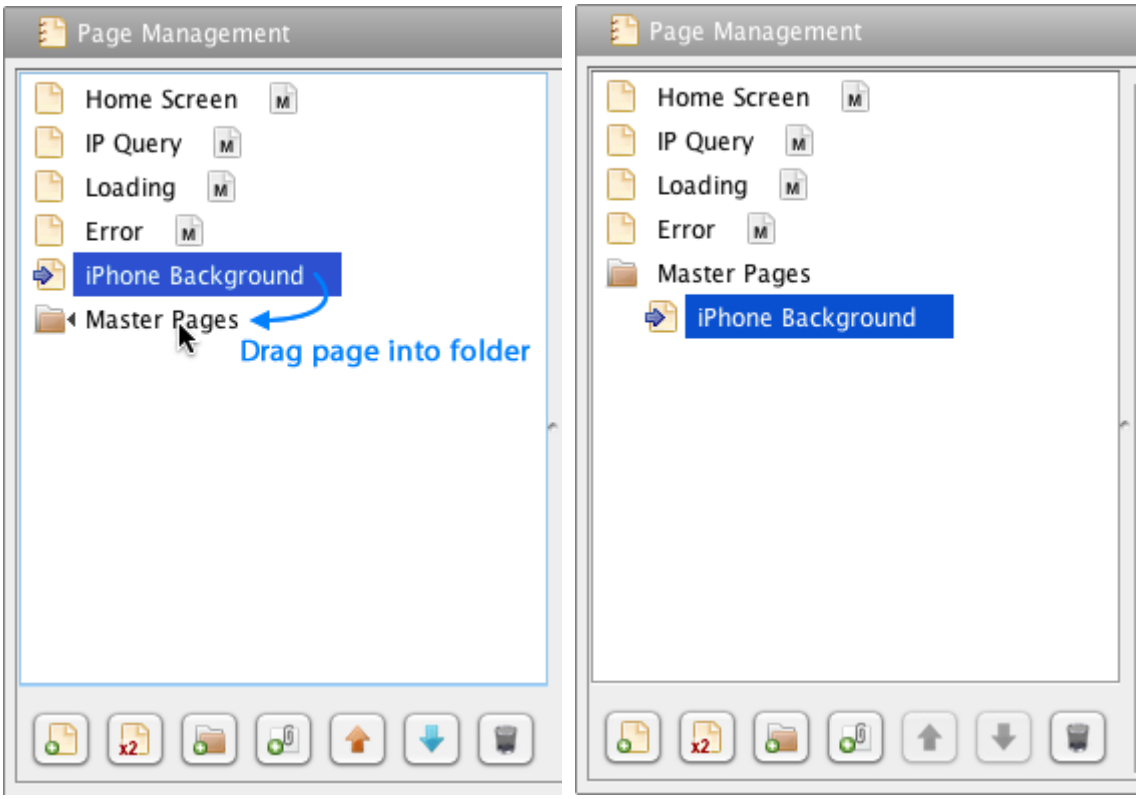
 Attach file(s) to the plot (will be included in HTML5 simulation).

 Move the current selected page up.

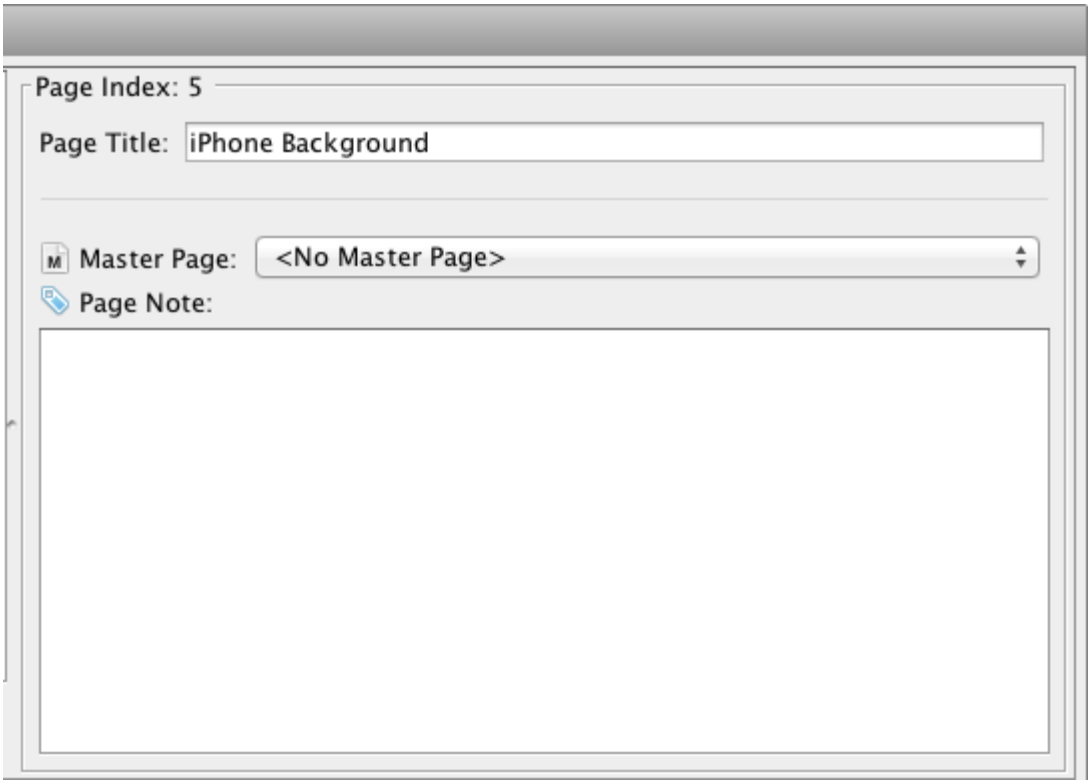
 Move the current selected page down.

 Delete the current selected page.

You can also drag and drop page or folder to make it nested into another page / folder.

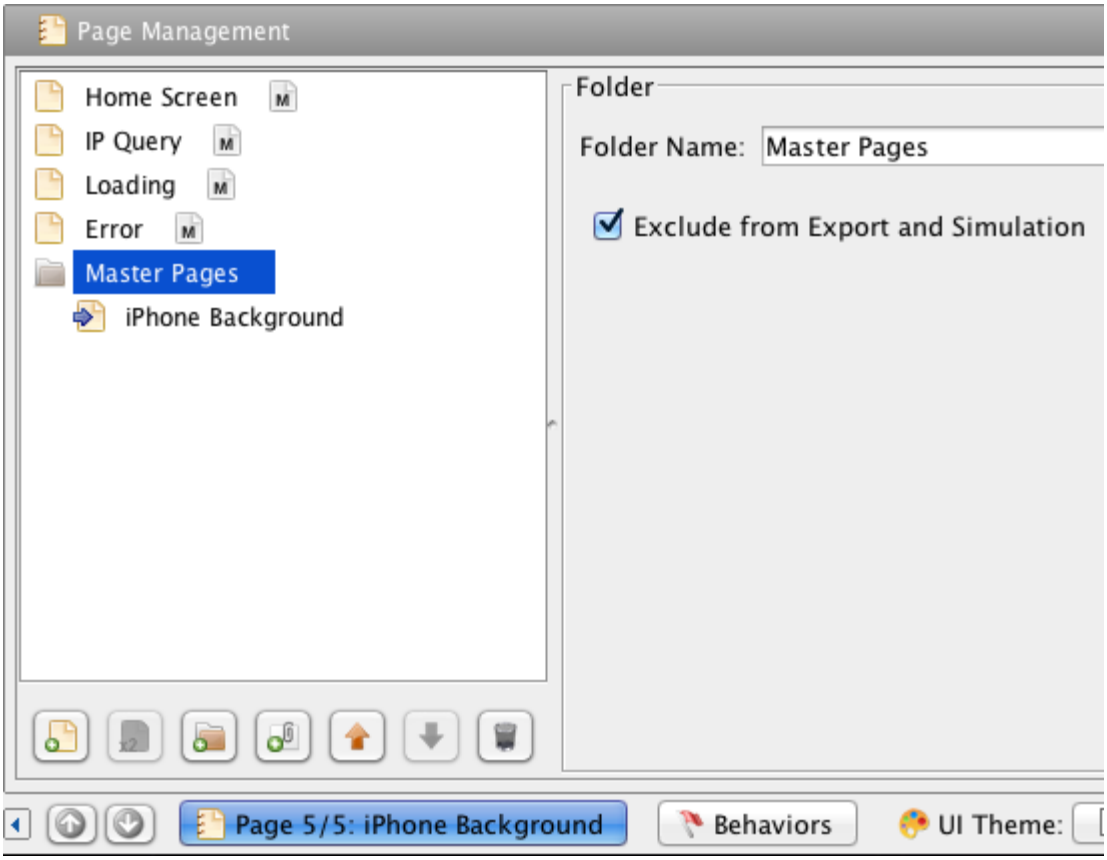


The panel on the right shows the detailed information about the select page / folder. You can change the title, assign the master page and input the page note for the select page.



You can select any other page as the master page for the current page. Thus your current page will inherit the content from its master page. You can find more details about this in the master page section.

When you select a folder in the tree view, you can rename it on the right panel. Also you can **exclude** it from the export and simulation, which means the pages under the folder (and its sub folder) will not be exported. In the example below, the "Master Pages" folder is to store master pages used by other pages, the folder is excluded from export and simulation since the master pages are used as page template, they are not completed pages.

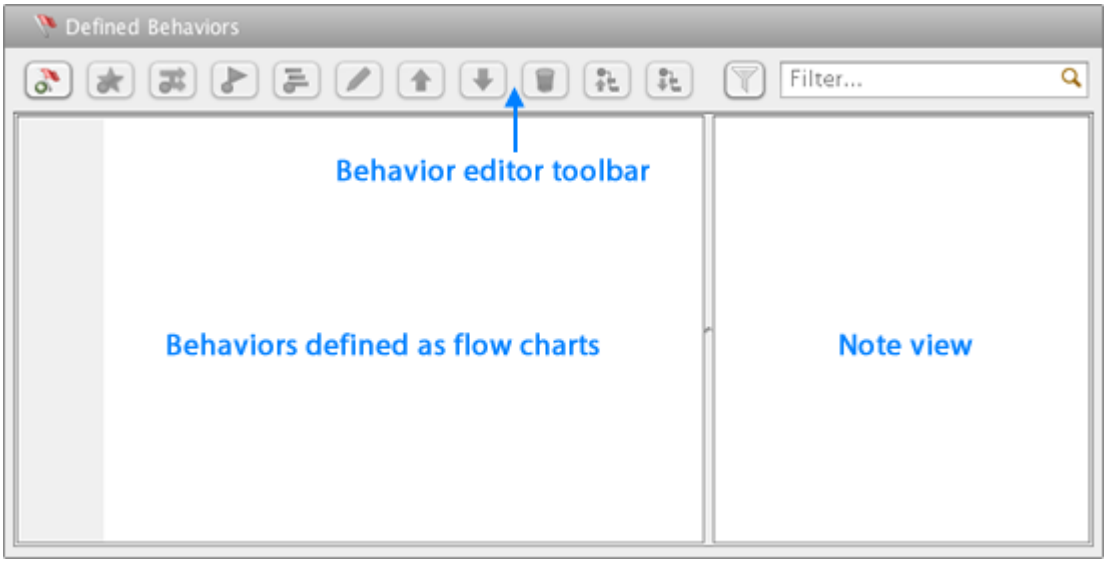


8.1 Behavior Editor View

Behavior editor can define the behavior of certain element(s) or page(s) in an intuitive way.

You can open the behavior editor view by clicking the "Behaviors" button in the bottom toolbar, or press Ctrl + D (Command + D in Mac OS).

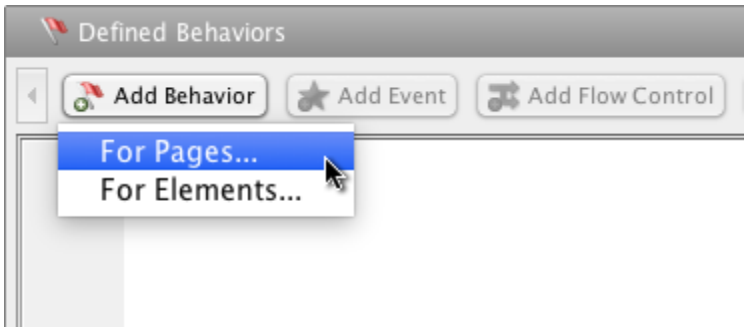
The behavior editor view looks like this:



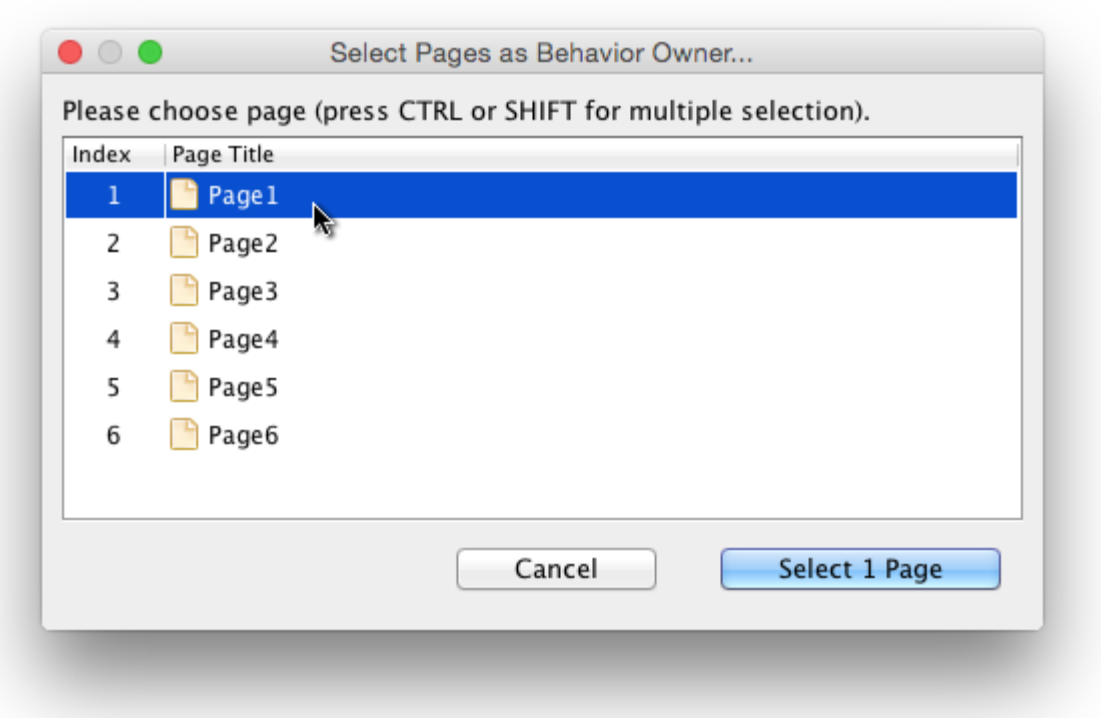
You can show/hide the text on toolbar button in the "Display" tab in the settings window. The text on toolbar are visible by default, you can hide it to save some space on the toolbar.

Add Behavior

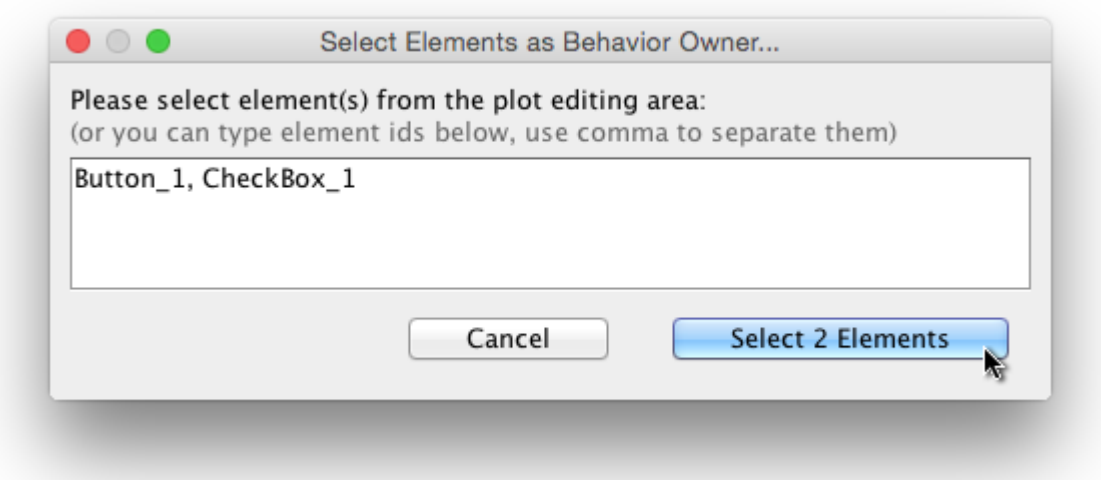
To add a behavior, you should start with clicking on the "Add Behavior" button in the toolbar. A menu will pop up and ask if the new behavior is for page or element:



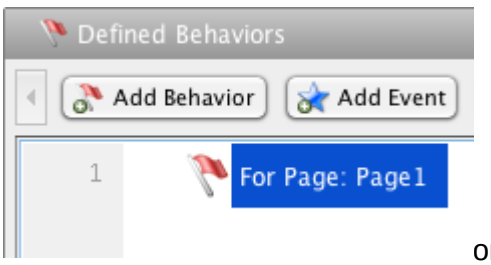
If you choose "For Pages...", a window will show up and prompt you to choose the page(s) as the owner of new behavior. You can choose multiple pages here, then those page will have the same behavior you define later.



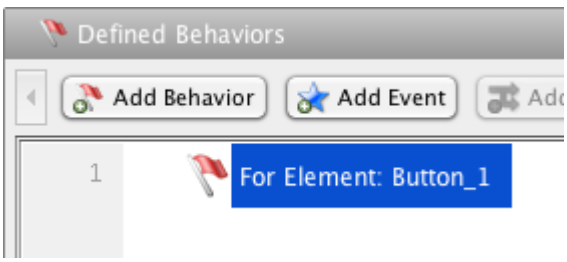
If you choose "For Elements..." instead, an "Element Chooser" window will be displayed and you can then select any element in your plot as the owner of new behavior. If you want, you can choose multiple elements here, and then they will have the same behavior you define later.



After selecting the owner of new behavior, you will see a new node created in the behavior editor, like this:



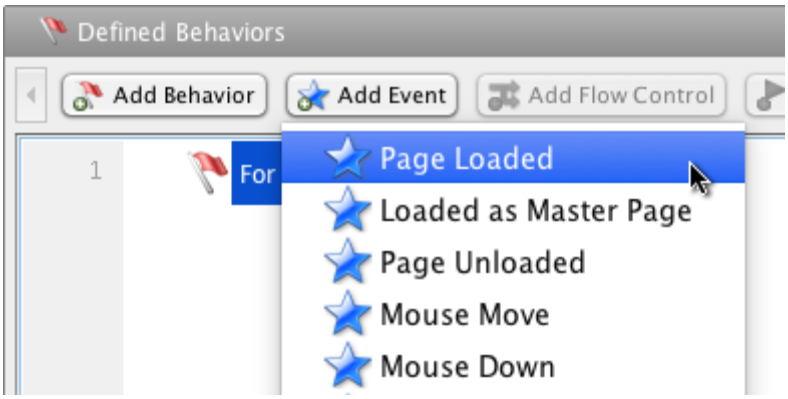
or



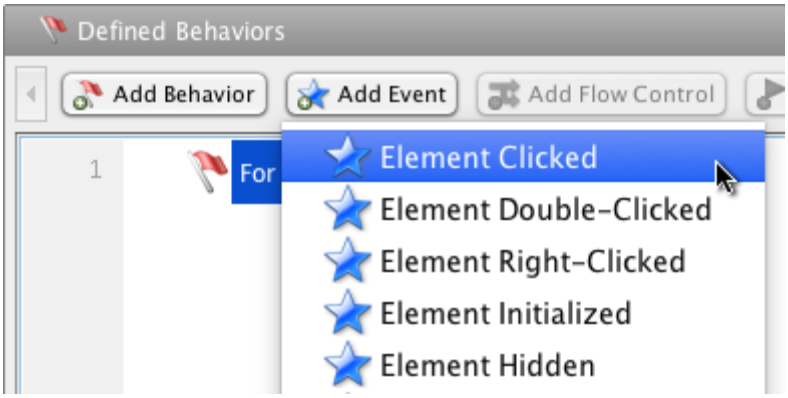
Add Event

The "Add Event" button is enabled when you select the behavior (node with red flag icon). That means you can continue defining the behavior by adding events into it. All available events (according to owner type,

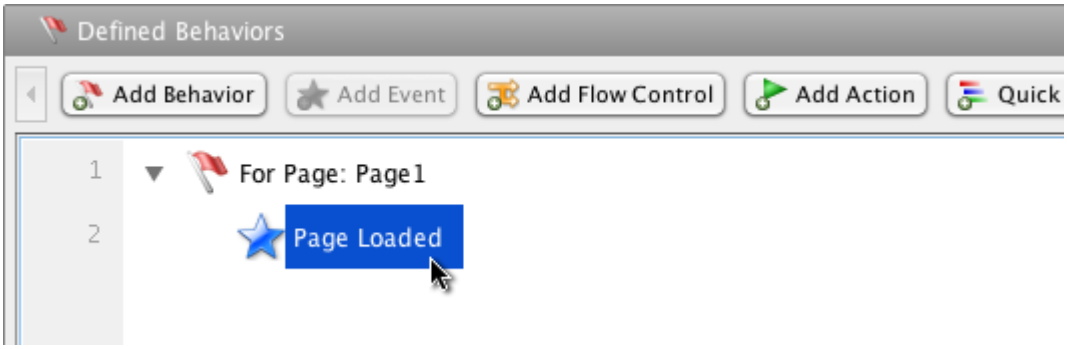
you can find all available events in Events Reference) will be listed in a popup menu when you click the "Add Event" button. If the behavior is for page, you will see:



If the behavior is for element, you will see:



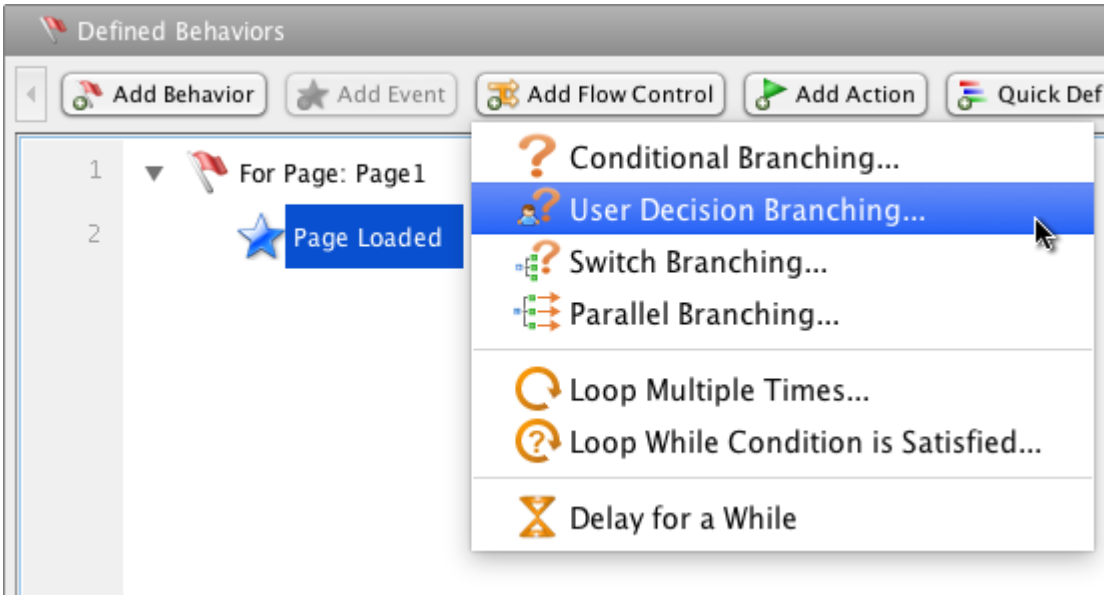
After adding the event, it will be selected by default and the "Add Flow Control" and "Add Action" buttons will be enabled now.



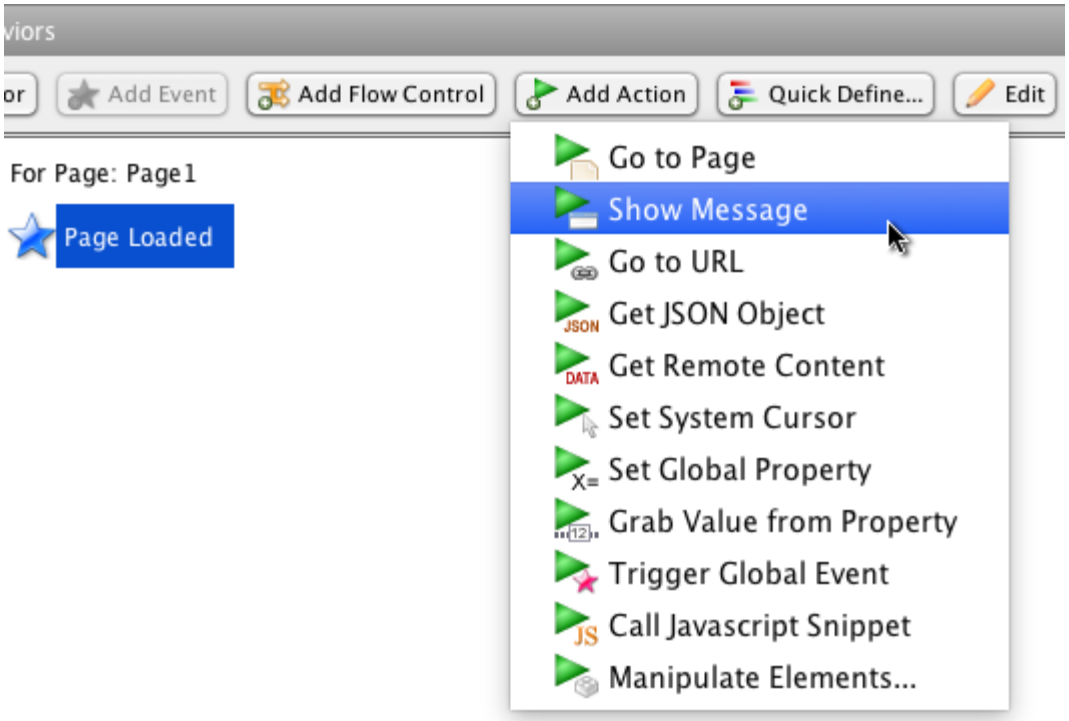
Add Flow Control or/and Action

Defining the behavior is just like creating flowchart. The event is the start point, and you can append flow controls (looping, branching, delay etc.) and actions under it.

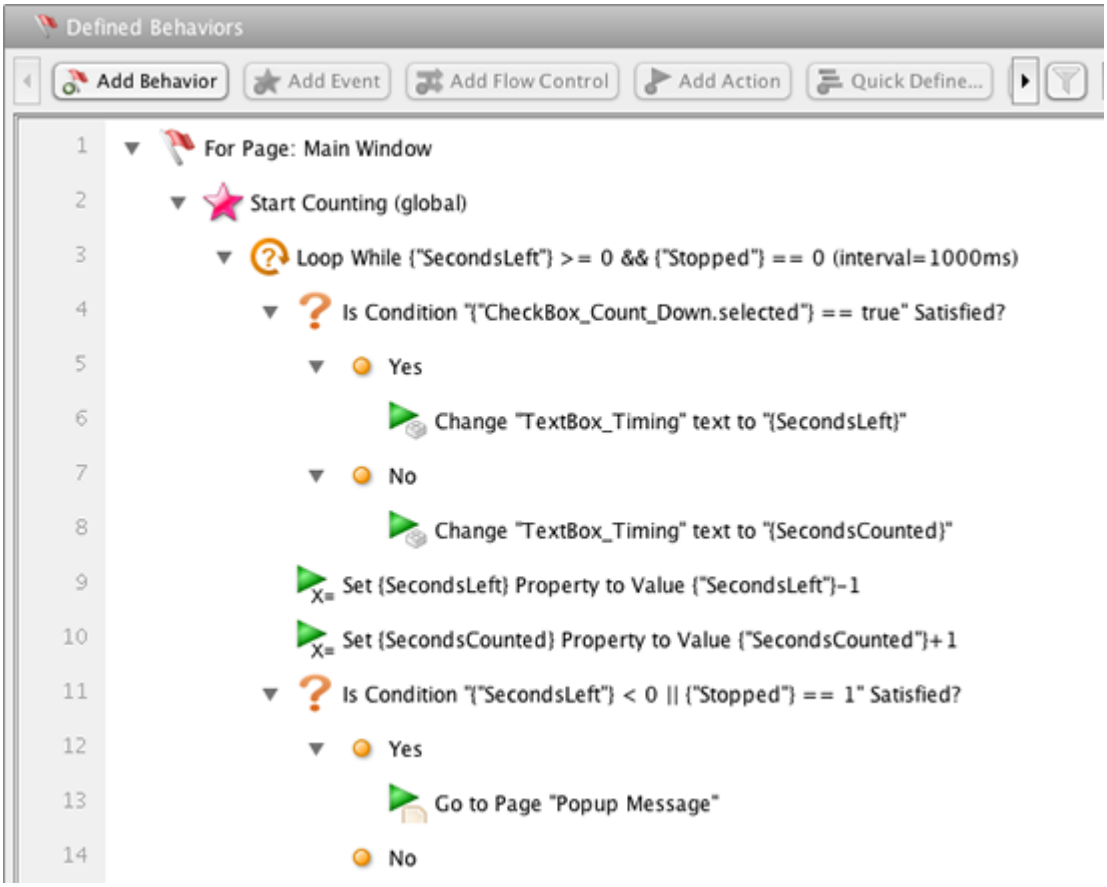
If you click the "Add Flow Control" button, all available flow controls will be listed in a popup menu (more details can be found in Flow Control Reference):



If you click the "Add Action" button, all available flow controls will be listed in a popup menu (more details can be found in [Action Reference]):



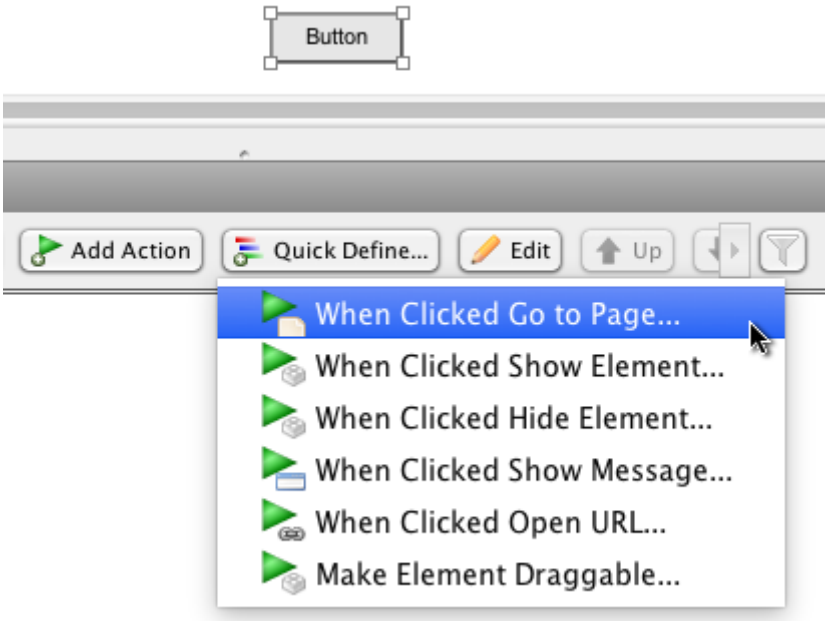
You can define "behavior tree" by adding Event, Flow Controls and Actions. Here is an example of finished behavior flow chart:



Quick Define Behavior

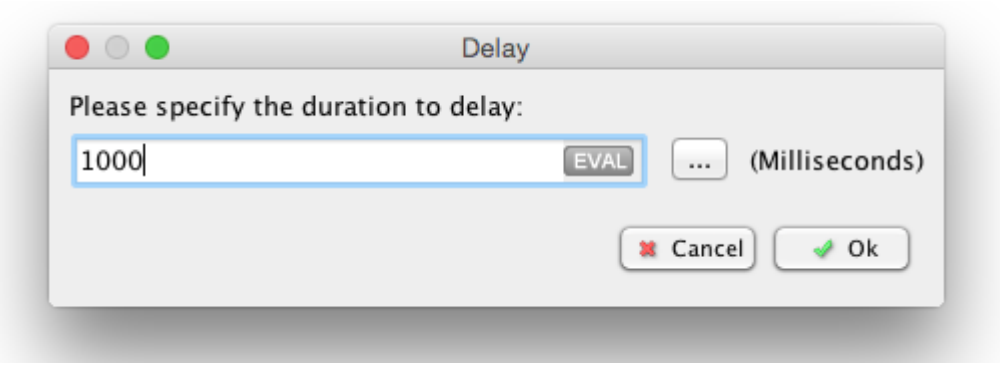
After selecting one or more elements in the working area, you will notice the "Quick Define..." button in behavior editor become enabled. Clicking on this button you will see a list of wizards that could quickly create a set of behavior for the selected element(s).

If needed, you will be asked to input some parameters, then the entire behavior will be generated automatically.



Edit Behavior Node

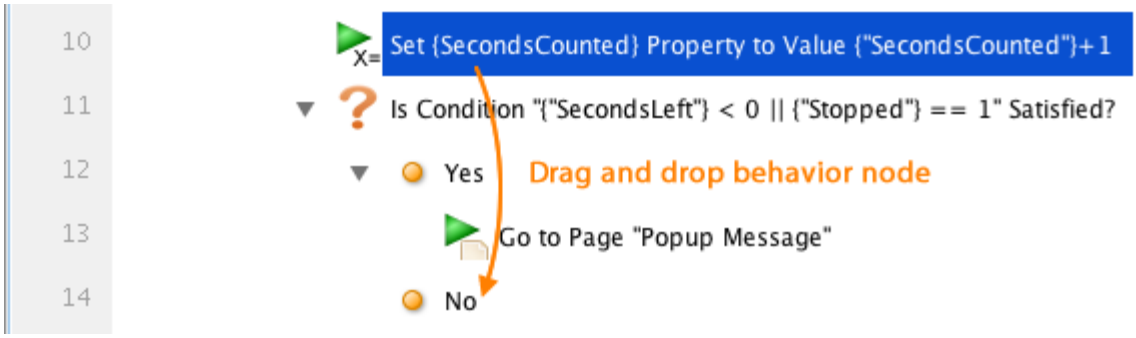
When adding behavior node (event, flow control or action), an editor window of corresponding node will show up and you can input some parameters for the node. Below is the editor for "Delay" flow control, you will see various editors when adding different nodes.



After adding the behavior node, you can double click or press the Edit button in toolbar to edit it.

Move Behavior Node

You can move the selected node via the Up and Down buttons in toolbar. It is also possible to move the selected node with drag and drop. If you have the Ctrl key pressed during the drag and drop, you will perform a copy of the selected node.



Remove Behavior Node

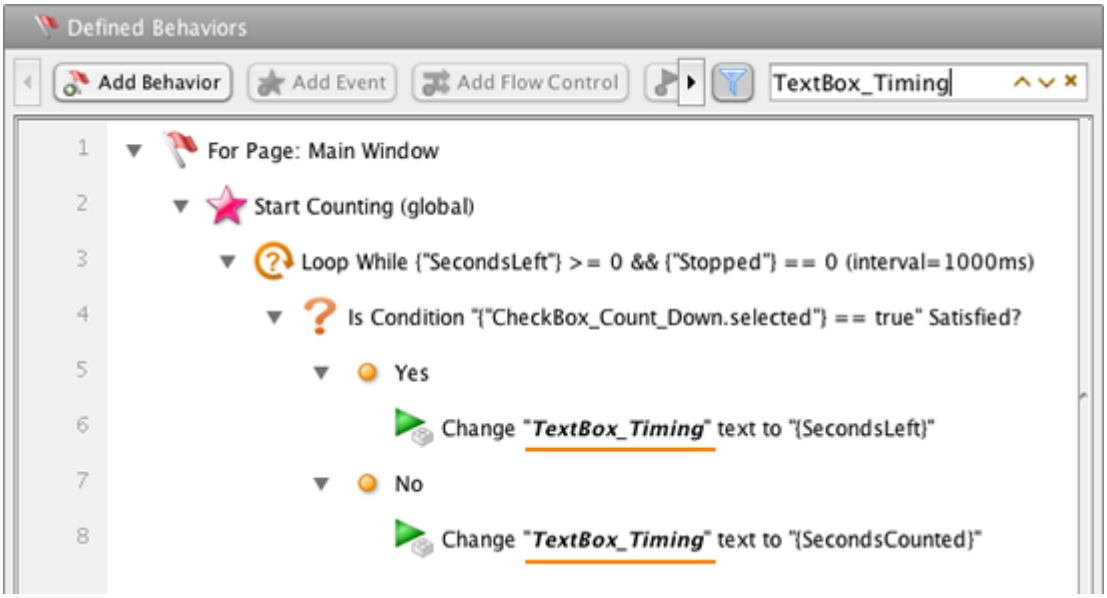
You can press Delete or Backspace key to remove the current selected behavior node. You can also click the Remove button to do the same.

Filter Behavior

On the right of the toolbar, you can see  or  button. It is a toggle button that control how to show defined behaviors in the behavior editor. If the button is selected (), only the behaviors for current selected elements (or current page) will be listed. If the button is unselected (), all behaviors will be listed.

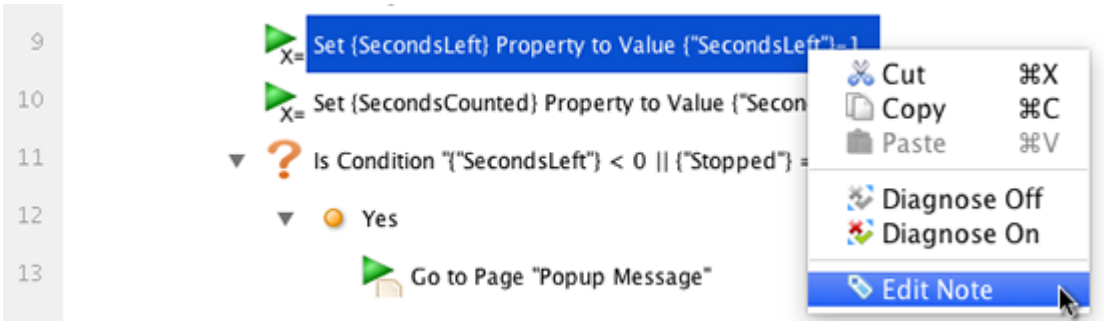
Remarks: when toggle button is selected ( ), behaviors for current page will not be listed until the element selection is cleared.

On the right of the toggle button, there is a filter box that accepts keyword to filter the listed behaviors further on. When filtering the behavior by keyword, the keyword will be highlighted with bold and italic style in the behavior view.



Use Context Menu

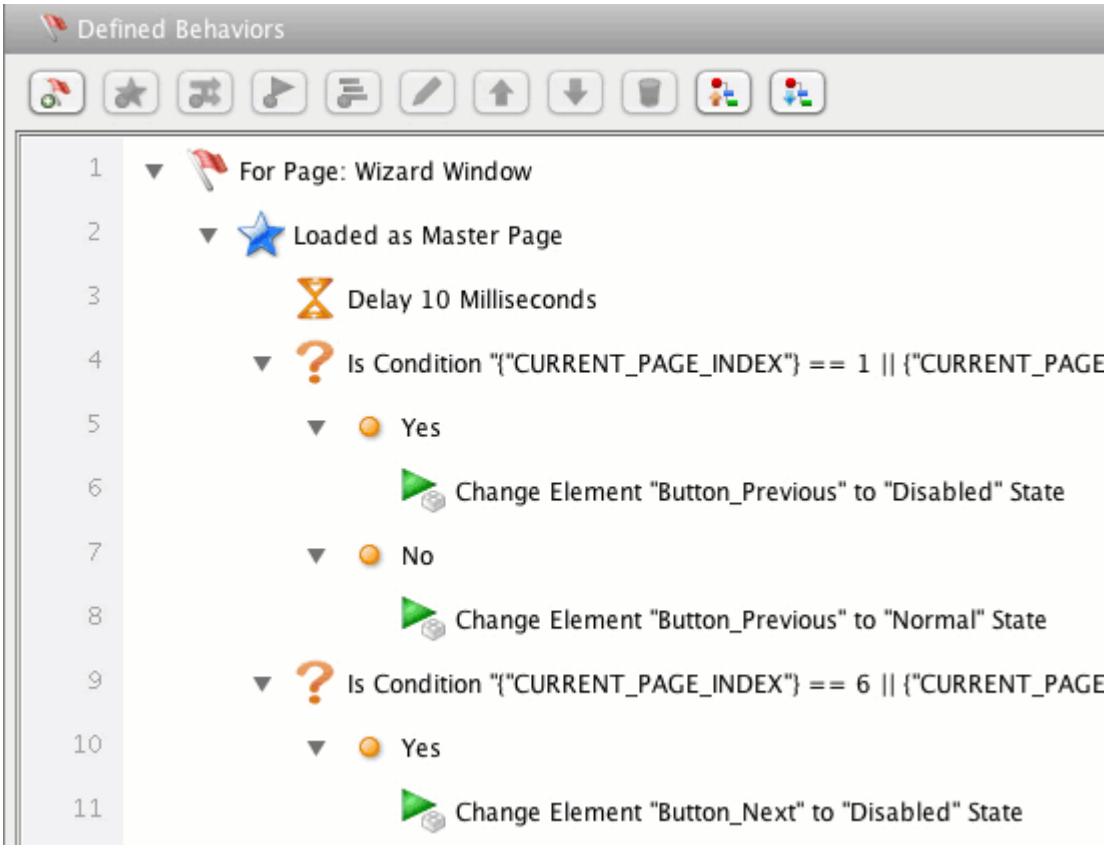
You can click right mouse button in the behavior view or note view to call out the context menu. In behavior view, the context menu allows you to cut, copy, paste and add note to the current selected behavior node. In the note view, you can add note for selected behavior node.



Also you can see the "Diagnose Off" and "Diagnose On" options. They are used to turn on/off the expression diagnosing on the node. If you define an expression and don't want ForeUI to check if it is correct, you can turn off the diagnose here.

Enable/Disable Behavior

You can disable/enable part of behaviors with a single click on the left of the row. Please see the demonstration below:



Once you disable an item, all its descendants will also get disabled (displayed in light green), and **they will not get exported to the HTML5 simulation**. If you doubt a part of behaviors cause problem, you could disable them and run simulation to see if the problem gets disappeared.

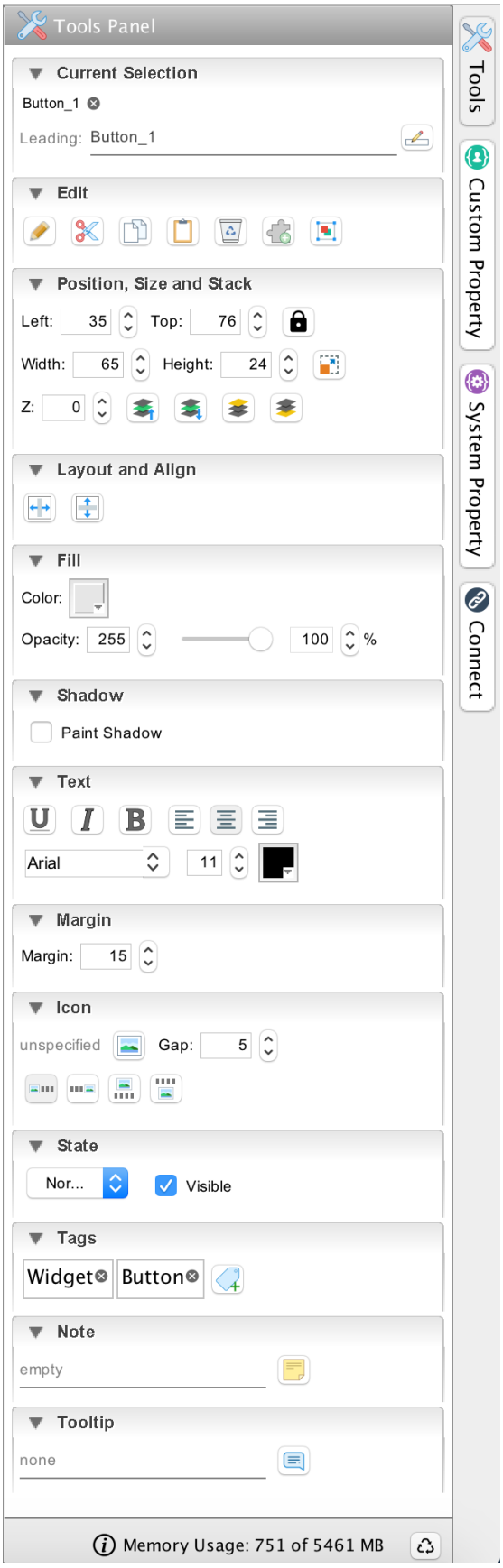
Clicking again on the same place can enable the item and its ancestors (if they get disabled already).

9.1 Tools Panel

Once you select one or more elements in the page, the tools panel will be displayed automatically. You can also show the tools panel by clicking the "Tools" button in right toolbar or pressing Ctrl + T (Command + T in Mac OS) even you have not select any element, in this case the tools panel will list some instructions of tool usage.

The actual tools displayed in the tools panel are changing according your current selected element(s). To learn about all available tools, please read Element Reference.

The figure below shows the tools panel and it has several categories:



The categories in the tools panel may vary when you select different element, or select multiple elements. Some categories are common, such as "Current Selection", "Edit", "Position, Size and Stack" etc., they are always there no matter what and how many elements you select. Some categories are dedicated for

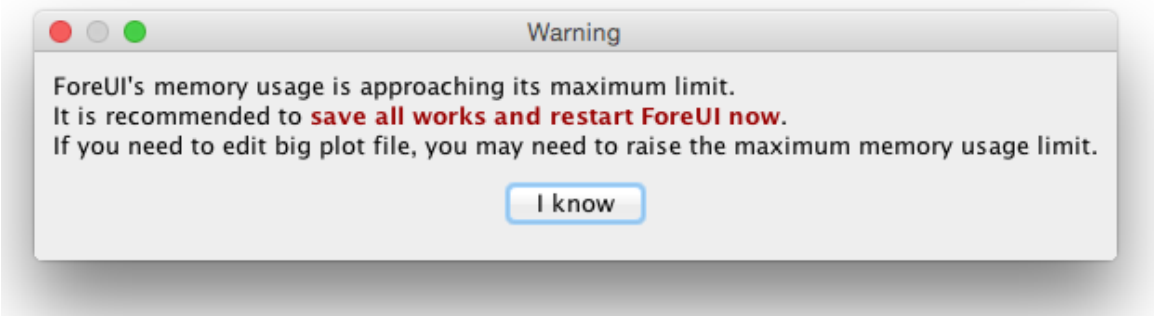
specific elements, such as the "Text" category is for elements that has text, such as Text Box, Button, Text Edit Box etc.

You can collapse specific category by clicking the ▼ icon on the left of category title, then its content will be hidden and it releases some space for other categories.

Memory Usage Information

At the bottom of the tools panel, there is a mini toolbar that displays the current memory usage information.

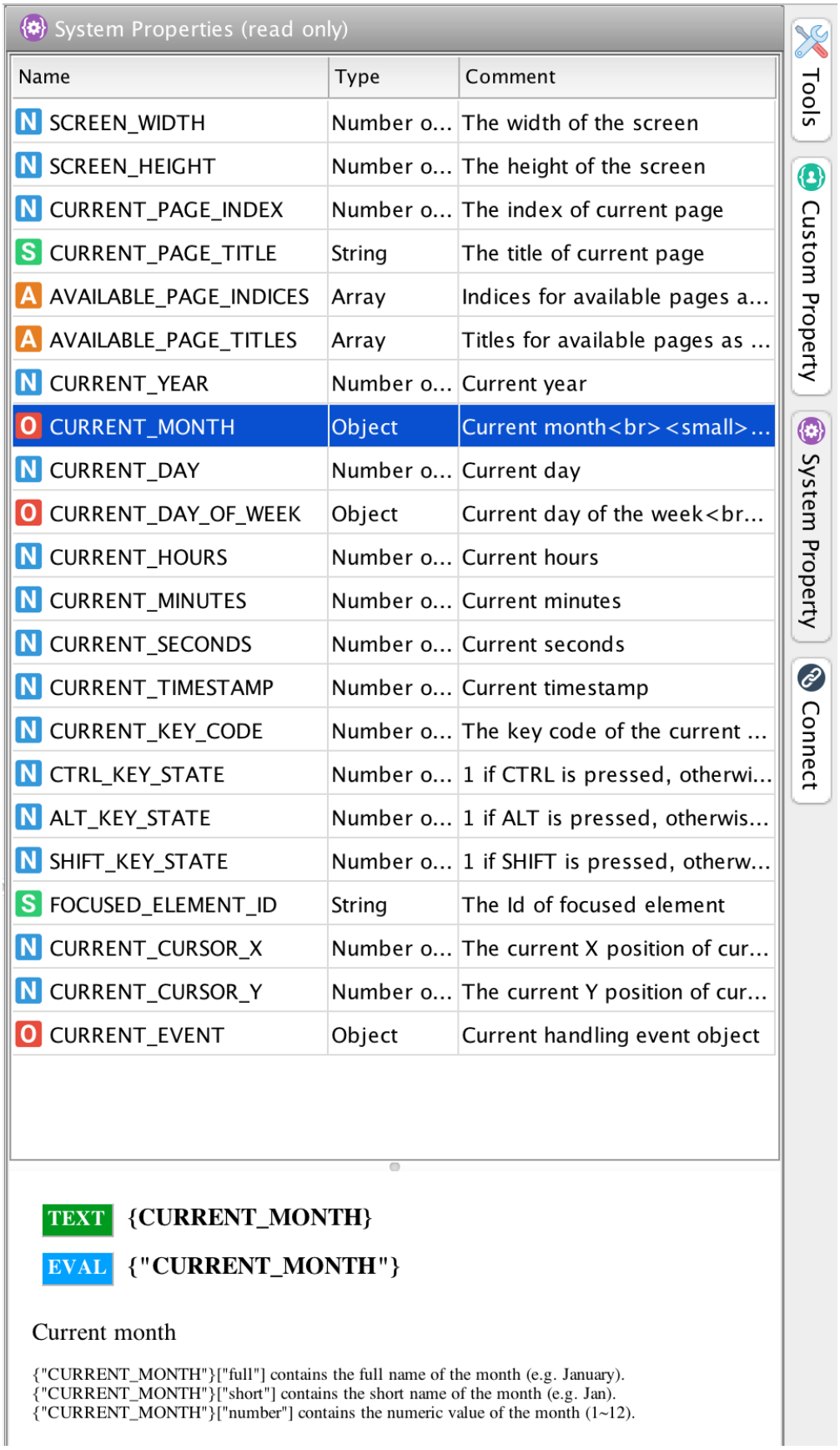
If ForeUI is running out of memory, try to click the button to see if it can release some memory. If that does not help, you may see a message



When you see this message, it is recommended to save all works and restart ForeUI.

10.1 System Property View

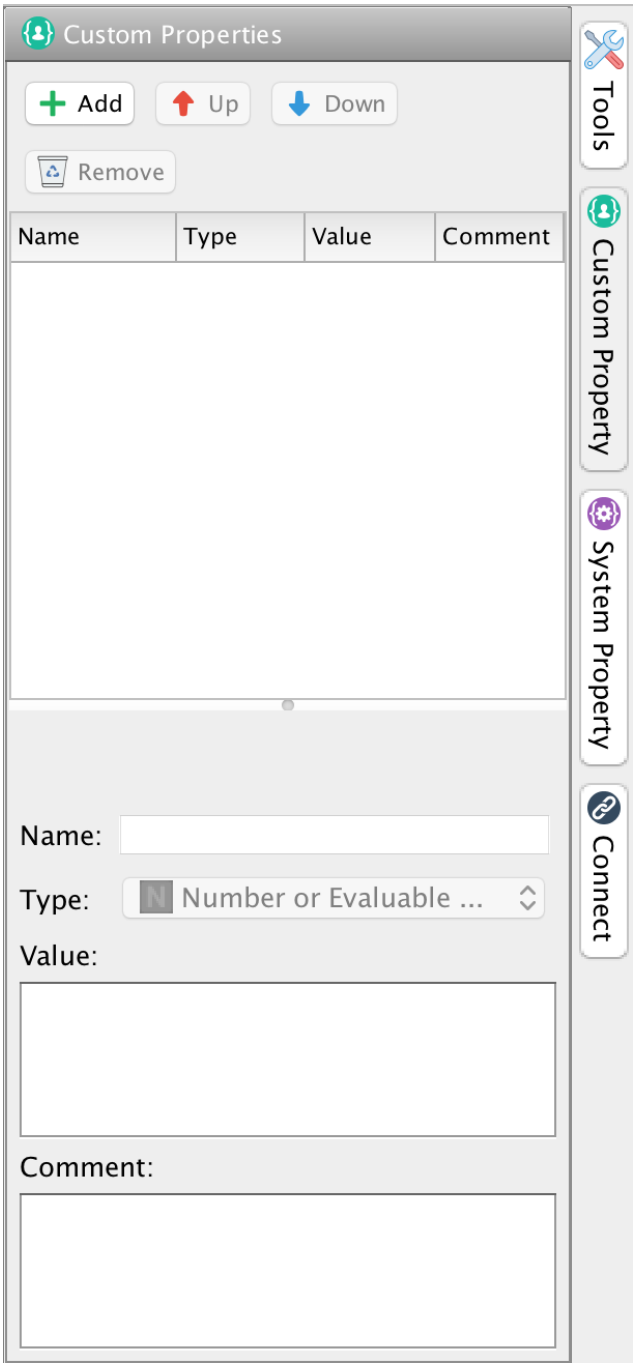
System property view lists all available system properties that could be used in expression. You can click the "System Property" button on the right toolbar, or press Ctrl+K (Command+K in Mac OS) to show/hide the system property view.



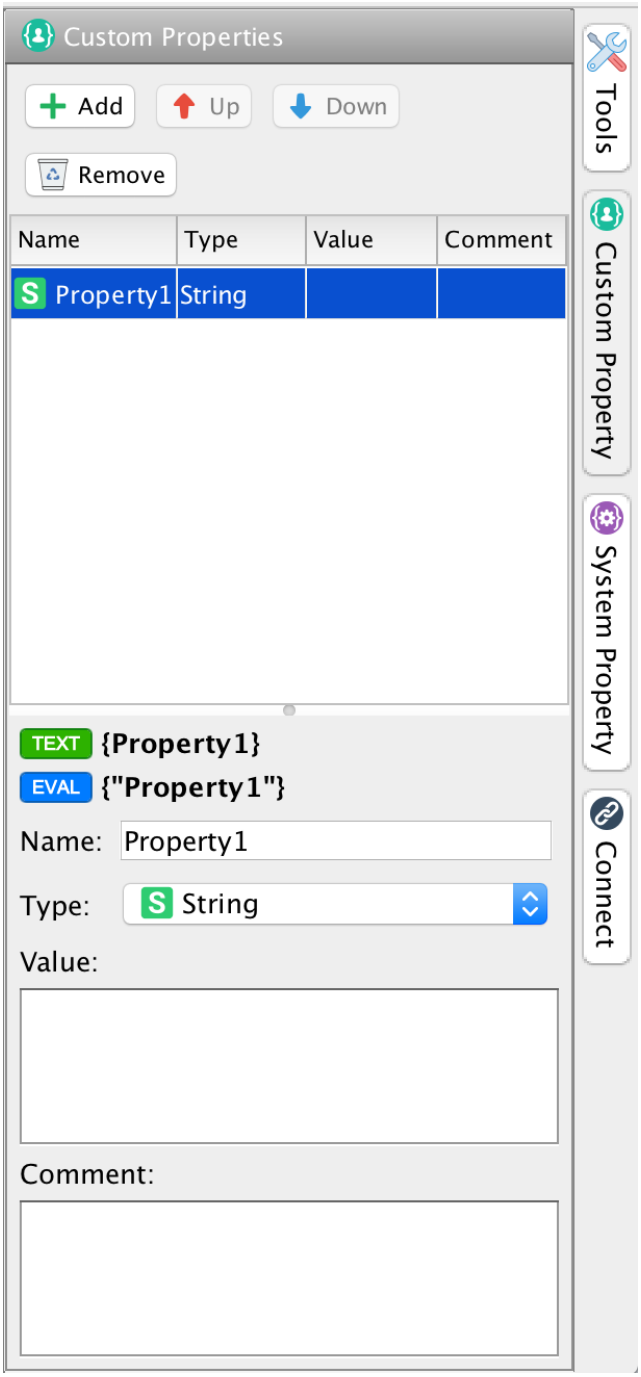
The table lists the name, type and description of every system properties. If you select a system property in the table, the sub view on the bottom will show how it looks within expression. The system property is read only, and you will not able to change its value via action.

11.1 Custom Property View

Custom property view will manage all custom properties (user defined properties) in the plot. You can click the "Custom Property" button on the right toolbar, or press Ctrl+J (Command+J in Mac OS) to show/hide the custom property view.



You can click the Add button to add a new custom property. You can then specify the details of the property in the bottom editor.

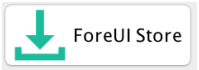


Please notice the property has different syntax in TEXT and EVAL parsing modes, and you can preview them here.

You can also choose the property type from available data types.

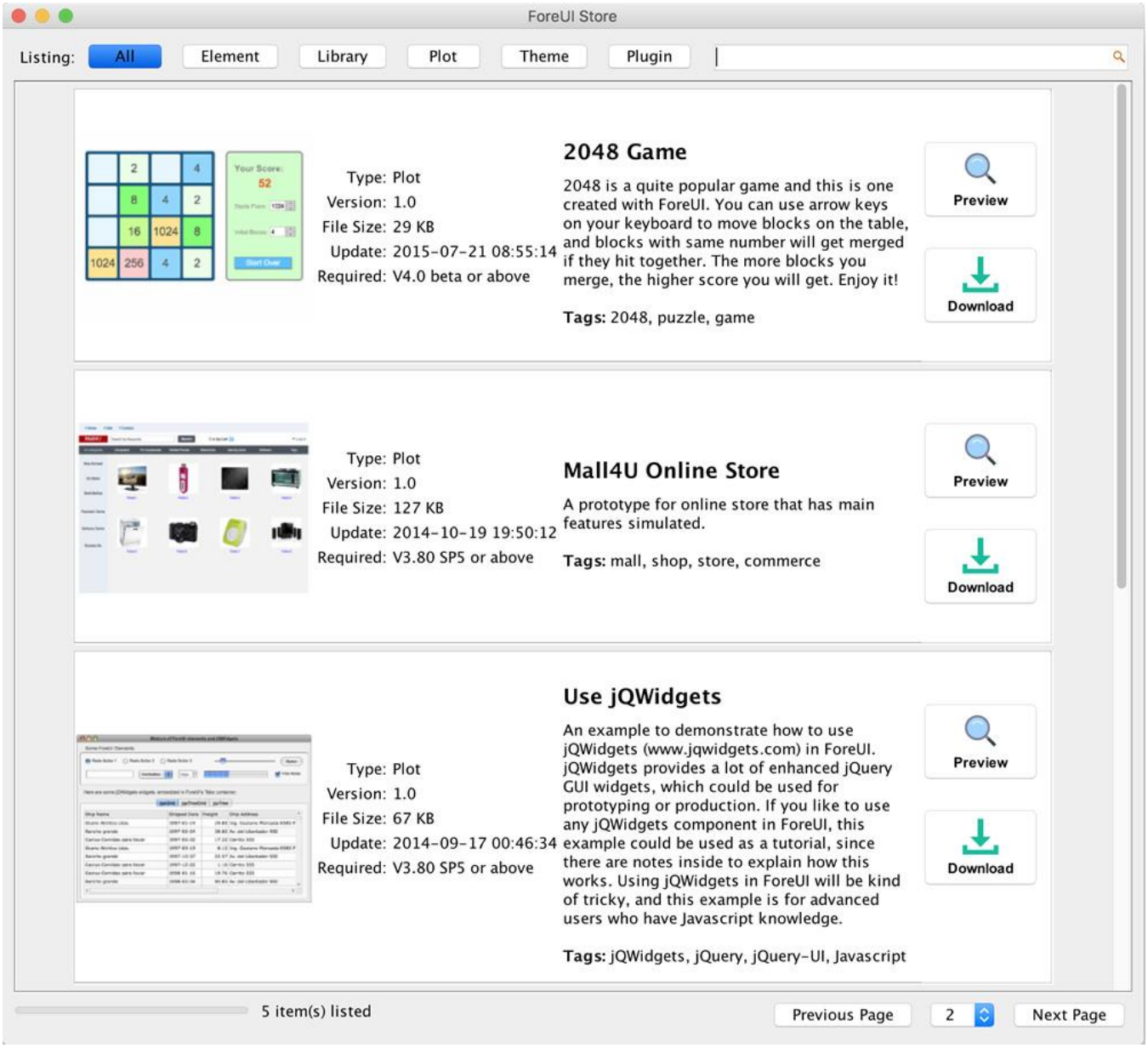
Once you define the custom property here, you will be able to insert them into expression.

12.1 ForeUI Store Window

ForeUI is integrated with its online resource store (http://www.foreui.com/store/). You will be able to download new resources (plot, custom element, library etc) from ForeUI store. You can open the ForeUI store window by clicking the 

button on welcome page, or by clicking the  button in the bottom toolbar of elements panel.

The ForeUI store window looks like this:



On top of the window, there are some filter buttons. You can list certain type of resources (all resources will be listed by default). On the top right corner, there is a filter box, inputting the tag can filter the resources as well.

For each listed resource, you can click the “Preview” button to run simulation in your default web browser, or you can click the “Download” button to download and deploy the resource (if needed).

You can click the "Previous Page" or "Next Page" button to switch the current page of resources. You can also use the drop-down list to choose the page to show.

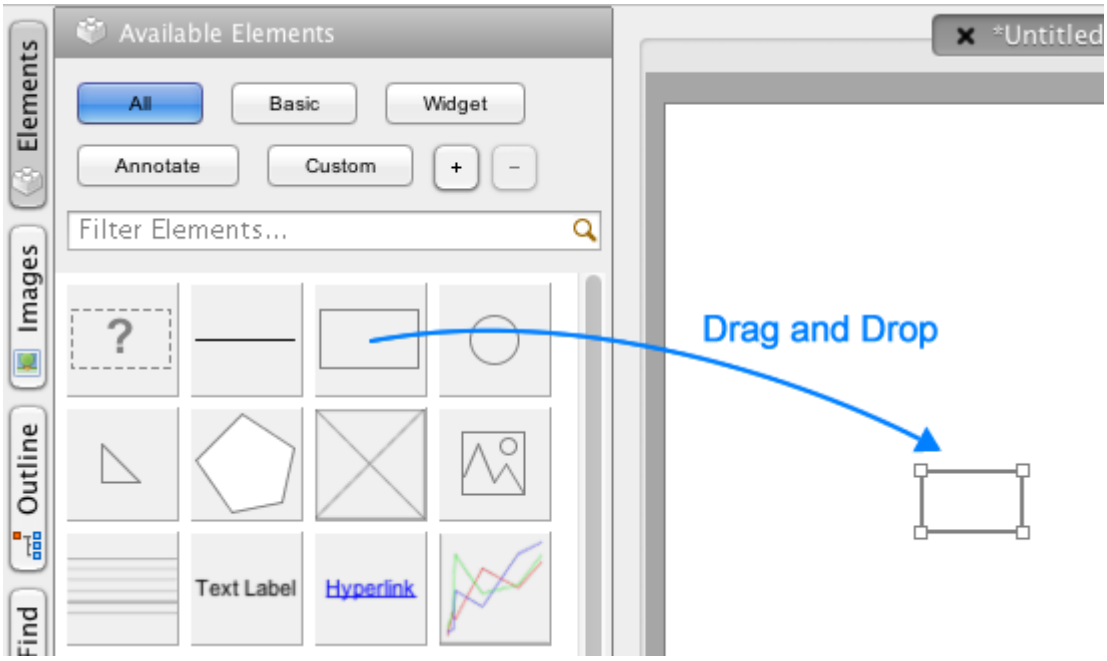
4. How-To…

1.1 Add, Move and Resize Element

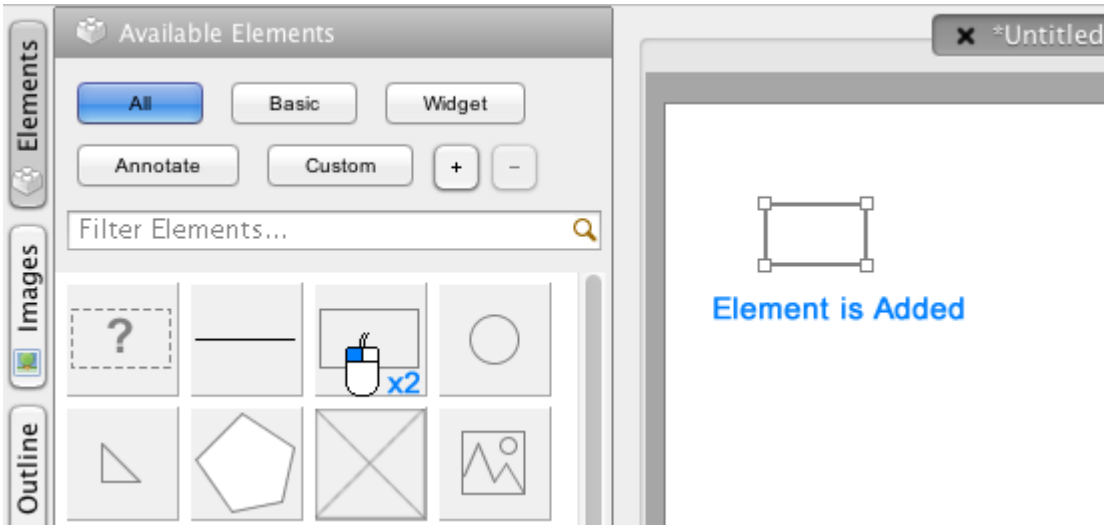
Add Element

To add an element into your page, you can try any of the three approaches.

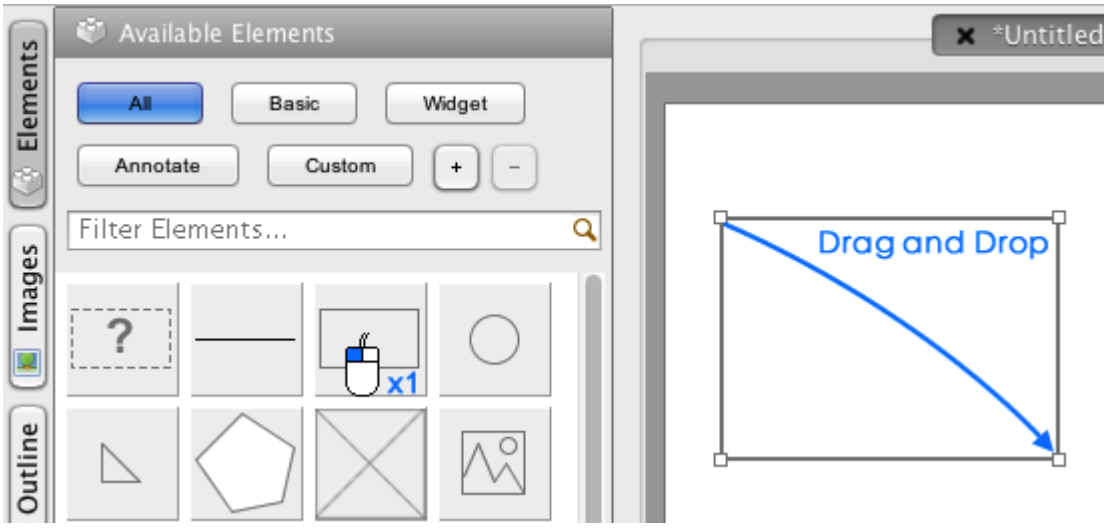
1. Drag element from elements panel into the working area. (specify element location only)



2. Double click element in the elements panel. (auto allocate location)

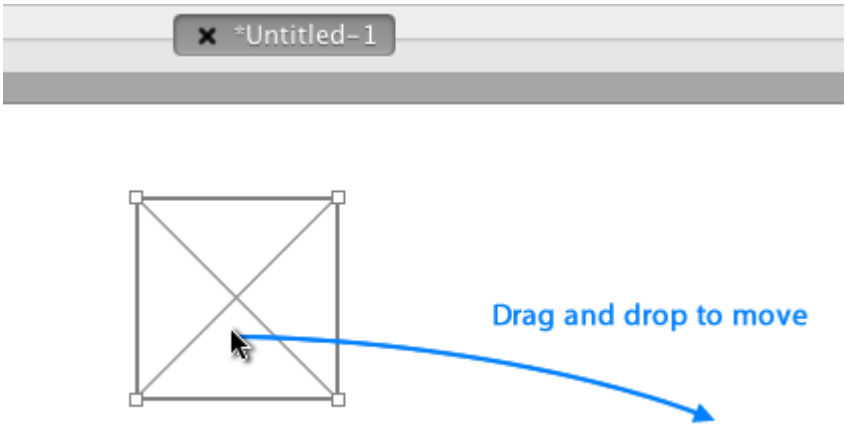


3. Single click element in the elements panel, then drag and drop in the working area. (specify element location and size)



Move Element

To move the element, you need to select it (by single left click) and then drag to move.

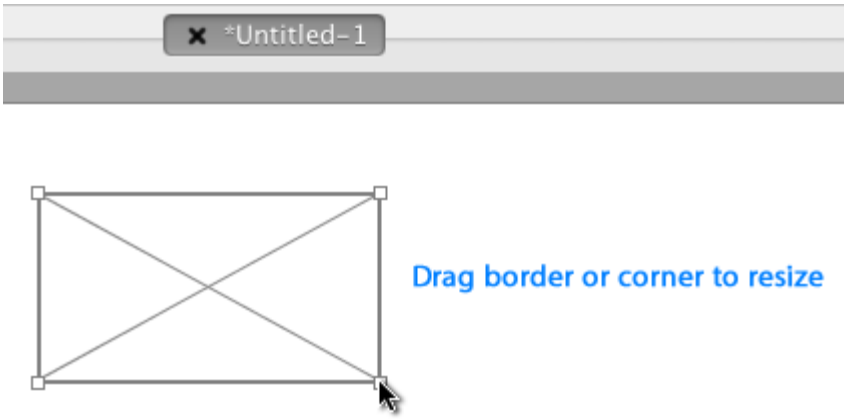


You can also move the selected element (pixel by pixel) with the arrow keys on keyboard. If you hold the SHIFT key and press the arrow keys, the selected element will be nudged for a longer distance, which can be configured under the "Edit" tab of settings window.

You can select multiple elements a time and then move them together.

Resize Element

To resize the element, you need to select it first, then drag its border or corner to resize.



You can also select multiple elements and then resize them together, in this case all selected elements will be stretched accordingly.

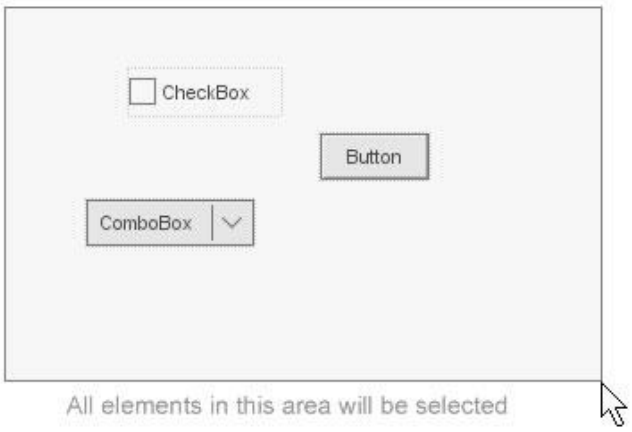
2.1 Select Multiple Elements

ForeUI supports multiple selections, which means you can select multiple elements a time and manipulate them together.

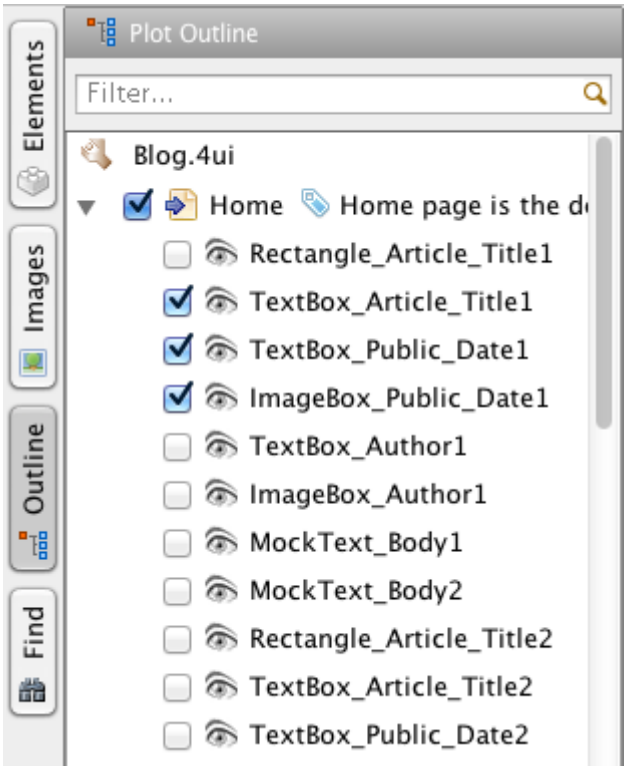
You can select an element by single clicking on it, and then you can press the **SHIFT or CTRL** key to turn on the "toggle mode" and click more elements to add/remove them into/from your selection.

You can also drag a rectangular bound that contains multiple elements; those elements will be selected together when you release the mouse.

Remarks: If you start dragging on an element, the default behavior is to move the element instead of selecting elements within the rectangle. In this case you could press the SHIFT key to force to drag selecting.

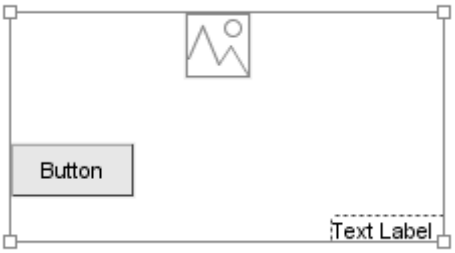


The outline view can also help you to pick elements that behind others. It lists the plot content in a tree structure,



Leading Selected Element

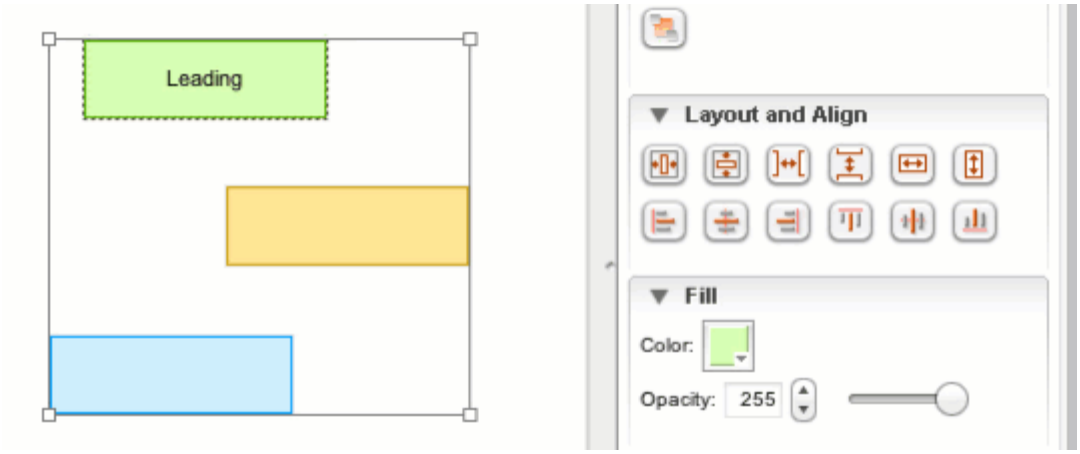
There is a “leading selected element” concept in ForeUI. When you select multiple elements a time, there will be an element that marked as “leading selected element”, which will have a dynamic dashed line border. We can change the leading selected element by clicking an element within the selection.



What is the leading selected element for? First, it allows you to tweak single element when you've selected multiple elements. You don't have to cancel the whole selection and select a single element for editing; you can just set it as the leading selected element, and then work on it in the tools panel. Secondly the leading selected element can work as the “reference substance” of some actions. Please take a look at the example below:

Example: Select three elements and then align left.

If the green box is the leading selected element, it will become the reference substance when performing the alignment.



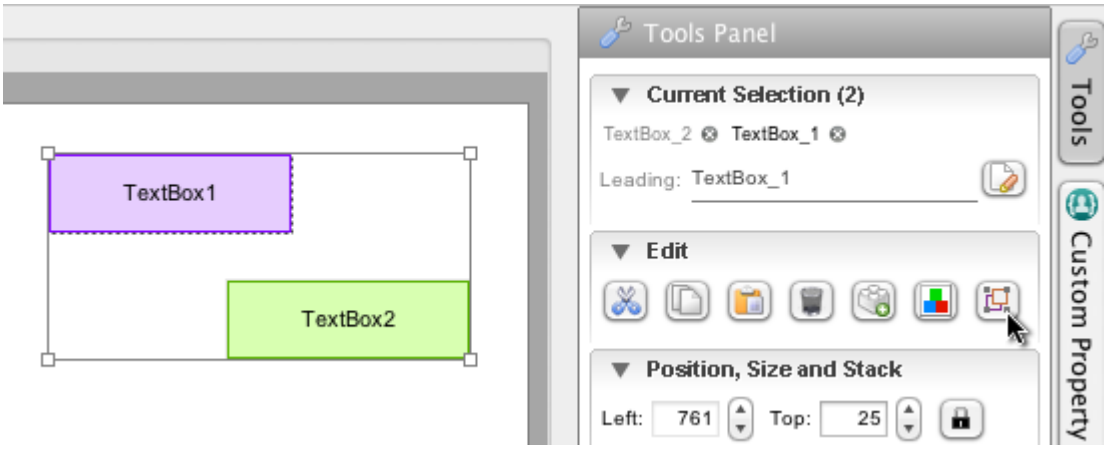
If the blue box becomes the leading selected element, the reference substance of alignment will change as well:

3.1 Conjoin Multiple Elements

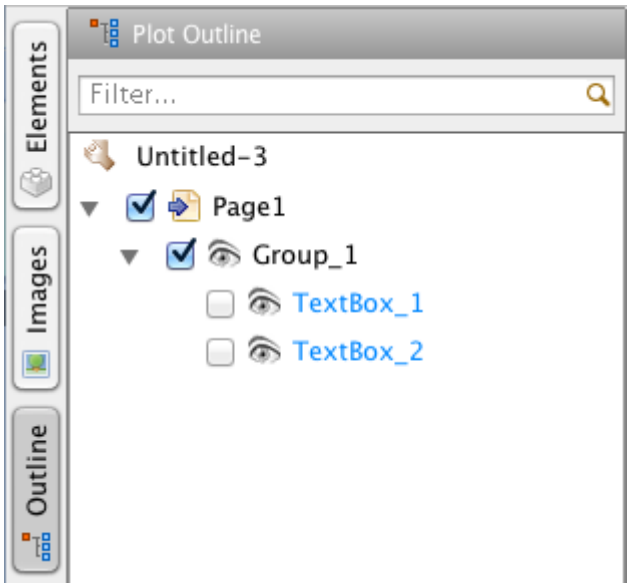
Sometimes you may need to conjoin multiple elements into one (so they can be easily moved together). ForeUI provides two approaches to achieve this: grouping and embedding.

Grouping

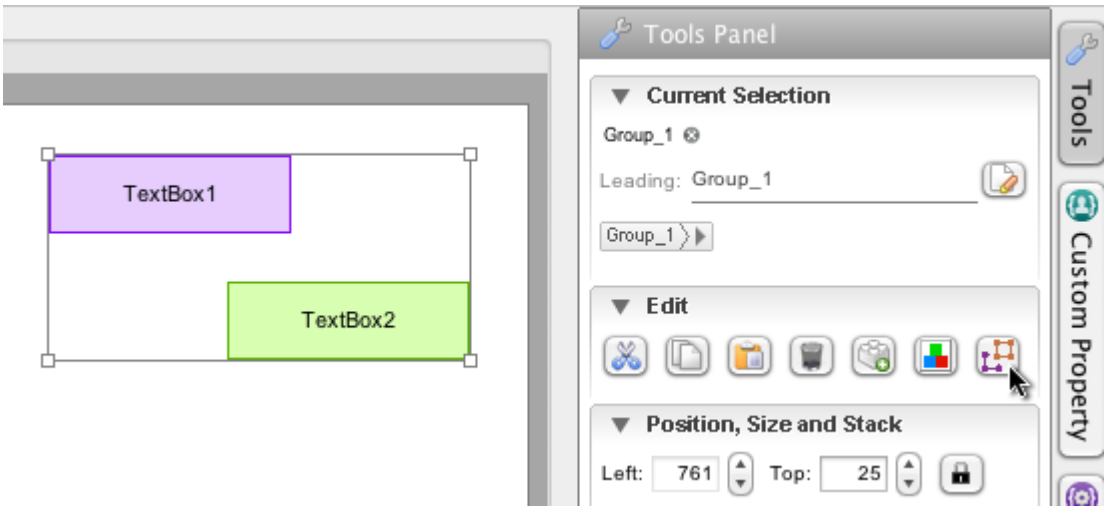
Grouping elements is very easy, just select them together and click the  button in the Tools Panel, then the elements will be conjoined into a single Group element.



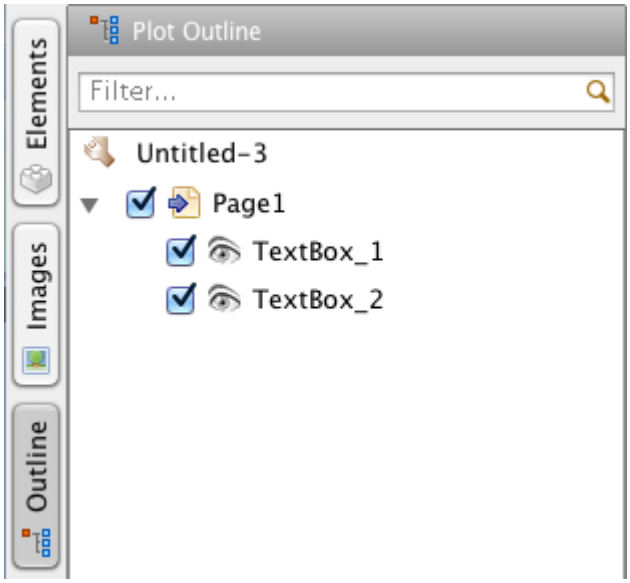
The member elements within the group can be reviewed in the outline view. The member elements are displayed in light blue color:



You can ungroup the group by clicking the button after selecting the group:



After ungrouping, the Group element is removed and the two member elements are restored.



Embedding

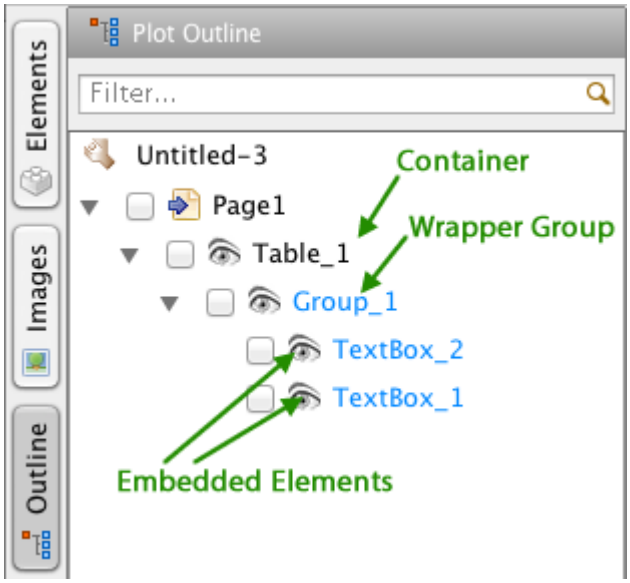
Some elements can work as containers and they allow you to embed other elements into them. Actually the Group element is a special type of container: all group members are "embedded" into the group.

These elements can be used as containers, and they can embed each other and build a nested structure.

- Group
- Table
- Scrollable Container
- Tabs
- Vertical Tabs
- Tree
- Window
- Accordion

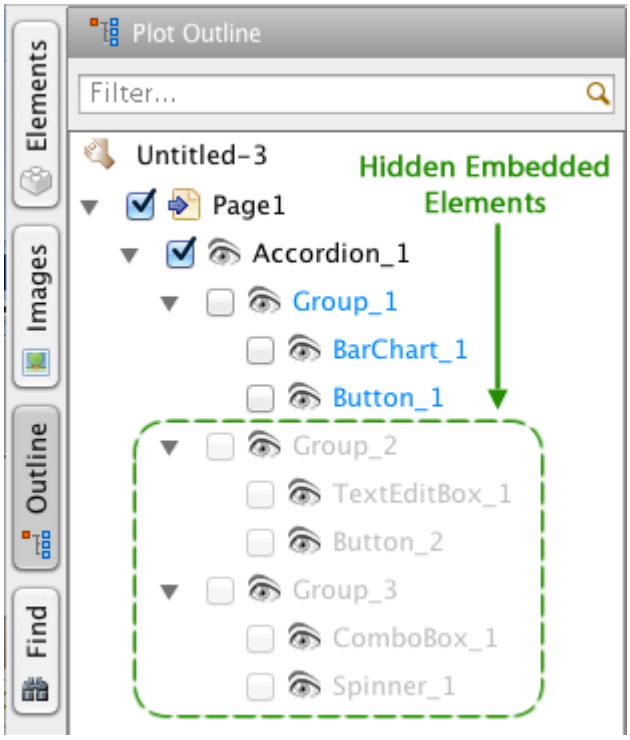
If you embed element A into B, then embed B into C, you will get a nested structure.

Remarks: The elements above (except Group) will automatically create a group to wrap its embedded elements, and you can clearly see that in the outline view.



Some containers have the ability to show/hide part of its embedded elements, such as:

- **Tabs & Vertical Tabs:** will hide the embedded elements in non-active tabs.
- **Tree:** will hide the embedded elements in collapsed nodes.
- **Accordion:** will hide the embedded elements in collapsed sections.



ClipArt

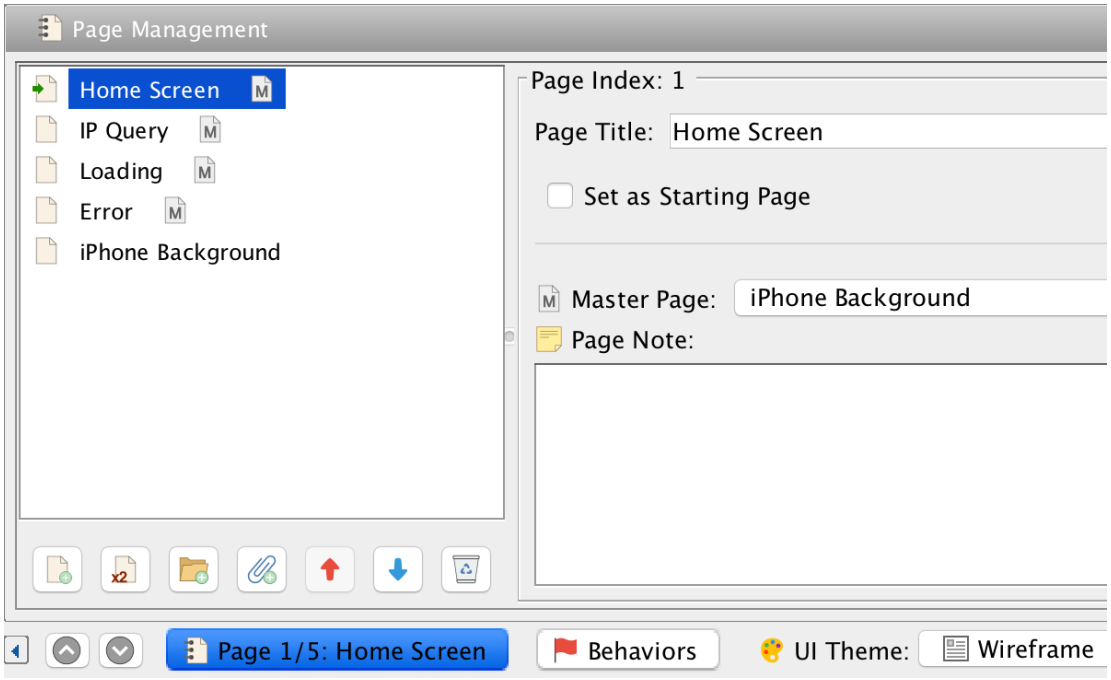
Since ForeUI V3.80, you can also conjoin multiple elements with ClipArt. The way to create a ClipArt is just like the way you create a Group; the difference is that you need to click the  button in the Tools Panel.

Please keep in mind that, all elements in ClipArt will become static images in the HTML5 simulation. So if you wish an element to be interactive, please make sure not to put it into a ClipArt.

4.1 Manage Pages

When you create a new mockup or prototype, it has one page by default. You can add more pages via the page manage view. You can click the page manage button at the bottom toolbar to open the page manage view:

The page management view looks like this:



The page manage view is separated as two sub-views (left and right). The tree view on the left allow you to create new page or folder, duplicate the selected page, move up / move down / delete the selected page (or folder). You can also drag and drop page or folder to make it nested in another page / folder. The panel on the right shows the detailed information about the select page / folder. You can change the title, assign the master page or input the page note for the selected page.

Create New Page

To create new page, just click the  button in the bottom left toolbar. The newly created page will be titled as "Page?" where ? is a number. You can rename the page title on the right view.

After creating a new page, the current editing page will be switched to the newly created page automatically.

Switch Current Editing Page

You can switch the current editing page by clicking the items on the left tree view, or click the  and  buttons on the bottom toolbar to switch the current editing page.

Duplicate Page

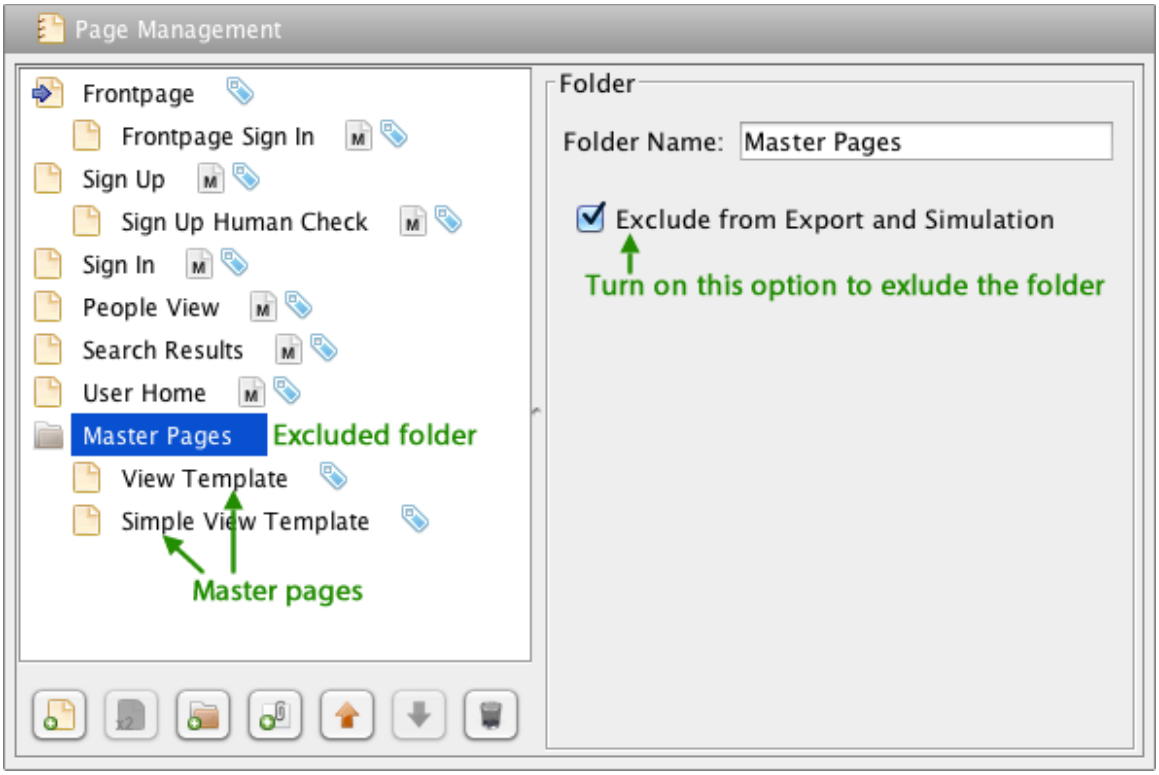
You can duplicate the current page by clicking the  button. All content within the page will be duplicated and the new page will be selected automatically.

Create Folder

You can create folder by clicking the  button. The newly created folder will be named as "Folder n" where n is a number. You can rename the folder on the right view.

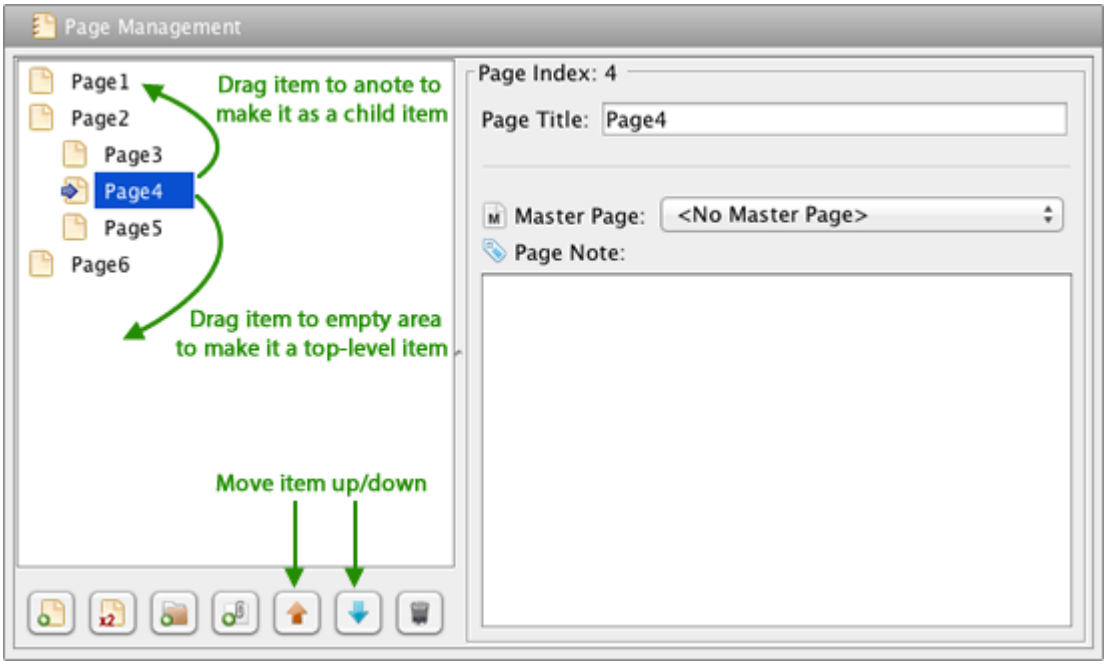
Here you can also set the folder to be excluded from the export and simulation. Once this option is selected, all pages under this folder (and its child node) will not be included in the image/PDF/HTML5 exporting. Such a folder will be useful to store master pages or other pages that do not want to be exported.

Remarks: Once a folder is excluded, all its child folders will be excluded automatically.



Organize Page Tree Structure

You can drag and drop the item within the tree view and place it under other page or folder, thus you can get a "site map". You can also move the selected item upper or lower by clicking the  and  buttons.



Delete Page / Folder

You can delete the current selected page or folder by clicking the  button.

Add Attachment to Plot

You can click  button to add files into the plot as attachment, which will be exported to HTML5 simulation as well.

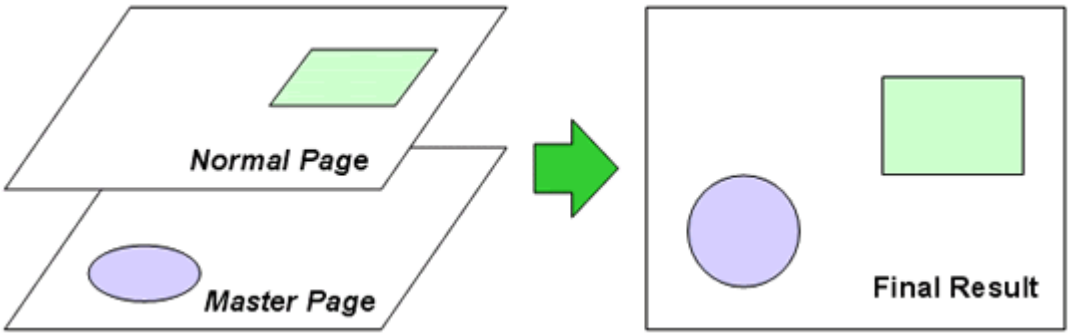
Specify Master Page

You can specify the master page for current editing page in the right view. One page can have only one master page, but the master page itself can have its own master page... So multilevel content inherit will be possible. To lean more details about master page, please read "Use Master Page" section.

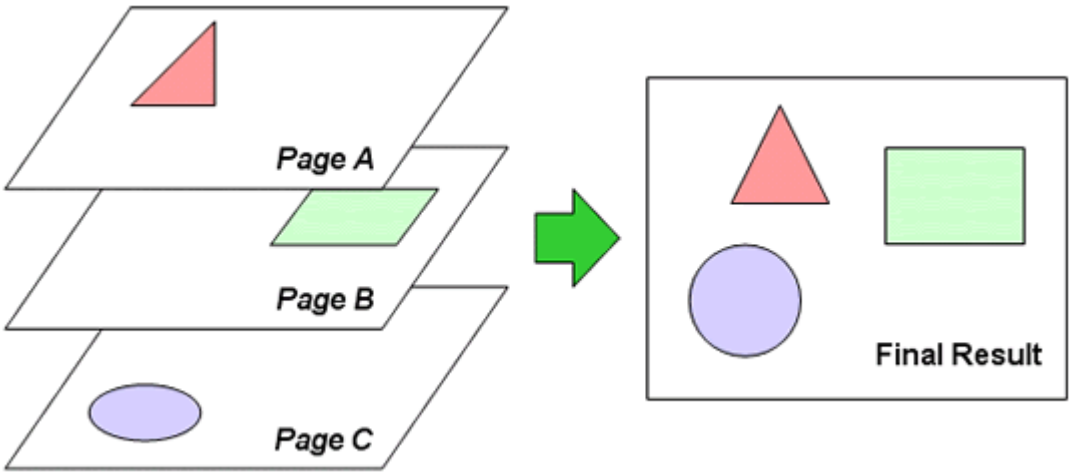
5.1 Use Master Page

What is Master Page?

The master page is a very useful feature since it can simplify your prototype / wireframe a lot. Please take a look at the figure below, the master page works like a "background" layer, once you assign a master page to a normal page, the content on master page will be merged into the normal page. What's more, a master page can be shared by multiple normal pages. That means you can put the common part of the prototype into the master page, thus it can be shared by several pages without copy and paste.



Here you can see two concepts: "Normal Page" and "Master Page". However they have no actual difference, they are just pages. In ForeUI, each page can be used as a master page (by other pages), you can assign any page as the master page for your editing page. So you can assign page B as the master page for page A, and then assign page C as the master page for page B, that will form a hierarchy structure like this:

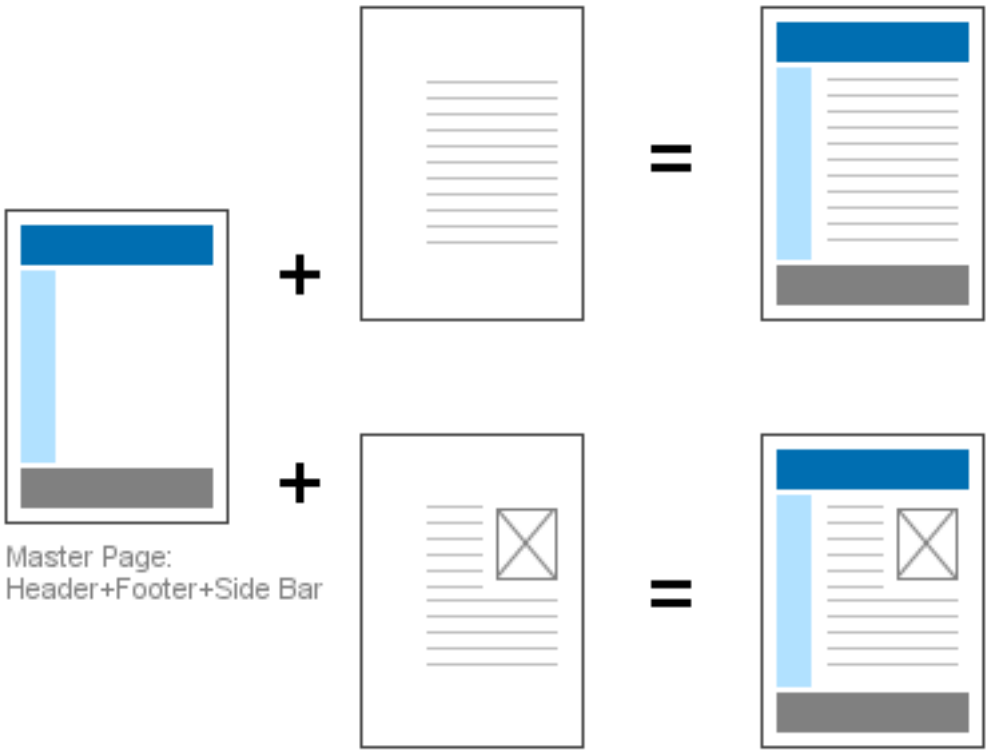


When export page A or run the simulation of page A, you will see the final result: the contents in both page B and page C are all merged into page A. So you can imagine how helpful will this feature be.

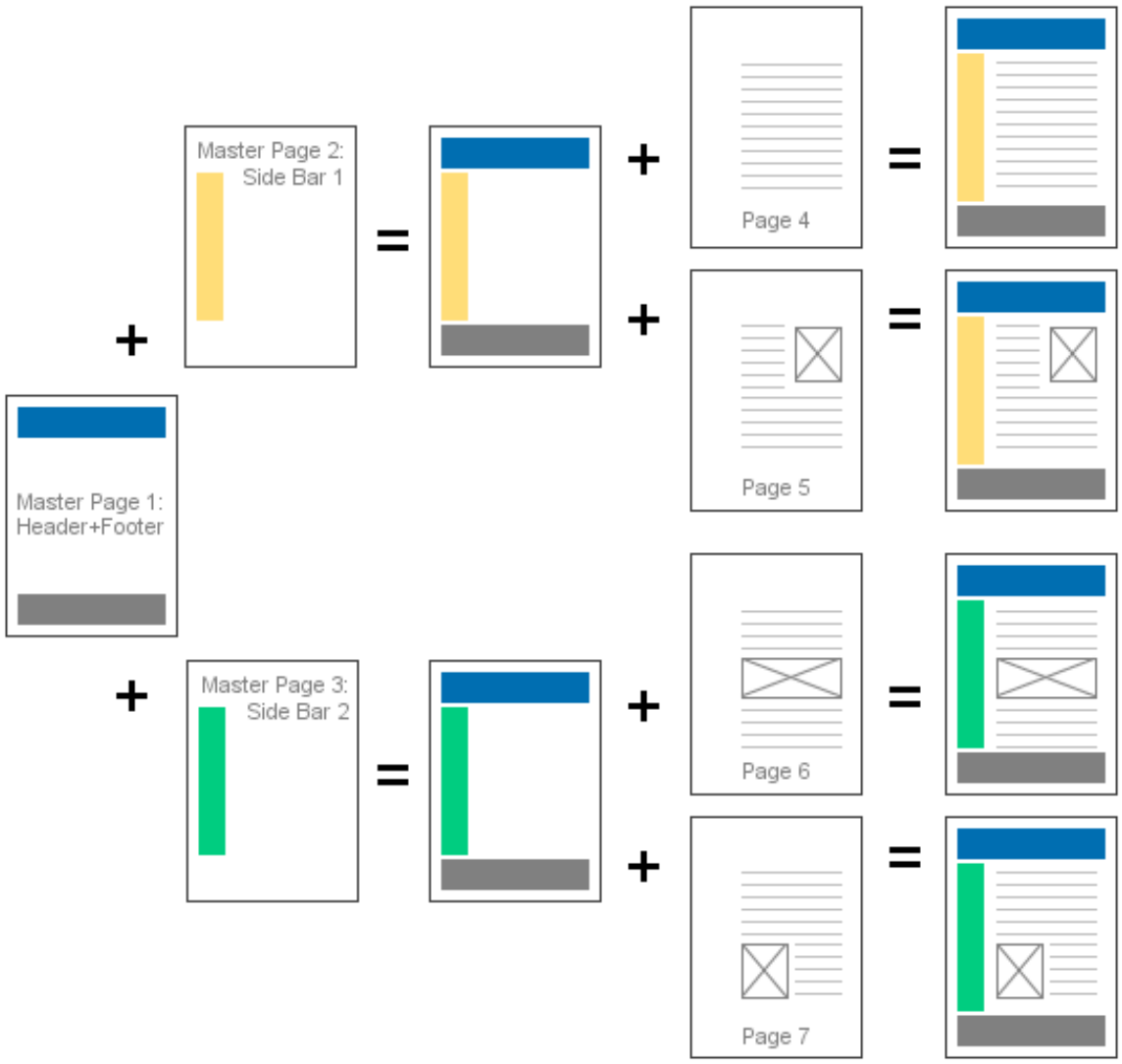
Why Use Master Page?

In the example above, although the three elements are on different pages, they can still interact with each other via predefined actions. So there is no drawback to move the common part of multiple pages to a shared master page, this will significantly simplify your mockup or prototype and will avoid a lot of content duplication between pages.

A very typical use case for master page is the header and footer of web site, each page of the web site has the same header and footer. You don't have to copy header and footer and paste them everywhere, you can just put the footer and header content on a single page, then assign this page as the master page for all other pages, that's quite simple and works well. Below is a simple example:

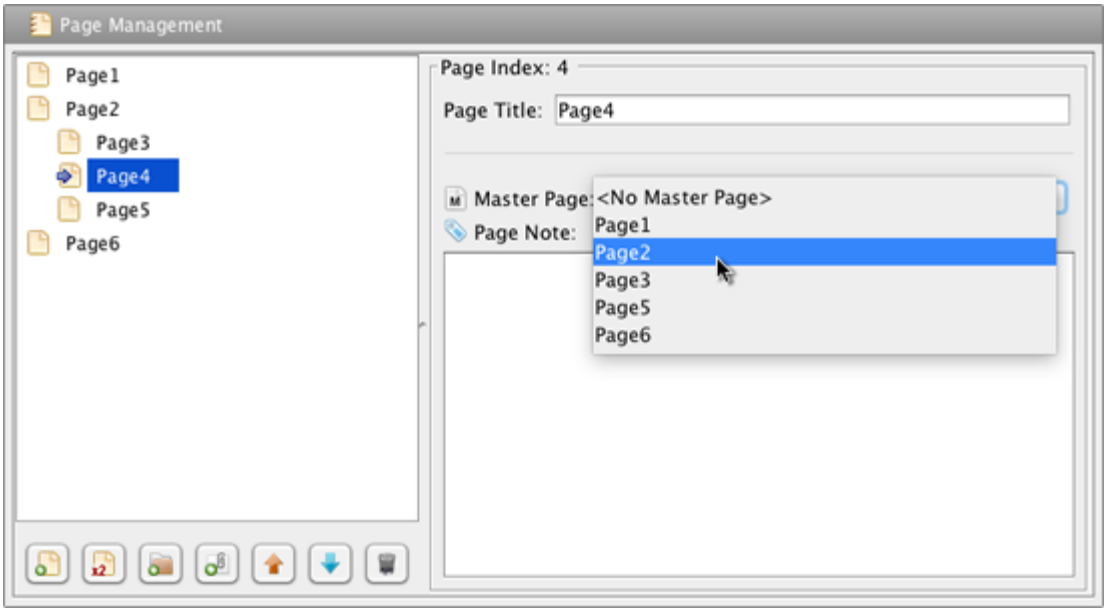


Here is a complex example with multi-level master pages.



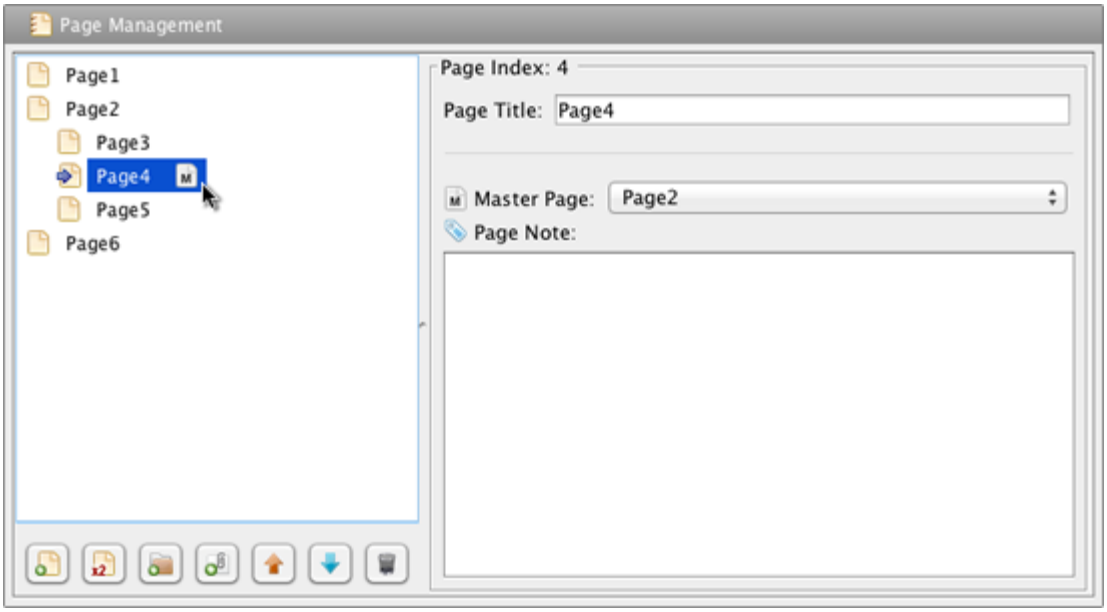
Assign Master Page

You can assign the master page of current editing page in the page manage view. Just choose a page in the "Master Page" drop down list on the right view.



Remarks: the current editing page will not exist in the drop down list, since a page cannot use itself as a master page.

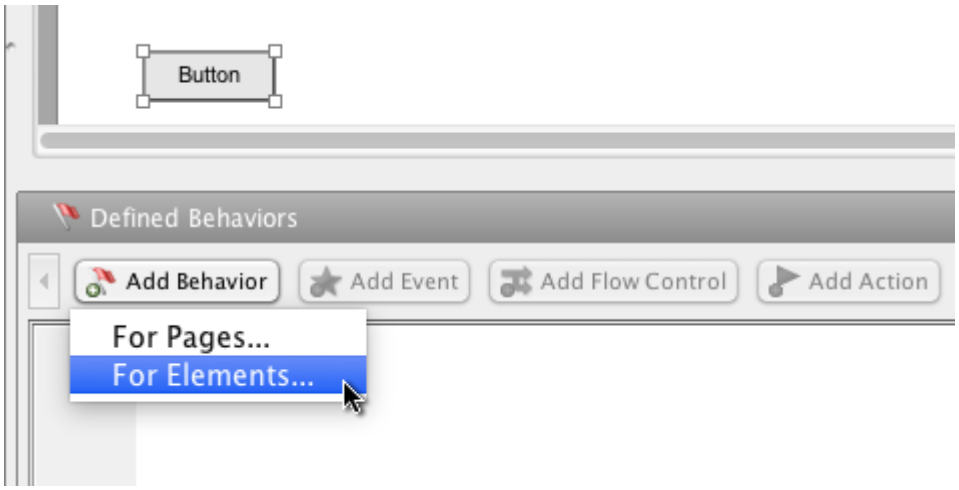
After assigning the master page, the item for current editing page will have a  icon on its right, like this:



6.1 Define Element's Behavior

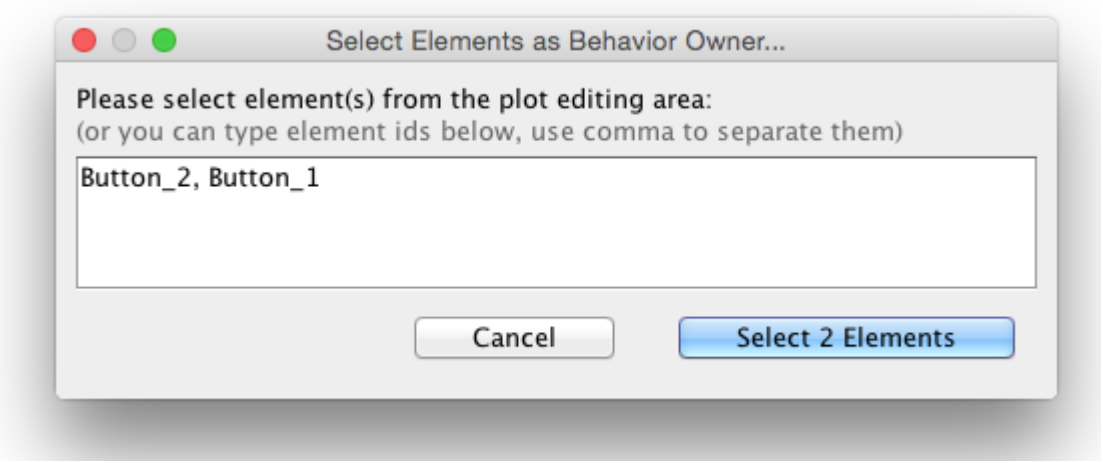
In ForeUI, behavior is defined as one or more event handlers, which consist of Event, Flow Control and Action. You can define the behavior for element by assigning the element as the owner of the event handlers. To do so, you can follow these steps:

1. Open the behavior editor view.
2. Click the "Add Behavior" button in the toolbar and then select "For Elements..."
3. Select the element(s). You can select multiple elements if you want to define behavior for them together.

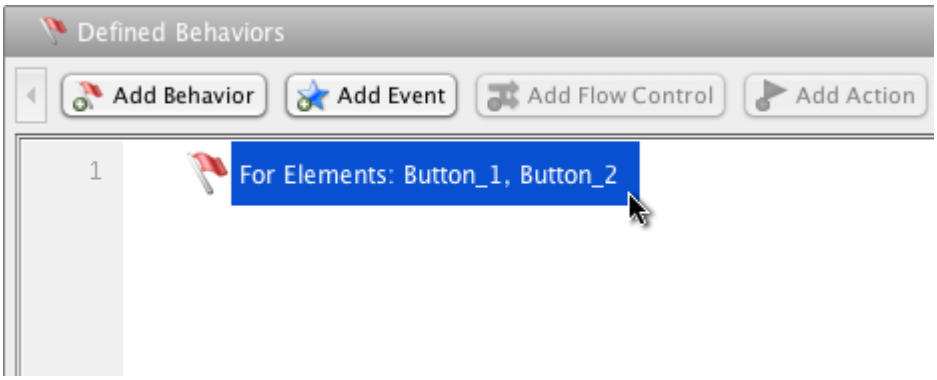


The step 1 and step 2 can also be achieved by pressing **Ctrl+D** (Command + D in Mac OS X).

In step 3, you will see the element chooser window:



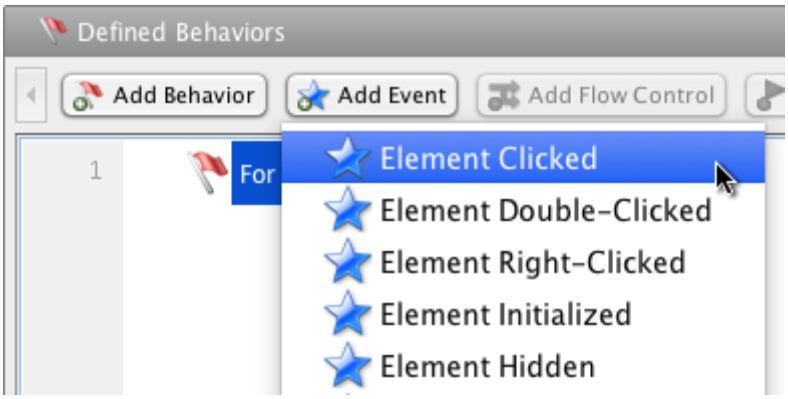
If you have any element selected before opening the element chooser window, the selection will be listed by default. You can also select element first, and then press Ctrl+D (or do step 1 and 2), you can finish step 3 by directly click the "Select n Elements" button (where n is a number). Then you will see the behavior owner in the behavior editor:



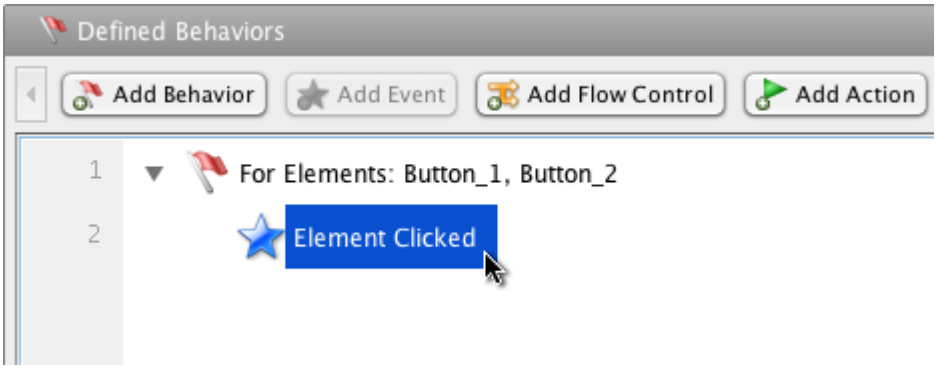
After that you can define the event handler by adding event, flow control or/and action.

Add Event

The "Add Event" button is enabled when you select the behavior (node with red flag icon). That means you can continue defining the behavior by adding events into it. All available events (according to owner type, you can find all available events in Events Reference) will be listed in a popup menu when you click the "Add Event" button.



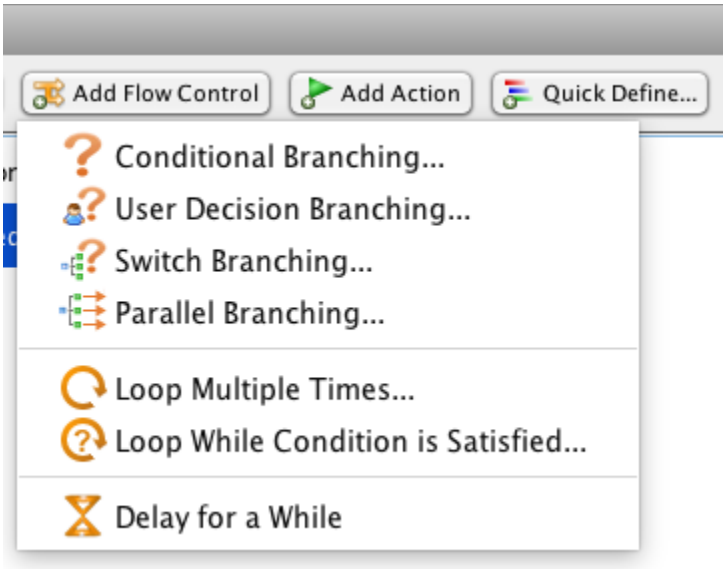
.After adding the event, it will be selected by default and the "Add Flow Control" and "Add Action" buttons will be enabled now.



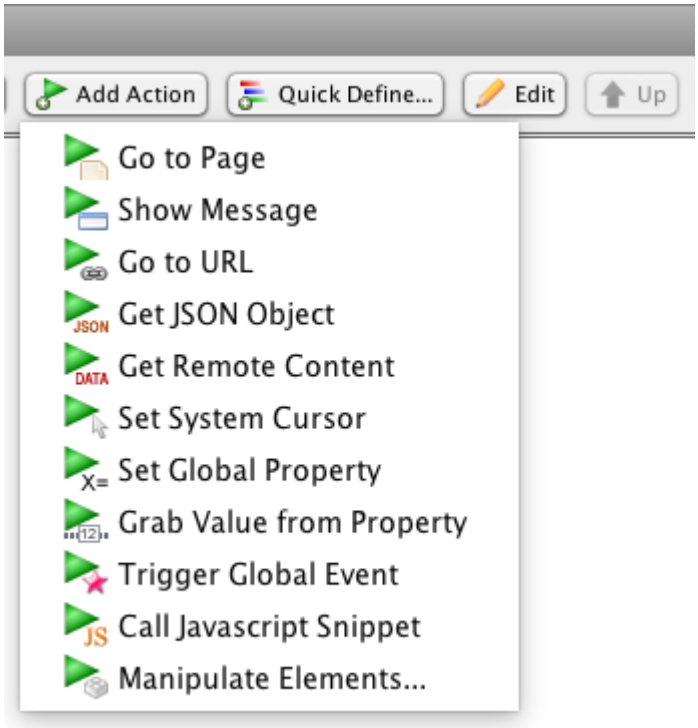
Add Flow Control or/and Action

Defining the behavior is quite similar with defining flowchart. The selected event is the start point, and you can append flow controls (looping, branching, delay etc.) and actions under it.

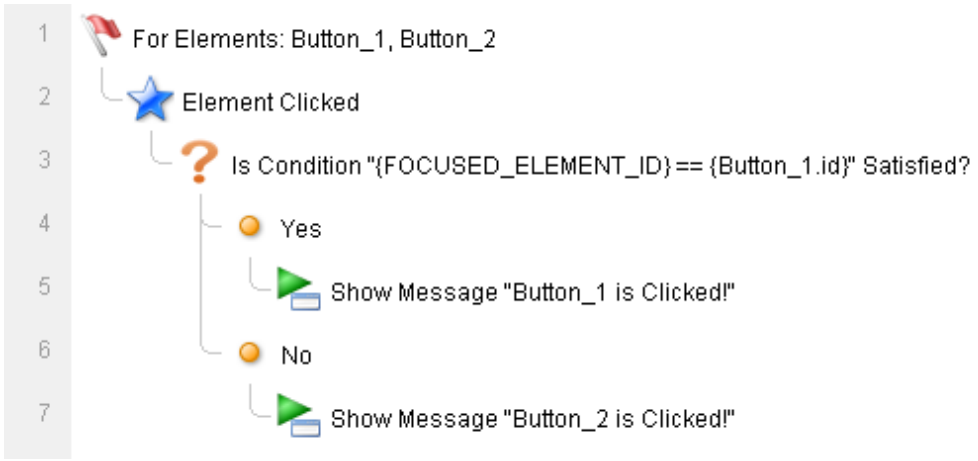
If you click the "Add Flow Control" button, all available flow controls will be listed in a popup menu (more details can be found in Flow Control Reference):



If you click the "Add Action" button, all available flow controls will be listed in a popup menu (more details can be found in Action Reference):



Here is an example. When clicking on Button_1 or Button_2 element, a message box will pop up and tell you which button is clicked. Here the {FOCUSED_ELEMENT_ID} is a system property, and {Button_1.id} is an element property.

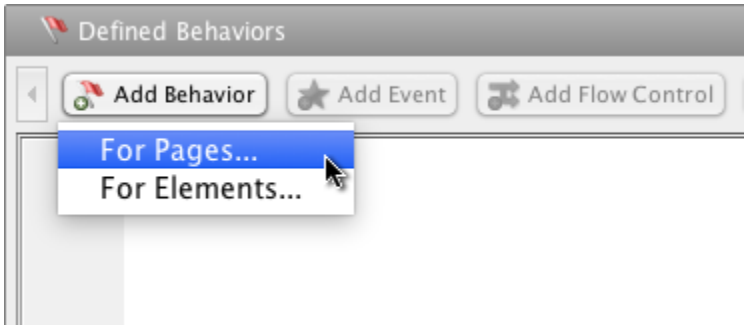


You can add as much events as you need, and define handlers for them, thus your plot can have very complex behavior/interaction. When running HTML5 simulation, all your defined behaviors will be converted to Javascript.

7.1 Define Page's Behavior

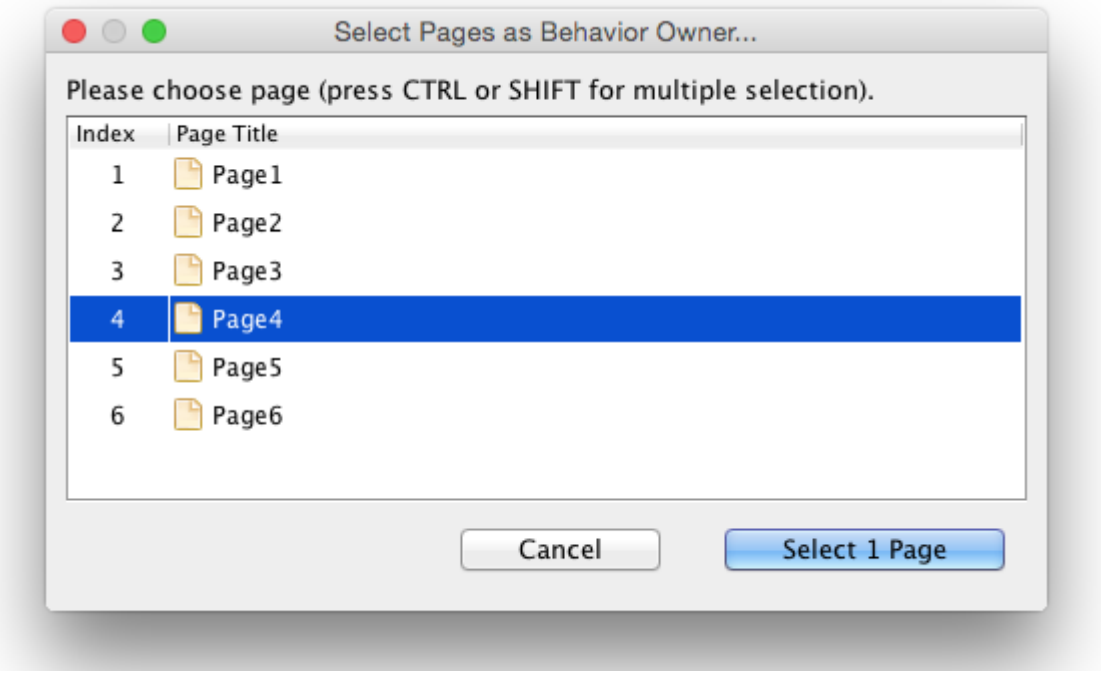
You can also define the behavior for the entire page. Just follow the steps below:

1. Open the behavior editor view.
2. Click the "Add Behavior" button in the toolbar and then select "For Pages..."
3. Select the page(s). You can select multiple pages if you want to define behavior for them together.

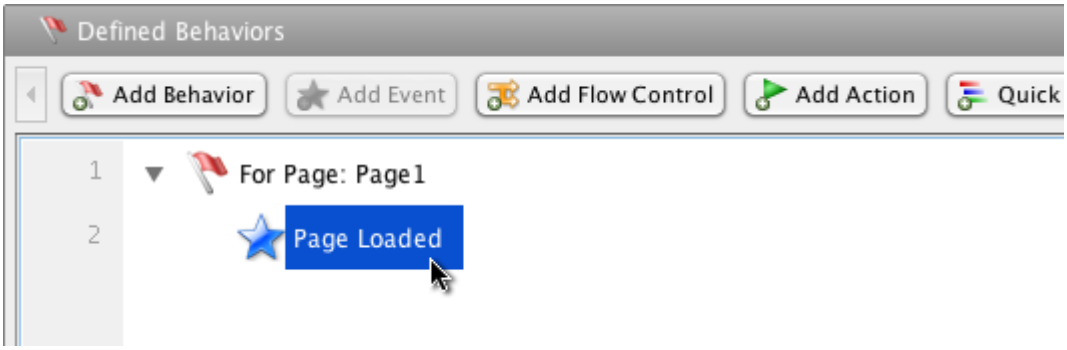


The step 1 and step 2 can also be achieved by pressing **Ctrl+D** (Command + D in Mac OS X).

In step 3, you will see the page chooser window:



The current page will be selected by default. You can switch to target page first, and then press Ctrl+D (or do step 1 and 2), you can finish step 3 by directly click the "Select n Page" button (where n is a number). Then you will see the behavior owner in the behavior editor:

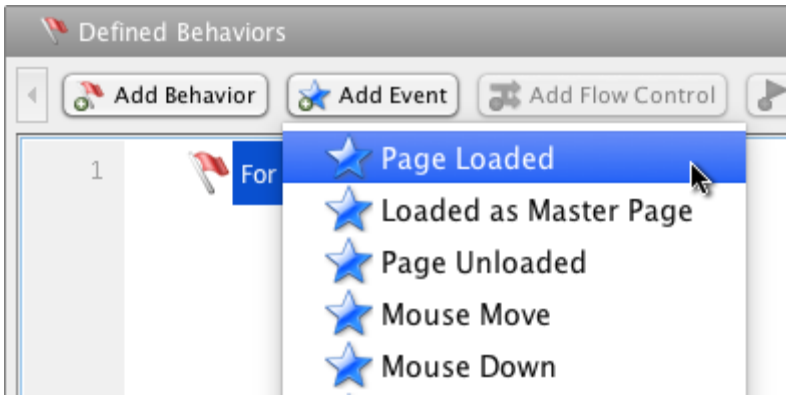


After that you can define the event handler by adding event, flow control or/and action.

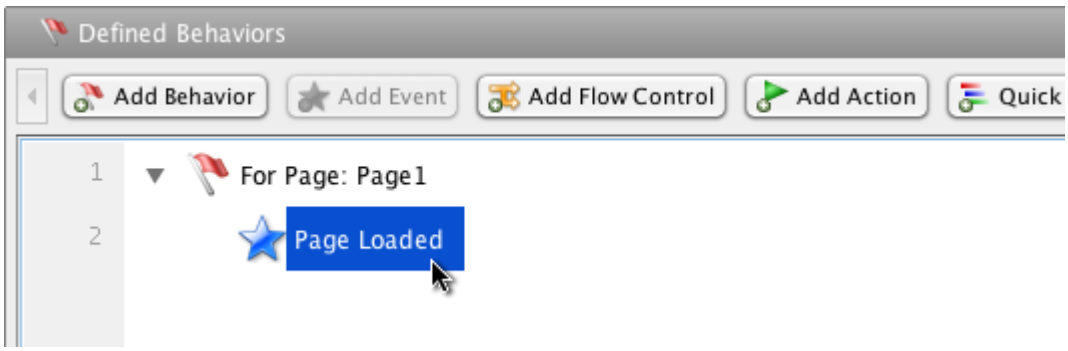
Add Event

The "Add Event" button is enabled when you select the behavior (node with red flag icon). That means you can continue defining the behavior by adding events into it. All available events (according to owner type,

you can find all available events in Events Reference) will be listed in a popup menu when you click the "Add Event" button.



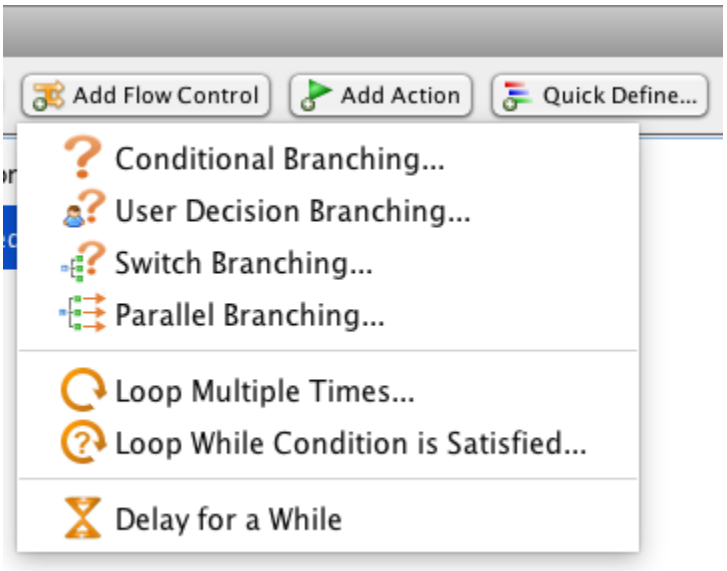
After adding the event, it will be selected by default and the "Add Flow Control" and "Add Action" buttons will be enabled now.



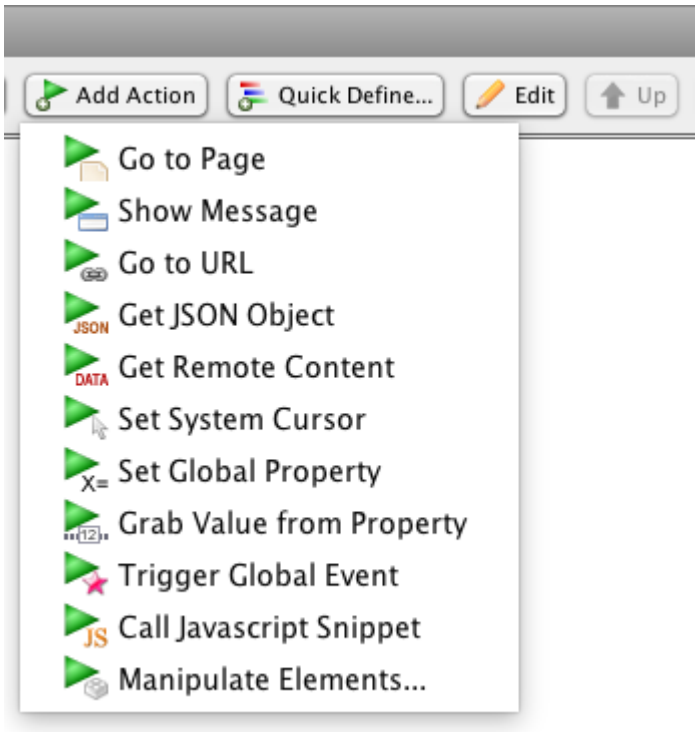
Add Flow Control or/and Action

Defining the behavior is just like creating flowchart. The event is the start point, and you can append flow controls (looping, branching, delay etc.) and actions under it.

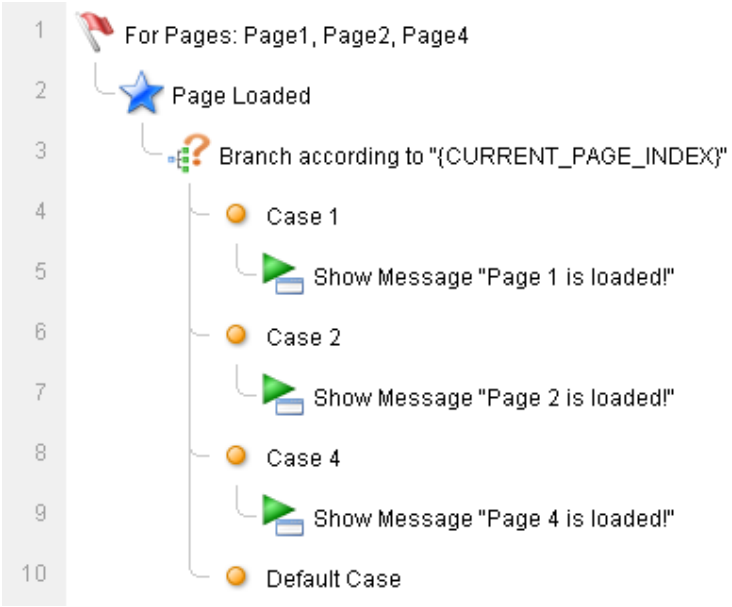
If you click the "Add Flow Control" button, all available flow controls will be listed in a popup menu (more details can be found in Flow Control Reference):



If you click the "Add Action" button, all available flow controls will be listed in a popup menu (more details can be found in Action Reference):



Here is an example. If page 1, 2 or 4 is loaded, a message window will popup and tell you which page is loaded.



You can add as much event handlers as you need for pages. When running HTML5 simulation, all your defined behaviors will be converted to Javascript.

8.1 Use Expression

When defining the behavior, we will be asked to input values frequently. Some input fields support expression (has ... button on the right), which means you can use a sentence to represent the value in design phase, and it will be replaced with the actual value in simulation.

According to the parsing mode, ForeUI supports two kinds of expressions: TEXT expression and EVAL expression. They have slightly different syntax and will be parsed with TEXT and EVAL parsing modes accordingly.

TEXT Expression

TEXT expression represents a text message, and it will be parsed with TEXT parsing mode (has TEXT icon in input field). You write TEXT expression just like writing normal text message, and only replace the variables in message with properties (element property, system property or user defined property) . Below are three examples of TEXT expressions.

You've input {TextEditBox_1.value}

We are in year {CURRENT_YEAR}

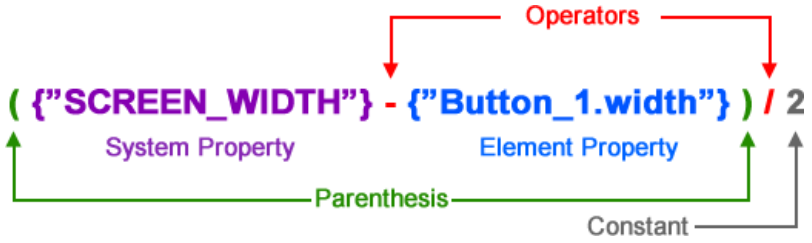
You have {msg_count} new messages!

As you can see, the properties in the expression have common form: {PropertyName}. Please notice that you don't need to quote the property name with double quote marks. However it is also acceptable if you do so (e.g. You have {"msg_count"} new messages!).

EVAL Expression

EVAL expression can represent many different things: a text message, a number, an array, an object or nothing but calling one or more Javascript functions. It will be parsed with EVAL parsing mode (has EVAL icon in input field).

An EVAL expression can contain constants, properties (element property, system property or user defined property) parenthesis and operators, and you can make calculations in it. The figure below shows an example of EVAL expression:



As you can see, the properties in EVAL expression are a little different than those in TEXT expression: the **property name must be quoted** with double quote marks. If you forget to do this, ForeUI will think it is a syntax error.

Comparing with the TEXT expression, EVAL expression is more like Javascript. Besides the properties within the expression, you can also use parenthesis, operators, constant etc. You can even call a Javascript function (see how to define Javascript function in ForeUI) directly in an EVAL expression. The only difference with Javascript is the property, which acts as variables.

EVAL expression can do much more tasks, but its syntax is also more complicated. If you want, you can use EVAL expression to replace TEXT expression. For example, the TEXT expression in previous example:

TEXT You have {msg_count} new messages!

You can rewrite it as EVAL expression like this:

EVAL "You have " + {"msg_count"} + "new messages!"

See the difference? In EVAL expression, you have to wrap the text string with double quote marks, including the property name. If there are multiple text string in expression, you need to use "+" operator to connect them.

Switch Between Two Types of Expression

In the major of cases, the text input box that accepts expression can accept both types of expression, and you can manually switch the type at your will.

You may see TEXT or EVAL icon on the right of the text input box. Clicking on it can toggle the parsing mode between TEXT and EVAL, hence switching the type of expression.

Sometimes the text input box can only accept one parsing mode, in that case the icon will become gray and you can not toggle the parsing mode.

Which Expression Should I Use?

It is true that EVAL expression can always replace TEXT expression. However in the example above you can also see the point to have TEXT expression: it is much simpler for building a text string with property value inside.

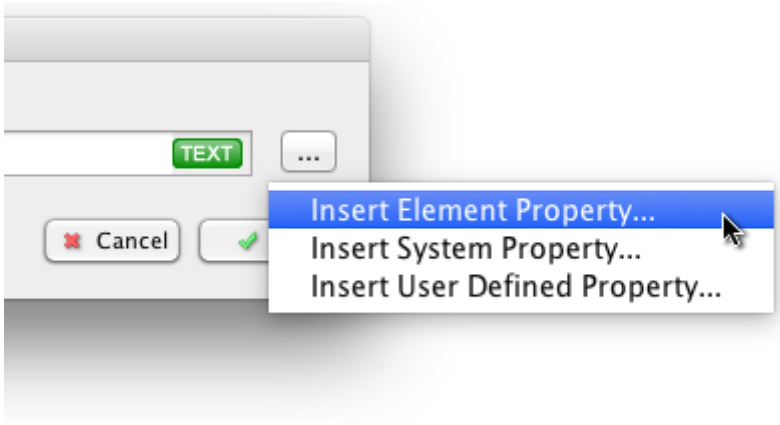
TEXT expression can do less but it is more friendly to everyone; EVAL expression is more like programing in Javascript.

So the conclusion is: if you need a text string, use TEXT expression; if you want to make some calculation or function calls, use EVAL expression.

Insert Property into Expression

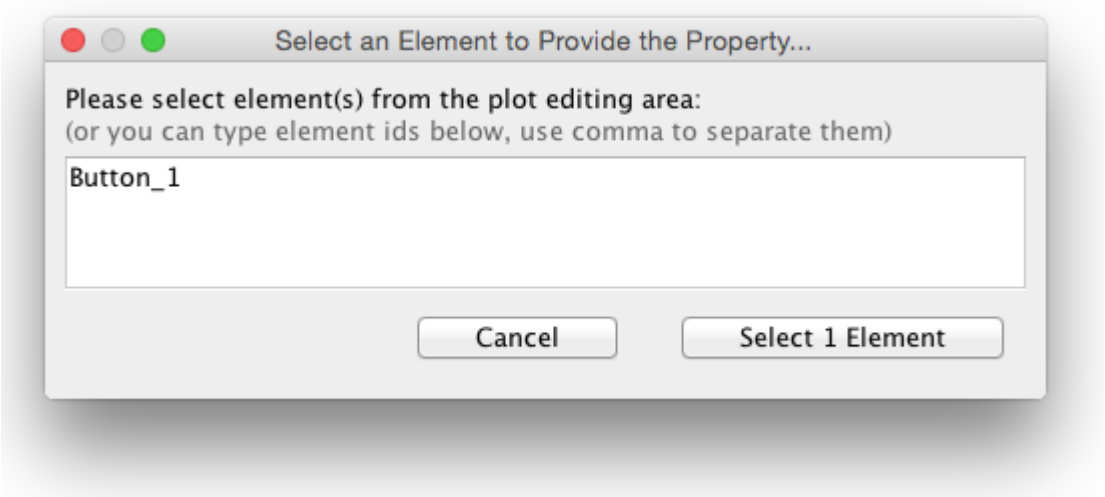
Although the expression is in code form, usually you don't need to input them letter by letter, ForeUI provides some facilities to help you create the expression quickly.

All input fields that support expression will has a ... button nearby. This button can bring out a popup menu that allows you to choose element property, system property or user defined property to insert into the input field.

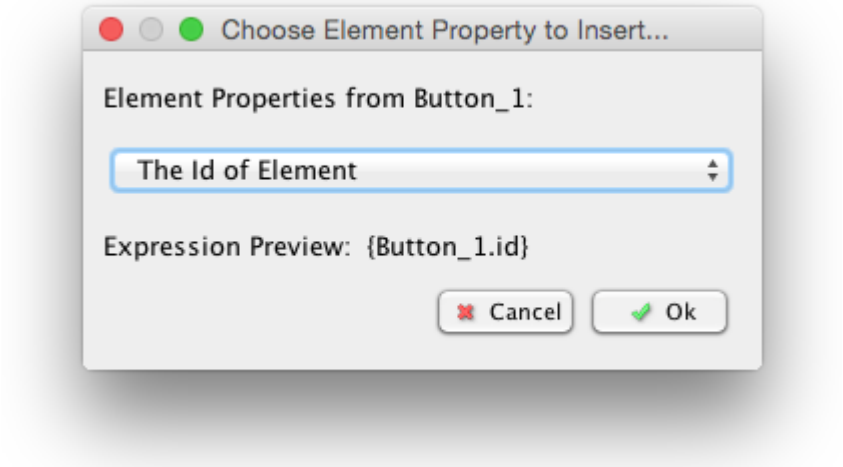


Insert Element Property

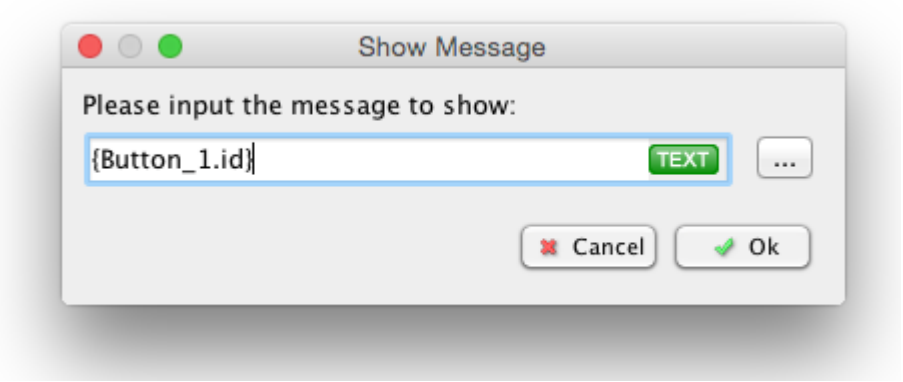
If you choose "Insert Element Property", you will be asked to pick an element that owns the property. You will see the element chooser window:



After picking the element and click "Select 1 Element" button, you will see the element property chooser window:



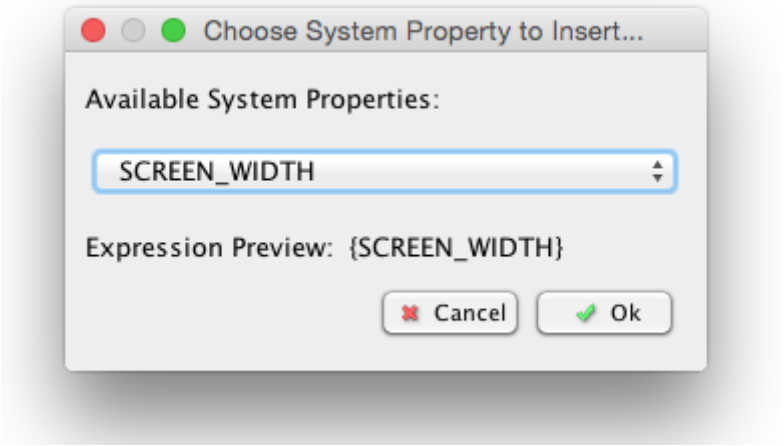
After choosing property from the drop-down list and click "Ok" button, the element property is inserted into the input field:



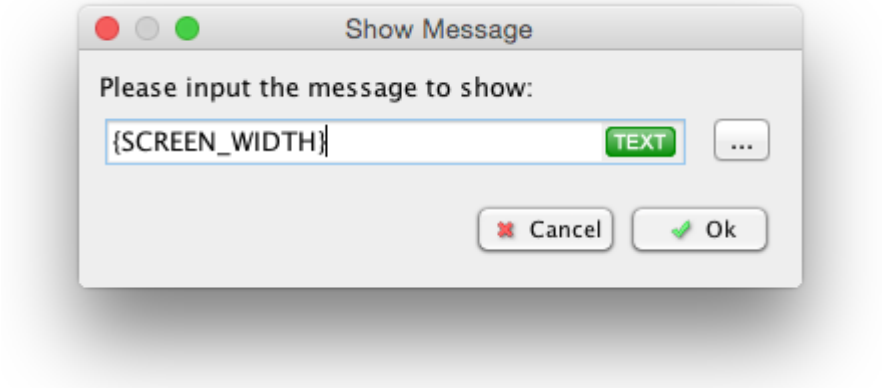
Insert System Property

All available system properties can be reviewed in the System Property View.

To insert a system property into the input field, just click the ... button and select the "Insert System Property..." from the popup menu. Then you will see the system property chooser window:



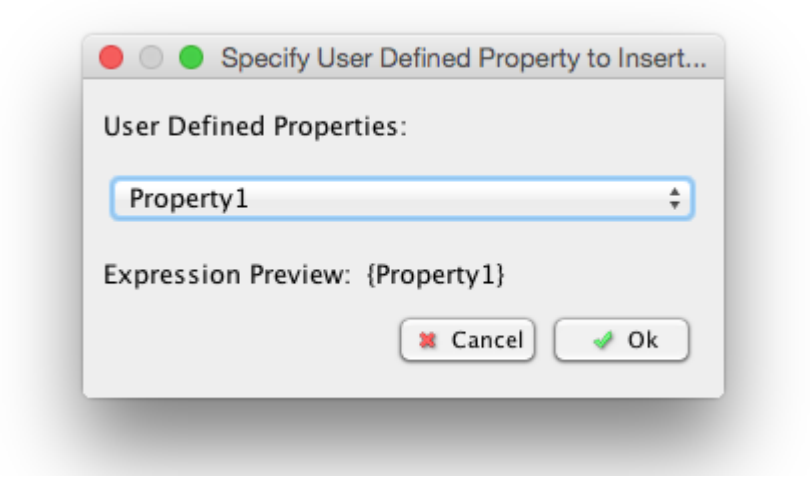
After choosing system property from the drop-down list and click "Ok" button, the system property is inserted into the input field:



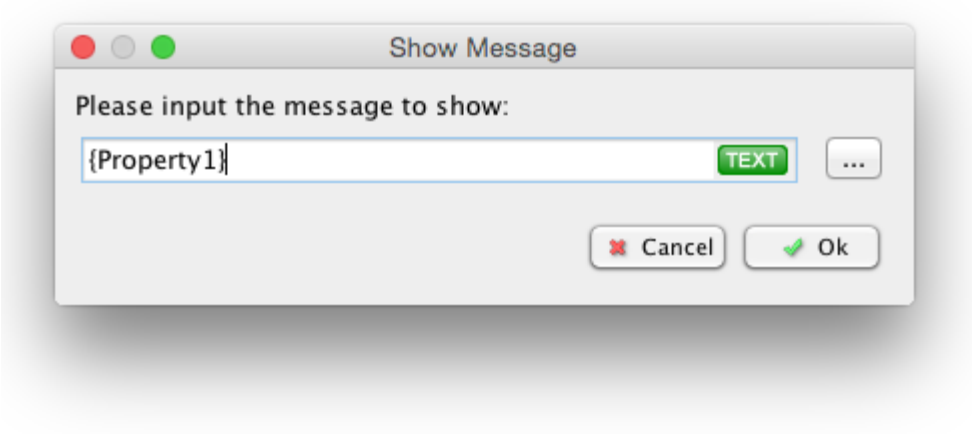
Insert User Defined Property (Custom Property)

You can manage or review user defined properties in the Custom Property View. Another way to create user defined property is to invoke the Set Global Property action, if the property with the given name is not defined yet, the action will create one for you.

To insert a user defined property into the input field, just click the ... button and select the "Insert User Defined Property..." from the popup menu. Then you will see the user defined property chooser window:



Here the drop-down list maybe empty if you have never defined any property beforehand. After choosing property from the drop-down list and click "Ok" button, the user defined property is inserted into the input field:

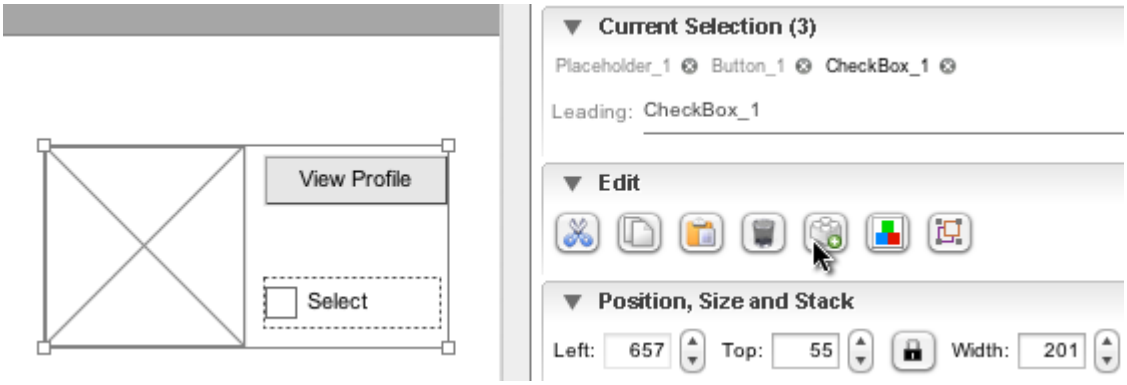


9.1 Use Custom Element

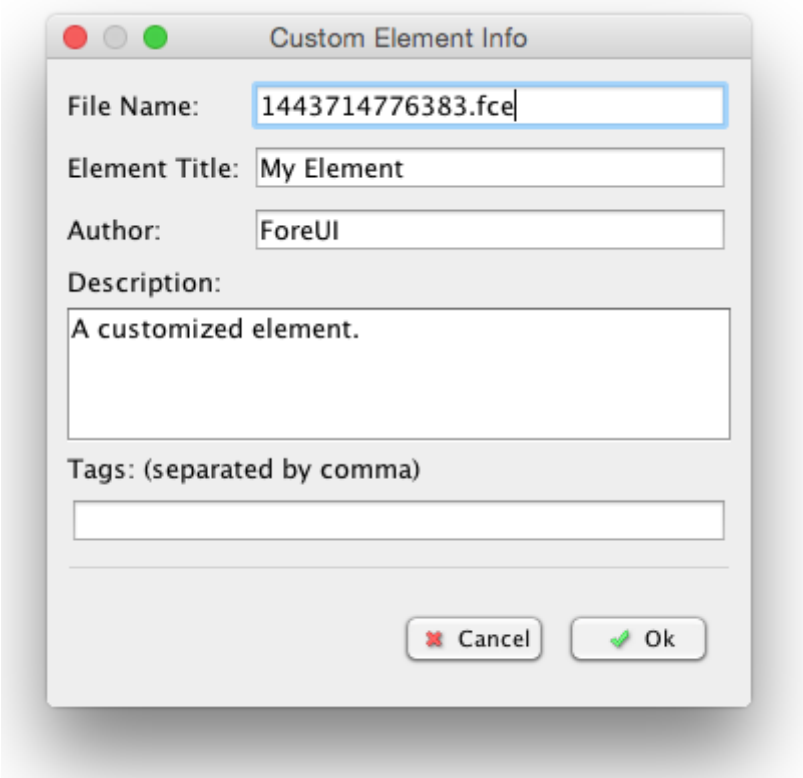
ForeUI provides many predefined elements for creating your prototyping. However you may still need more elements when working on actual project. In this case, you can create your own element with the predefined elements, and save it for future usage. The customized elements can also have their own behavior, which means you can create some functional elements, which will be really useful.

Create Custom Element

To create a custom element, just select the content that you want to pack as custom element in the Plot Editing Area, and then click the button in the Tools Panel, or the button in bottom toolbar of the Elements Panel.



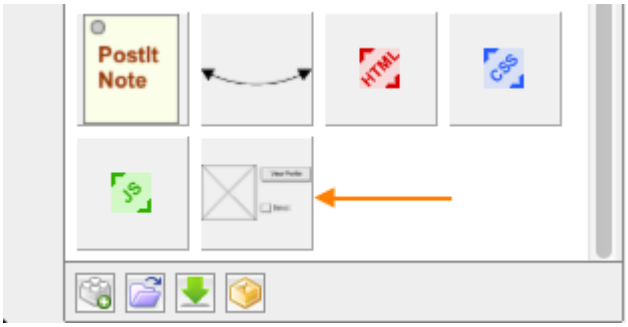
Then you will see this window:



Here you can input the basic information of the custom element.

- **File Name:** the name of the custom element. You can find the saved .fce file in "<UserDir>/.foreui/customize" directory.
- **Element Title:** this title will be displayed in element button (big button) or tool tip (small button).
- **Author:** the author of the element (you).
- **Description:** describe this element.
- **Tags:** the tags input here will become the keywords of this element, and you can use the tag to find your element quickly.

After clicking the "Ok" button, your custom element will be created and listed in your Elements Panel, and you can use it like other elements.



If you move your mouse cursor over the element button for custom element, you will see three small buttons:

button can remove the custom element from the panel, but **will NOT delete the .fce file**.

button can view/edit the basic information of the custom element.

button can open the .fce file and edit its content in plot editing area.

Load Custom Element into Panel

If you removed the custom element and want to load it back to the panel, just click the button in the bottom toolbar of elements panel and choose the .fce file to load.

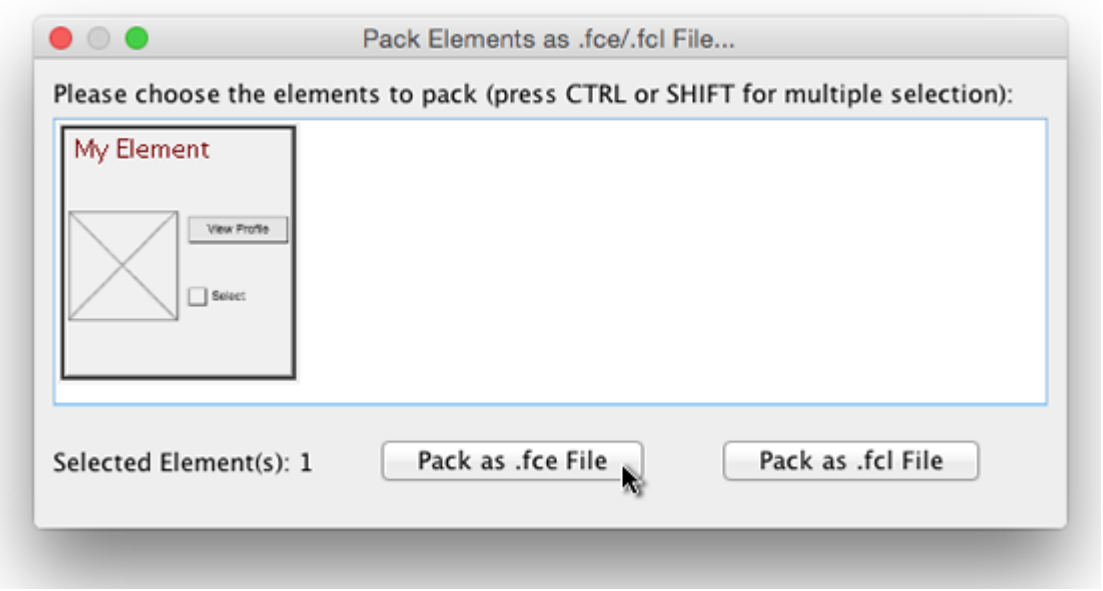
If you get some .fce files from others, you can load them into your elements panel with the same way.

Download Custom Elements

You can download more custom elements from ForeUI Store by clicking the button in the bottom toolbar of elements panel.

Export .fce File

If you have your custom element in your elements panel and want to share it with others, you can click the button in the bottom toolbar of elements panel. You can then select your element and click the "Pack as .fce file" button, and export it to any location.

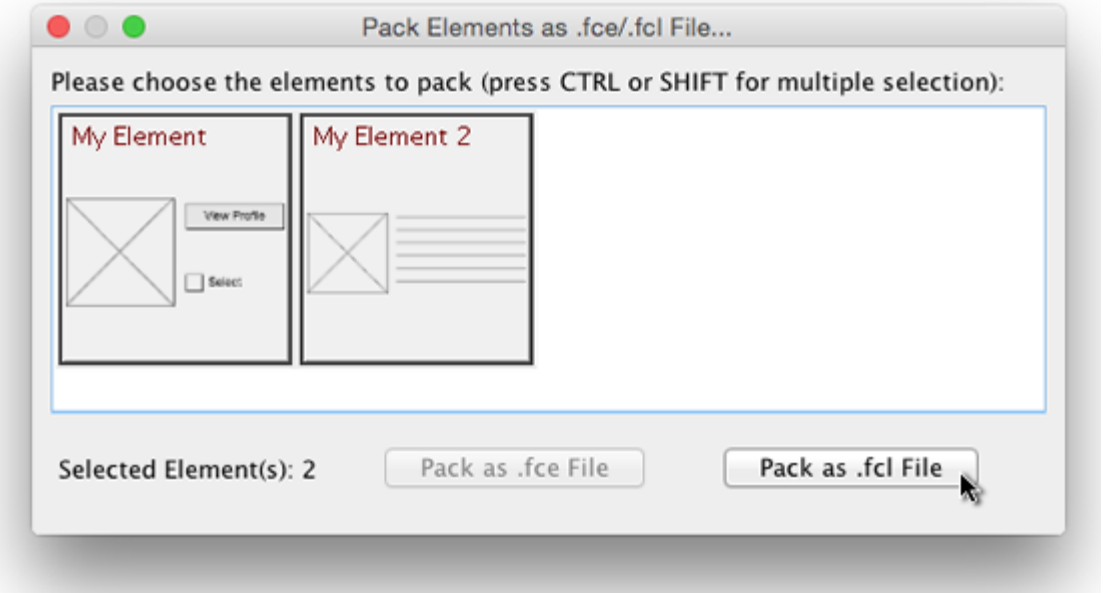


10.1 Use Custom Element Library

If you have several similar custom elements, you may consider packing them as a library.

Create Custom Element Library

Just click the  button in the bottom toolbar of elements panel. You can then choose the elements and click the "Pack as .fcl file" button, and export these elements as a single library (.fcl) file.



The .fcl file is just a ZIP file that contains all .fce files (the custom elements). You can simply share it with others via email or FTP.

Load Custom Element Library

If you want to load the .fcl file into your elements panel, just click the  button in the bottom toolbar of elements panel and choose the .fcl file to load. All custom elements in the .fcl file will be loaded into the panel.

Download Custom Elements Library

You can download more custom element libraries from ForeUI Store by clicking the  button in the bottom toolbar of elements panel.

11.1 Switch UI Themes

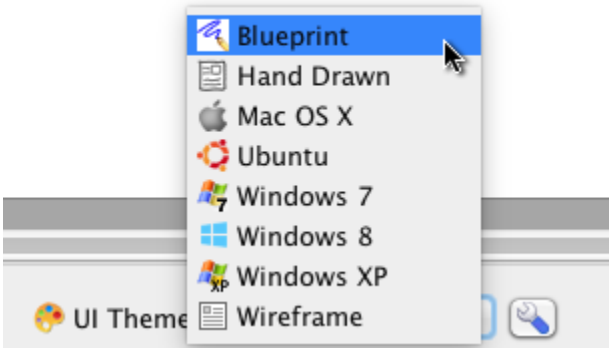
You can change the style of your prototype by simply switch its UI theme.

So far ForeUI support 5 UI themes:

- Hand Drawn
- Wireframe
- Windows XP
- Mac OS X
- Windows 7

The Wireframe UI theme is the default one for newly created plot, which can be changed via the preference configuration.

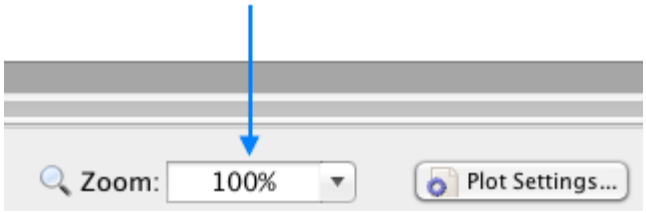
To change the UI theme, just simply choose the UI theme from the drop-down list at the bottom:



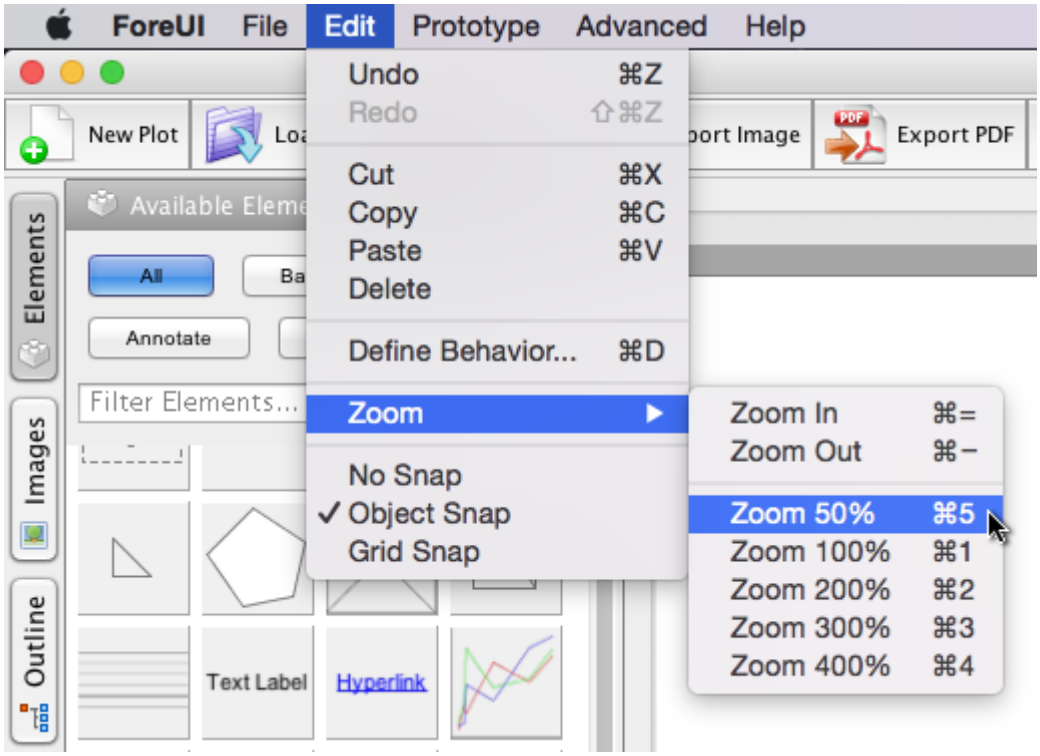
Once you switch the UI theme, the current editing prototype will change its look, some elements in the elements panel will change their look as well.

12.1 Zoom In and Zoom Out

Sometimes you may want to zoom in the plot to tweak small elements, or you need to zoom out the plot to get an overview of the whole design, you can choose the zooming scale from the drop down list at the bottom right corner, you can also input any percentage value directly for zooming:



It is also possible to change the current zooming scale from the menu:



As you can see in the popup menu, you can also use hockey to change the scale as well:

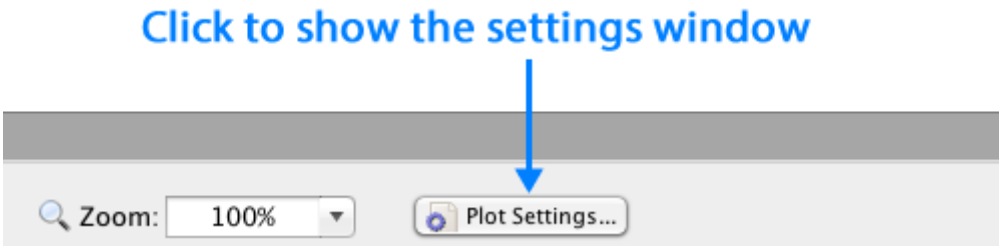
- Ctrl+Equals: Zoom In (Command+Equals in Mac OS)
- Ctrl+Minus: Zoom Out (Command+Minus in Mac OS)
- Ctrl+1: Zoom to 100% (Command+1 in Mac OS)
- Ctrl+2: Zoom to 200% (Command+2 in Mac OS)
- Ctrl+3: Zoom to 300% (Command+3 in Mac OS)
- Ctrl+4: Zoom to 400% (Command+4 in Mac OS)
- Ctrl+5: Zoom to 50% (Command+5 in Mac OS)

Remarks: You could also hold the CTRL key and scroll the mouse wheel to zoom in/out.

13.1 Configure Plot Parameters

This section is about configuring the parameters for the plot, if you want to configure preferences of the software, you should read Configure Software Preferences instead.

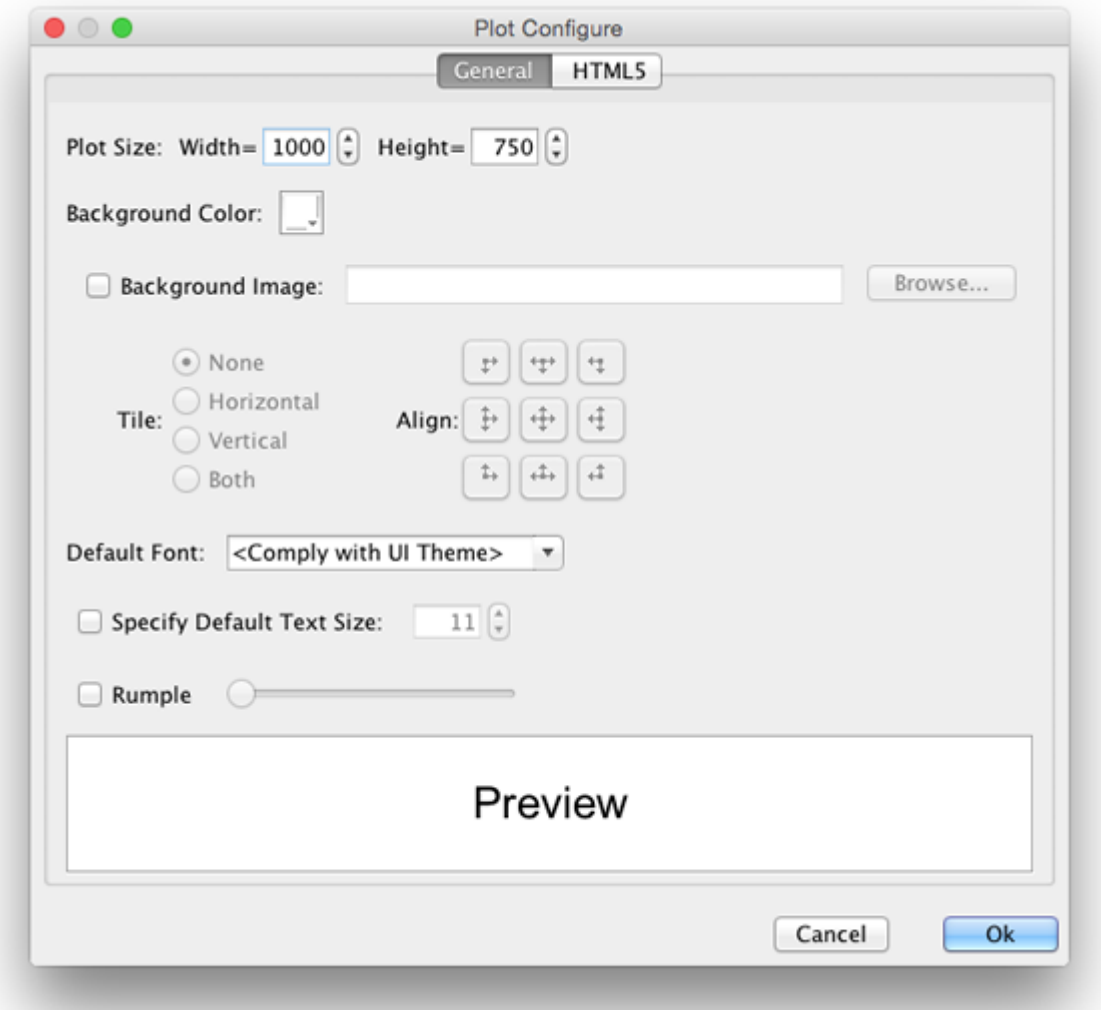
You can always bring out the plot configure window by clicking the button on the right-bottom corner:



The plot configure window can help you to:

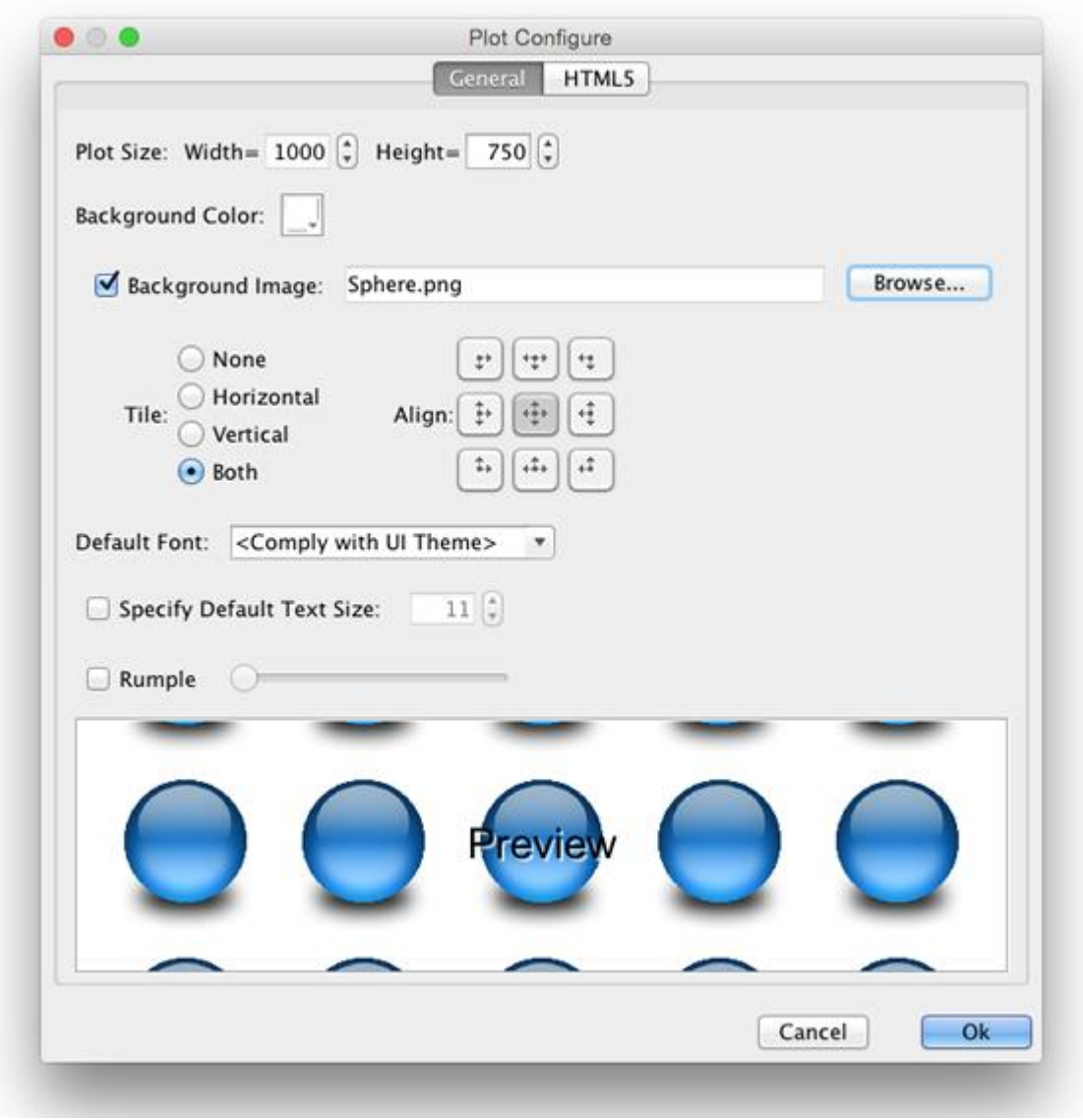
- Change the plot dimension
- Change the background color
- Change the default font
- Specify background image and its placement
- Adjust optional rumple effect
- Tweak HTML5 export/simulation parameters

The plot configure window looks like this, and the "General" tab is shown by default:



Here you can make configuration for the current editing plot.

If you specify a background image, you have the chance to define how to tile it and how to align the background image with the plot background.



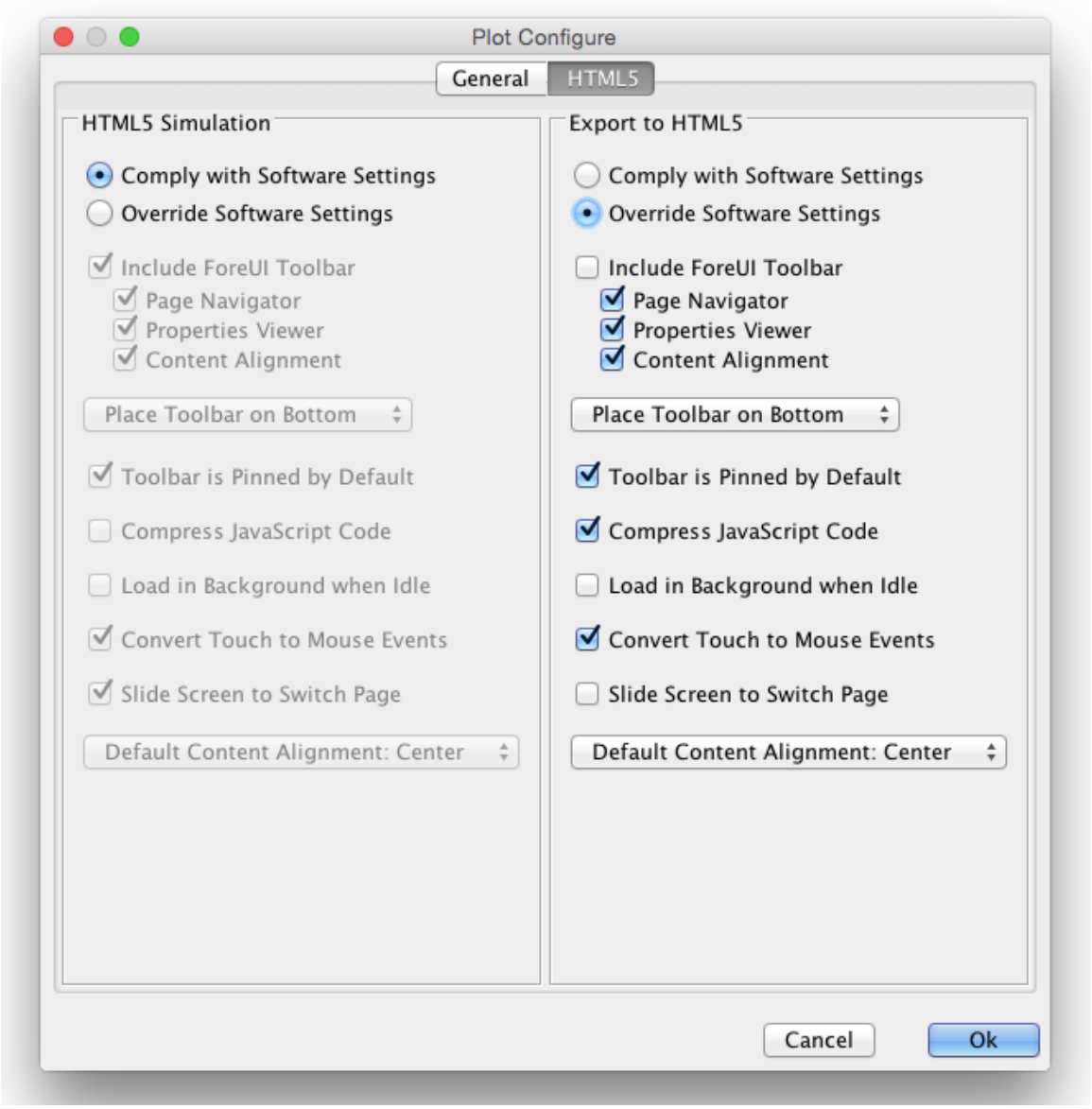
The **default font** will be applied to every text element. The default value is "<Comply with UI Theme>", which means the default font will be specified by UI theme (for example, the "Hand Drawn" UI theme use "Comic Sans MS" as the default font). If you choose a specific font here, the default font defined in UI theme will be overridden.

There is an option named "**Rumple**", which can make your prototype looks more unfinished. You can drag the slider on the right to change the strength of the rumple effect. Please keep in mind that the rumple effect will only exist in editing mode, exported PDF and images, it will not exist in HTML5 simulation.

Remarks: If you like to change the preferred parameters for newly created plot, you should do it in the settings window.

If you click the "HTML5" tab, you can tweak the parameters for HTML5 export/simulation, and the configuration here will override those in the settings window. By default the "Comply with Software Settings" option is selected and the plot will use the default software settings for HTML5 simulation/export.

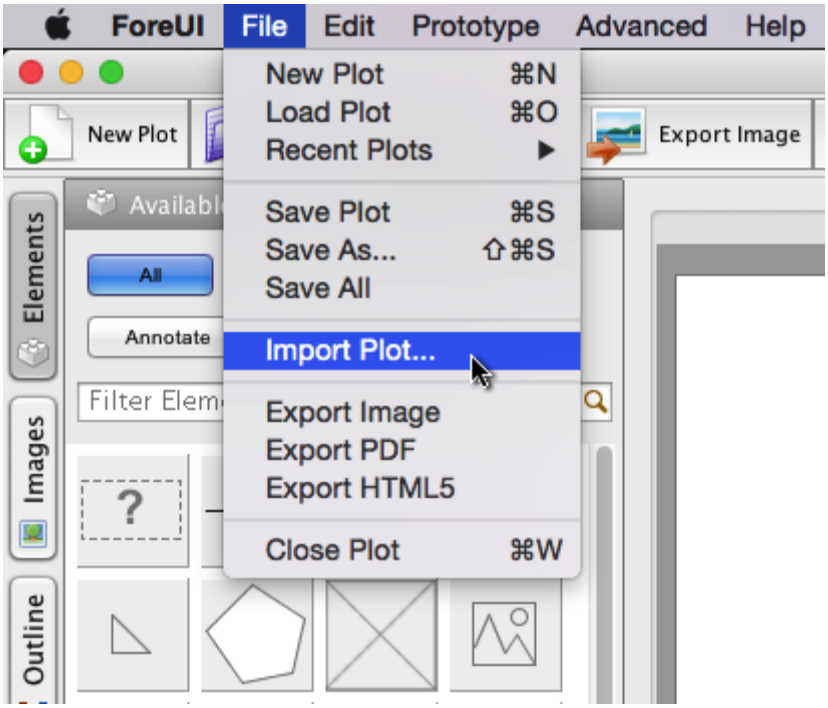
However, if you want, you can choose "Override Software Settings" option here, and then specify the parameters for HTML5 simulation/export. Thus each plot can have different settings for HTML5 simulation/export.



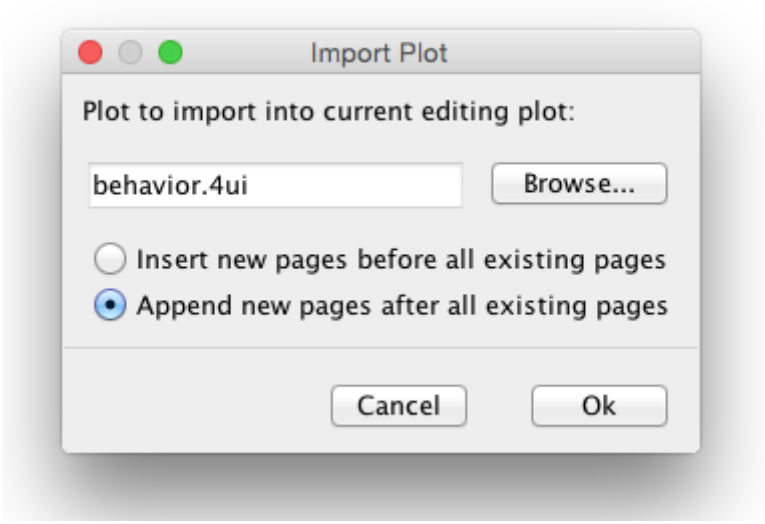
14.1 Merge Multiple Plots

If you are a team and the members are working on different parts of the design, you may want to merge your work in the future. There is an "import plot" feature, which will help you to merge multiple plots into one.

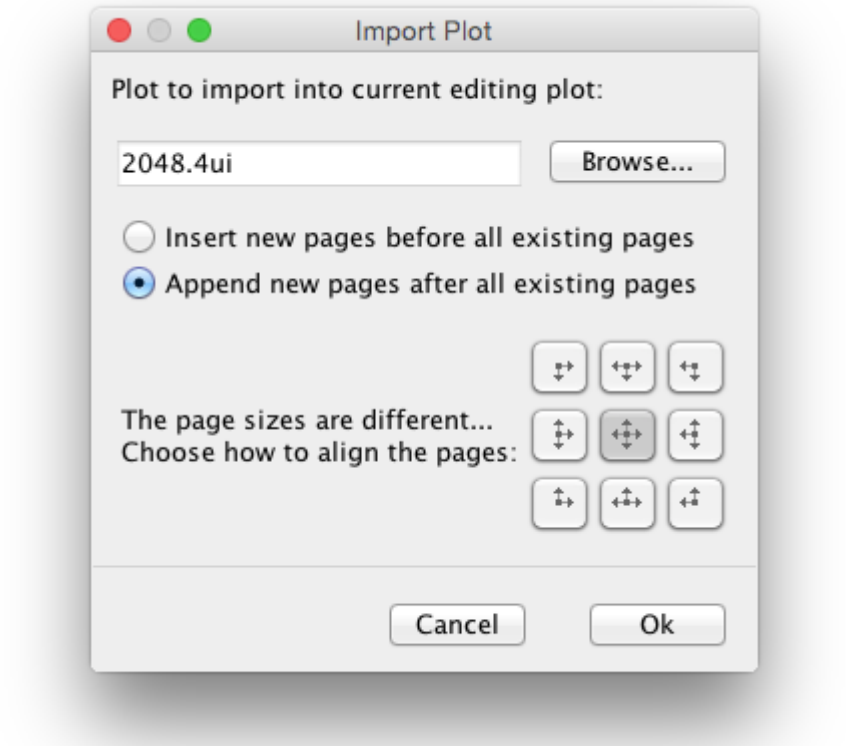
Open the "File" menu you will see the "Import Plot..." option, this option will be enabled if you have opened at least one plot for editing.



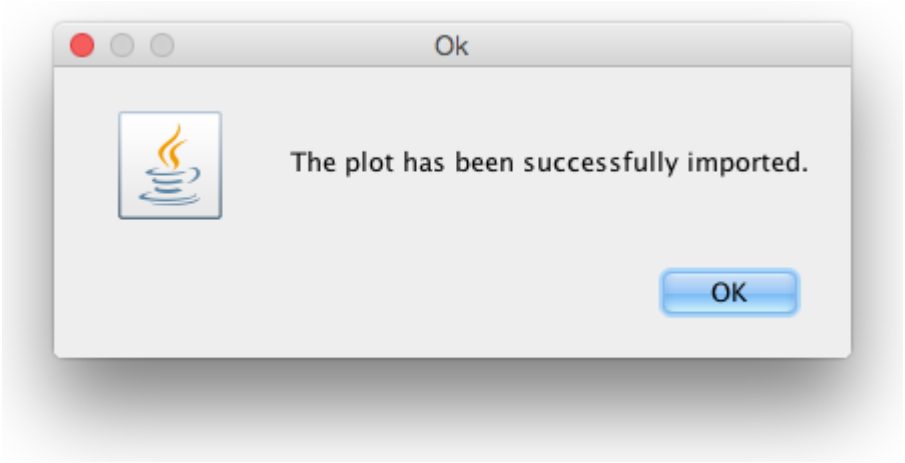
This option allows you to choose another plot to import into the current editing plot. After specifying the plot, you can decide how to add the new pages into the current editing plot.



If the chosen plot has different size than the current editing plot, you can also specify how to align the new pages with the current existing pages.



After clicking the "Ok" button, the plot will be imported and you will get notification when it is done.

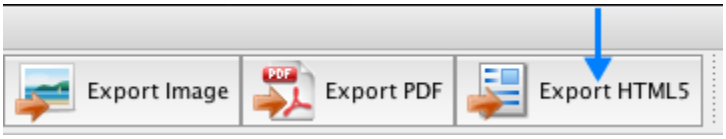


If you need to merge more plots, just repeat the steps and you will get all of them merged into the current editing plot. When you are working as a team, you don't have to define a convention for naming of elements, since the plot importing feature will take care of the naming of elements, usage of images and the behavior definition.

15.1 Export Plot to HTML5 Simulation

Export HTML5 Simulation via GUI

To export your plot as HTML5, just click the "Export HTML5" button in the toolbar or click menu "File->Export HTML5".



Then you need to specify the target directory to store the HTML5. If you input a directory that does not exist, the directory will be created automatically.

The target directory will contain three folders and an "**index.htm**" file. You can load the "index.htm" file in your web browser after the export, which equals to run HTML5 simulation in your web browser.

You can also upload the exported directory to your web site, thus all visitors can review your prototype.

Remarks: if you just want to run HTML5 simulation, you can click the "Run Simulation" button in the toolbar.

Export HTML5 via Command Line:

If you want to perform the HTML5 export silently in the background, without showing the GUI, you can use the command below:

executable plotFilePath -Export:HTML5 targetDirectoryPath

The **executable** will be "launch.bat" in Windows system, or "sh launch.sh" in other platforms. The parameters in **red are required**, so all parameters are mandatory.

Remarks: please run the command in ForeUI's install directory.

Here is an example in Windows system:

launch.bat c:\plots\test.4ui -Export:HTML5 c:\simulations\test

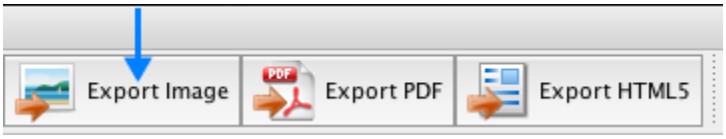
And here is another example in Linux system:

sh launch.sh /User/me/plots/test.4ui -Export:HTML5 /User/me/simulations/test

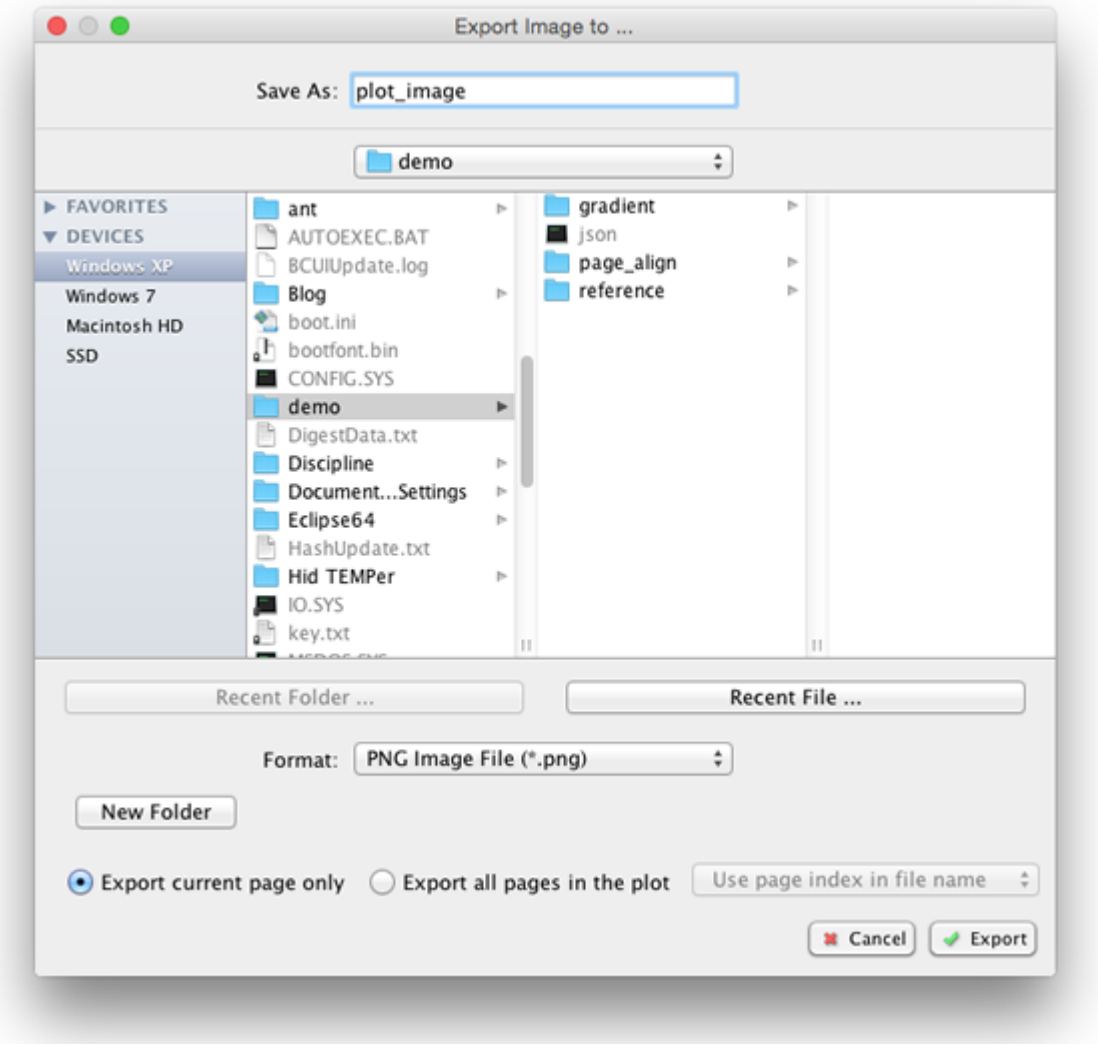
16.1 Export Plot to Images

Export Image via GUI

You can export your plot as PNG, JPEG or BMP images. Just click the "Export Image" button in the toolbar or click menu "File->Export Image".



You will see the image export window like this:



You need to input the file name of the target image file. If you select the "Export all pages in the plot" option, all pages (except the pages that under excluded folders) will be exported. The target image files will be named according the naming strategy you selected:

- **Use page index in file name:** the target image file name will be **FileName_Index.Ext**, where the **FileName** is the string you input in the "File Name:" field; the **Index** is the page index; the **.Ext** is the image file extension (will vary according to the image type).
- **Use page title in file name:** the target image file name will be **FileName_Title.Ext**, where the **Title** is the page title.

Export Image via Command Line:

If you want to export your plot silently in the background, without showing the GUI, you can use the command below:

executable plotFilePath -Export:Image -Format:JPG -Pages:1,3,5 -FileNamePrefix:Img -NamingWith:Index targetDirectoryPath

The **executable** will be "launch.bat" in Windows, or "sh launch.sh" in other platforms. The parameters in **red are required**, parameters in **green are optional**.

Parameters (except the source plot path and target directory path) are in **-Name:Value** format.

The available formats include **JPG, PNG, GIF and BMP**.

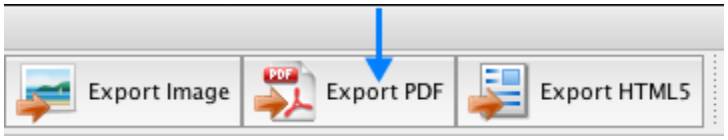
The target image file names will be formatted as: **FileNamePrefix_NamingWith.jpg**

Remarks: ForeUI will use the value below if the optional parameters are omitted:

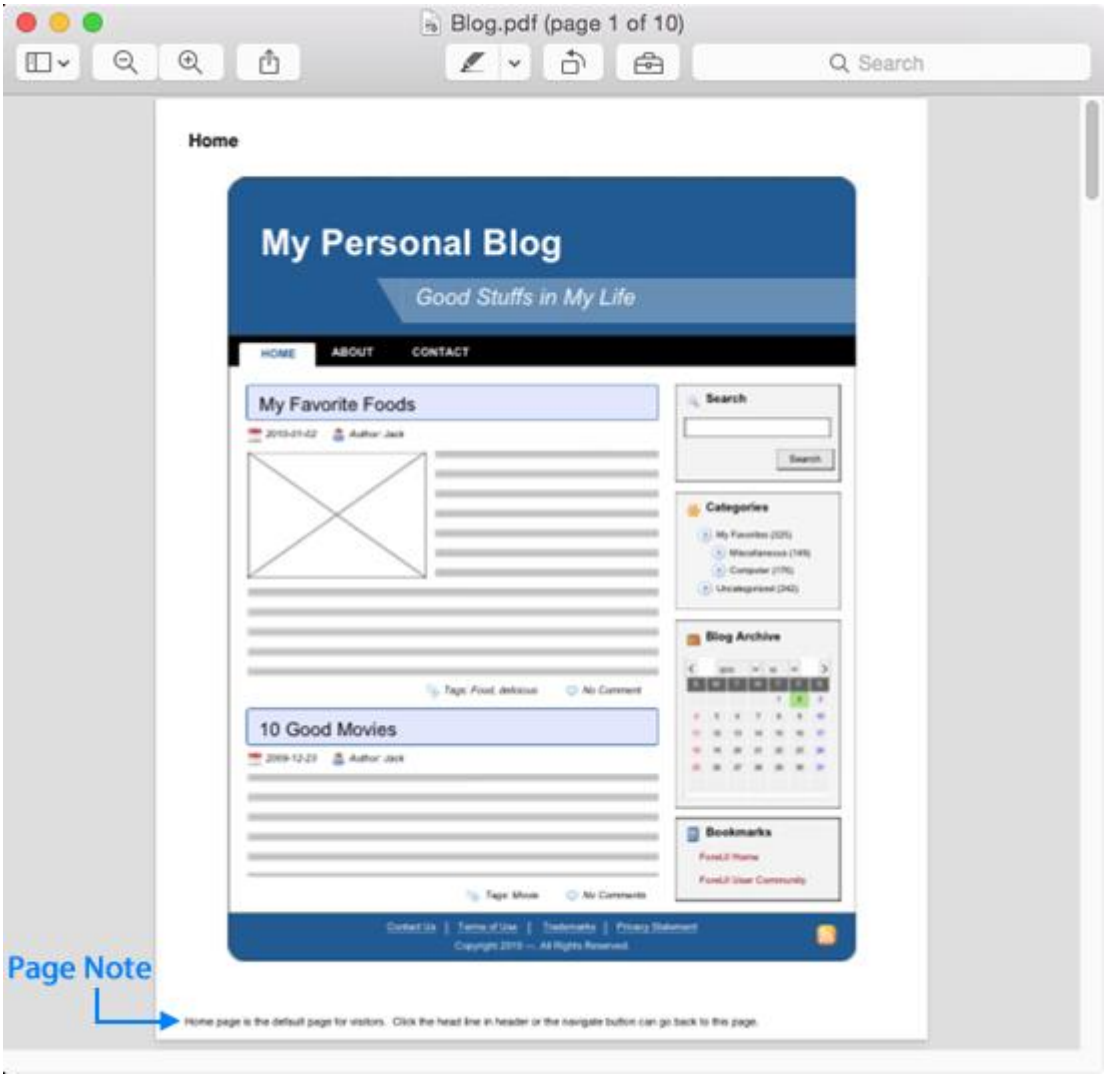
- -Format: PNG
- -Pages: <all pages that are not under excluded folders>
- -FileNamePrefix: <same with the plot file name>
- -NamingWith: Index

17.1 Export Plot to PDF

You can export your plot as PDF document. Just click the "Export PDF" button in the toolbar or click menu "File->Export PDF".



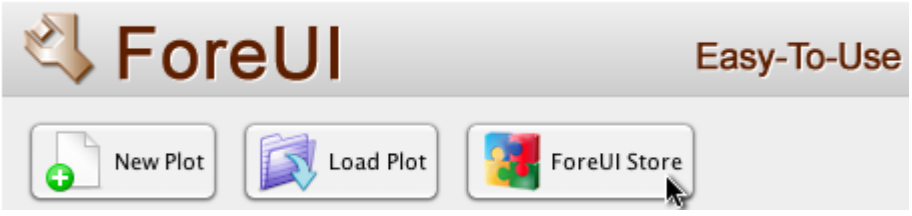
You will be asked to input the file name of the PDF document. The page note will be converted to footnote.



18.1 Download More Examples and Elements

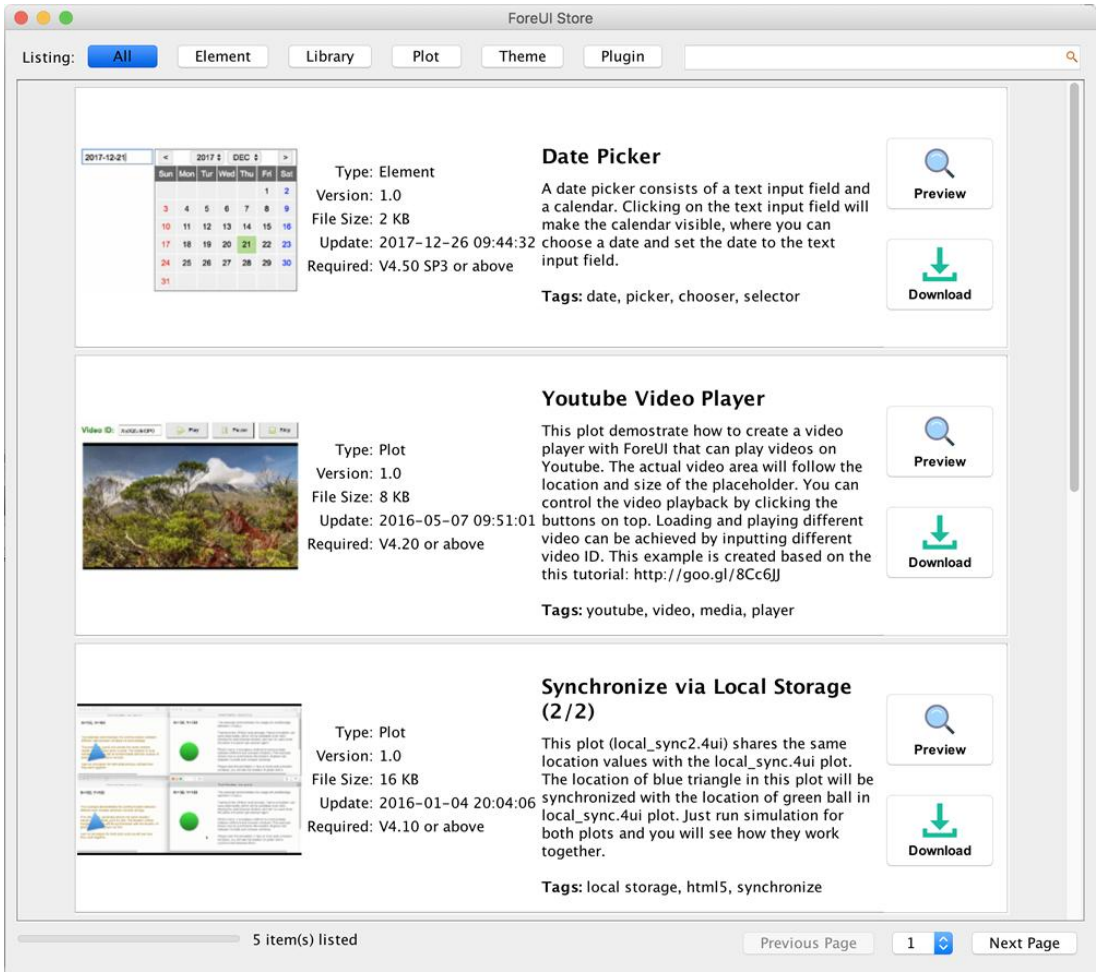
ForeUI has integrated with ForeUI store (http://www.foreui.com/store/), which allows you to easily download more resources from the software. Here the resource means the example plots, custom elements and custom element libraries. You will be able to download UI theme and plugin for ForeUI in the future.

You can click the "ForeUI Store" button in the Welcome Page to open the ForeUI Store Window.



Or you can click the  button in the bottom toolbar of Elements Panel to open it.

The ForeUI store window looks like this:

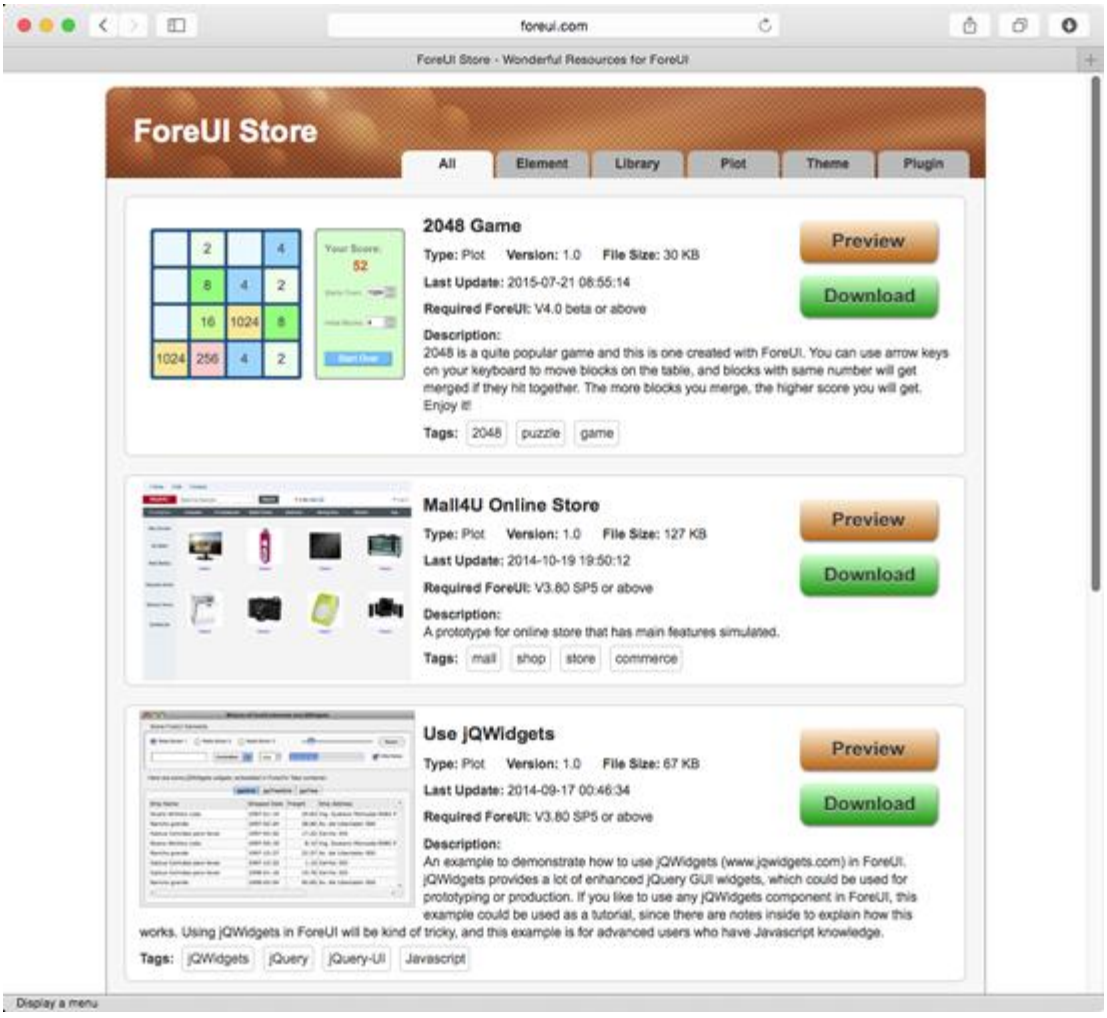


You can preview the resource by clicking on the “Preview” button, which will launch the preview in your default web browser.

Clicking on the “Download” button will download the resource automatically. The resource will be deployed automatically if needed.

More details about how to use ForeUI Store Window can be found here.

You can also download the resource from ForeUI store (http://www.foreui.com/store/) via web browser.



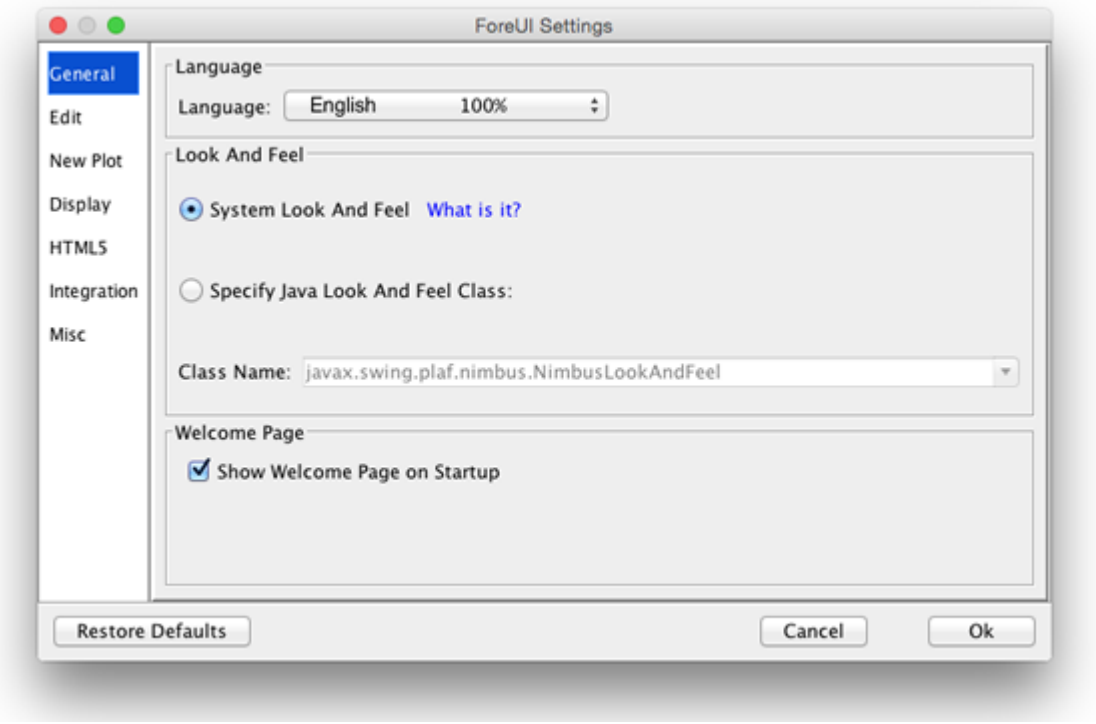
19.1 Configure Software Preferences

This section is about configuring the preferences of the software, if you want to configure parameters for plot only, you should read Configure Plot Parameters instead.

You can choose the menu "Advanced->Settings" to open the settings window. You can make some configurations here so ForeUI can serve you better.

There are 7 tabs in the settings window:

General



Language:

Change GUI's language. So far ForeUI supports English, German, French, Spanish, Brazilian Portuguese and Russian languages. These translations are contributed by volunteers. More details are available in http://www.foreui.com/translation.htm.

Look and Feel:

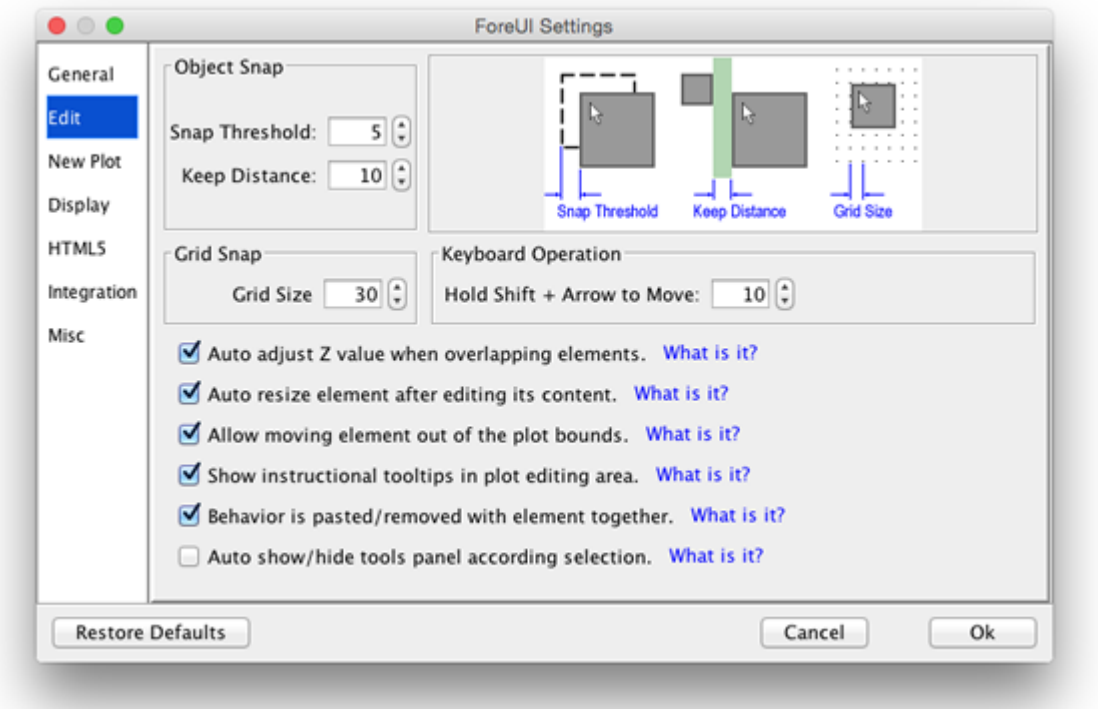
You can choose the look and feel for ForeUI software itself. The default one is the system look and feel, which will let your ForeUI looks native and have similar UI style with other applications in the system.

If you want, you can also choose different look and feels for ForeUI here.

Welcome Page:

Here you can turn on/off the Welcome Page when no plot is opened for editing yet.

Edit



In the "Edit" tab you can configure some important parameters for editing.

Object Snap: Here you can define the parameters for object snap. Snap Threshold is the maximum distance between two elements that can trigger the object snap. Keep Distance is the distance between two elements when their location is adjusted by the snapping system.

Grid Snap: Specify the grid size for grid snap. (Grid snap could be overridden if you drag element with right mouse button hold)

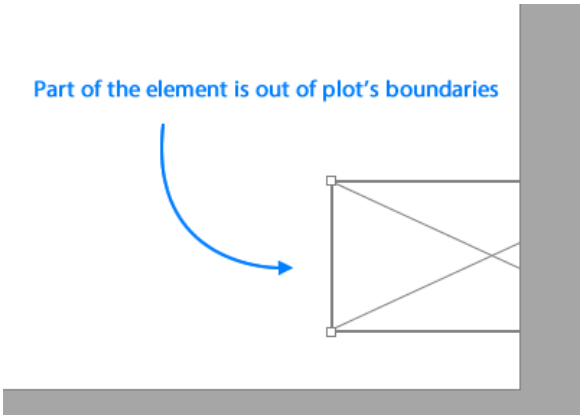
Keyboard Operation: You can configure the step length when nudging element with Shift + Arrow keys pressed.

Shift+Arrow Button to Nudg Element

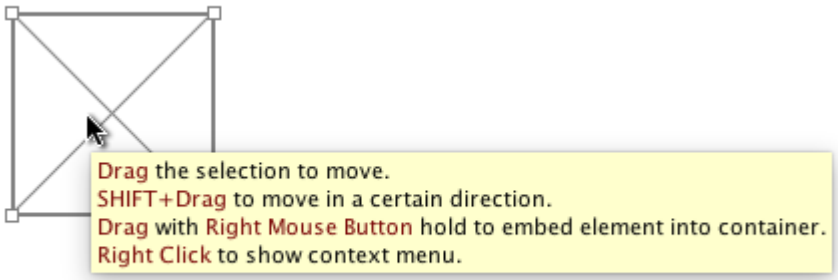
Auto adjust Z...: If you don't want ForeUI to change the Z values of elements when they are overlapped, you can uncheck this option.

Auto resize ...: If you don't want ForeUI to resize the element after editing the element content (text, image etc.), you can uncheck this option.

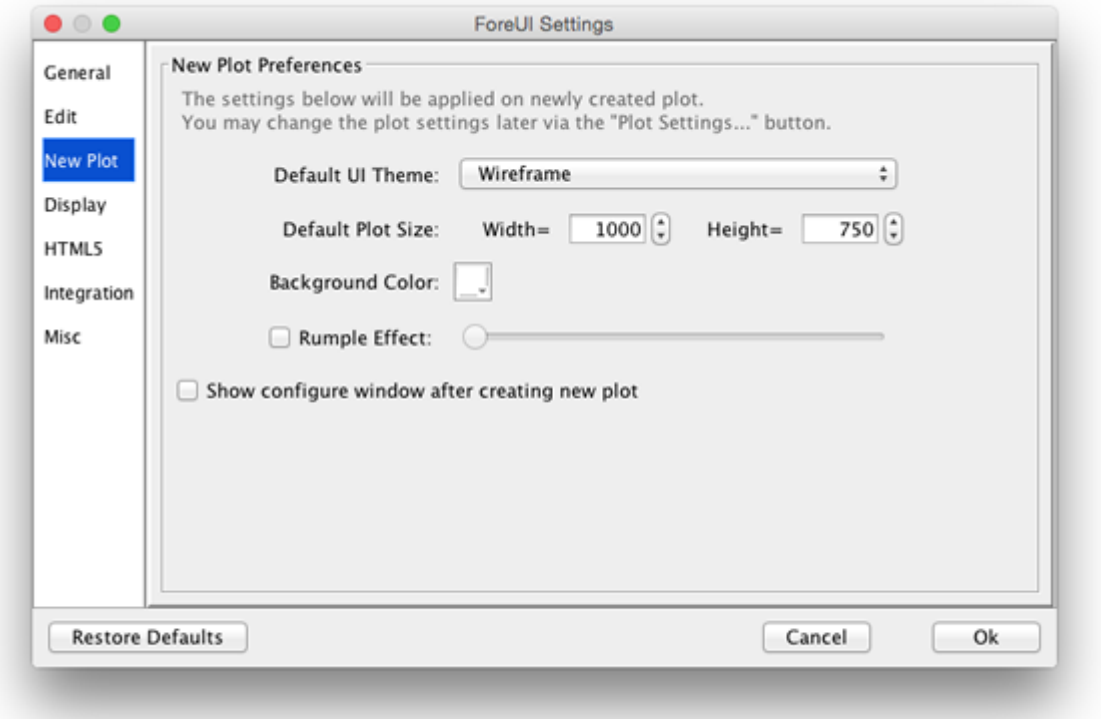
Allow moving...: If you don't plan to place a part or whole element out of the plot's boundaries, you can uncheck this option.



Show instructional tooltips...: If you don't wish to see the floating instructional tool tip, you can uncheck this option. The instructional tool tip is the small floating window like this:



New Plot



Here you can change the preferences for newly created plot.

Default UI Theme: the UI theme that will be used by newly created plot.

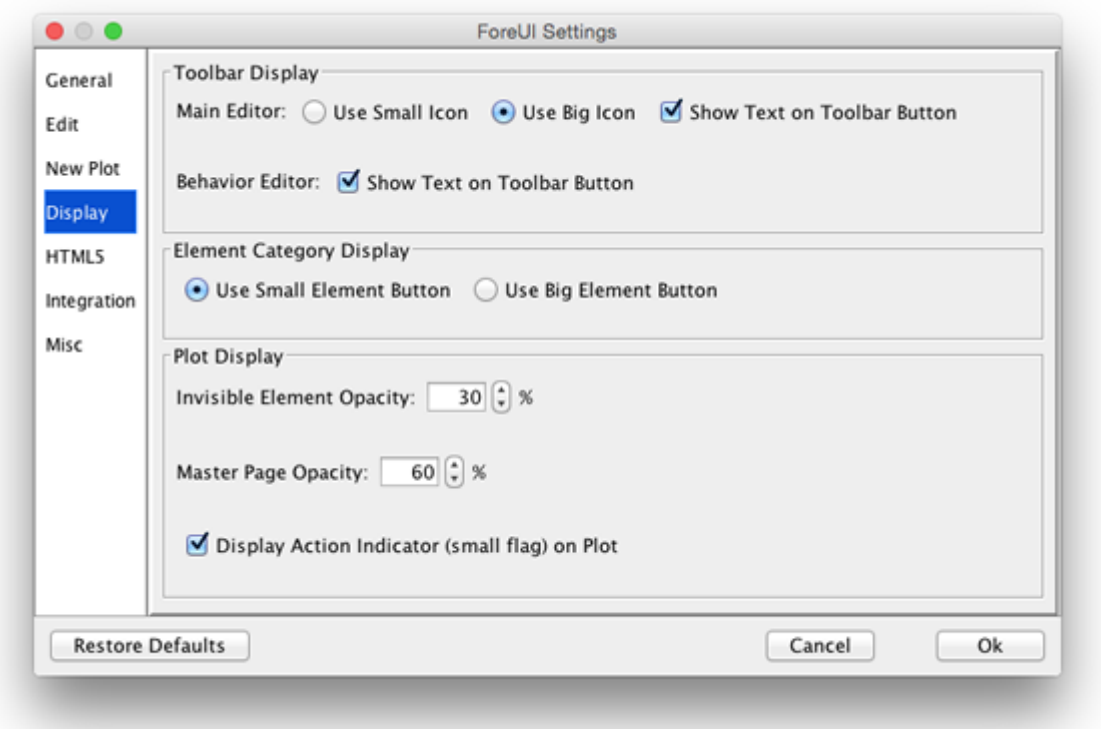
Default Plot Size: the initial plot size for newly created plot.

Background Color: the background color of the newly created plot.

Rumple Effect: turn on/off/adjust the rumple effect in newly created plot.

When you create a new plot, ForeUI will try to use these parameters. After the plot is created, you can still configure the plot's parameters.

Display

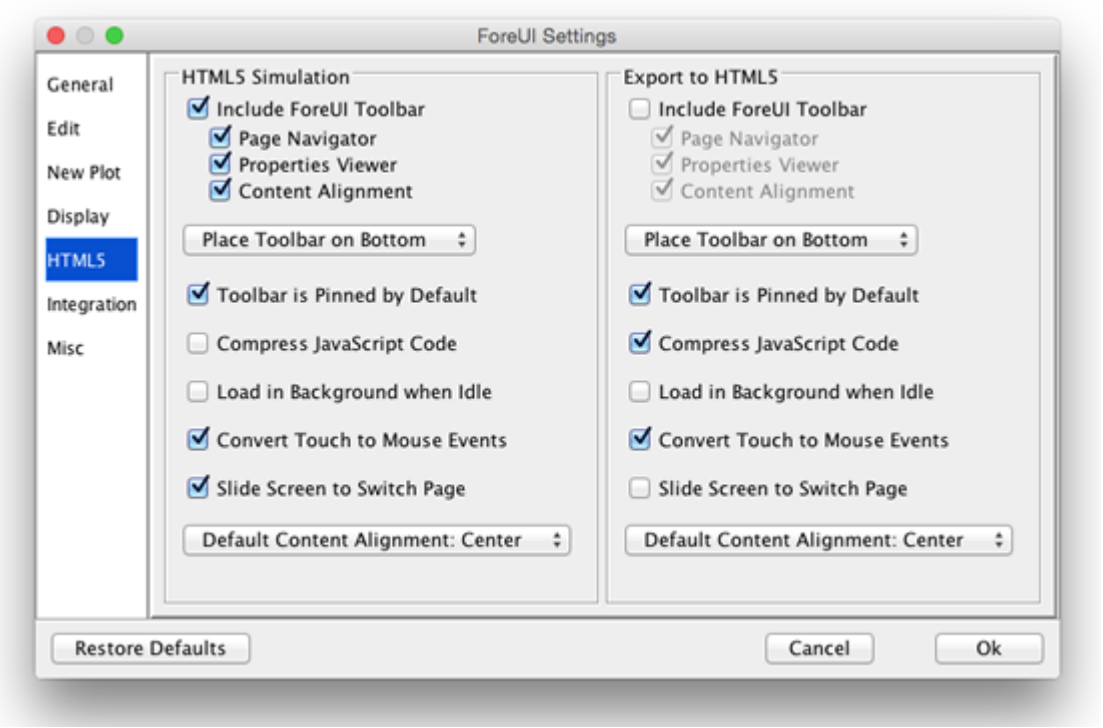


There are quite a lot options in this tab, they are all about the display / rendering.

- **Toolbar Display:** you can customize the look of the toolbar in main window.
  - **Use Small Icon:** show small icons (16 x 16) in the toolbar.
  - **Use Big Icon:** show big icons (32 x 32) in the toolbar.
  - **Show Text on Toolbar Button:** whether to show text in toolbar button.
- **Element Category Display:** you can change the button size of element categories.
  - **Use Small Element Button:** show small buttons (50 x 50) for elements listed in category.
  - **Use Big Element Button:** show big buttons (100 x ??) for elements listed in category.
- **Plot Display:** you can decide how to paint the invisible elements, master pages and the action indicators.
  - **Invisible Element Opacity:** 0% for totally invisible, 100% for totally visible. Default value is 30%.
  - **Master Page Opacity:** 0% for totally invisible, 100% for totally visible. Default value is 60%.

- **Display Action Indicator:** whether to display small red flag on top right of the element that has action defined.

HTML5



In this tab you can make some customization on HTML5 export/simulation. The parameters here will be default for all plots, however you can override these parameters in the plot settings.

Remarks: the options in this tab are available for paid users only.

Include ForeUI Toolbar: when enable this option, the HTML5 simulation will include a Toolbar, which contains:

- **Page Navigator:** allows you to switch to any page in the plot.
- **Properties Viewer:** provides a view for all system/user defined properties.
- **Content Alignment:** set the content alignment to left, center or right.

Toolbar Placement: you can decide where the toolbar should be placed (top, left, bottom or right).

Toolbar is Pinned by Default: if this option is selected, the toolbar will be pinned by default, otherwise the toolbar will be hidden automatically when idle.

Compress Javascript Code: the Javascript code will be compressed if this option is selected, which can give you better loading/running speed.

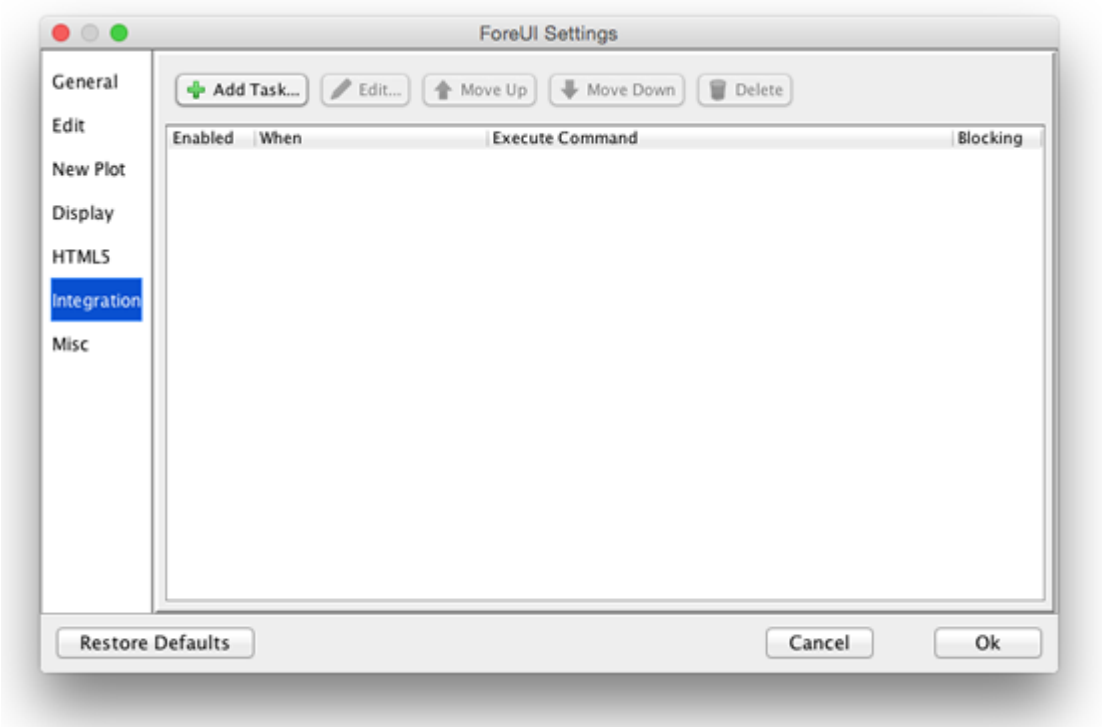
Lazy Loading: this option allows the simulation to show the GUI ASAP and keep loading the remaining resource in the background. (very useful for big plot)

Convert Touch to Mouse Events: if you plan to run the HTML5 simulation in touch screen devices (smart phone or tablet), you may want to turn this option on. This will convert the touch events to normal mouse events, thus you can get similar experience than running simulation on desktop computer.

Slide Screen to Switch Page: this option is also for running simulation in touch screen devices. When it is turned on, you will be able to switch page by sliding the touch screen.

Default Content Alignment: set the default alignment of the content here. By default it is center aligned.

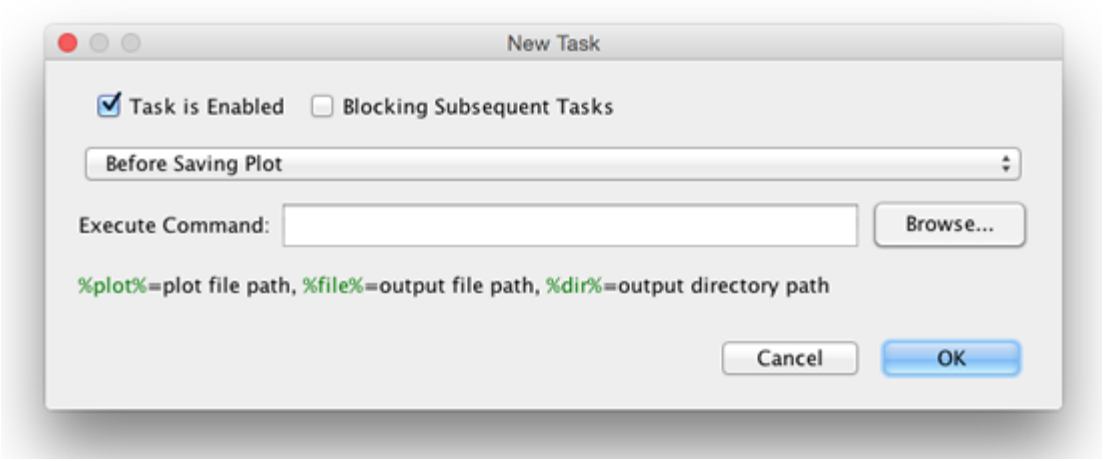
Integration



In this tab you can configure task that will be executed when certain event happens. This is very useful for integrating ForeUI with other tools.

Remarks: the options in this tab are available for paid users only.

You can click "Add Task..." button to add a new task:



The new task will be set to "Enabled" by default, which means it will take effect immediately after the settings. You can also disable it and reserve it for future usage.

If the task is set to "Blocking" mode, it will block the subsequent task(s).

Here you can specify the event and command that needs to be executed when the event happens. In the command, you can use **%plot%** to represent the current editing plot file path, **%file%** to represent the output file (HTML5 file) path, and **%dir%** to represent the output directory path.

You can define as many task as you need, and use the buttons on toolbar to manage them.

Miscellaneous

Memory Usage: You can specify the maximum memory usage of ForeUI (the default 500 MB should be enough in the major of cases). You can also enable the "Cache Image Data in Memory" option here, which will cache the image data in memory instead of file system. This will take more memory but it is needed if any Antivirus system disallows ForeUI to use the file cache.

Auto Backup: If the auto backup is enabled, the unsaved plot will be backup for certain intervals.

Workspace: Here you can change the location of the workspace, which will cache the data for editing plot temporarily.

Recent File Record: You can also specify the maximum number of recent file records (which will be listed under "File" menu).

Proxy for Internet Access: You may need to specify a proxy here if you need that for internet access.

20.1 Use Hotkey in ForeUI

You can use hotkey to accelerate your operations in ForeUI. Currently the hotkeys below are available:

Hotkey	Effective Region	Description	Comment
Ctrl + N	Main Window	Create a new ForeUI plot file and open it.	Command + N in Mac OS X
Ctrl + O	Main Window	Browse an existed ForeUI plot file to open.	Command + O in Mac OS X
Ctrl + S	Main Window	Save the current editing plot file.	Command + S in Mac OS X
Shift + Ctrl + S	Main Window	Save plot as ...	Shift + Command + S in Mac OS X
Shift + Ctrl + Y	Main Window	Launch slide show.	Shift + Command + Y in Mac OS X
Ctrl + R	Main Window	Run simulation in your default web browser.	Command + R in Mac OS X
Ctrl + Z	Main Window	Undo the last editing action.	Command + Z in Mac OS X
Ctrl + Y	Main Window	Redo the last undo action.	Shift + Command + Z in Mac OS X
Ctrl + X	Main Window, Text Editor, Behavior Editor	Cut the selected content.	Command + X in Mac OS X

Ctrl + C	Main Window, Text Editor, Behavior Editor	Copy the selected content.	Command + C in Mac OS X
Ctrl + V	Main Window, Text Editor, Behavior Editor	Paste the selected content.	Command + V in Mac OS X
Delete or Backspace	Main Window, Text Editor, Behavior Editor	Remove the selected content.	Backspace in Mac OS X
Ctrl + D	Main Window	Open behavior editor for selected element or current page.	Command + D in Mac OS X
Ctrl + Right	Main Window	Select next element in plot.	Command + Right in Mac OS X
Ctrl + Left	Main Window	Select previous element in plot.	Command + Left in Mac OS X
Ctrl + A	Main Window	Select all elements in plot.	Command + A in Mac OS X
Ctrl + Q	Main Window	Quit ForeUI.	Command + Q in Mac OS X
Page Down	Main Window, Simulation in Browser	Switch to next page.	Fn + Down in Mac OS X
Page Up	Main Window, Simulation in Browser	Switch to previous page.	Fn + Up in Mac OS X
Left, Right, Up, Down	Plot Editing Area	Move the selected elements in plot.	
Hold Shift + Single Click	Plot Editing Area	Add/remove element to current selection.	
Hold Shift + Single	Line and Polygon Editor	Draw horizontal, vertical	

Click		or 45 degree line.	
Hold Ctrl + Single Click	Plot Editing Area	Add/remove element from current selection.	
Hold Shift + Drag & Drop	Plot Editing Area	Force to drag an area to select elements inside	
Hold Shift + Drag & Drop	Selection Border	Resize selected elements, with aspect ratio locked.	
Right Click	Plot Editing Area, Behavior Editor	Show context menu.	
Enter	All windows that have "Ok" button	Emulate pressing the ok button.	
Escape	All windows that have "Cancel" button	Emulate pressing the cancel button.	
Ctrl + Equals	Main Window	Zoom In	Command + Equals in Mac OS X
Ctrl + Minus	Main Window	Zoom Out	Command + Minus in Mac OS X
Ctrl + 1	Main Window	Zoom to 100%	Command + 1 in Mac OS X
Ctrl + 2	Main Window	Zoom to 200%	Command + 2 in Mac OS X
Ctrl + 3	Main Window	Zoom to 300%	Command + 3 in Mac OS X
Ctrl + 4	Main Window	Zoom to 400%	Command + 4 in Mac OS X

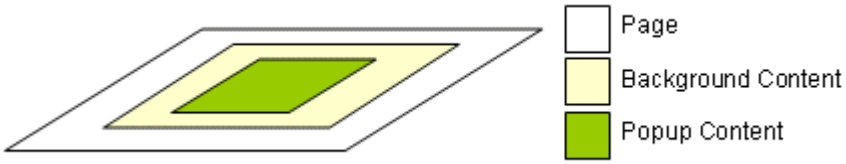
Ctrl + 5	Main Window	Zoom to 50%	Command + 5 in Mac OS X
Ctrl + Mouse Wheel	Main Window	Zoom In / Out	
Ctrl + E	Main Window	Show/Hide the Elements Panel	Command + E in Mac OS X
Ctrl + I	Main Window	Show/Hide the Images Panel	Command + I in Mac OS X
Ctrl + L	Main Window	Show/Hide the Outline View	Command + L in Mac OS X
Ctrl + F	Main Window	Show/Hide the Element Search Panel	Command + F in Mac OS X
Ctrl + P	Main Window	Show/Hide the Page Management View	Command + P in Mac OS X
Ctrl + B	Main Window	Show/Hide the Behavior Editor View	Command + B in Mac OS X
Ctrl + T	Main Window	Show/Hide the Tools Panel	Command + T in Mac OS X
Ctrl + J	Main Window	Show/Hide the Custom Properties View	Command + J in Mac OS X
Ctrl + K	Main Window	Show/Hide the System Properties View	Command + K in Mac OS X
Ctrl + G	Main Window	Group selected elements.	Command + G in Mac OS X
Ctrl + Shift + G	Main Window	Ungroup selected elements.	Command + Shift + G in Mac OS X

21.1 Simulate Popup Window

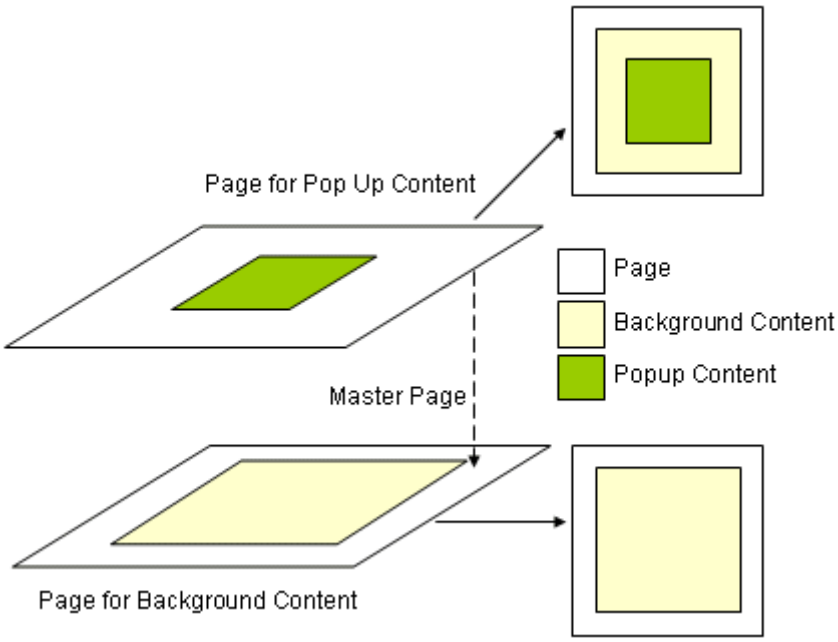
When we create the prototype for desktop or web applications, sometimes we need to simulate the popup window.

Basically there are two methods to simulate the popup window:

1. Place the invisible popup window in the same page, change it to visible when need to pop up the window.



2. Place the popup window in another page, which use the current page as master page. Switch page when need to pop up the window.



Both methods can work as expected.

The first approach can do all things in one page; it will be good if you like to include the pop up window in a custom element (as custom element cannot have multiple pages). But you will feel hard to make changes to the content that covered by the popup window.

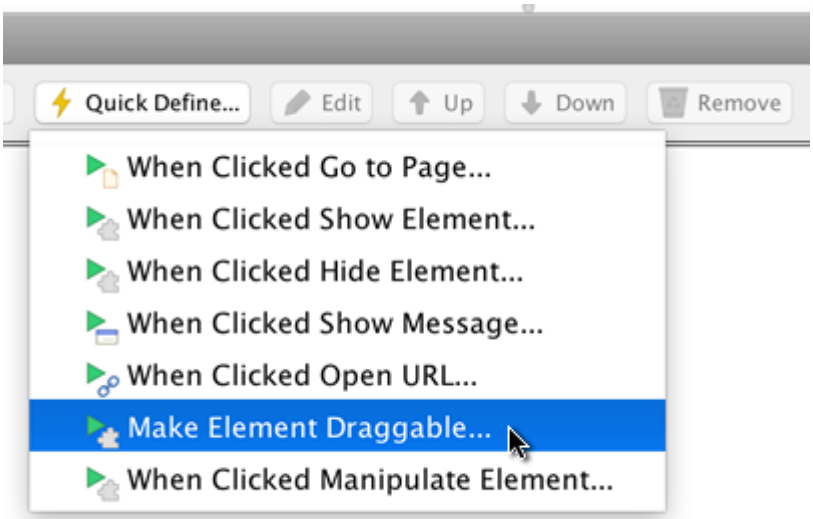
The second approach is using two pages to do the job. It makes pages very clean since there is no overlapped element. But it is not possible to pack the pop up behavior into a custom element.

22.1 Simulate Drag and Drop Behavior

If you want to simulate a draggable window, just check the "Draggable" option for the Window element and that's all.

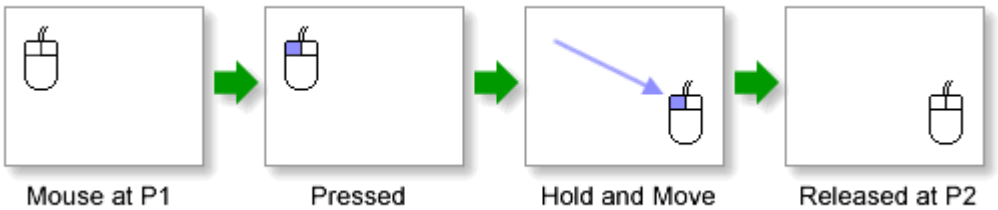
What if you need to simulate the drag and drop behavior on other elements? That's also possible. All elements can become draggable, as long as you handle the mouse events properly.

Thanks to the “Quick Define…” button in the behavior editor, you can make any element draggable very easily:



It will automatically generate the behavior for the element, so it will become draggable in the HTML5 simulation. This is a very easy to use and you can just use it without understanding how it works. However, if you want to understand it and maybe fine adjust its behavior, please read on.

Before going deep into the details, let’s take a look at the analysis of drag and drop behavior (drag from point P1 and drop to point P2):

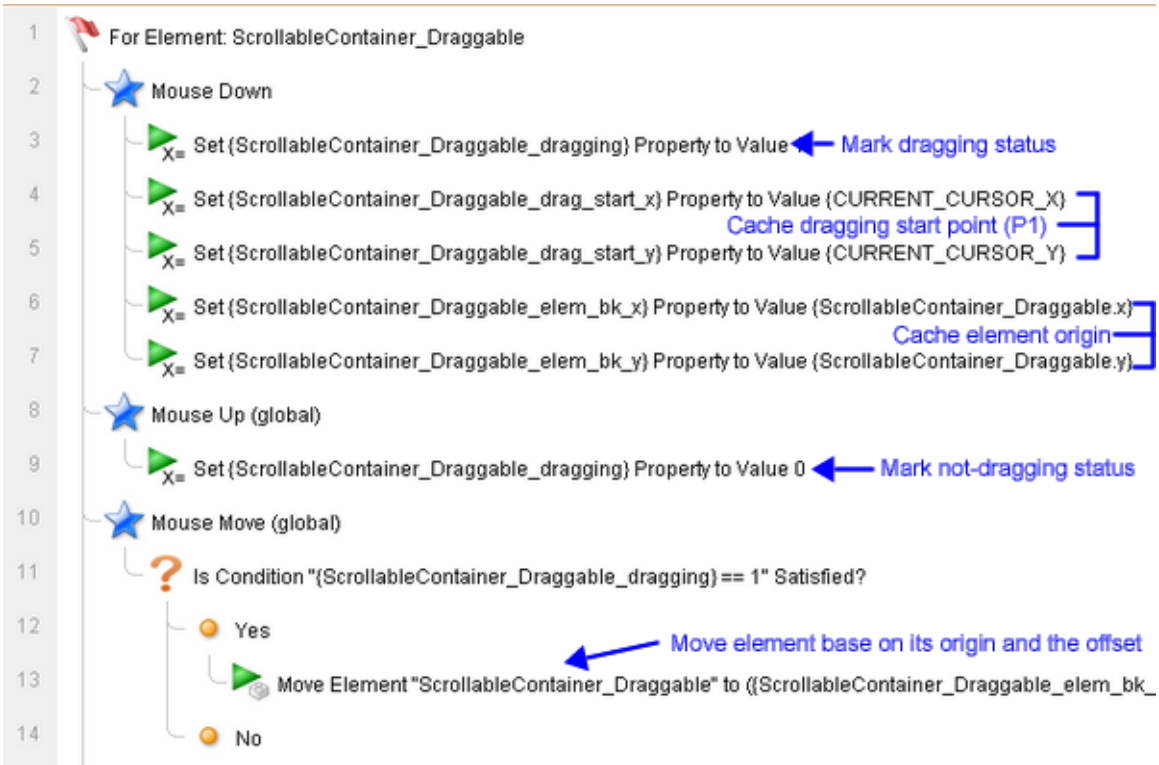


During this process, there will be three mouse event triggered orderly: Mouse Down, Mouse Move and Mouse Up. To simulate the drag and drop behavior, we need to handle these mouse events in this way:

- Mouse Down:
  - Mark the current status as "dragging".
  - Cache the current location P1 as the dragging start point
  - Cache the current location of the element that being dragged as the element origin
- Mouse Move:
  - Calculate the current offset against the dragging start point (P1), and move the element base on its origin and the offset.
- Mouse Up:
  - Mark the current status as "not dragging".

You can find a good example of drag and drop simulation in ForeUI Store, it is a custom element named "Draggable Container" that supports the drag and drop in simulation. You can see how it works in your browser.

If you download that custom element and open it with ForeUI, you will see it becomes draggable because it handles the mouse events listed above. Here is its behavior definition:



As you can see, the dragging status, dragging start point and element origin are stored in global properties. The most important action is the "Move Element" action, it move the TextBox to this new location:

NewX = ElementOriginX + CurrentCursorX - DraggingStartX

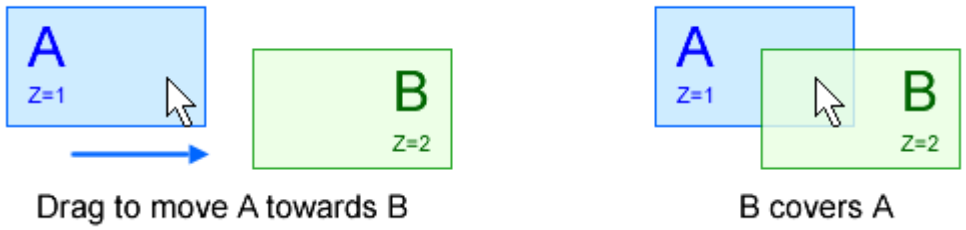
NewY = ElementOriginY + CurrentCursorY - DraggingStartY

You may notice that some of the three mouse events are global mouse event, while others are not. Specifically speaking the "Mouse Move" and "Mouse Up" events are global mouse events, while the "Mouse Down" event is a native mouse event. The table below compares the differences between global and non-global mouse events:

Events	Native Mouse Event	Global Mouse Event
Mouse Down	Mouse is pressed on the element.	Mouse is pressed anywhere.
Mouse Move	Mouse is move on the element.	Mouse is moved anywhere.
Mouse Up	Mouse is released on the element.	Mouse is released anywhere.

You can see the global event will be triggered no matter where the mouse cursor is, while the native mouse event must be triggered on the element. Here is an example to explain why we need to use global "Mouse Move" and "Mouse Up" events instead of native ones:

Let's say we have two elements: A and B. B has a higher z value. When we drag element A towards B, B will cover A since B has higher z value. In this case, element A cannot receive the "Mouse Move" and "Mouse Up" events since element B intercept them. To avoid this bad situation, we should use global "Mouse Move" and "Mouse Up" events.

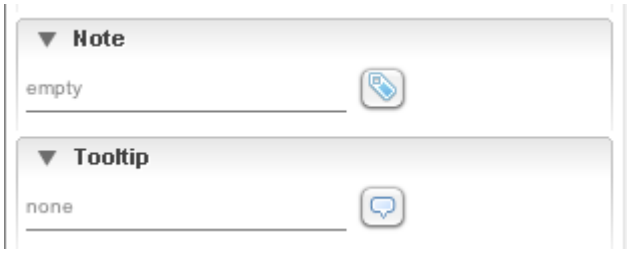


Once you understand the mechanism to simulate drag and drop, you will find that all elements can be dragged if you'd like them to.

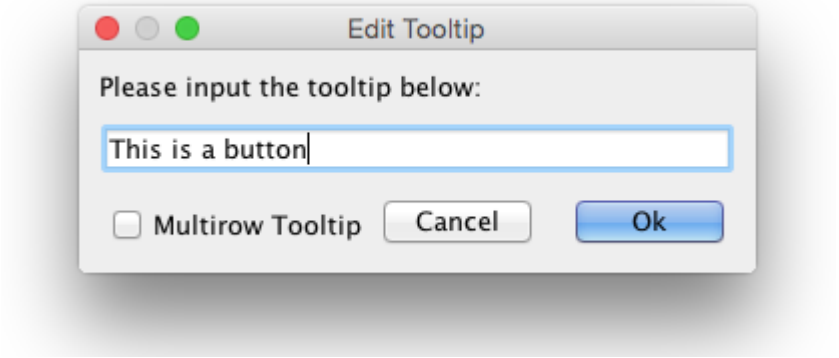
23.1 Define Element's Tooltip and Note

Any element can have its own tooltip and note. The difference between tooltip and note is that, tooltip will be displayed in the HTML5 simulation, when you move your mouse cursor hovering on the element, while note will not be displayed and it just helps you to remember some information about the element.

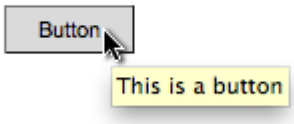
In the tools panel, you can see two categories named "Tooltip" and "Note":



You can click the  button on the right to input the tooltip for element.

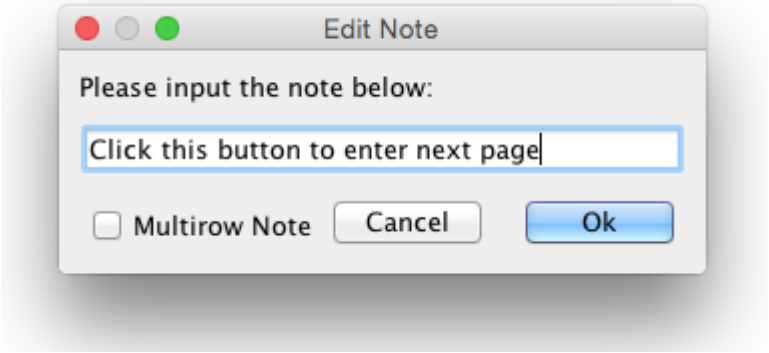


If the tooltip of element is not empty, the tooltip will be displayed in simulation, when you hover your mouse cursor on the element:



Tooltip of element could be set or removed via action Set Tooltip.

If you click the  button, you can input the note for the element.



24.1 Make Your Plot/Simulation Smaller

This topic seems to be prepared for advanced ForeUI users, as your plot/simulation would not go big until you use ForeUI for a while. However the best practice is to learn those tricks to keep your plot/simulation small and neat at the beginning, otherwise you will feel very frustrated to spend a lot of time on optimizing your big plot in the future. Please keep in mind that, 90% of the "big plot" issue could be avoided or fixed by refactoring your plot, so the better you know of these tricks, the less chance you will face the issue.

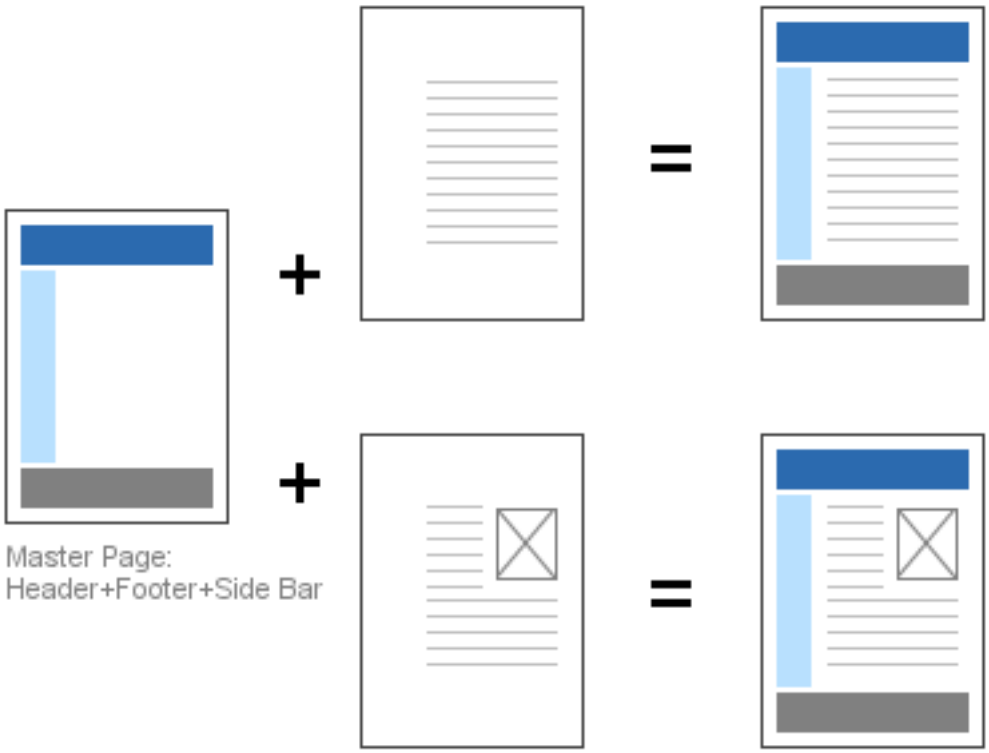
How big is big? When your plot takes more than 10 seconds to save, or the simulation takes minutes to load in web browser, you know your plot/simulation is really big. You'd better not to add more content into the plot before optimizing its size. Otherwise, your efficiency will go down a lot as the saving the plot can take a minute, or loading a simulation can take a hour. You will find it worth every minute you spend on optimizing your plot size.

Here are some tricks to make your plot smaller:

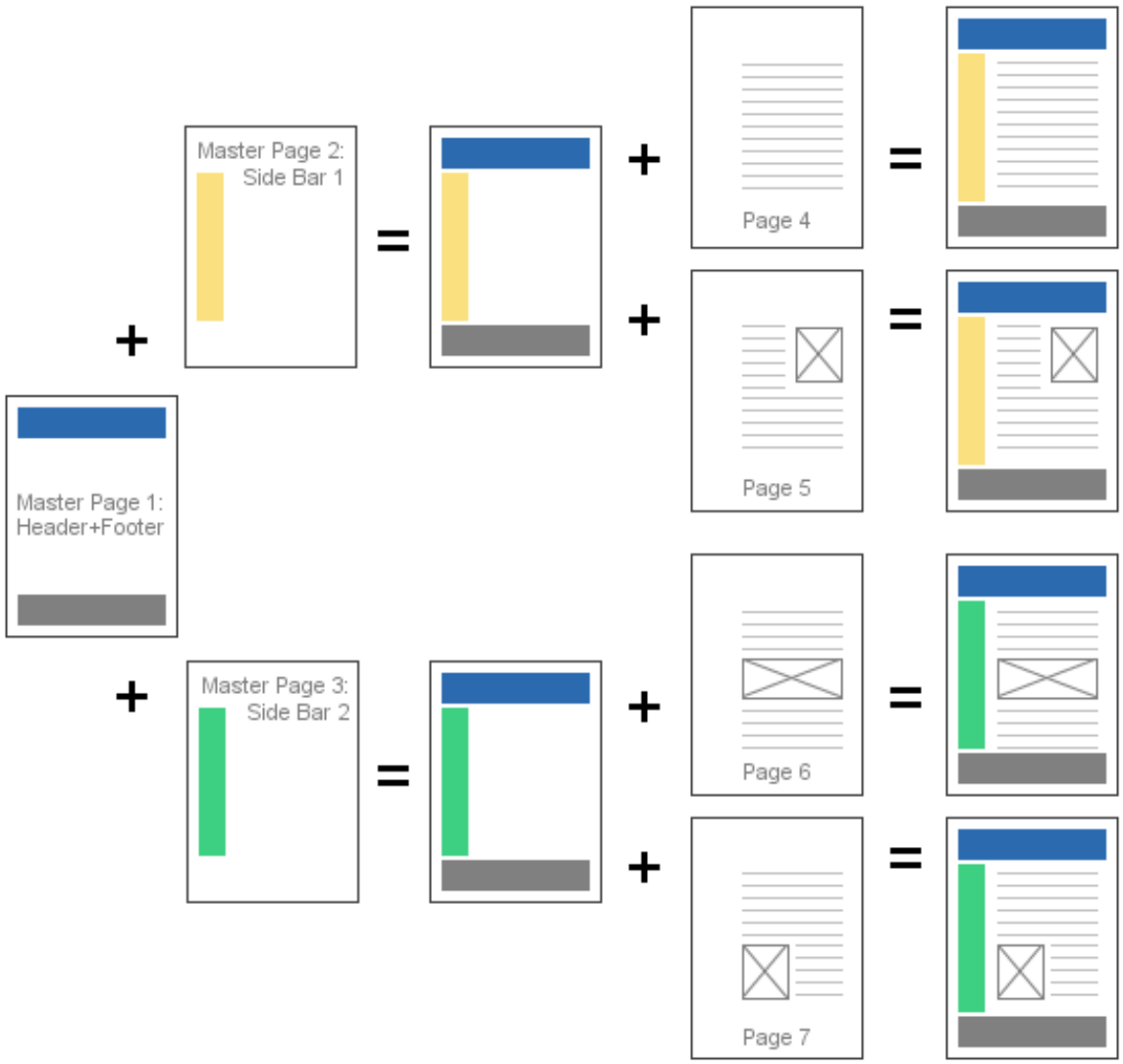
Use Master Page to Avoid Duplicated Content

Usually there will be some common parts between pages in your plot. Many people choose to duplicate elements or even the entire page to make sure every page have those common parts. However that's a very bad idea since it generates a lot of unnecessary duplication of content (thus make your plot big), and you will have to synchronize those duplicated content if you want to change a part of it.

You are encouraged to move those common parts to a master page, which can be shared by other pages. This trick is frequently used for pages that have the same header, footer or side bar. For example, all pages in your plot have the same header, footer and sidebar. So you can place them in a page, and all other pages use this page as the master page. Thus you can just manage the center content, and no need to worry about the header, footer and sidebar for every page. In case you want to change the style of header, footer or sidebar, you just do so in the master page, and all pages will have the updated content. If you don't know how to set master page, here is the how-to.



Continue on the example above, what if some pages have different side bars, while they have the same header and footer? Please notice that master page is not a special kind of page, it is just a normal page that get referenced by other pages. So a master page can have its own master page as well. You can consider using this structure:



In the example above, three pages are used as master pages. Master Page 1 is shared by Master Page 2 and 3, while Master Page 2 is shared by two pages (Page 4 and 5) and Master Page 3 is shared by another two pages (Page 6 and Page 7). This hierarchy should meet the requirements for most web site/application prototyping.

By using master pages, you can avoid duplicating content in the plot, and significantly reduce your plot size.

Use Reference Element to Avoid Duplicated Elements

Besides using master page, there is another way to reuse existing content: the Reference element. The Reference element is listed in the Elements Panel, and has the icon like this:

After adding the Reference element into the plot, you will need to specify its target element.

By specifying the target element, the Reference element will then copy the look and behavior of the target element. Different than master page, a Reference element can be placed anywhere in your plot, so it is very suitable for reusing content in different locations.

You can choose only one element as the reference target. In case you want to reference more elements a time, you could group them before setting as the target.

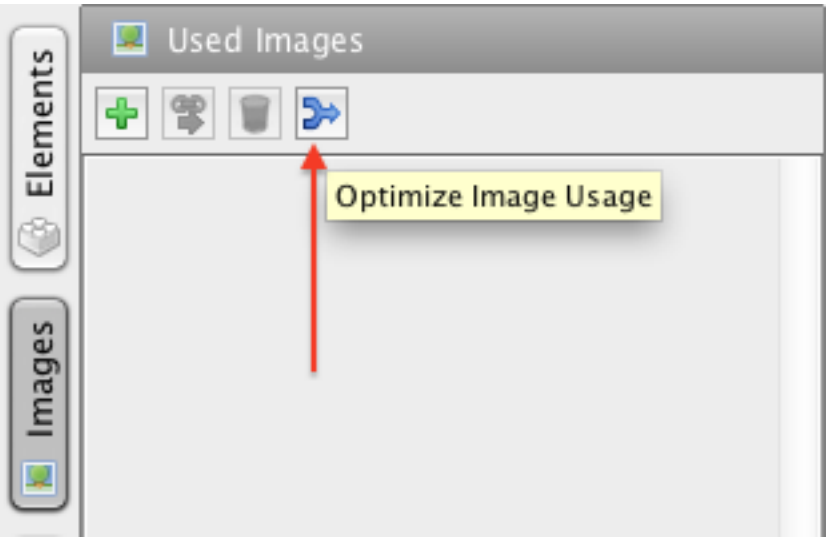
Using Reference element can also reduce the duplicated content in your plot, thus make your plot smaller.

Optimize Image Usage

We noticed that many plots is big because they use a lot of (big) images. So if you wish to keep your plot small, please make sure your plot only includes the images that are necessary. Nobody would like to include unnecessary images in the plot, but sometimes you may do this unconsciously. Consider this case: you add an Image Box element and specify an external image file for it, after a while you delete that Image Box and forget it. But the image is still stay in the Images Panel, and will be saved to plot file even it is not used anymore. If you did this a lot, your plot will become big, because it contains many unused images in Images Panel.

What's more, sometimes you may include duplicated images in your plot, which is imported from the same image file in your hard drive. The better way is to remove the duplicated images and redirect all the links to the only remaining copy.

Since ForeUI V3.0, you have a tool in the Images Panel to optimize the image usage. Just click the button and it will detect all the duplicated and unused image, and fix them for you. It is easy and cost nothing, so it please remember to give it a try if you feel your plot is becoming big.



Convert Static Elements into ClipArt

You can convert selected elements into ClipArt. Many elements in your simulation are actually static, if you convert them into a single picture, your simulation will be much smaller and the DOM structure will be significantly simplified, and the loading time will be reduced a lot.

The most suitable scenario to use ClipArt is to create charts with those basic shape elements, which are all static. Instead of letting ForeUI to generate the DOM objects for each static element, generating a big static picture to hold all of them is much better.

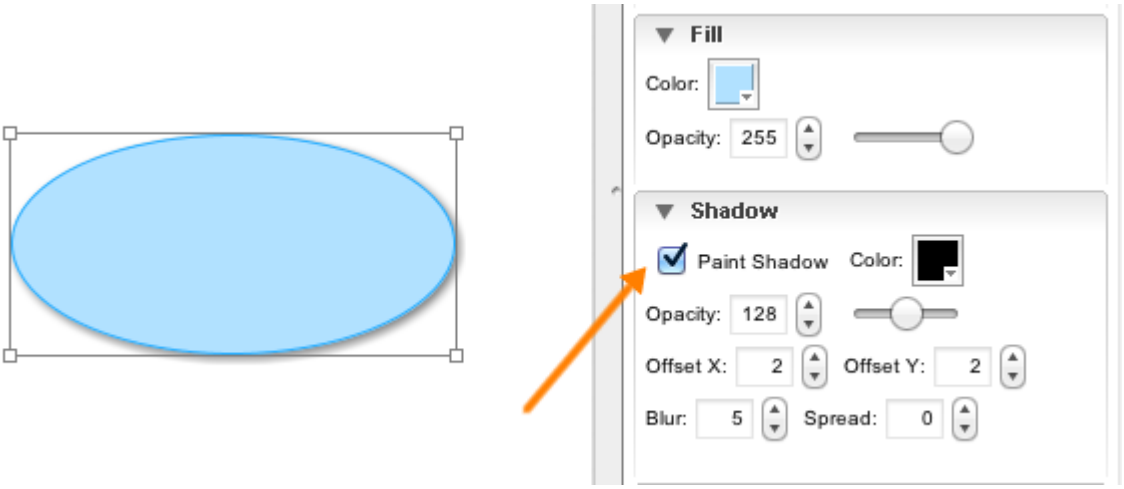
Here are two simulations for the same content, the only difference is that, one of them converts elements into ClipArt, while the other one doesn't.

Simulation without ClipArt Conversion

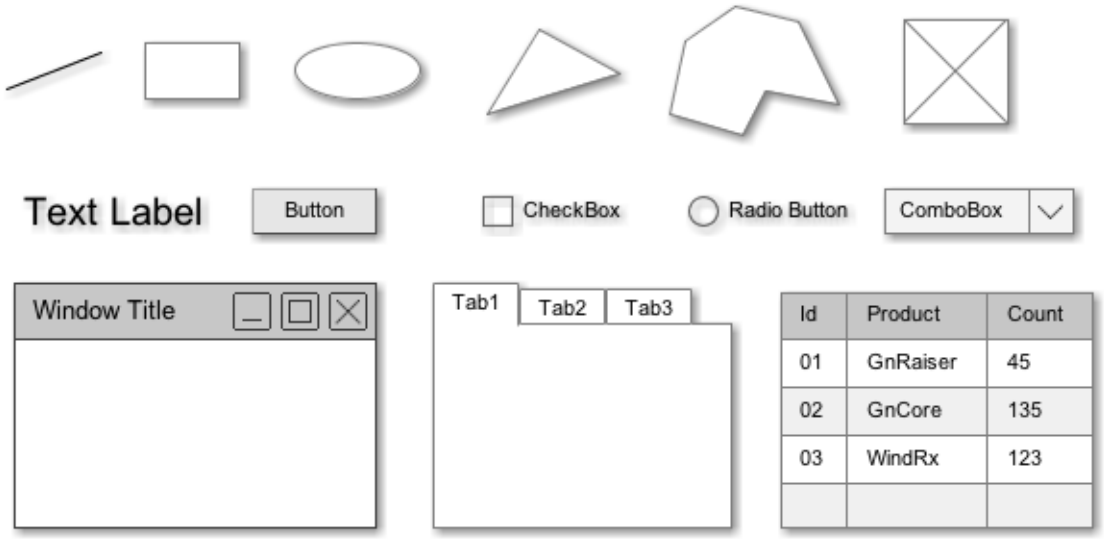
Simulation with ClipArt Conversion (much smaller and faster)

25.1 Apply Shadow on Elements

ForeUI supports shadow on all elements. In the tools panel you will see a new “Shadow” category, and there is a “Paint Shadow” checkbox that allows you to enable the shadow rendering. Once you enable shadow on element, you can also change the shadow color, opacity, offset, blur and spread.



All elements, including basic shapes and interactive controls can have their shadow. The shadow will also be applied in HTML5 simulation (thanks to the shadow attributes in CSS).



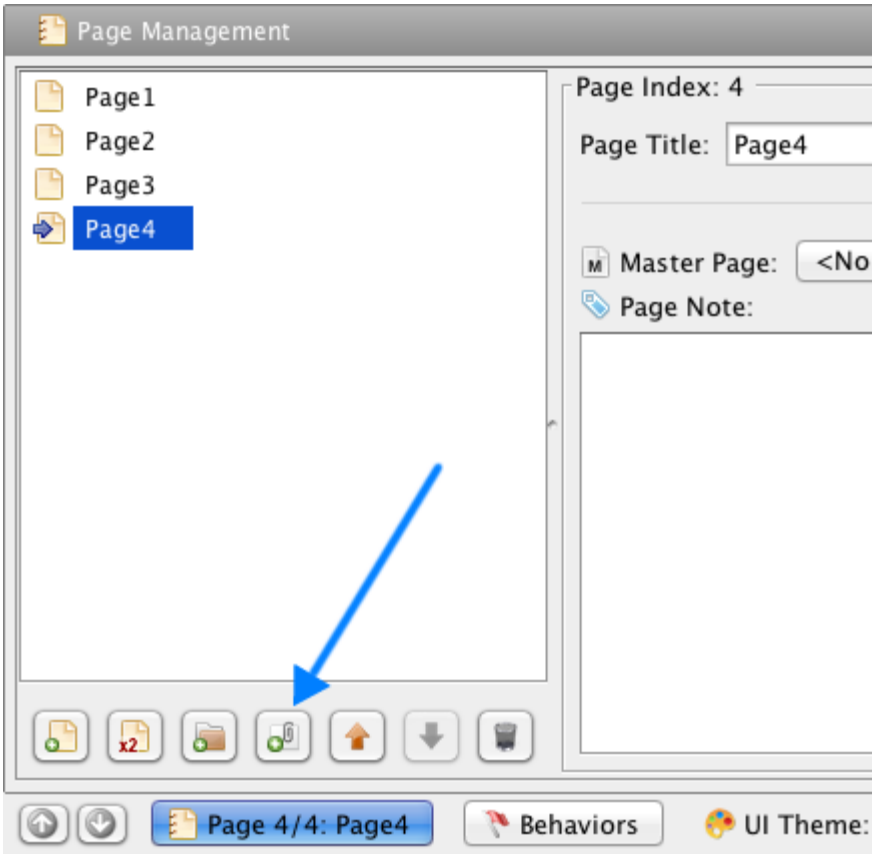
Some UI themes may also apply shadow to certain elements by default, such as Window, Post-It and Balloon elements.

26.1 Add External Files into Your Plot

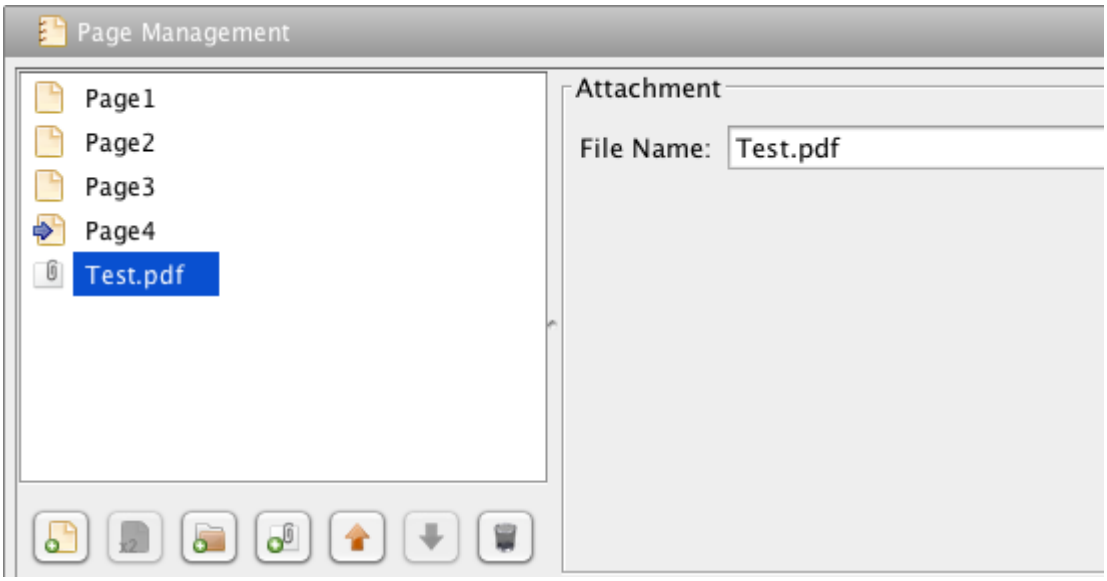
Since V4.0, ForeUI allows you to add any file into your plot, such as video, PDF document, and ZIP archive etc.. Once you add the file into your plot, it will be stored in your .4ui file and get extracted to output directory, when you export your plot to HTML5 simulation.

How to add File?

The new “Add Attachment” button in the page management view is added for this purpose. Once you click this button, you will be asked to choose the file to add. Any type of file can be added, and there is no limitation on the files number. However, please keep in mind the files will be saved into your plot file (.4ui file). If you put too many files, or too big file into your plot, your .4ui file might become much bigger.



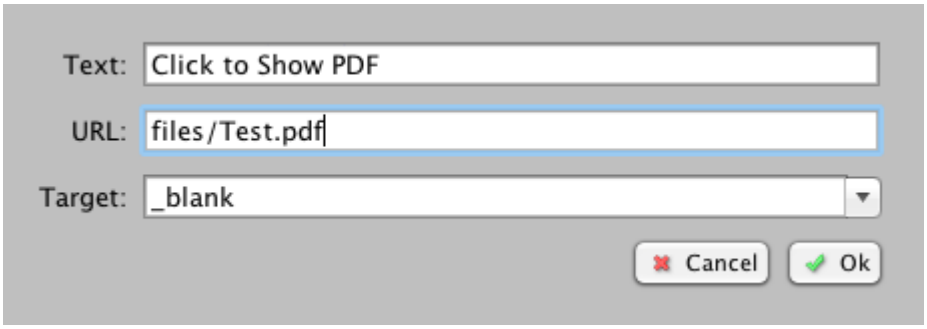
After you add the file into your plot, you can still change its file name. The file name must be unique in the plot, since all attached files will be exported to the same directory.



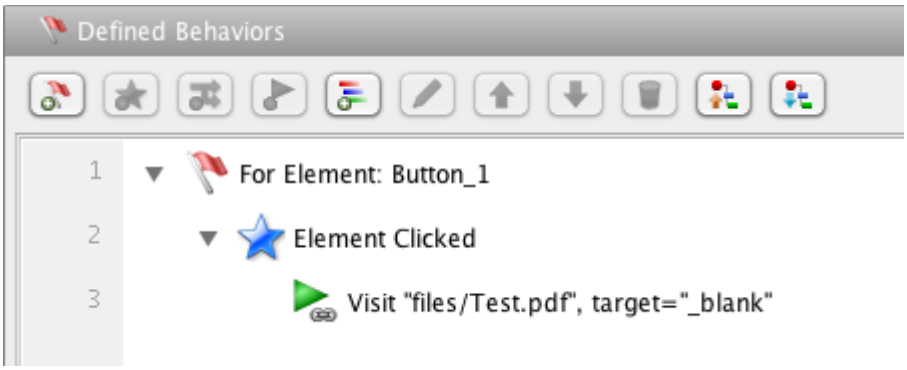
How to Access the Attached File in HTML5 Simulation?

Once you run HTML5 simulation, or export your plot to HTML5 simulation, all attached files will be extracted to the "files" folder under the simulation directory. So it is possible to access them via relative URLs in the simulation.

If you want to make a hyperlink to the attached file. You can use the Hyperlink element and specify its URL to "files/<file_name>". Please see the example below:



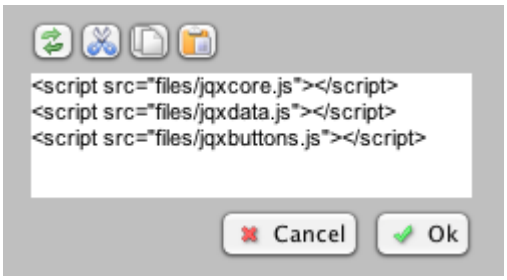
Or if you want to use Button element, it is possible to link an attached file to the button with “Go to URL” action. You just need to set its URL to “files/<file_name>”:



Of course you can link the file to any other elements with the same approach.

Attach Javascript Library File

Thanks to this new feature, now you can easily add Javascript library file as attachment, and then use Html element to import it into your HTML5 simulation. Below is an example for Html element that add jQWidgets library files into HTML5 simulation, all these Javascript files are attached into the plot beforehand:



Make Good Usage of It...

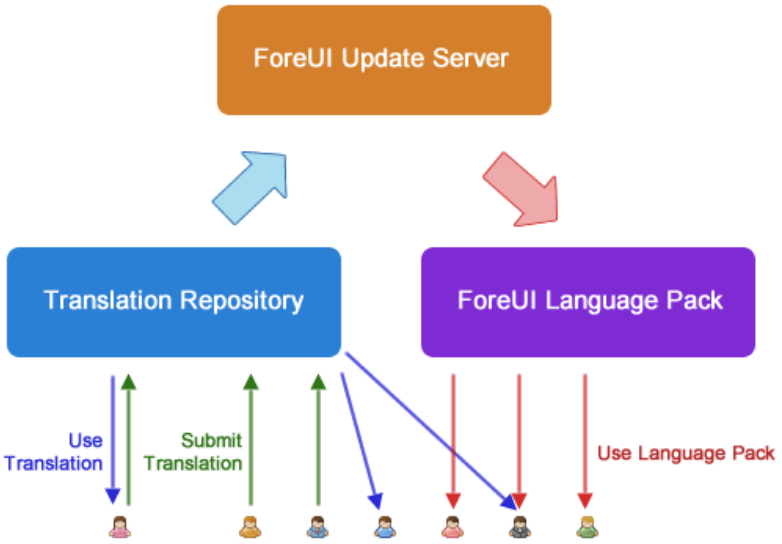
This feature is very useful, since it allows you to add any type of file into your plot (and HTML5 simulation).

With this feature, you can let people to download video or document from your HTML5 simulation, without manually modifying anything in the code.

You can even use this feature to add new Javascript library into your HTML5 simulation, so you can use the feature provided by other Javascript libraries.

27.1 Translate Text in ForeUI

You can localize ForeUI by yourself, which means you can translate the text you see in ForeUI into your native language. We intend to build a platform for it, so everyone can very easily contribute his/her translation. The figure below shows how this platform will work.



All translations will be public shared. When you try to translate a sentence or phrase, you may see that someone already did it before, and you can use his work instead of translating by yourself. Your translation may also be used by others, the more users choose your translation, the more credit points you will get, and those credit points can be used to get discount on buying/renewing ForeUI license in the future.

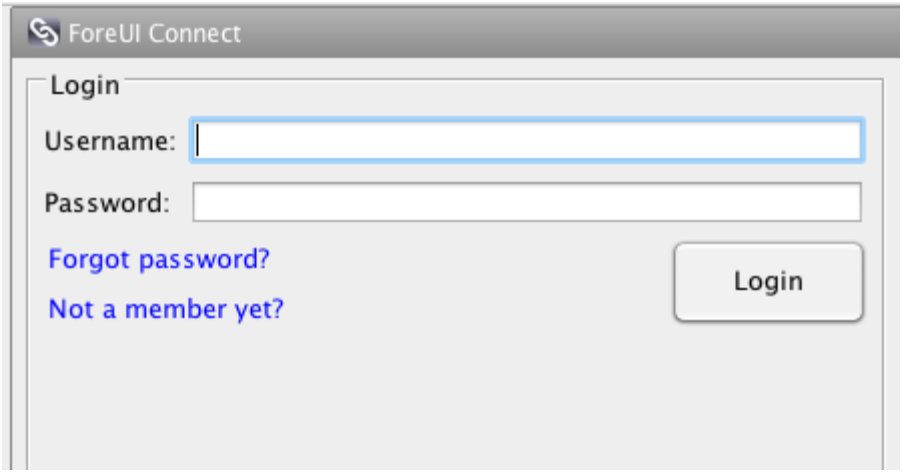
We will put those translations that are widely used into language pack and push it to all users during software upgrade.

How to use it?

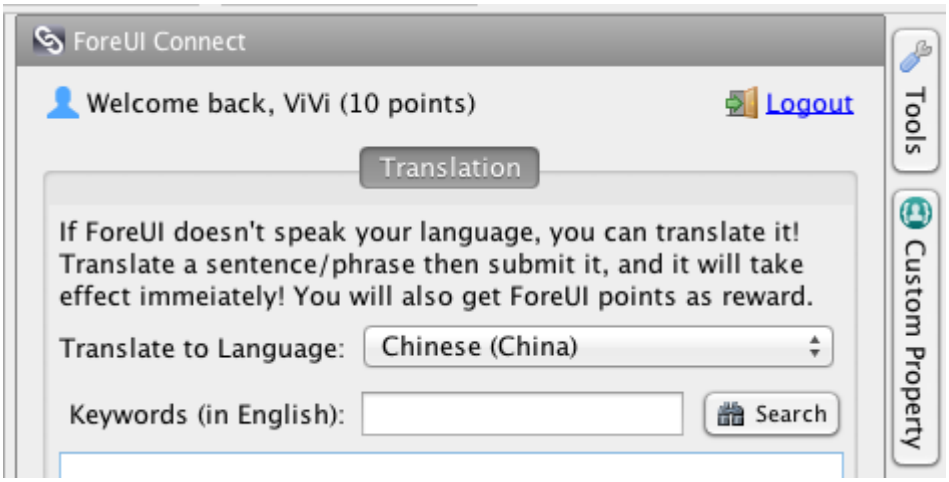
You may already notice the new tab 

on the right side of ForeUI's main window, and the view has title "ForeUI Connect", which is the entrance of this feature. ForeUI Connect is a platform for online activity, and translating ForeUI is just the first feature we implement on it.

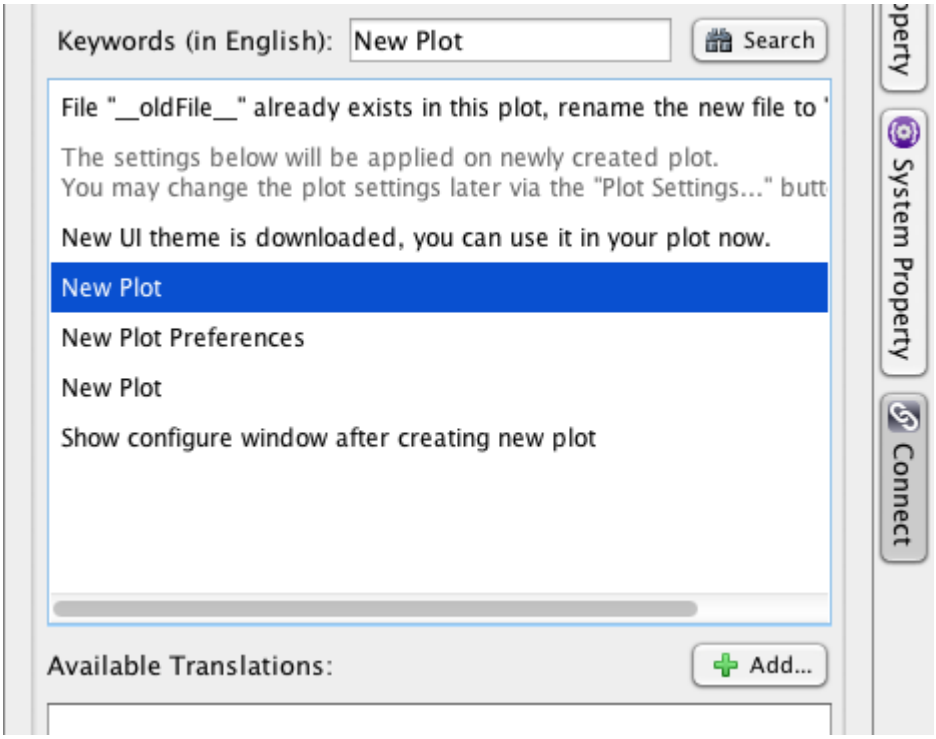
You will need to login before using this feature. The login account/password is the same you used to login our website. If you don't have an account yet, you can register one here, or click the "Not a member yet?" link in the view.



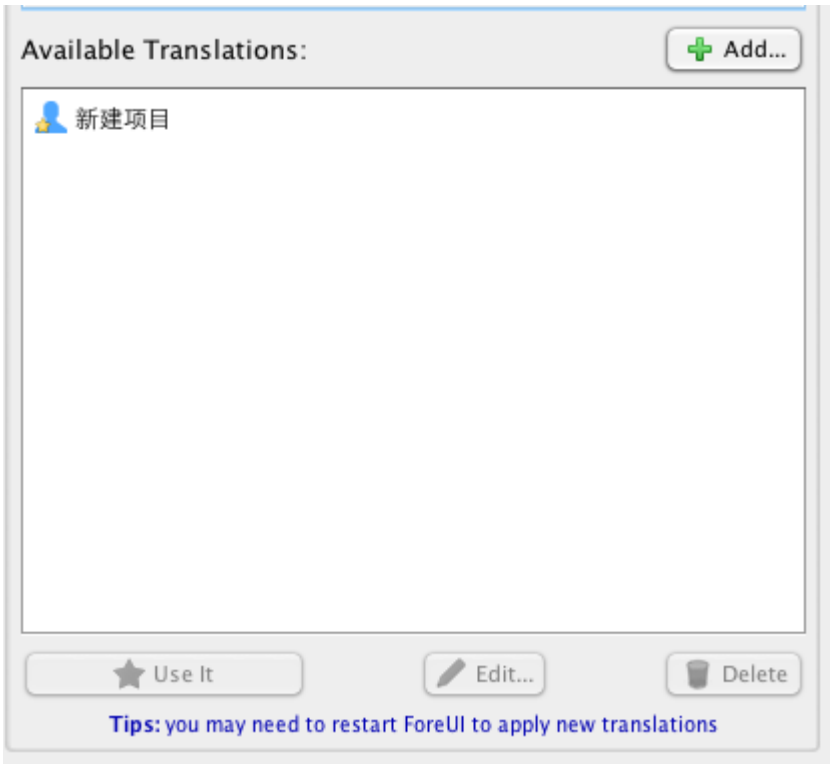
After login you will see the “Translation” tab inside the view, and you are ready to go:



Here you need to specify which language do you want to translation ForeUI to. Then you can input a keyword in English to find the text to translate. Below is an example, I input “New Plot” as the keyword and then click “Search” button, and 7 results a listed:

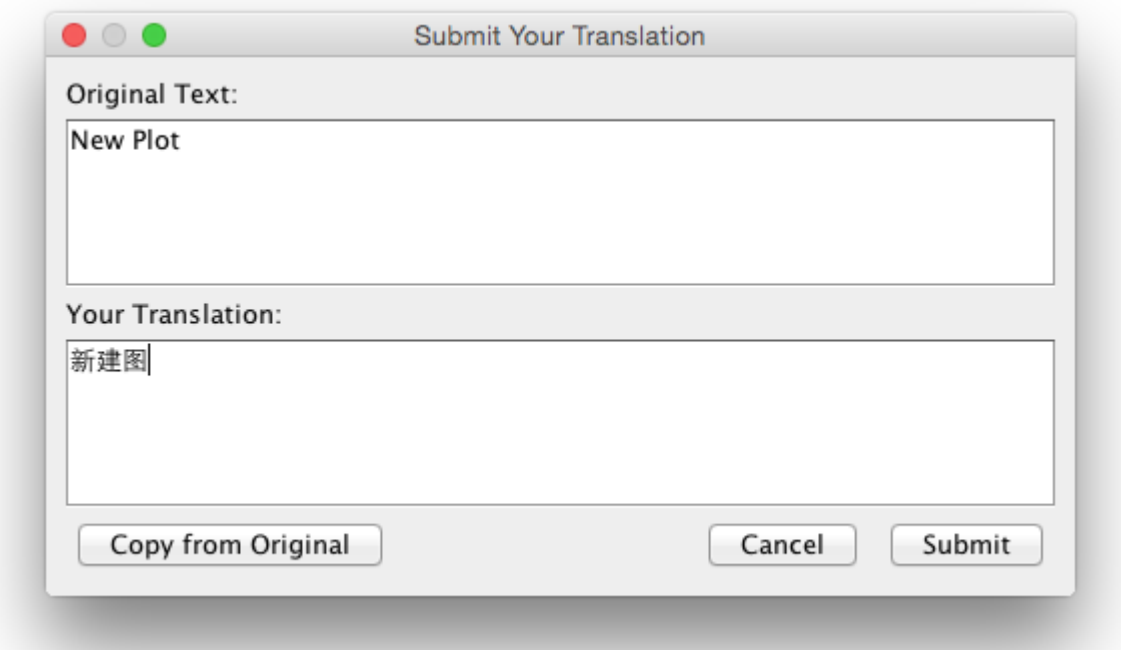


There are two “New Plot” phrases in the list, because we have two places in ForeUI that use this phrase (one is for menu/toolbar/welcome page, and the other is for the settings window). Now you can choose the phrase to translate. If someone else (or you) has ever submitted a translation for this phrase, you will see it in the “Available Translations:” list. As shown below:



If you like this translation, you can select it and the click the “Use It” button. The more users use a translation, the higher chance that it will get included in future language pack. If the translation was submitted by you, you can also edit or delete it here. If you don’t find any existing translation can satisfy you,

you can submit your own by clicking the “Add…” button above the list. A window will pop-up and accept your new translation.



After clicking the “Submit” button, congratulations! You just contributed one translation to the repository! Meanwhile you will be able to switch to the language that you are translating ForeUI to, and see the translated text in ForeUI immediately.

5. Examples

This section includes some useful examples, which help you to make use of ForeUI step by step. You will also learn some important skills and tricks to make better prototype with ForeUI.

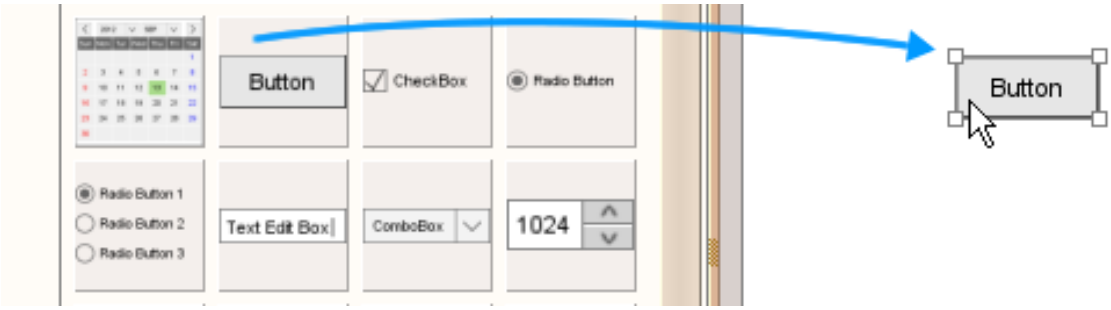
We strongly recommend you to go through these examples in the order they are defined, that will help you to follow up.

5.1 Example 1: Hello World Button

Run This Example in New Window

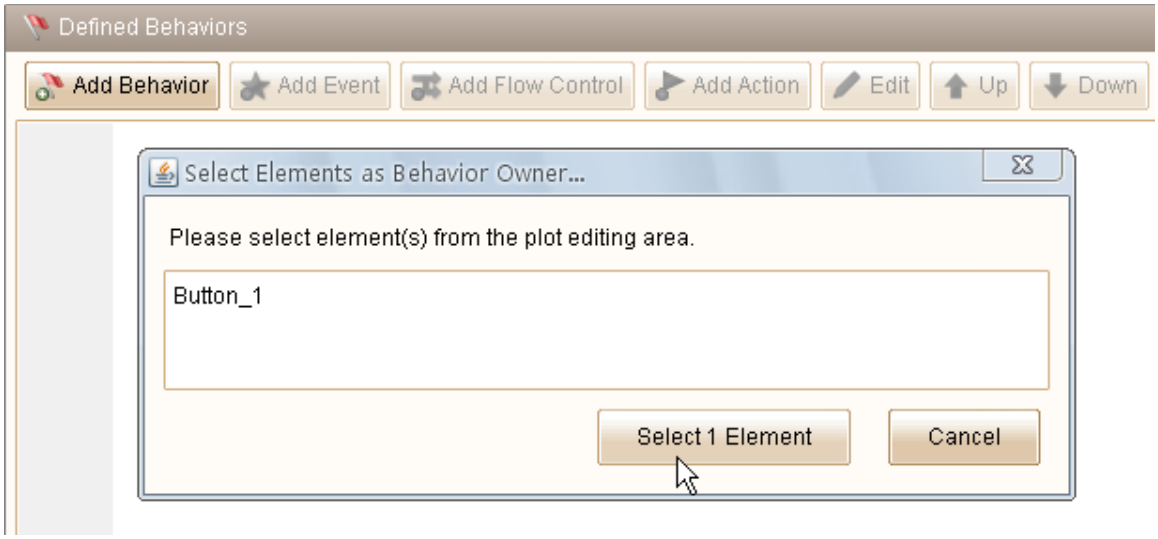
Now let's create a simplest example: when clicking the button, popup a window to say "Hello World!".

First of all we need to drag a button element to the editing area.

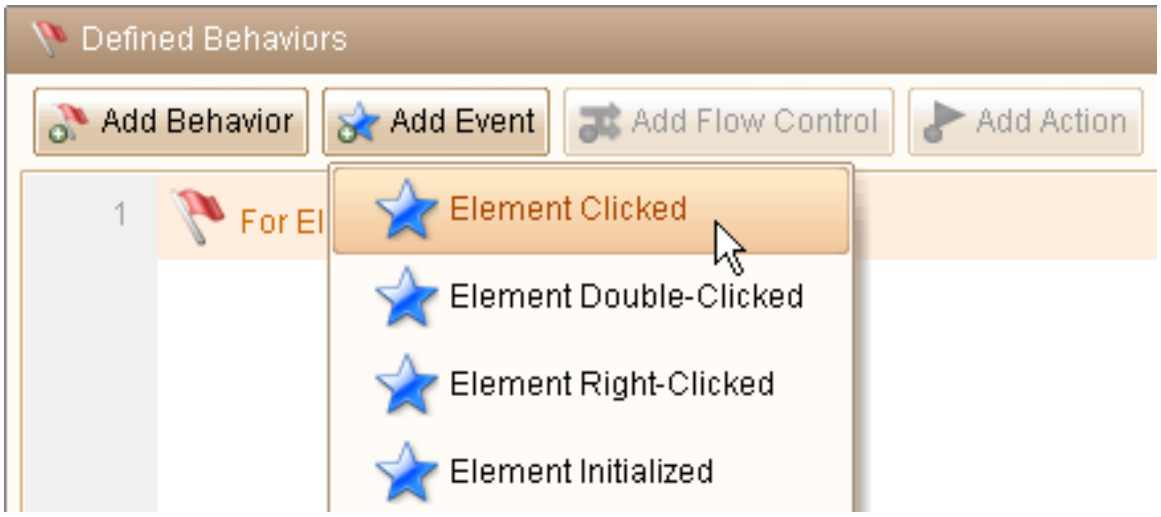


Then double click the button to change its text to "Click Me", then click the "Ok" button to finish editing.

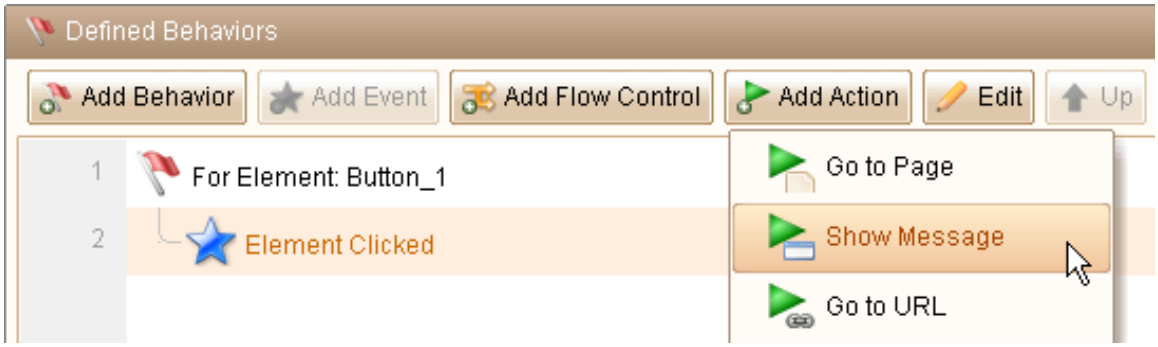
Press Ctrl+D (Command+D in Mac OS) to open behavior editor and define behavior for the button.



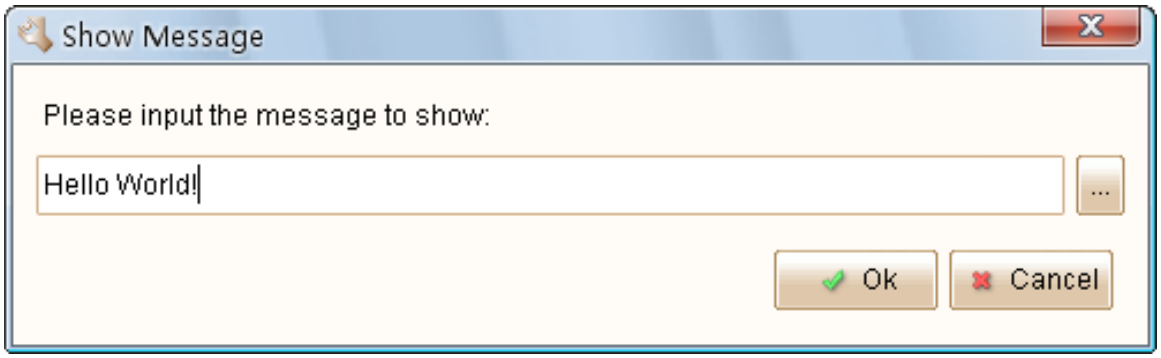
Click the "Add Event" button in the toolbar and choose the "Element Clicked" event.



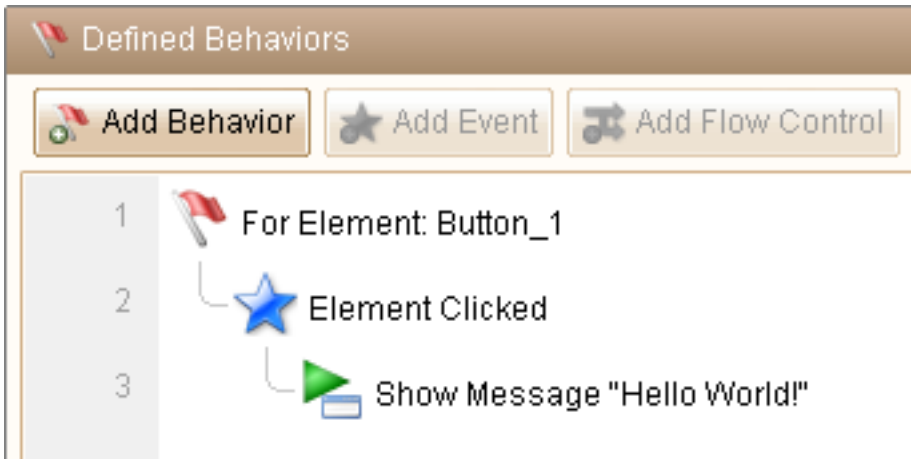
Click the "Add Action" button in the toolbar and choose the "Show Message" action.



Input the message "Hello World!" in the popup window, click "Ok" to finish.



After these steps, the behavior tree should looks like this:



Now let's run this simplest example, just click the 

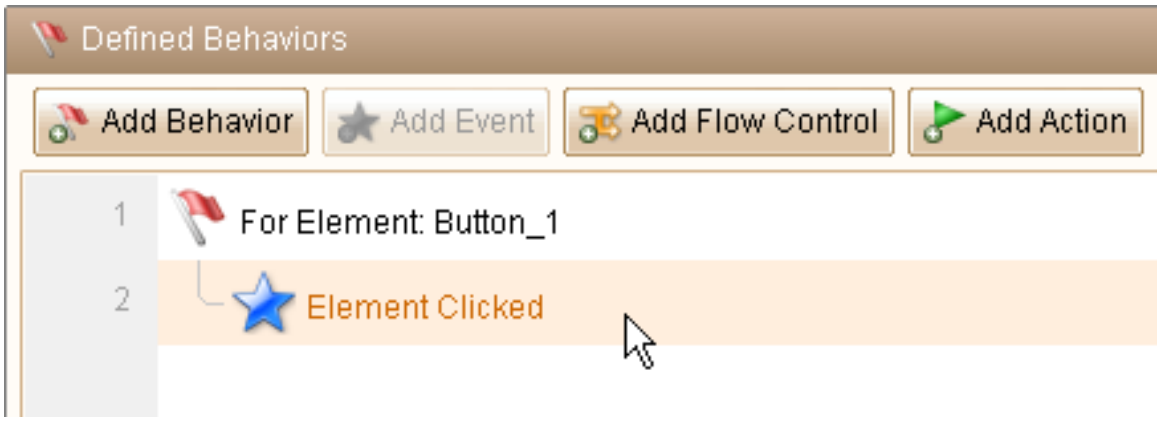
button in the toolbar of main window, or launch the simulation from menu "Prototype->Run Simulation". You will see a browser window popped up with a button inside, clicking the button will bring up a message "Hello World!". You can view this online example to see the result, or download the plot file.

5.2 Example 2: Branching the Work Flow

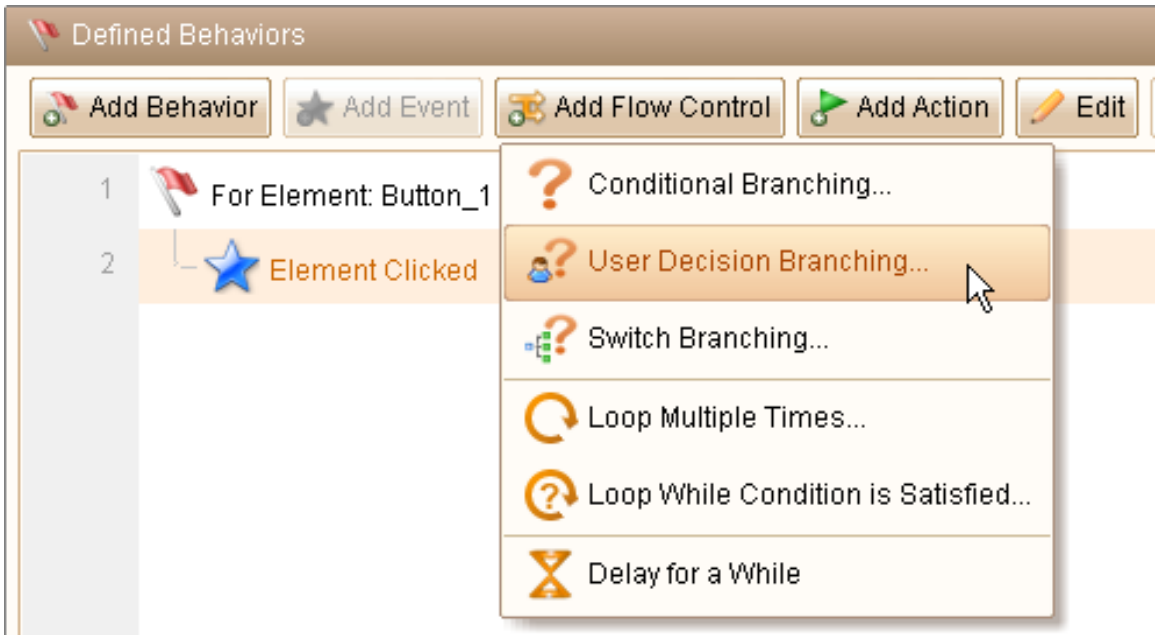
Run This Example in New Window

In previous example, the work flow is very straightforward: when event happens, execute an action. Now let's make it a little complex: branch the work flow and let user to decide which branch to go.

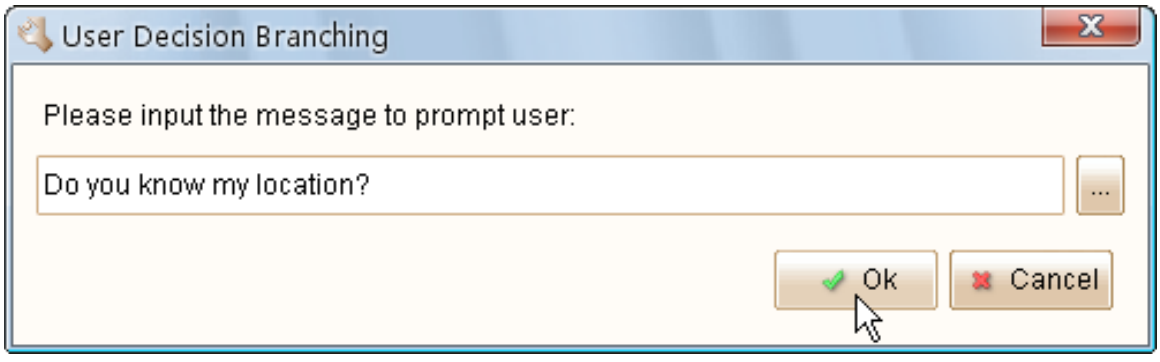
Just like what we did in previous example: add a button element into the plot, open behavior editor, define a behavior for the button, and add the "Element Clicked" event.



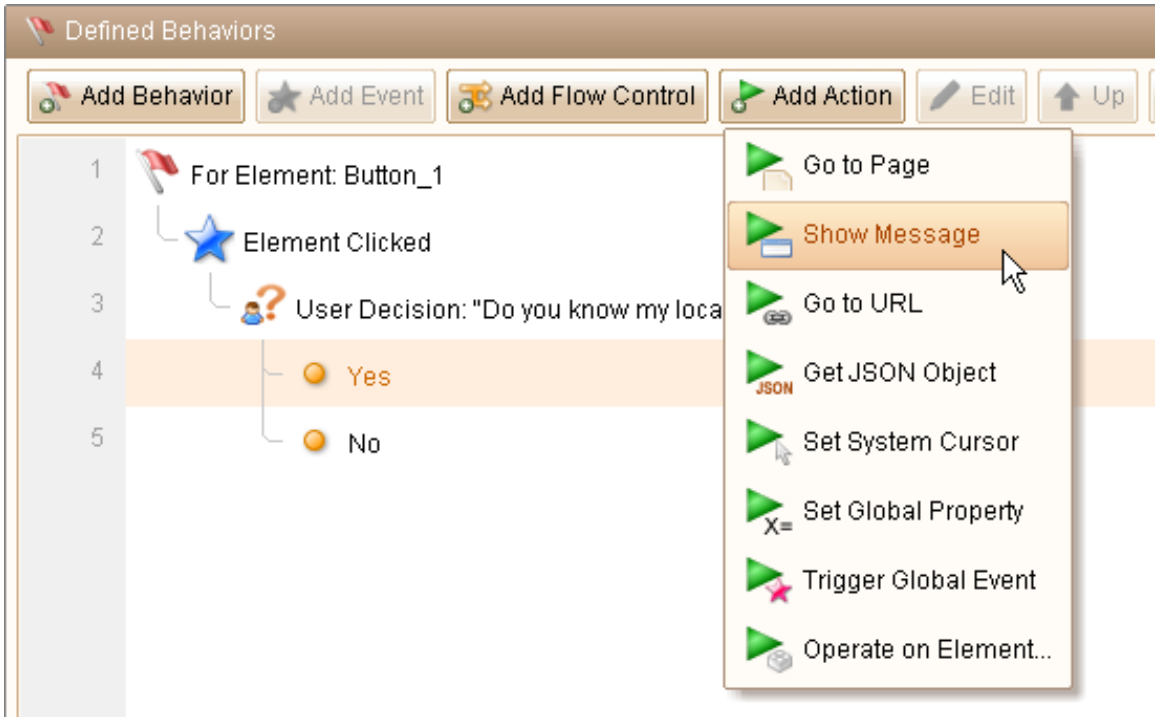
Click the "Add Flow Control" button in the toolbar and then choose the "User Decision Branching..." in the pop-up menu.



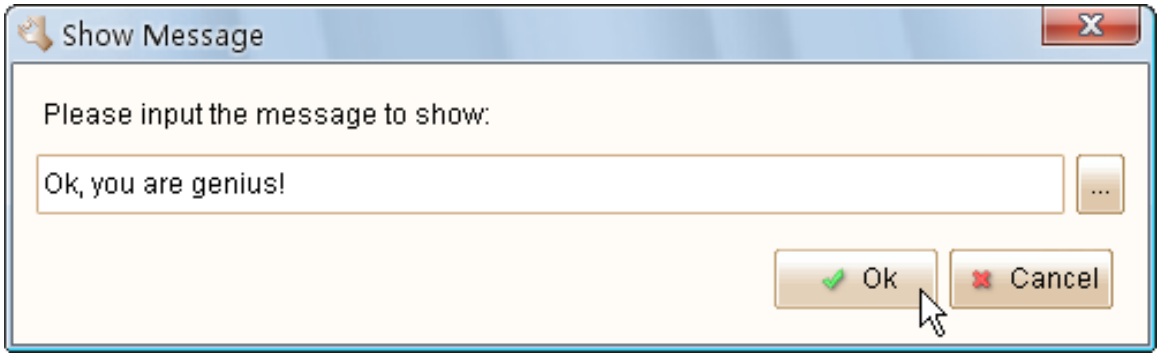
In the popup window, input the question that will prompt user: "Do you know my location?".



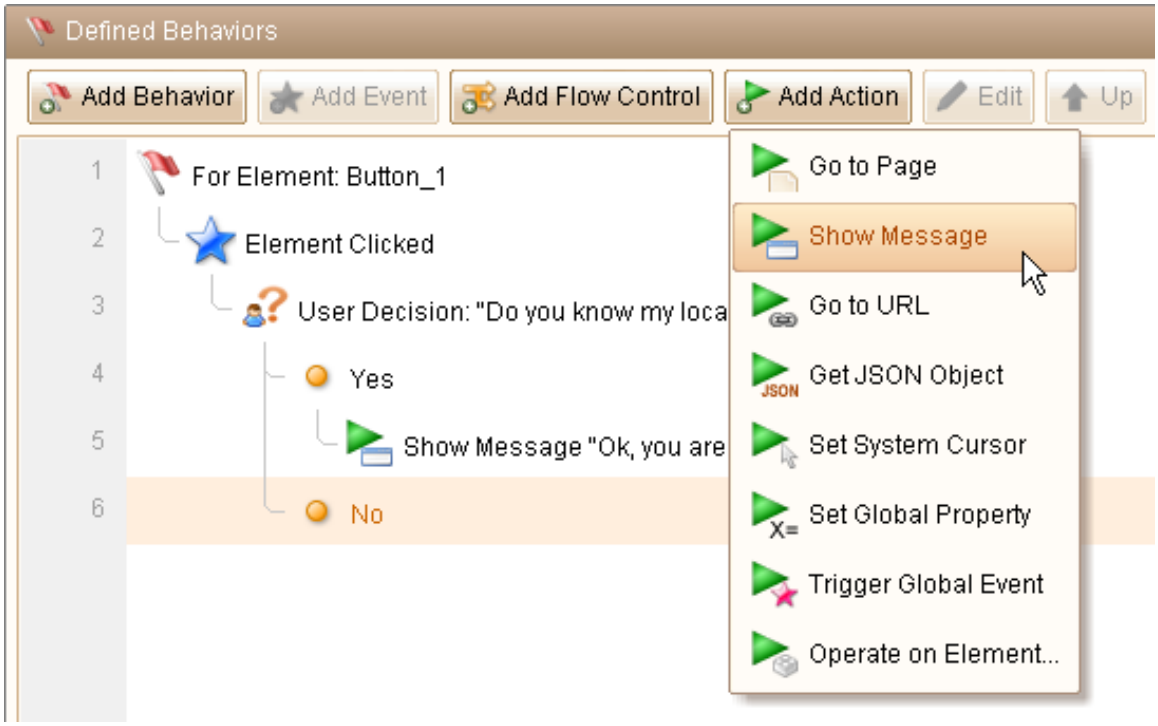
Select the "Yes" node in behavior editor and click the "Add Action" button to add a "Show Message" action.



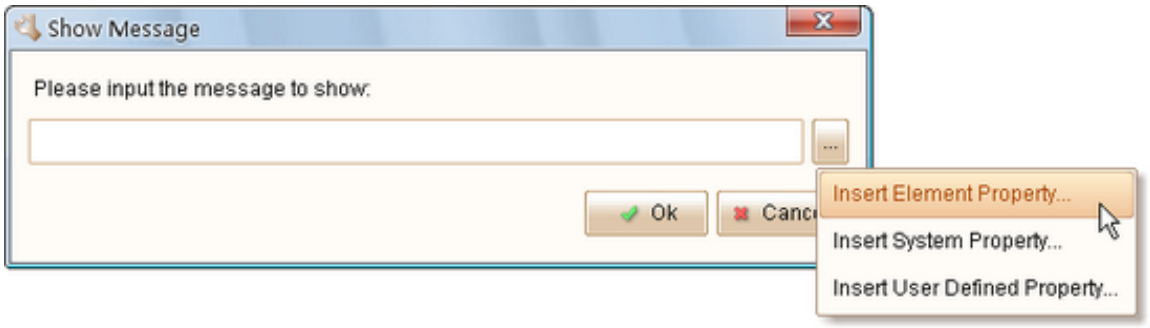
Here we input a praising message: "Ok, you are genius!"



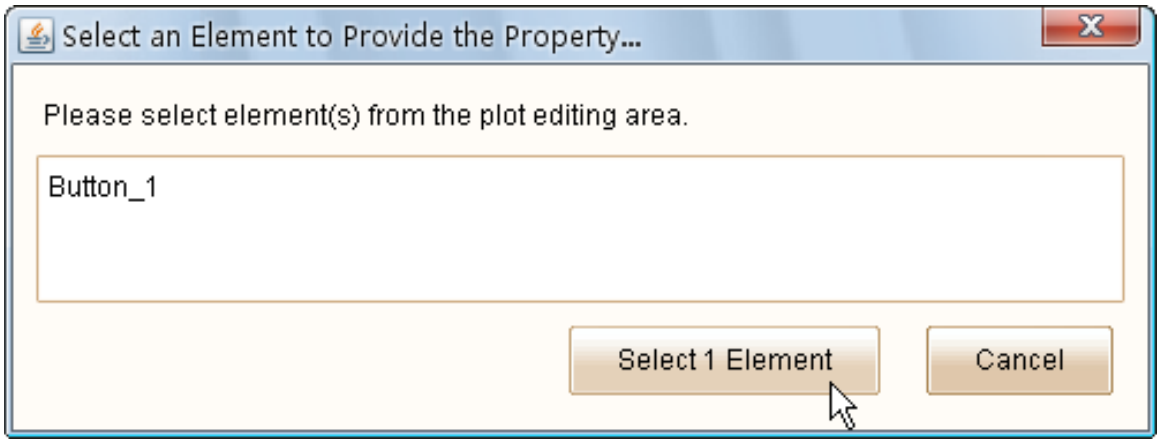
Select the "No" node, and click the "Add Action" button to add a "Show Message" action.



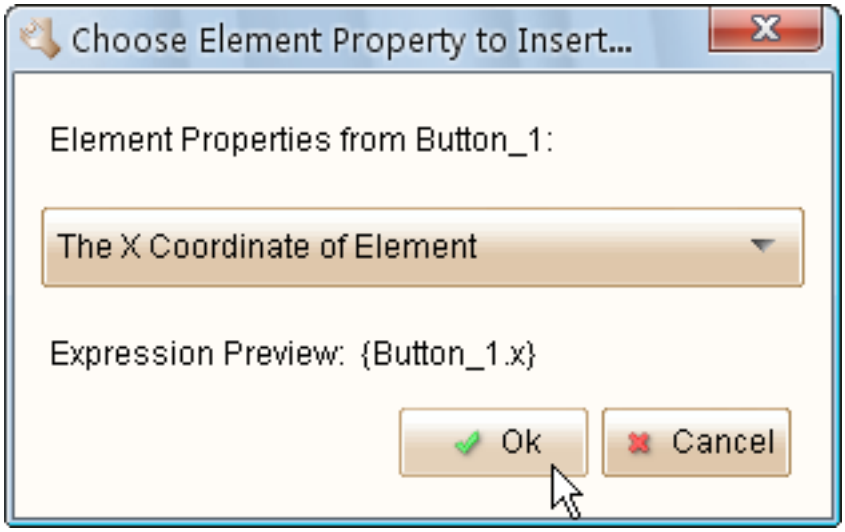
In the "Show Message" edit window, we click the ... button to insert two element properties to represent the current location of the button.



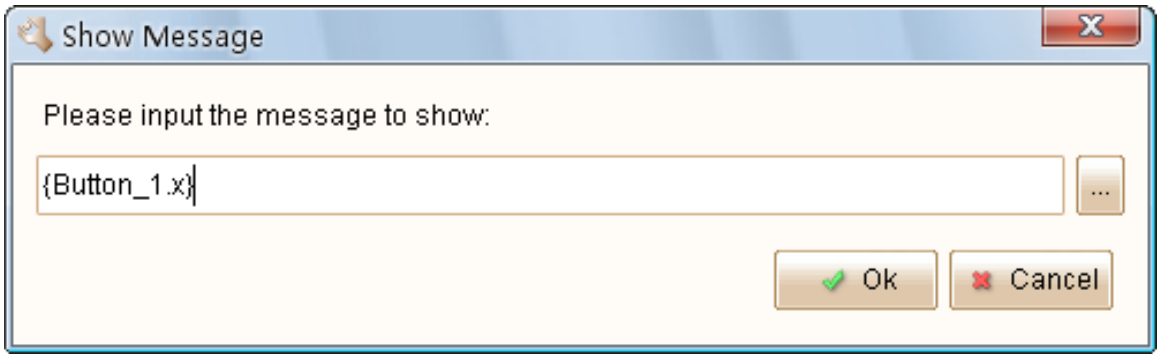
You will be asked to choose an element to provide the property, we choose Button_1.



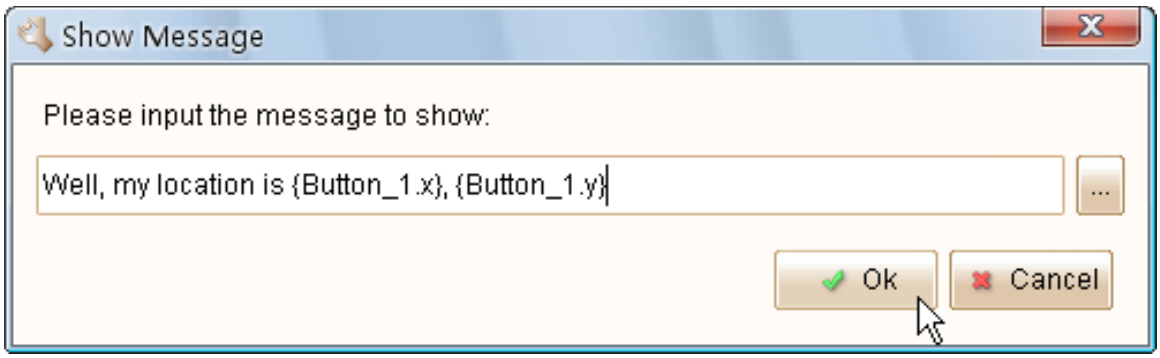
Then in the property chooser window, we can find the property that represent the X coordinate of Button_1.



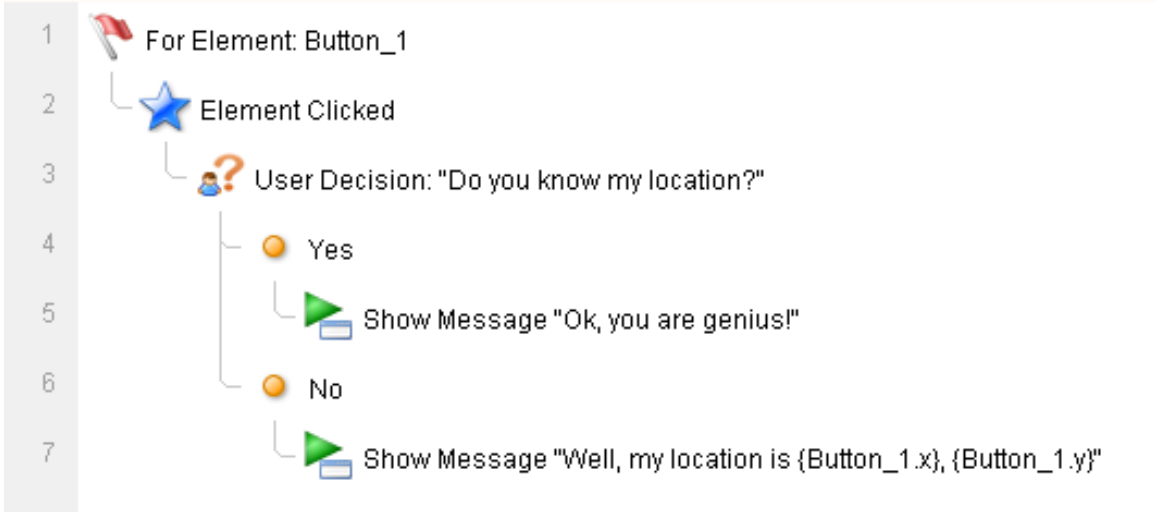
The inserted element property looks like this:



We can do the same thing for inserting another property that represent the Y coordinate of Button_1, and complete the message by adding a few words:



The final behavior for Button_1 should looks like:



Now you can run the simulation by clicking the 

button in the toolbar of main window. This example shows how the branching works and the ability to access the element property. You can view this online example to see the result, or download the plot file.

In this example we use the "User Decision Branching", which can branch the work flow according user's decision. ForeUI also support another two types of branching:

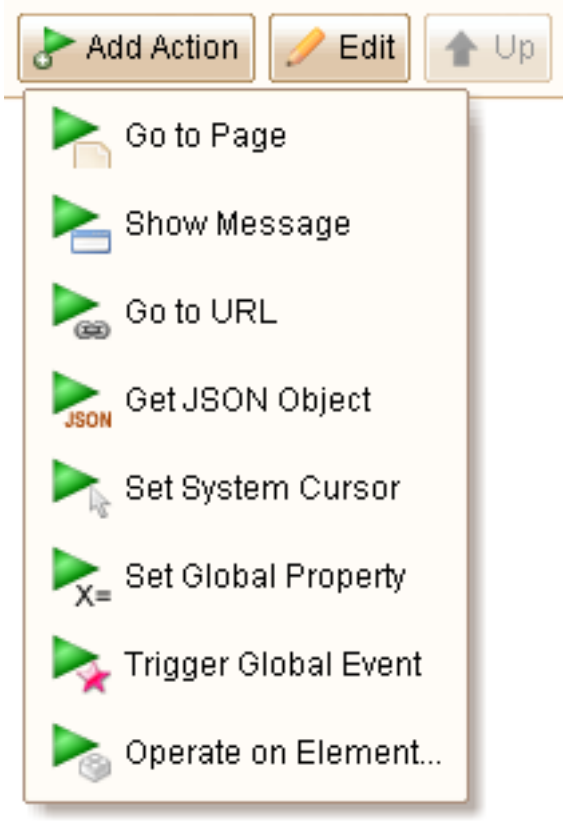
- **Conditional Branching:** do branching according to the conditions.
- **Switch Branching:** do branching according to the value of expression.

Please read the Branching section for more details.

5.3 Example 3: Manipulate Element in Runtime

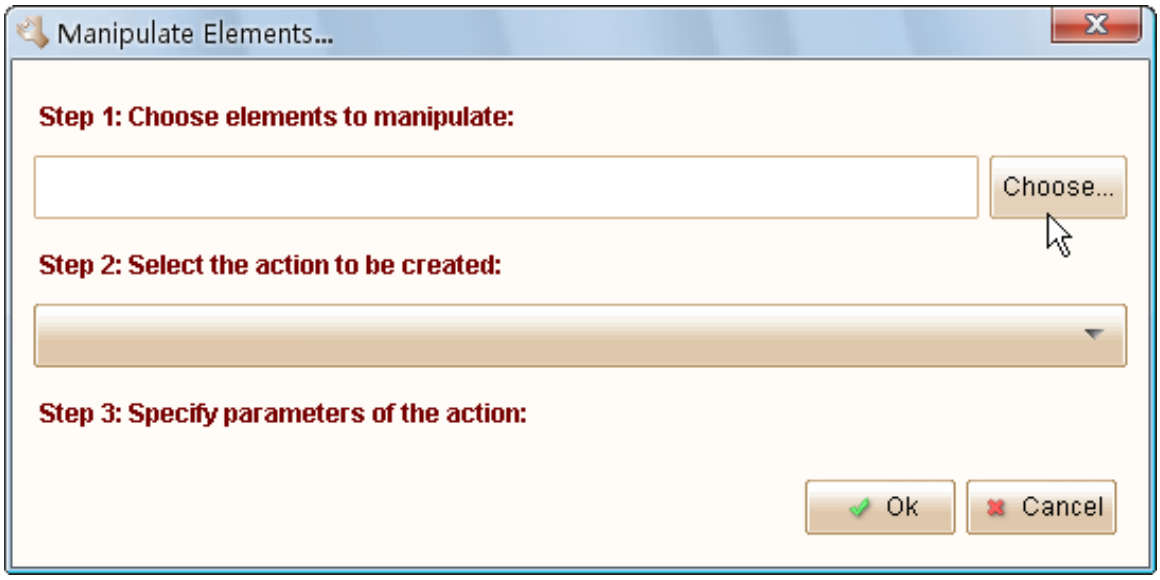
Run This Example in New Window

We already know that clicking on the "Add Action" button in the toolbar of behavior editor will bring up a list of available actions.

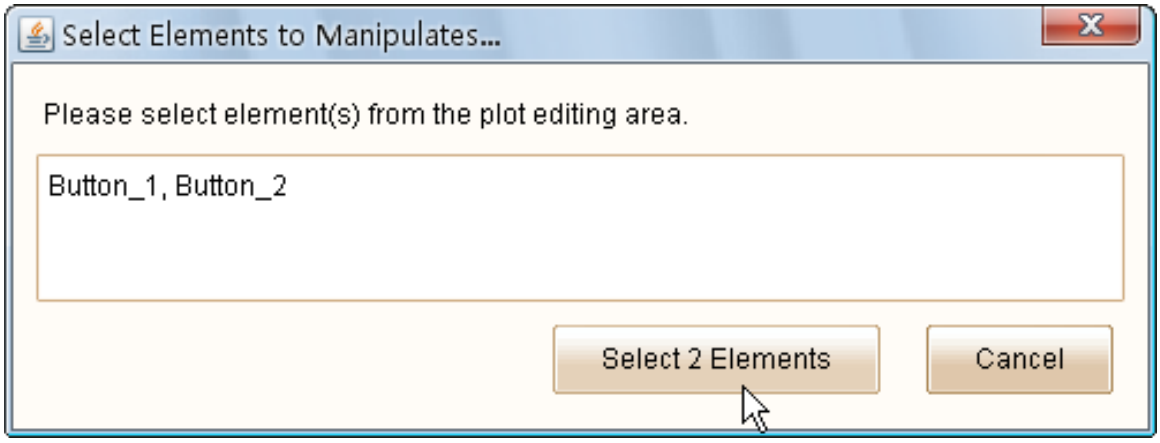


Actually there are more actions, which are not listed here. ForeUI provides a set of actions that belongs to the "Operate on Element..." category. These actions can be selected when you choose the element to operate on.

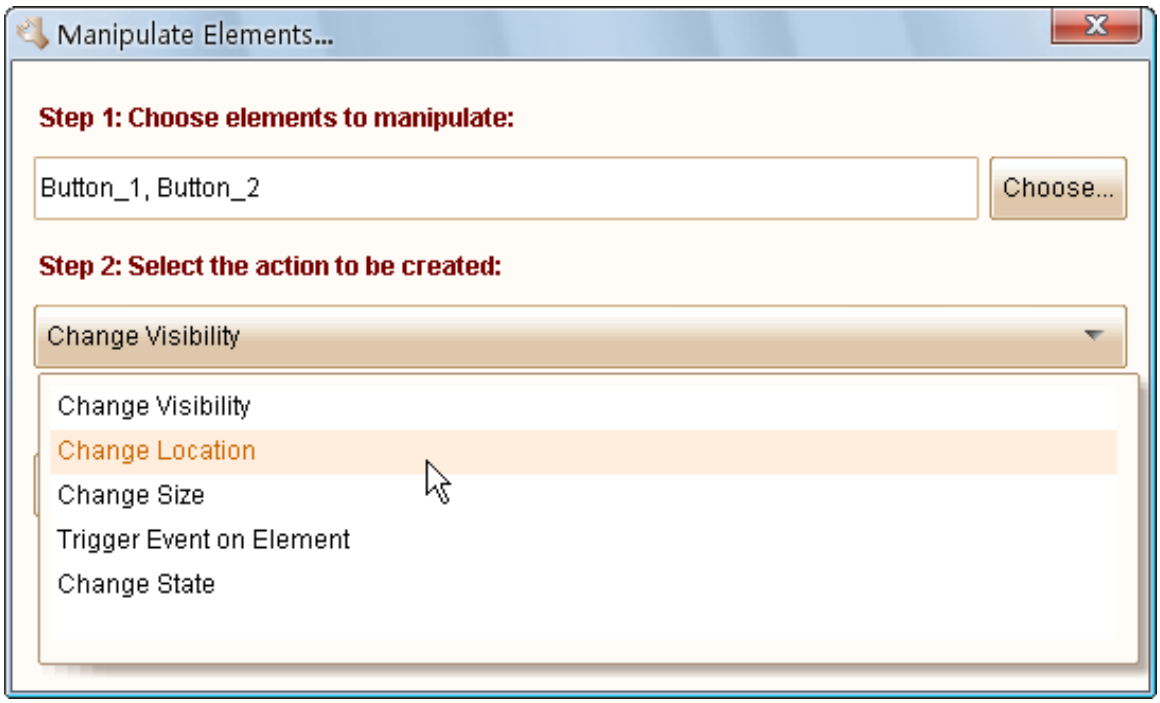
After clicking on the "Operate on Element..." menu item, you will see this window:



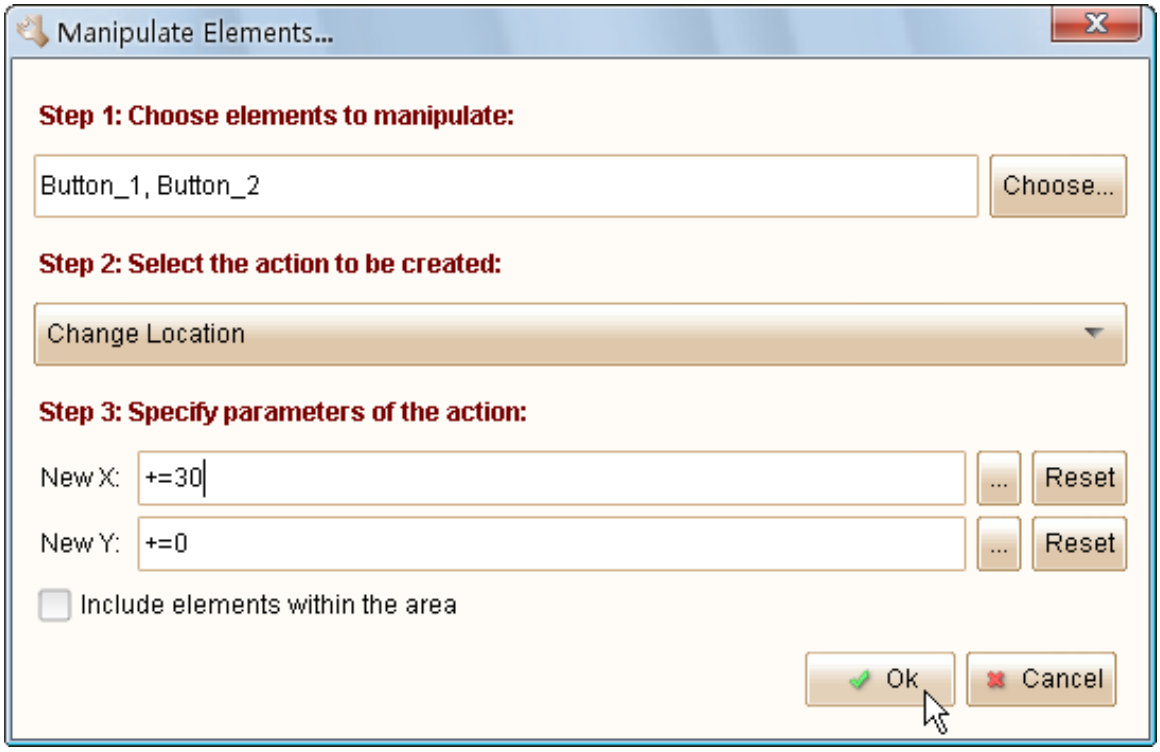
Here you should click the "Choose" button to pick one or more elements to manipulate. You will see the element picker like this:



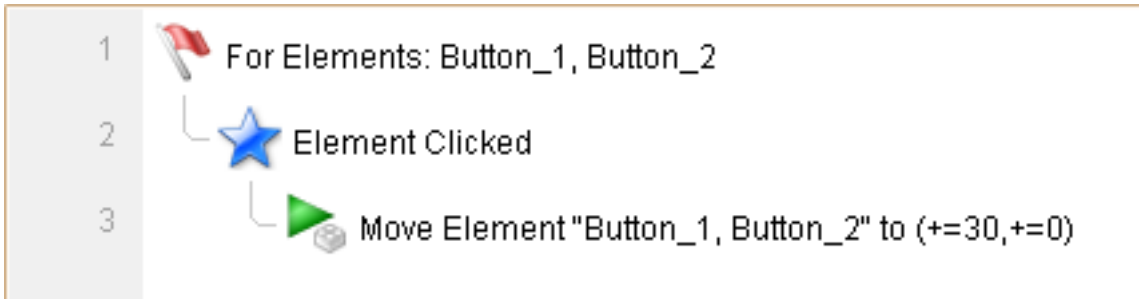
As usual you can choose element in the plot editing area (hold SHIFT to select more elements). Let's select two buttons (Button_1 and Button_2) and then click the "Select 2 Elements" button, you will see all available actions for tweaking the elements in a drop-down list.



After selecting the "Change Location" action, you can input the action parameters. Here we use "+=30" as the new value of X coordinate, means the button will move right 30 pixels a time.



Here is the final behavior definition:



Now let's run it, it works as we expected! Each time we click any of the two buttons, they move right a bit (30 pixels). You can view this online example to see the result, or download the plot file.

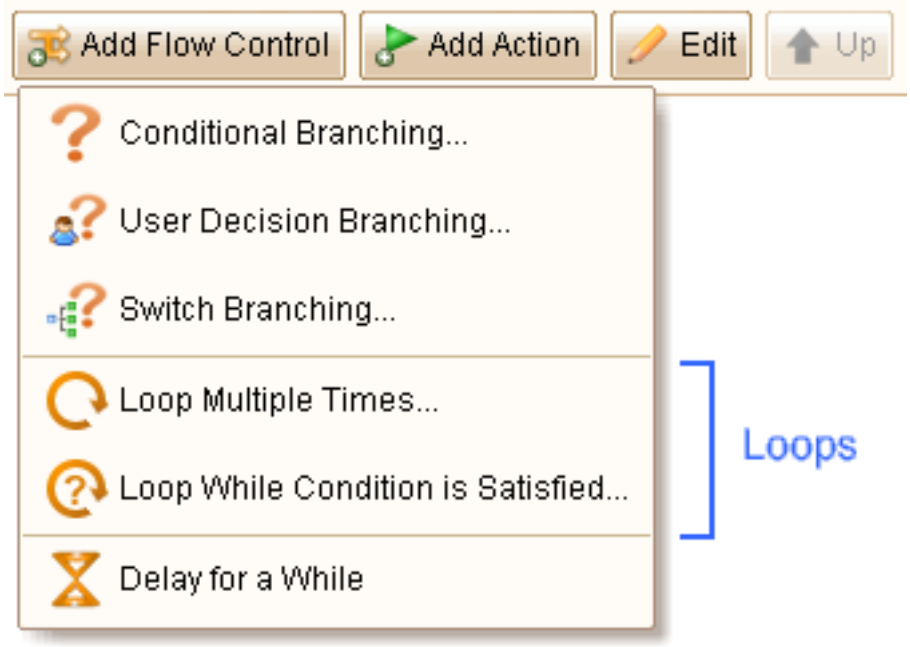
5.4 Example 4: Using Loop

Run This Example in New Window

Sometimes we need to execute some tasks for multiple times, we will need to use loop in this case.

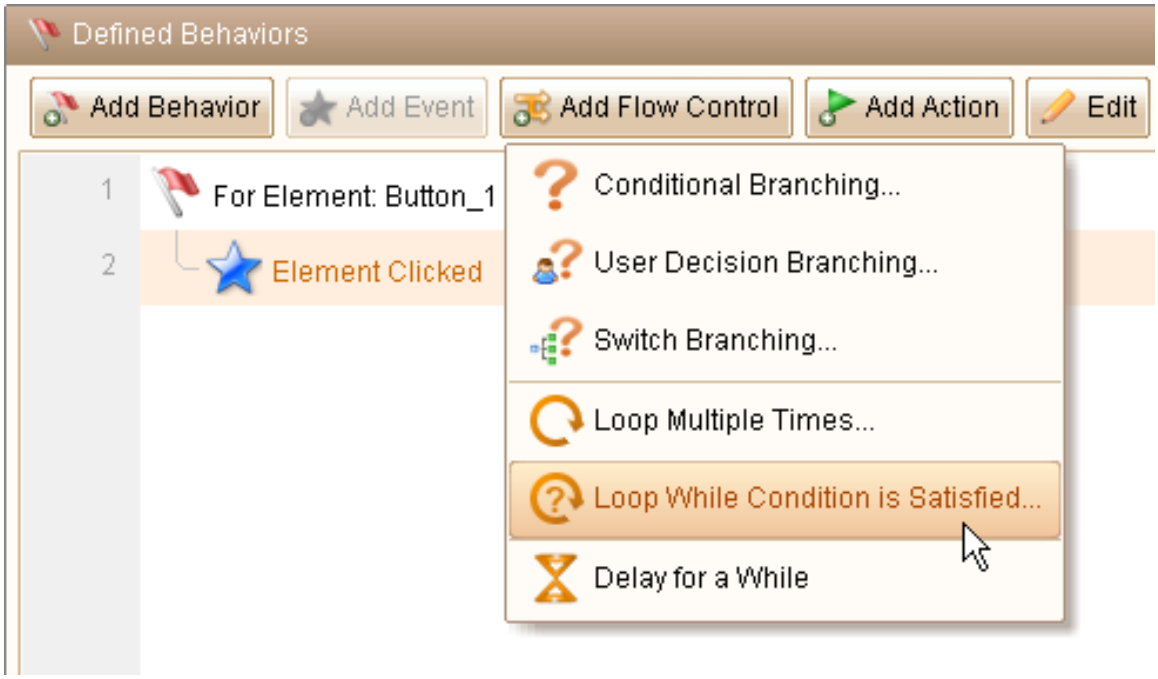
ForeUI supports two kinds of loop:

- **Loop Multiple Times:** will loop for certain times.
- **Loop While Condition is Satisfied:** will keep looping as long as the condition is satisfied.

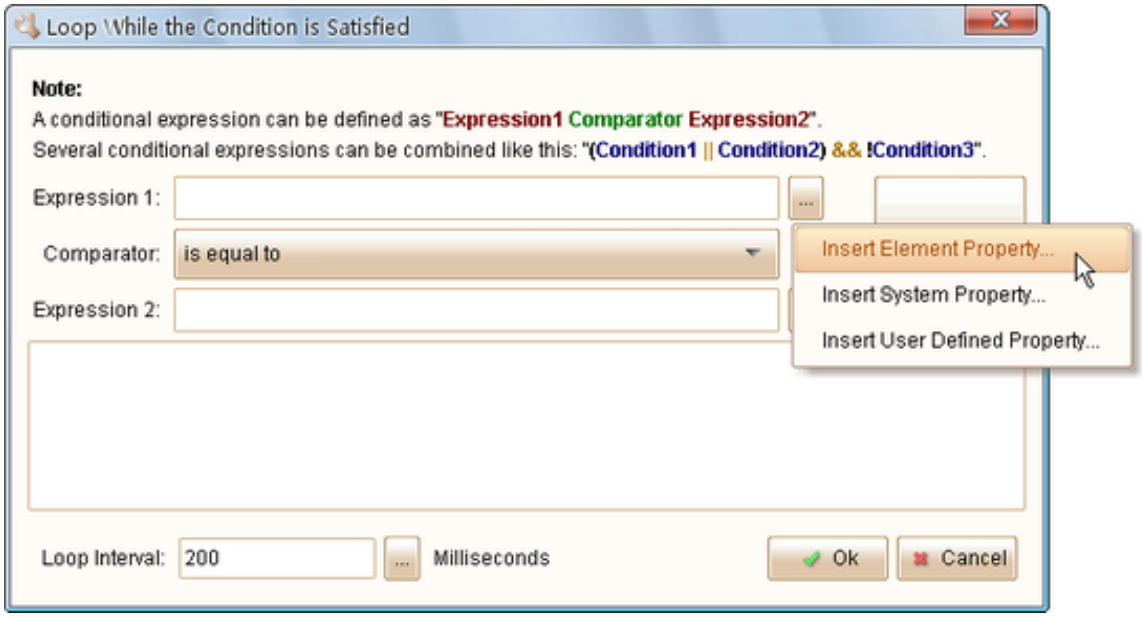


Now let's make such an example that make use of loop: when you click the button, the button will keep moving to the left, until its x coordinate is less or equals than 100.

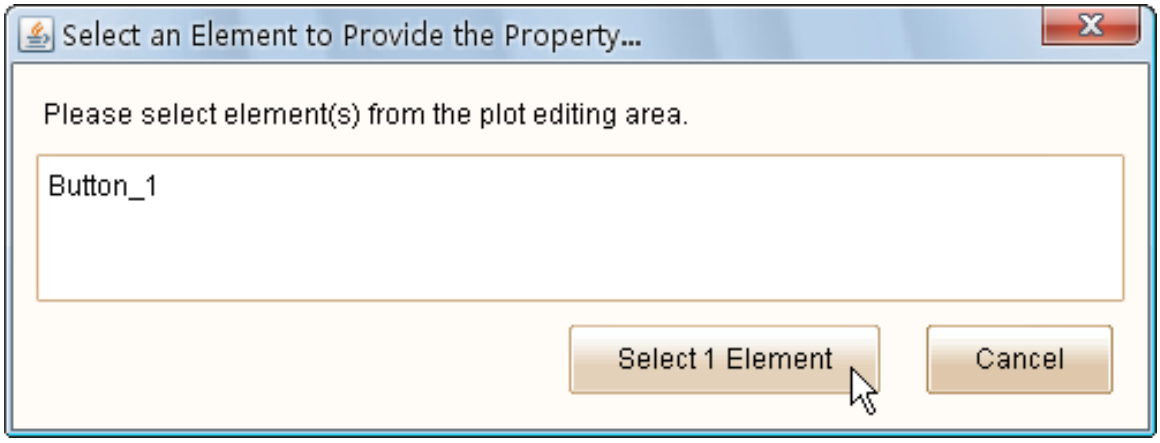
The first step is to place a button on somewhere, for example, at (400, 100). After defining the behavior and create the "Element Clicked" event for it, we can add the loop by selecting the "Loop While Condition is Satisfied..." item under the "Add Flow Control" button in toolbar.



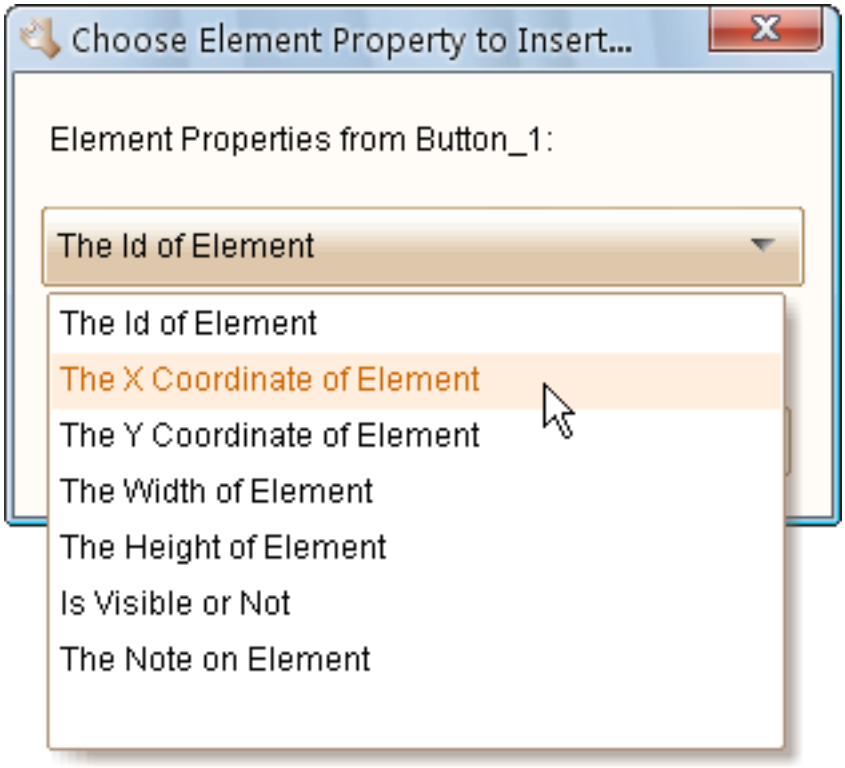
In the popup window we can specify the condition. Since we expect the button to move to left until x <= 100, we need to specify a condition like "Button's x coordinate is greater than 100" for looping. So we need to insert the property that represent button's x coordinate, just click the ... button and choose "Insert Element Property...":



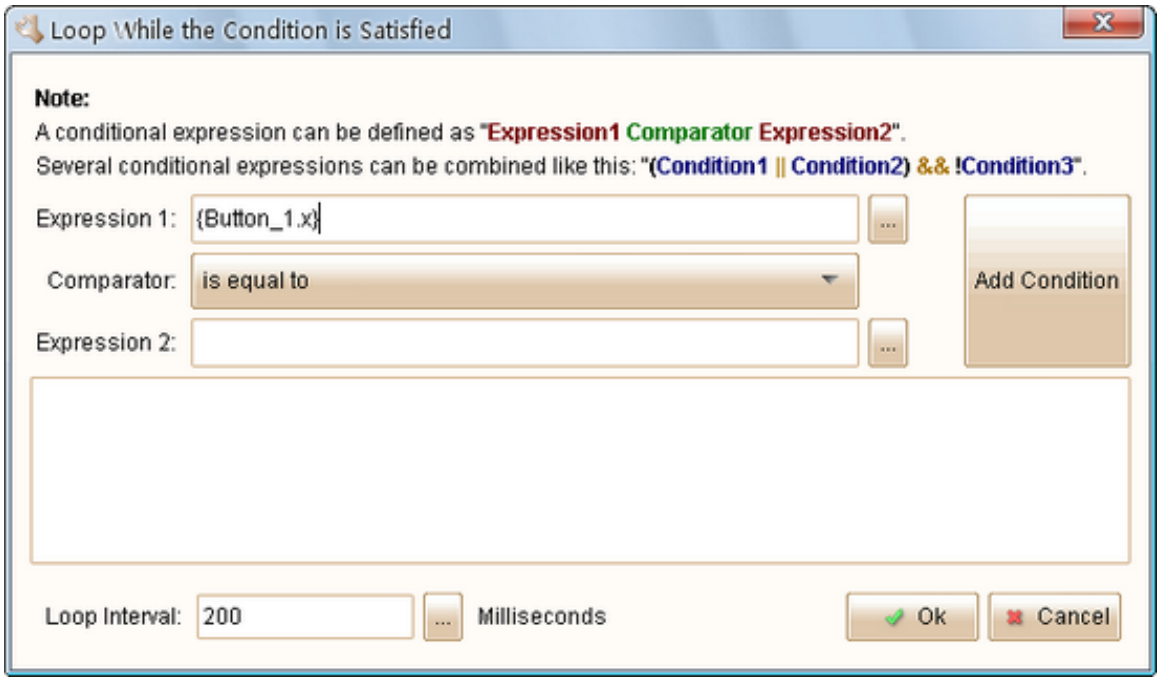
The element chooser will show up and we choose our only button (Button_1):



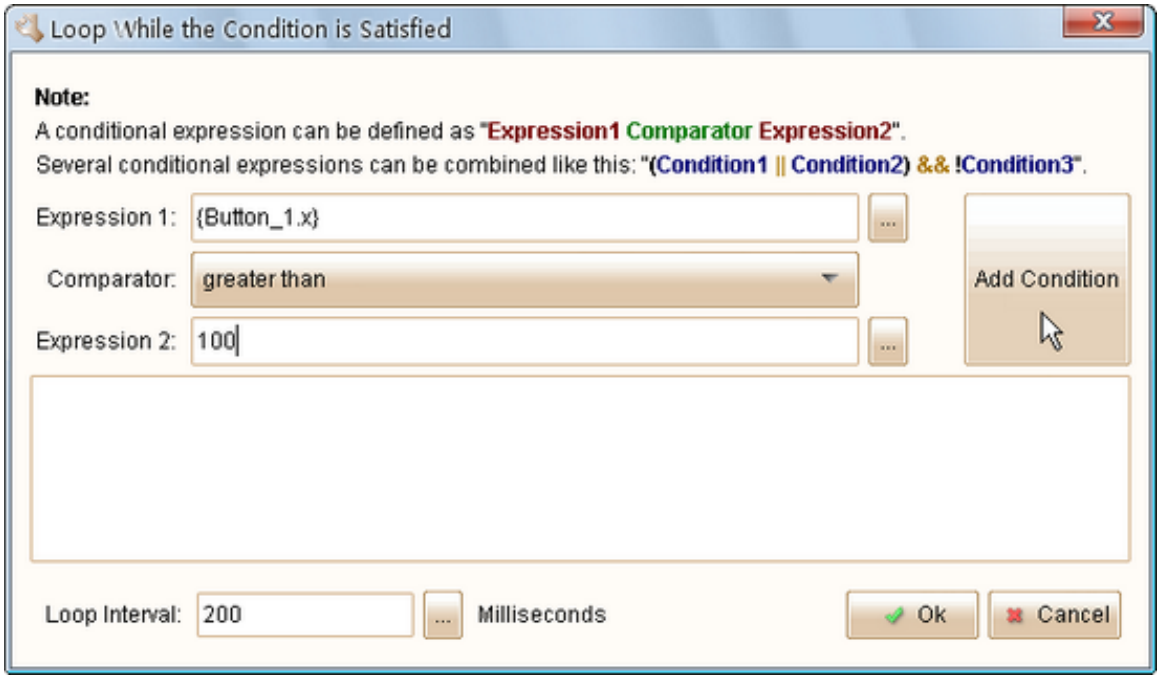
Then in the property chooser window, we choose "The X coordinate of Element" and click "Ok" button:



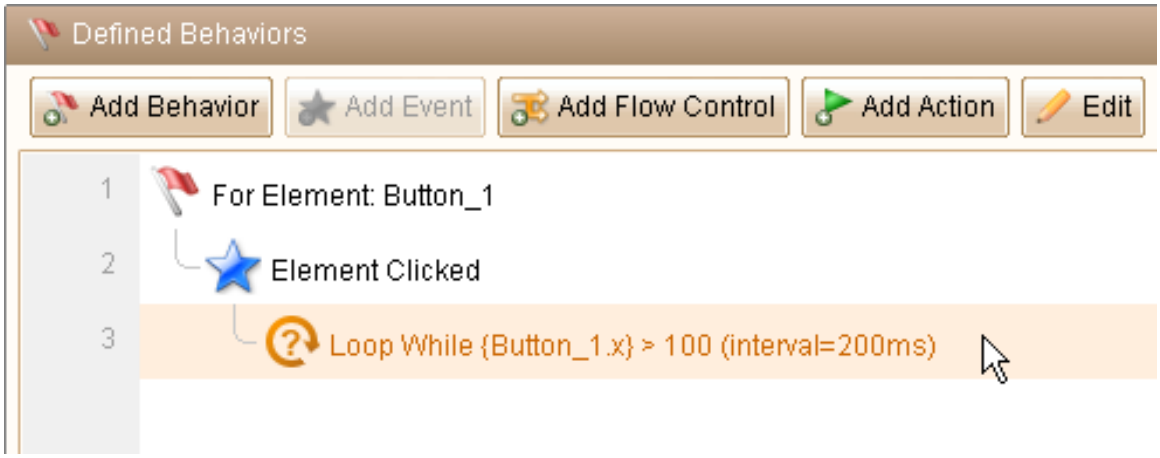
Then you will see how the property looks like in the expression:



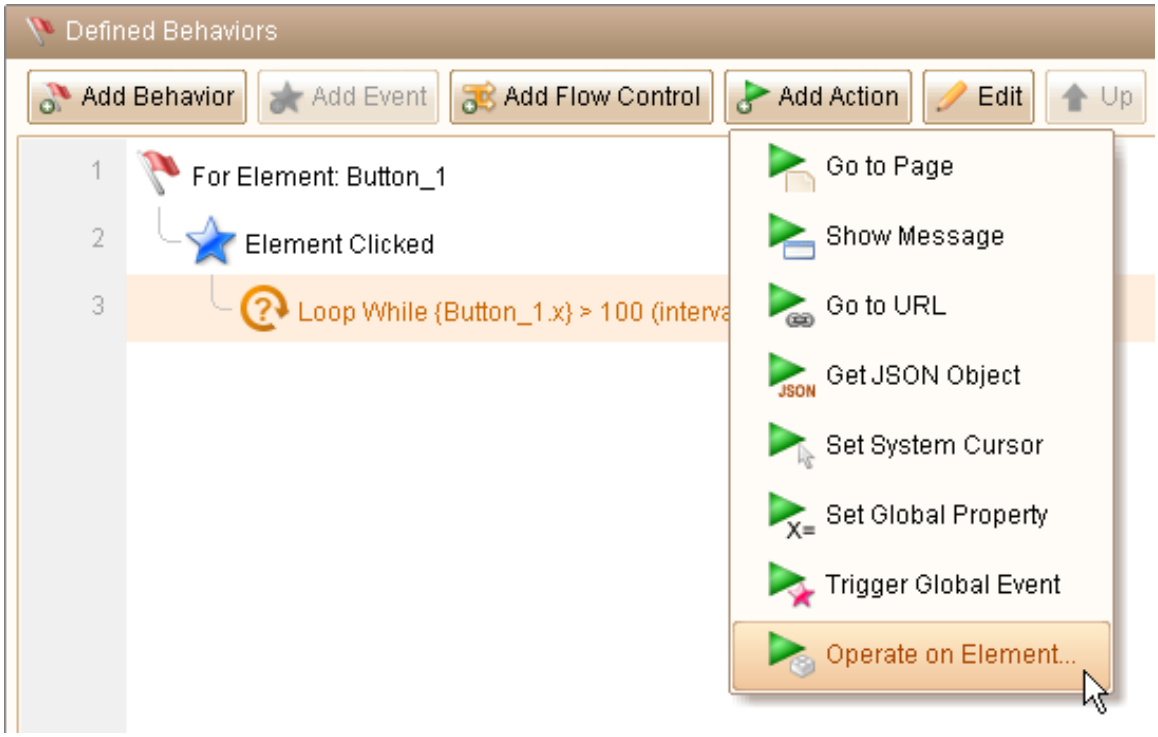
To finish the condition, please choose "greater than" comparator and input value "100" as the expression 2, then click "Add Condition" button:



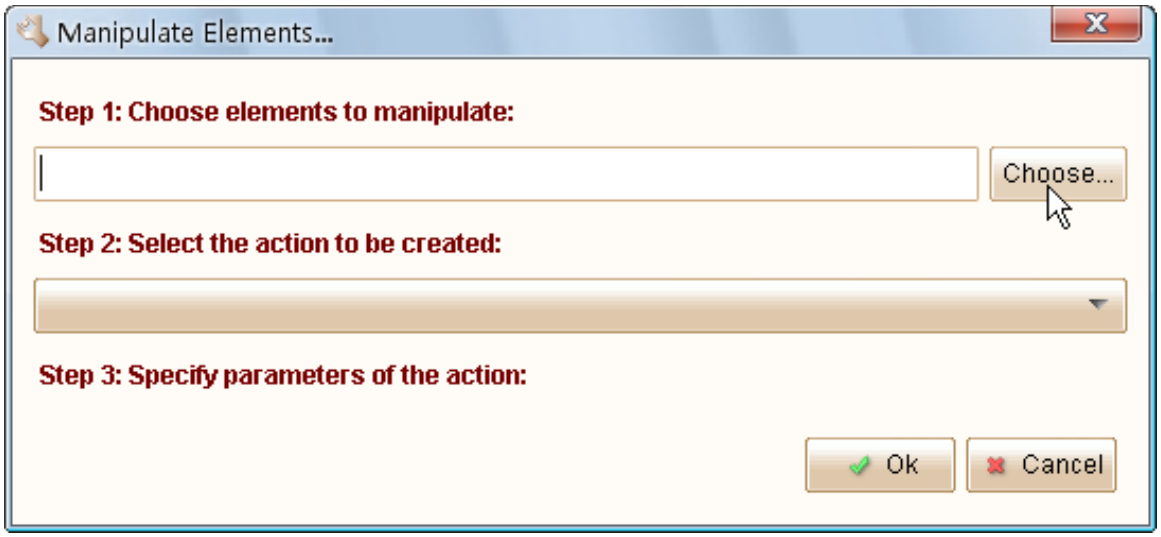
You can change the "Loop Interval" to smaller value if you want the button move faster. After clicking the "Ok" button, the conditional loop is created.



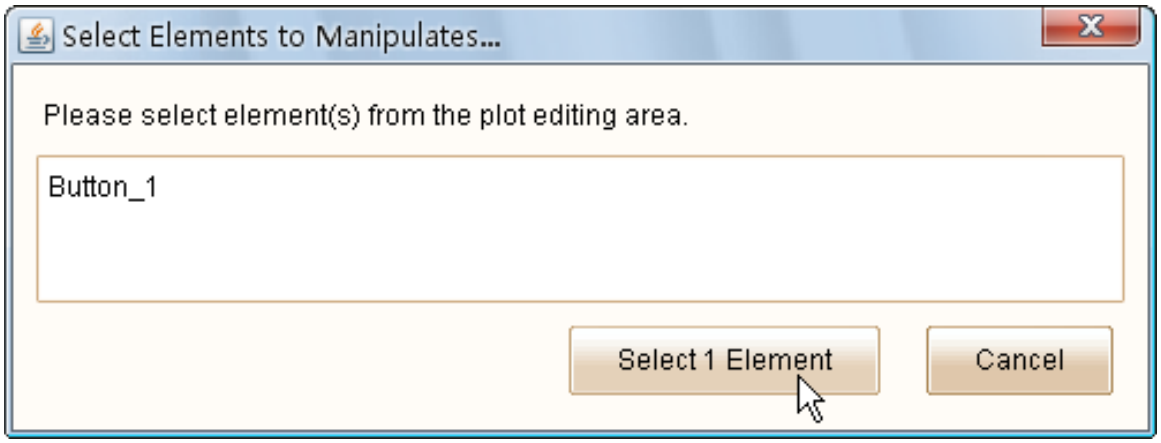
Now we can add the action to move button. Click the "Add Action" button and choose "Operate on Element...":



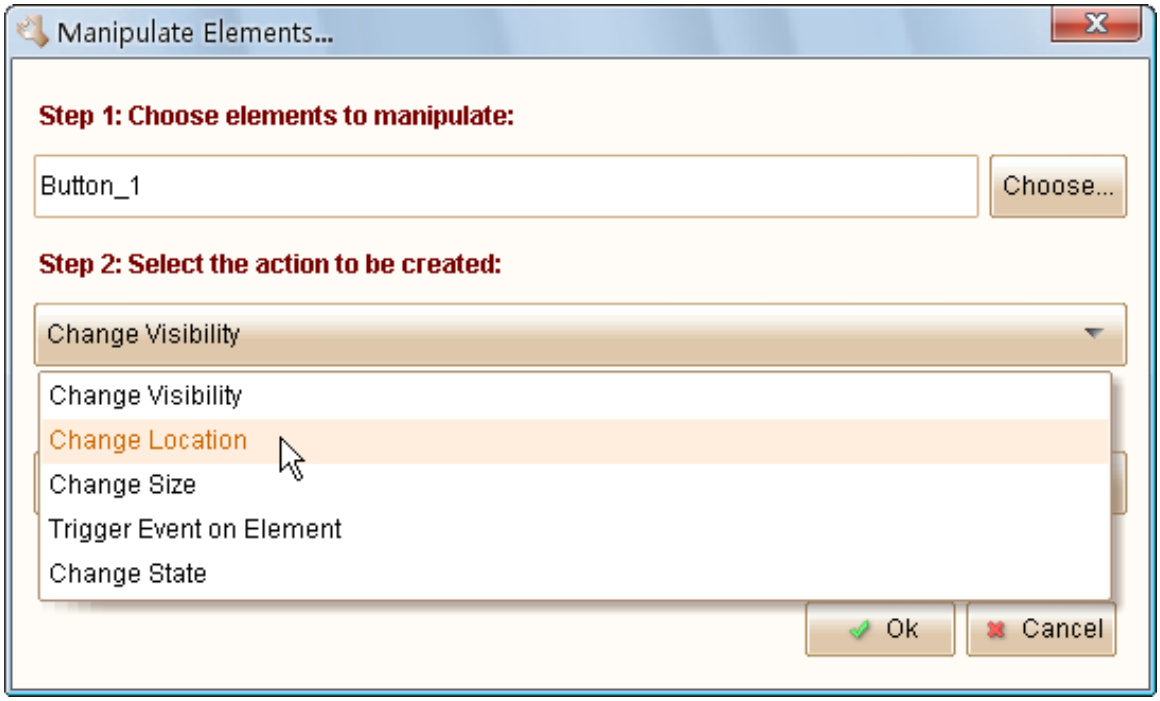
We need to choose the target element first:



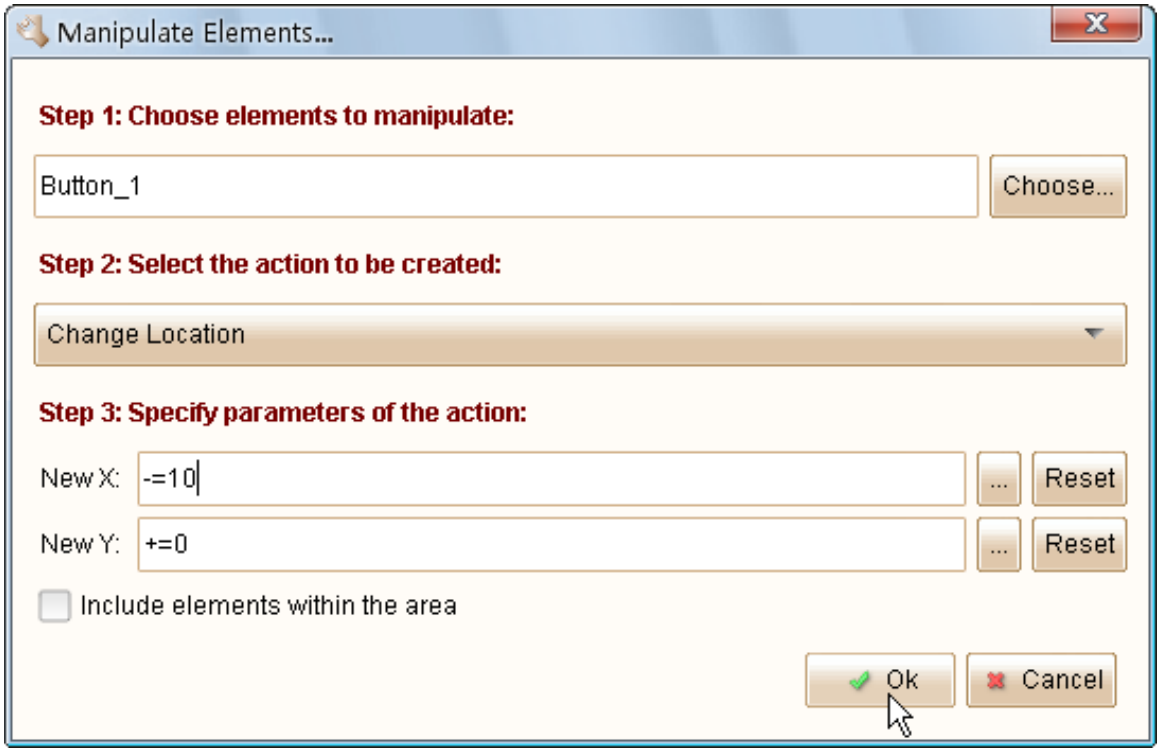
Here just choose the only button (Button_1) as the target element:



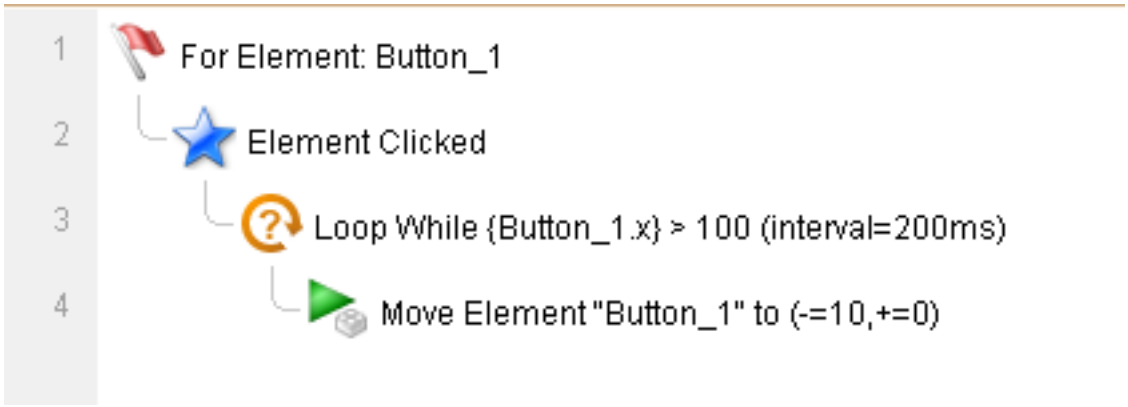
Then we can choose "Change Location" action for moving the button:



Input the new x value as "-=10", which means moving button to left for 10 pixels a time:



The final behavior looks like this:

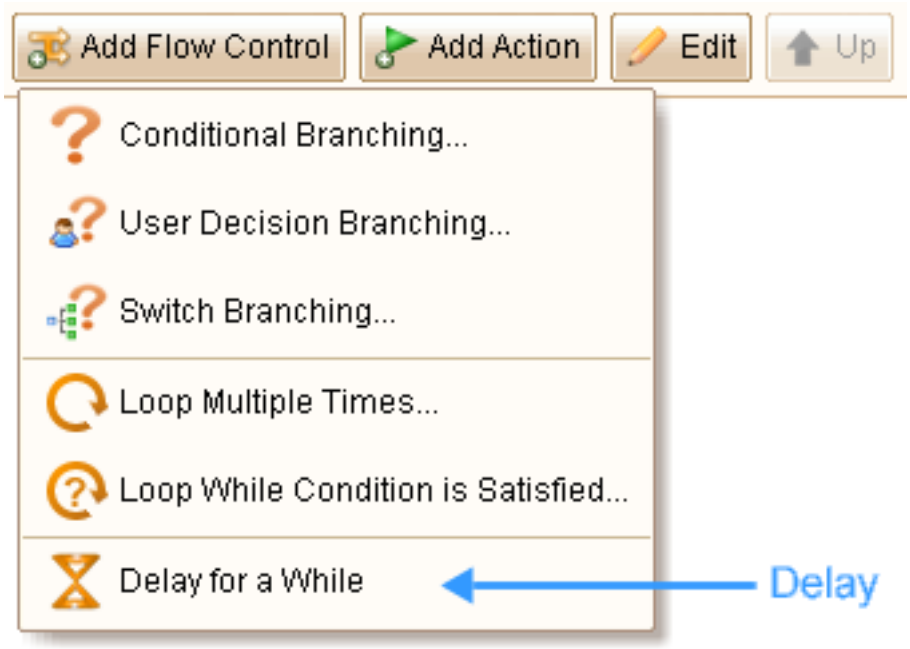


We can run it now. When we click the button, it keeps moving left a little until its x position equal or less than 100. You can view this online example to see the result, or download the plot file.

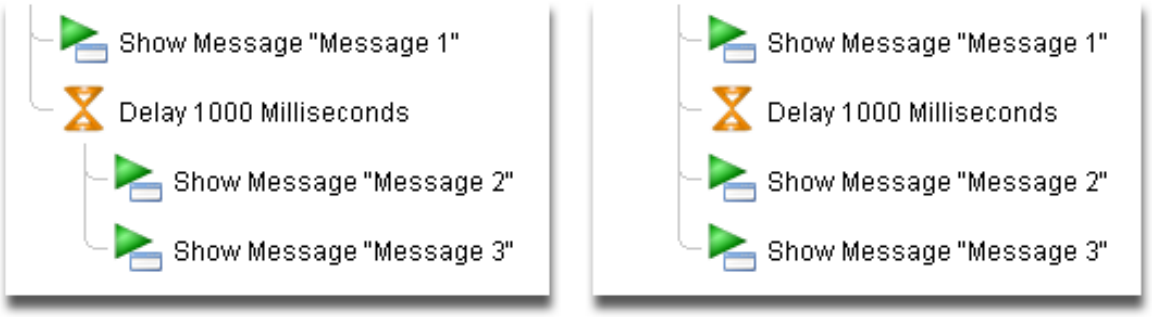
5.5 Example 5: Using Delay

Run This Example in New Window

Sometimes you may want to pause a while between two actions. Delay is a flow control that designed for this use case.

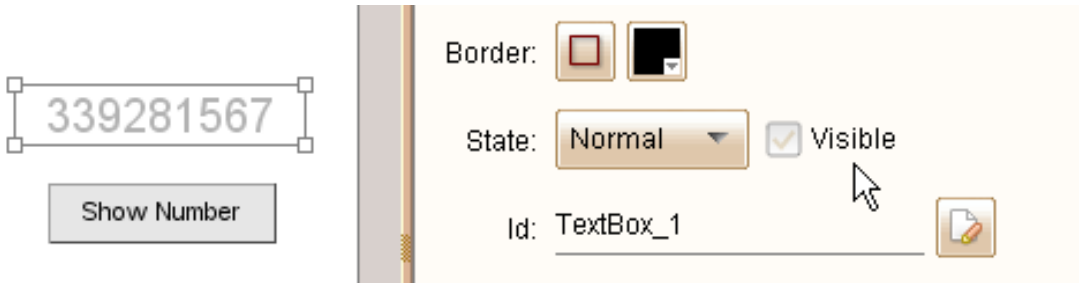


Delay can be placed under event or other flow controls. You can put some actions under the delay, or after the delay, these two ways are equivalent, you can choose any one you like:

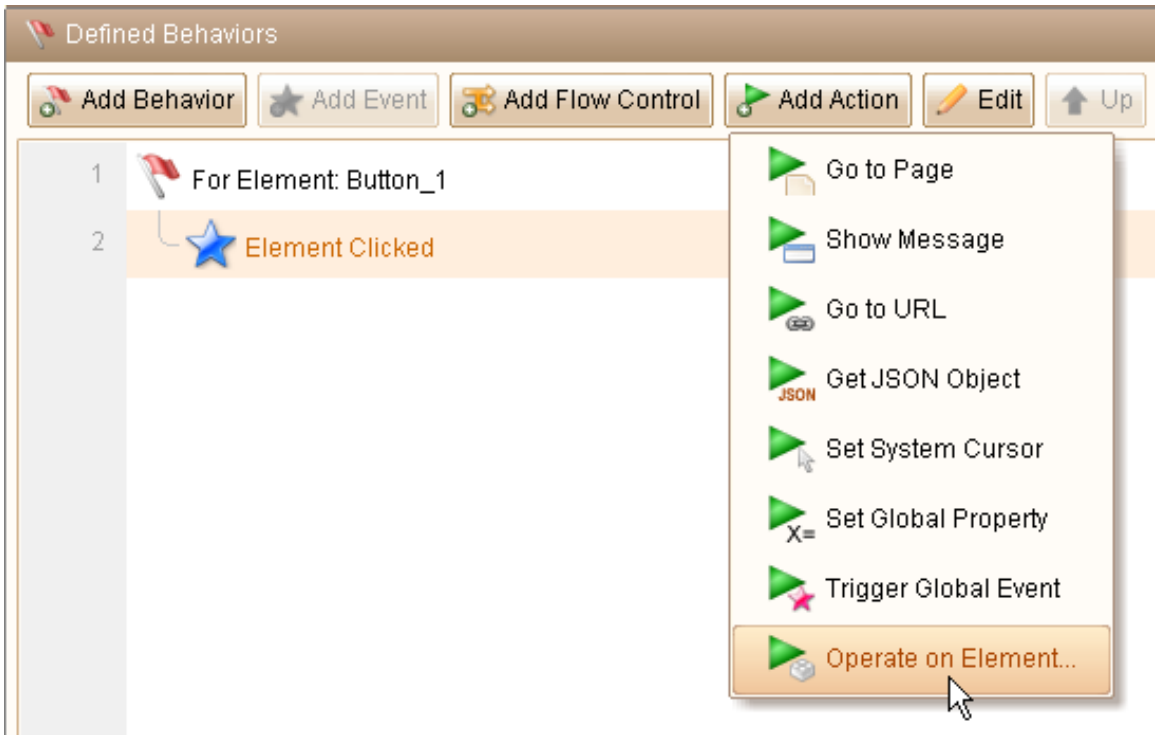


Let create an interesting example: show a number when clicking on a button, hide it after 2 seconds.

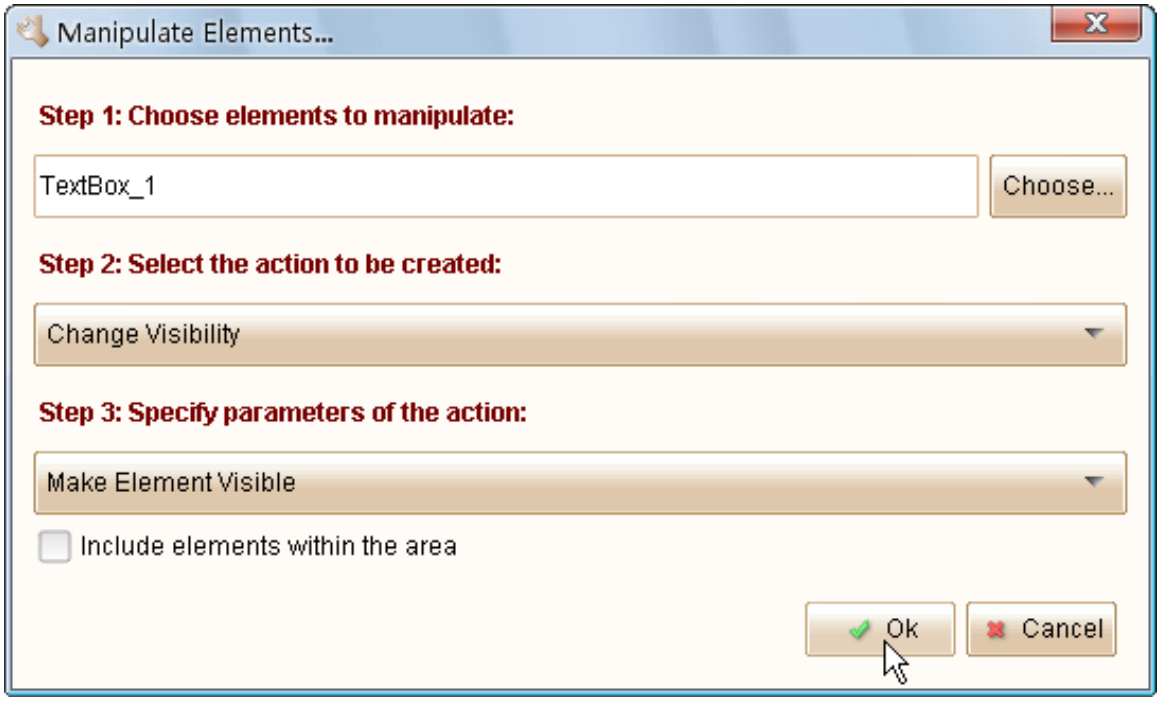
First we need to place a TextBox and a Button element in the page. Put a long number in the TextBox and change the text of Button to "Show Number". Don't forget to unselect the "visible" checkbox for the TextBox element (so it will be hidden by default):



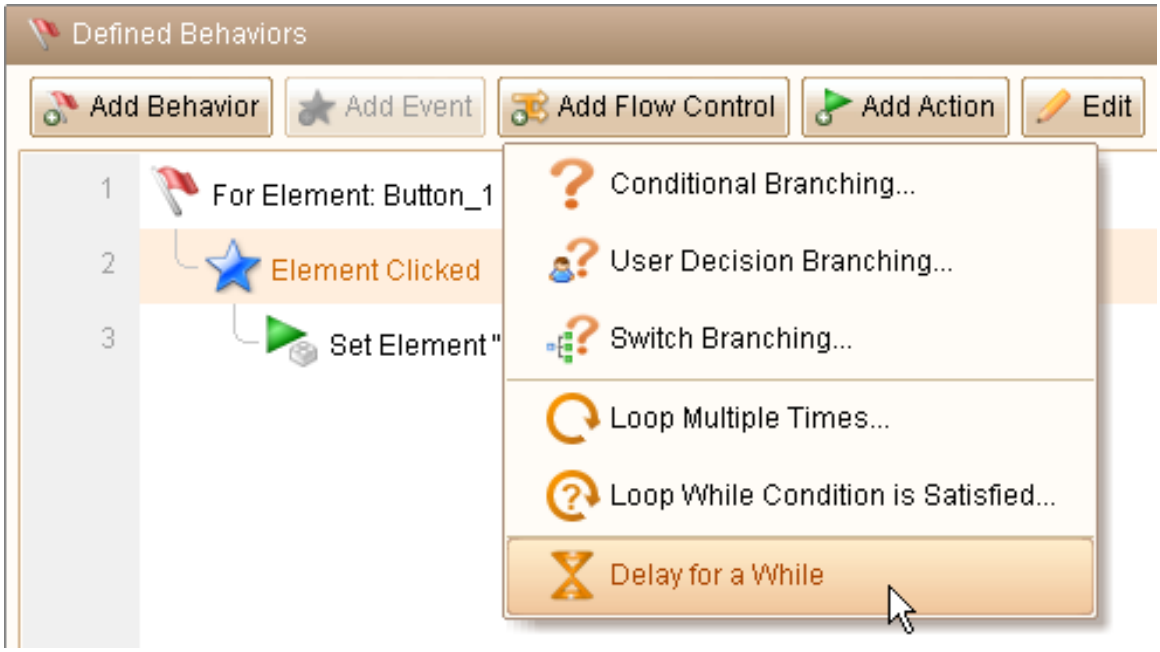
Create the behavior for the button (Button_1) and add "Element Clicked" event. Choose "Operate on Element..." after clicking "Add Action" button in toolbar.



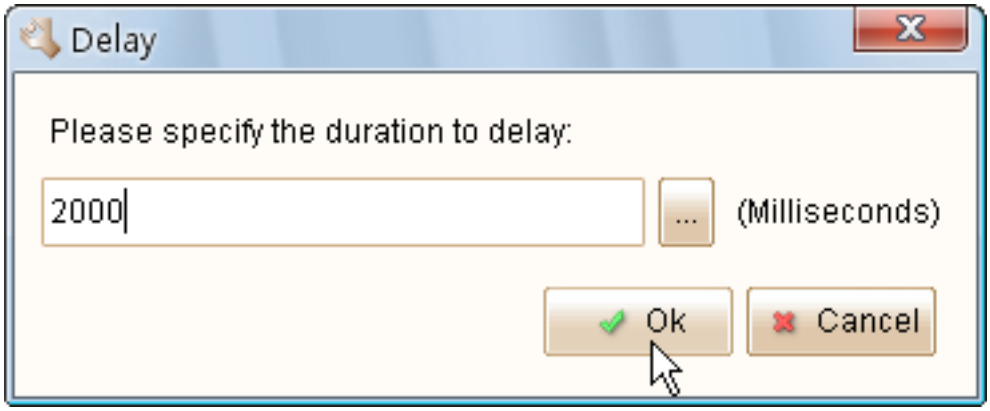
Choose the TextBox element (TextBox_1) as the target element, select "Change Visibility" action and then select "Make Element Visible" as parameter. This action will make the TextBox element visible.



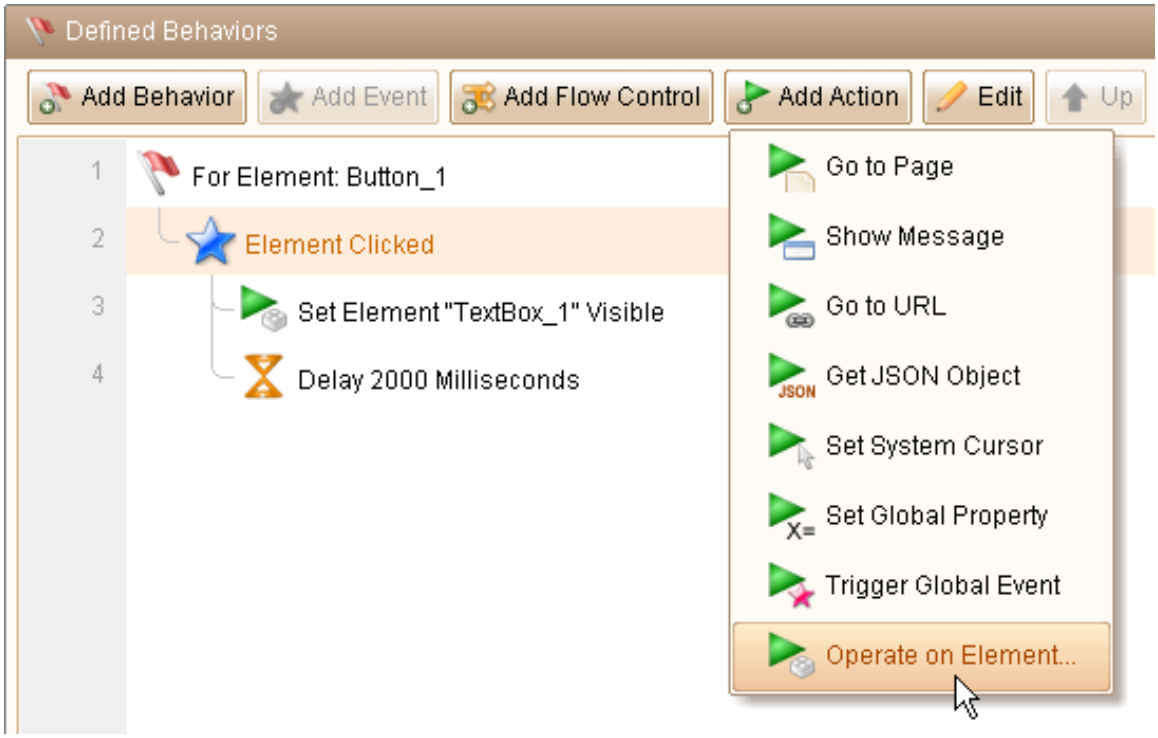
After creating the action, we can select the "Element Clicked" event again and add a delay by clicking the "Add Flow Control" button in toolbar.



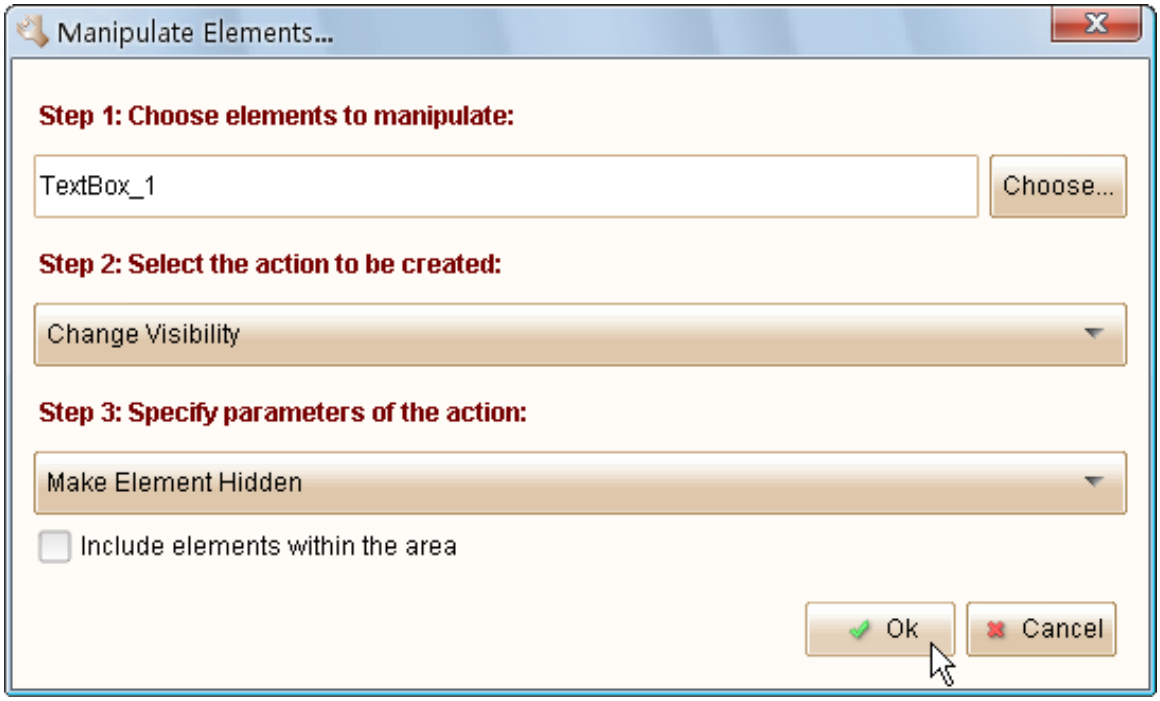
You will see the window below if you select "Delay for a While". Please input 2000 (2 seconds) and click "Ok" button.



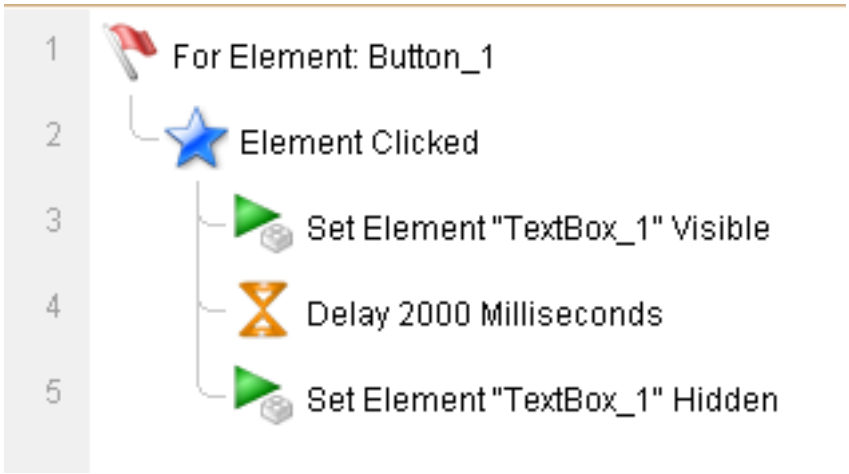
Then we add another action to hide the TextBox element again.



Fill the action editor window like this:



Ok, below is the final behavior definition:



Now you can run the simulation and see how it works. You can view this online example to see the result, or download the plot file.

5.6 Example 6: Using Custom Event

Run This Example in New Window

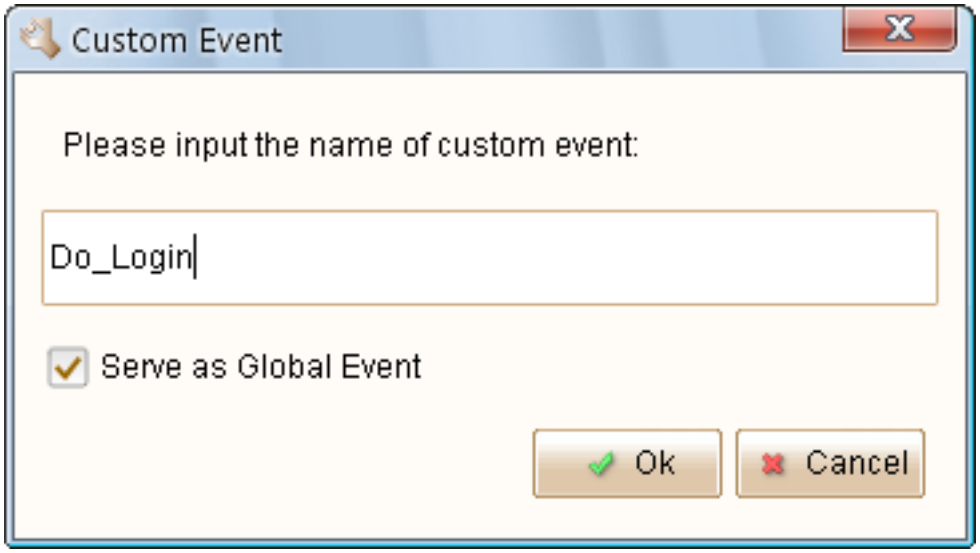
If you want to define the same behavior for different events, you may consider copying/pasting the content between event handlers. That works but it is not the best solution, the best practice is to define the behavior in a handler of "Custom Event", and trigger this custom event when you want the behavior be executed.

If you have programming experience, you will realize the way to use custom event is very similar with function call: you define a function, and call it at some places.

Now lets create a simple example to demonstrate how to use custom event: input password "1234", and press ENTER key or click button to login.

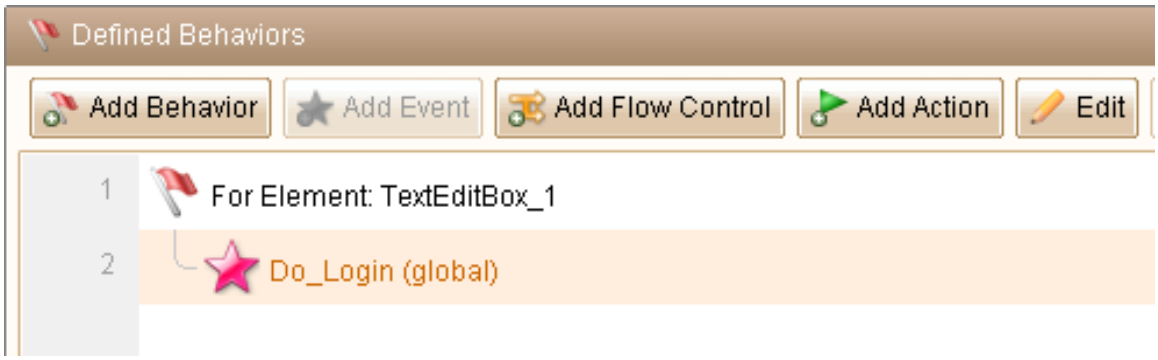
First we need to place a TextBox, a TextEditBox and a Button element into the page to construct a login UI. The TextEditBox should have the "Encrypt" option selected since it is used to accept password.

Since clicking the login button and pressing ENTER key in TextEditBox have the same behavior, we can define a custom event for that. Have the TextEditBox selected and press Ctrl+D to define behavior for it, then click "Add Event" button in toolbar and choose "Custom Event...", you will see this window:

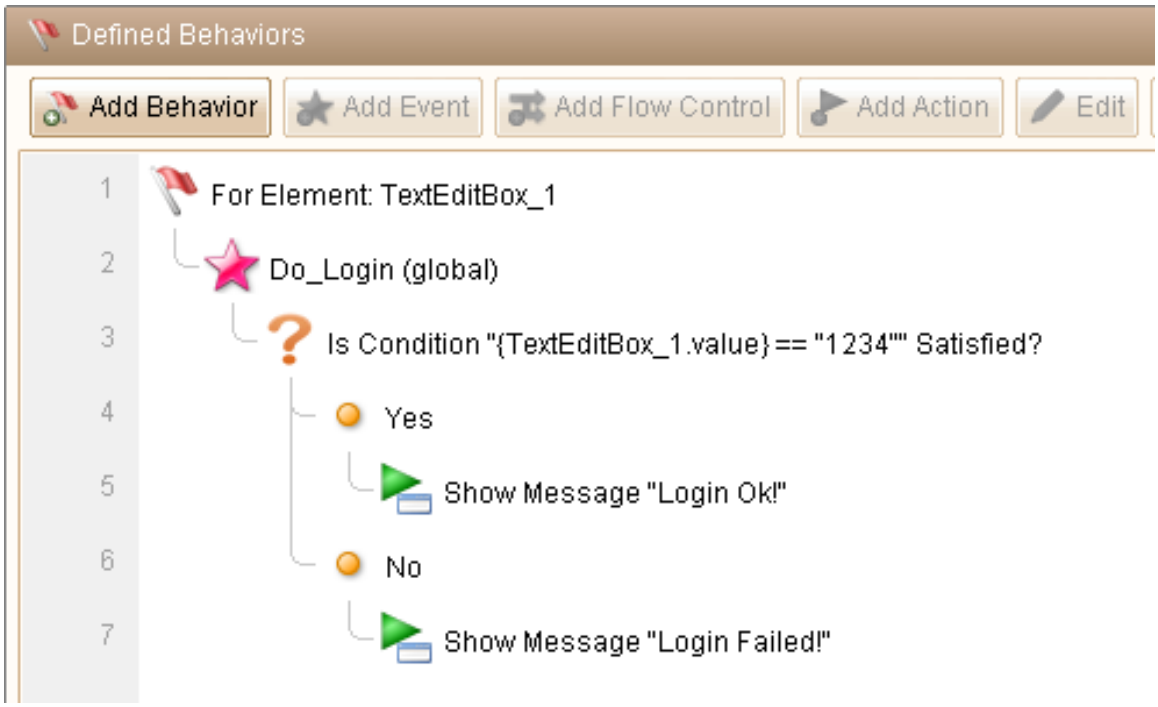


Here we need to input the name of the custom event, only letters and underline can be accepted. Let's use "Do_Login" as the event name. Here the "Serve as Global Event" option is selected by default, which means the event can be triggered anywhere and the event handler will be invoke whenever the event is triggered. In most cases we can let it work in this way, but it may bring some problems when we have duplicated custom event names in the same plot. We will talk about this at the end of this example.

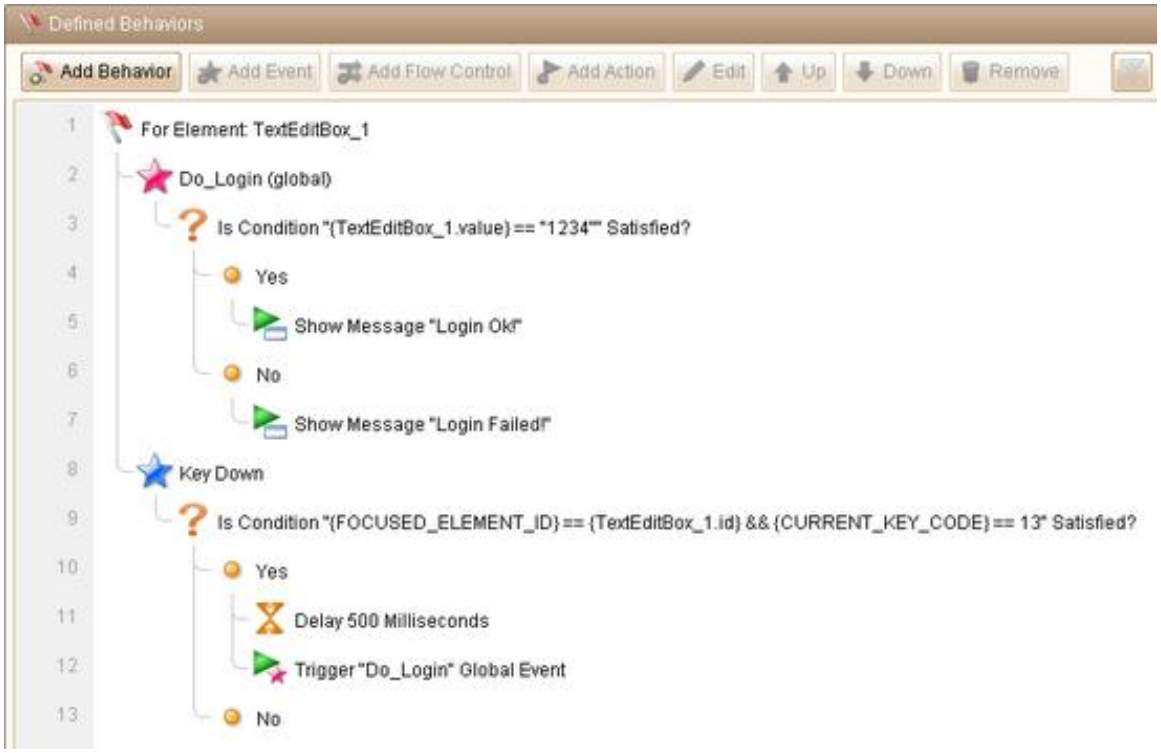
The defined custom event looks like this:



Lets finish the event handler for this custom event by adding conditional branch and show message actions, as shown below:



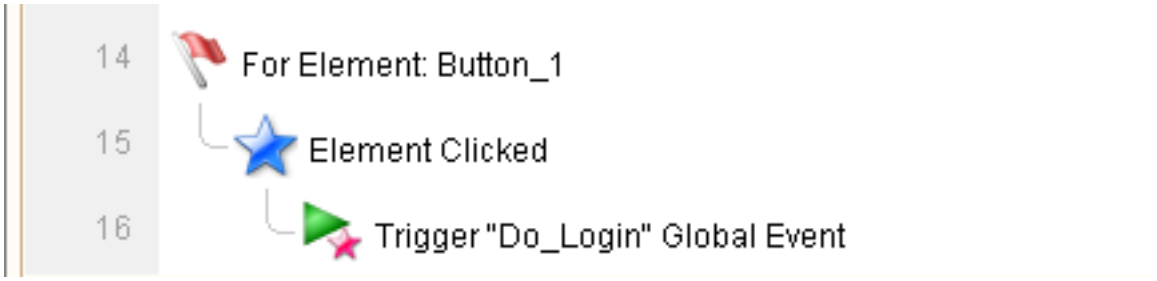
Now we want to trigger this custom event when user press ENTER key in the TextEditBox. We need to add a "Key Down" event for the TextEditBox, and invoke the "Trigger Global Event" action to trigger the "Do_Login" custom event when the TextEditBox has focus and the pressed key is ENTER. Please check out the figure below for details:



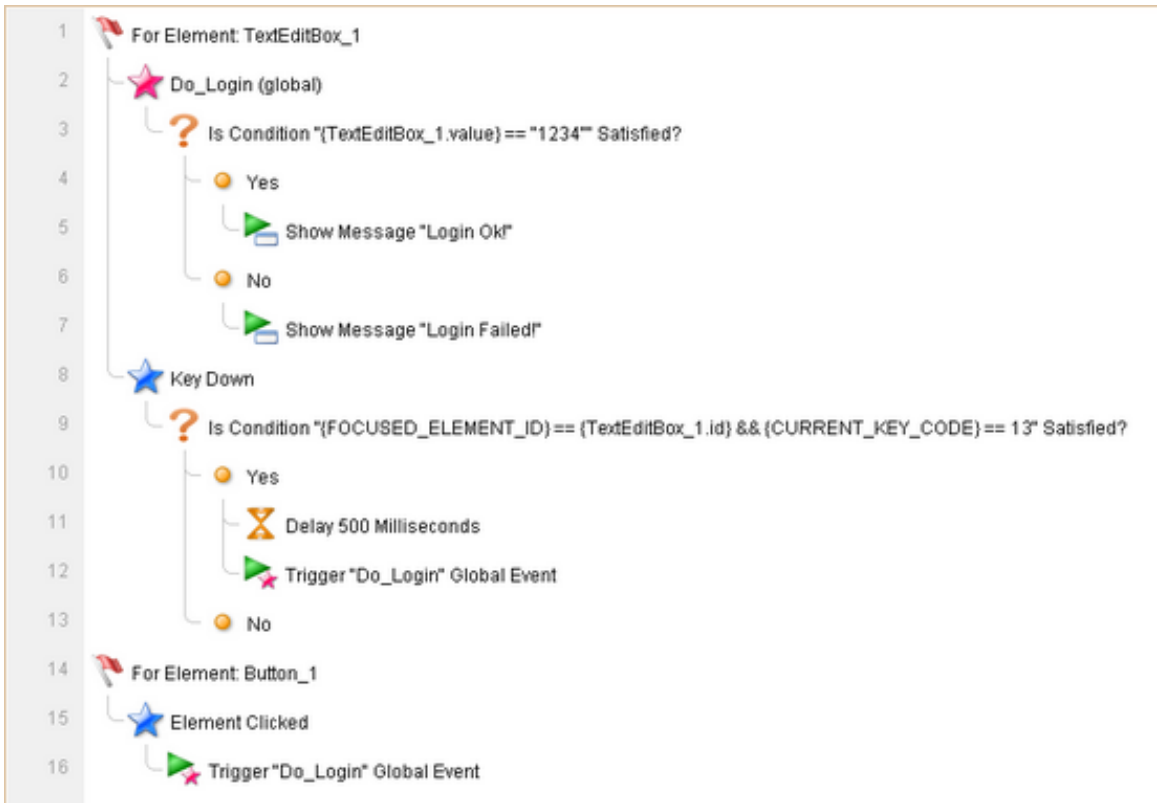
You may notice there is a "Delay" before actually triggering the custom event, it is related to the keyboard event. When you press the ENTER key on keyboard, the "Key Down", "Key Up" and "Key Clicked" events

are triggered in the browser. When the "Key Down" event handler invoke the custom event handler, the login result will be popped up as a message box. If the "Key Clicked" event comes after showing the message box, it will automatically close that message box and you will never see the login result. So, we put a Delay here, to delay the invoking of custom event handler, and avoid the pop up window to be closed automatically.

Also we need to trigger the same custom event when user click the "Login" button. This is quite easy:



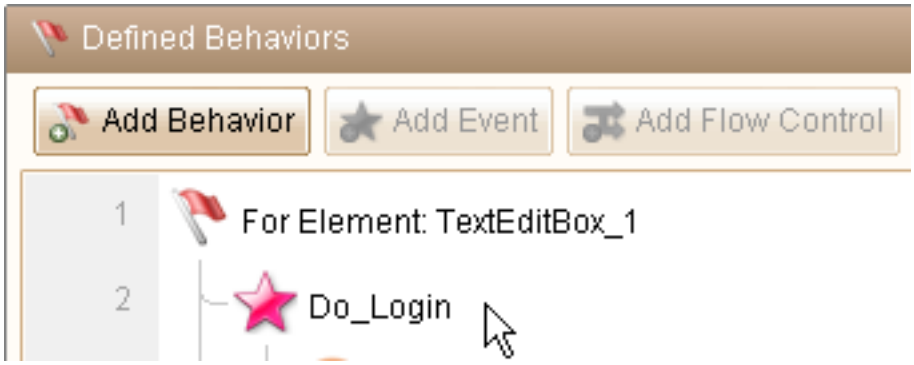
So the final behavior looks like this:



If you run the simulation, you will see it works as we expected: you input "1234" and press ENTER or click "Login", it says "Login Ok!", otherwise it says "Login Failed!".

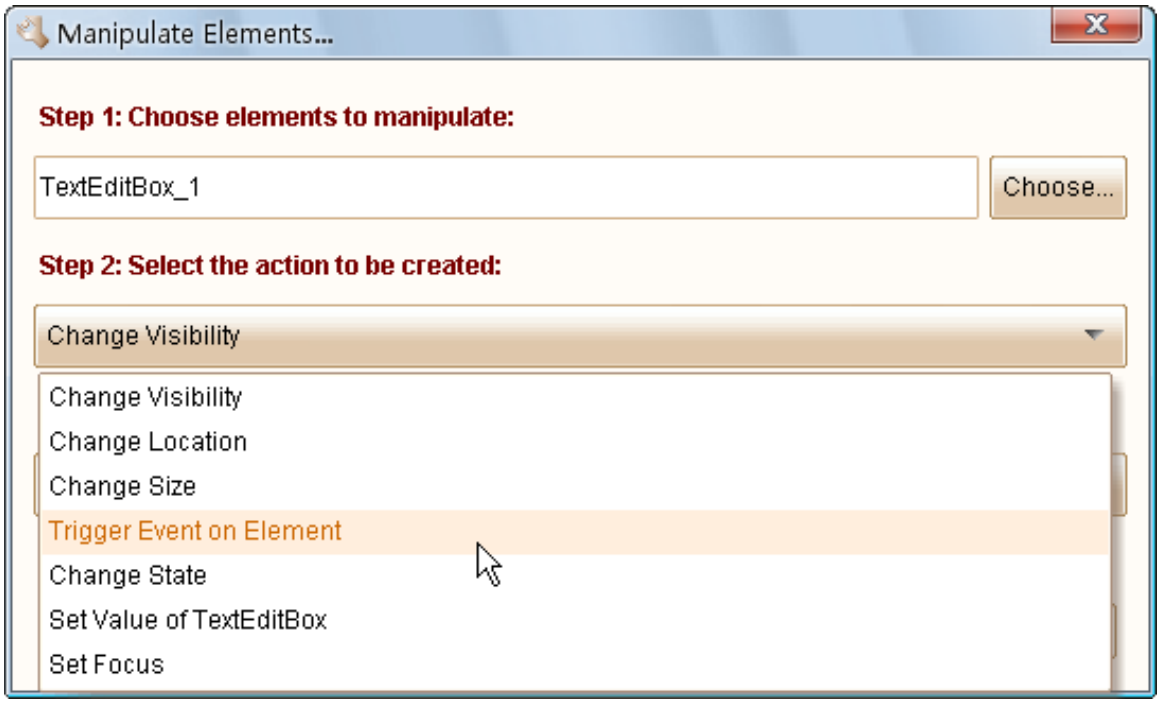
What happen if you have multiple handlers for the same (global) custom event?

Good question. In this case all handlers will be invoked. If it is not the behavior you want, you may consider using non-global custom event. Do you still remember the "Serve as Global Event" option when creating the custom event? You can double click the custom event to change it after creation. If you unselect that option, the custom event become "non-global", which means it belongs to the element that you add the custom event. As shown below:

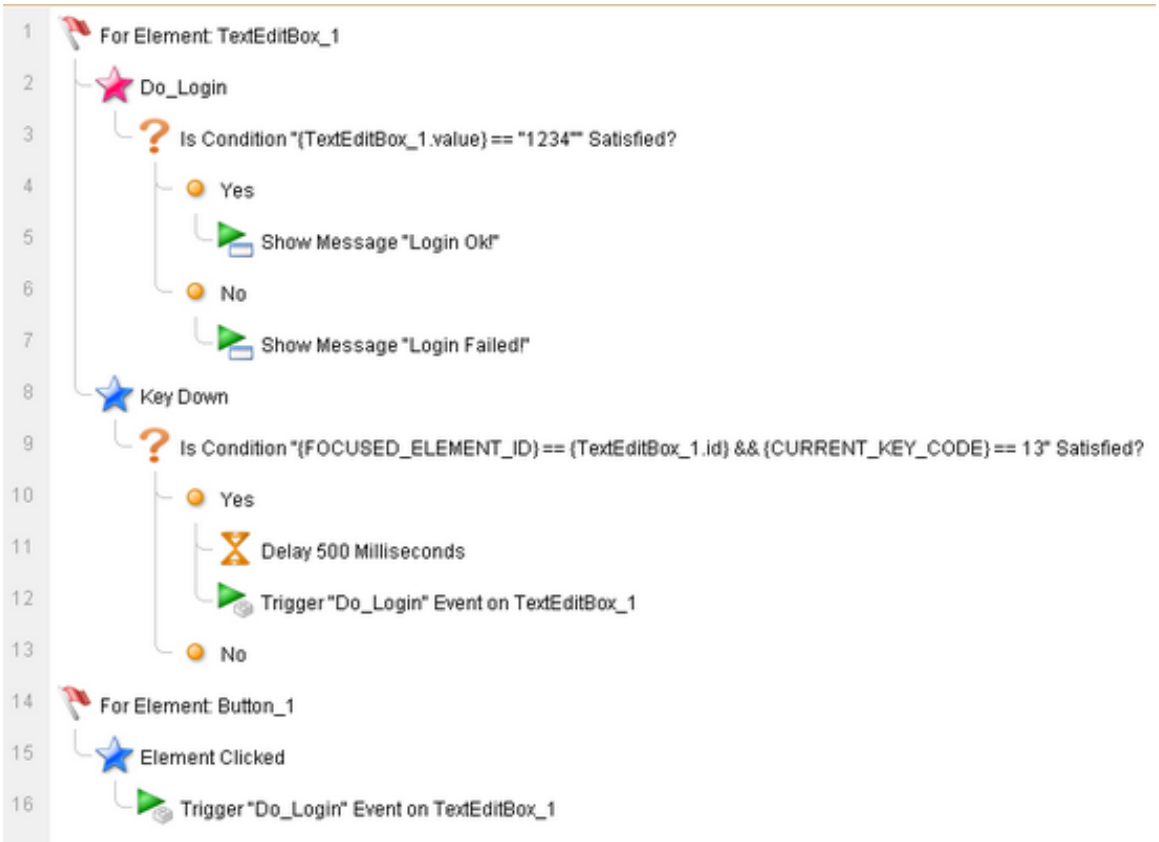


You will notice the "(global)" text disappear now, indicating that custom event is not global now, it belongs to the TextEditBox_1 element only.

After doing this, you could not trigger this custom event by using the "Trigger Global Event" action, instead you have to trigger this custom event via the "Trigger Event on Element" action, which is under the "Operate on Element..." category:



By doing so, you can define multiple custom events with the same name. They will not affect each other as long as you add them to different element. The final behavior with non global custom event is shown below:



Now you can run the simulation and see how it works. You can view this online example to see the result, or download the plot file.

6. Reference

This section includes the reference for all:

- Elements
- Events
- Flow Controls
- Actions
- Properties

6.1 Elements

There are many types of elements in ForeUI. Once you select an element in the plot, the tools panel will show up and list all applicable facilities (buttons, checkbox, slider etc.) for the selection. Different elements will have different facilities to tweak, different events to handle, and different actions to perform.

Reference

Reference is a basic element that can emulate another element's look and behavior. Or say, it can work as an Element Reference.

You can find the "Reference" element in the elements pane, and add it into your page. It looks like:

That means it has no reference target yet. You can specify its target from the tools panel.

After specifying the target element, the Reference element will emulate the look of target element.

Since the Reference element is just a reference of the target element, resizing the Reference element will not affect the target element.

Element Events

Element Clicked, Element Double-Clicked, Element Right-Clicked, Element Initialized, Element Hidden, Mouse Over, Mouse Out, Mouse Move, Mouse Down, Mouse Up, Key Down, Key Up, Custom Event

Element Actions

Change Visibility, Change Location, Change Size

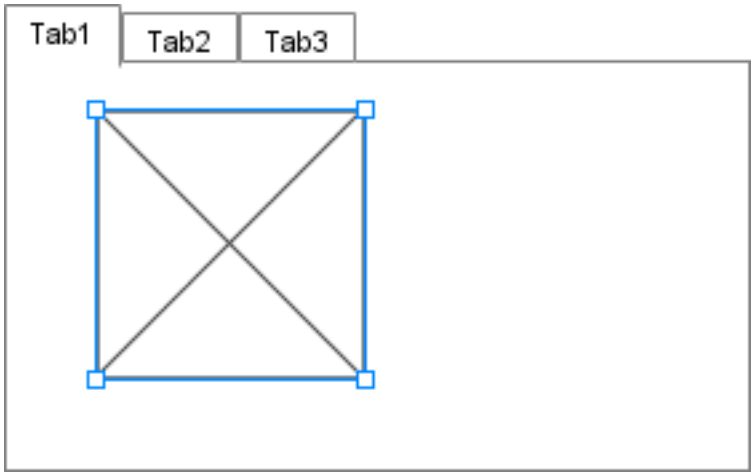
Element Properties

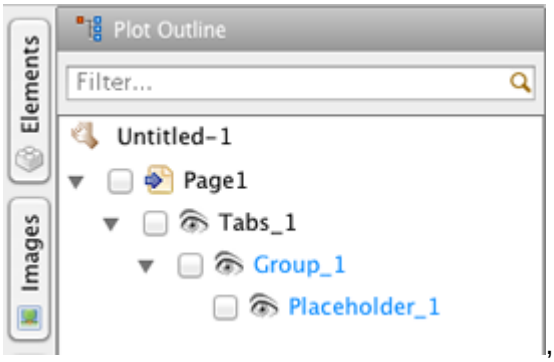
Id, X Coordinate, Y Coordinate, Width, Height, Visible, Note

Group

Group is a very special element to conjoin multiple elements. You could not find it from the elements panel, instead ForeUI automatically generates it and wraps the current selected elements inside when you click the  button in the tools panel.

Group is also used as the wrapper of elements that embedded into the container. For example, when you embed a Placeholder element into a Tabs element, a Group element will be generated and wrap the Placeholder element inside.





Element Events

Element Clicked, Element Double-Clicked, Element Right-Clicked, Element Initialized, Element Hidden, Mouse Over, Mouse Out, Mouse Move, Mouse Down, Mouse Up, Key Down, Key Up, Custom Event

Element Actions

Change Visibility, Change Location, Change Size

Element Properties

Id, X Coordinate, Y Coordinate, Width, Height, Visible, Note

ClipArt

ClipArt is very similar with Group, or say it is a extended version of Group. Besides all the features that a Group has, it enforces all its members to be converted as a static picture in the HTML5 simulation. Converting elements that are "static" in simulation to a ClipArt can reduce the DOM objects in your HTML5 simulation and hence accelerate the loading.

The same as Group, ClipArt is a special element to conjoin multiple elements, and you could not find it from the Elements Panel. After selecting one or more elements you can click the  button in the Tools Panel to convert it/them into a ClipArt.

Although ClipArt will be output as static picture in the simulation, you can still edit its member in the editing mode, just like they are within a Group.

Element Events

Element Clicked, Element Double-Clicked, Element Right-Clicked, Element Initialized, Element Hidden, Mouse Over, Mouse Out, Mouse Move, Mouse Down, Mouse Up, Key Down, Key Up, Custom Event

Element Actions

Change Visibility, Change Location, Change Size

Element Properties

Id, X Coordinate, Y Coordinate, Width, Height, Visible, Note

Line

Line is a basic element, besides in the Hand Drawing theme, its appearance doesn't change for other UI themes.

UI Theme:	Hand Drawn	Blueprint	Other UI Themes
Legend			

Line element can simulate vertical/horizontal splitter, or common lines for any purpose.

Element Specific Facilities

1. Line Type
2. Line Direction
3. Line Thickness (in pixels)

Remarks: the "dashed" and "dotted" line types are not supported in HTML5 simulation, and they will be rendered as solid line.

Element Events

Element Clicked, Element Double-Clicked, Element Right-Clicked, Element Initialized, Element Hidden, Mouse Over, Mouse Out, Mouse Move, Mouse Down, Mouse Up, Key Down, Key Up, Custom Event

Element Actions

Change Visibility, Change Location, Change Size

Element Properties

Id, X Coordinate, Y Coordinate, Width, Height, Visible, Note

Rectangle

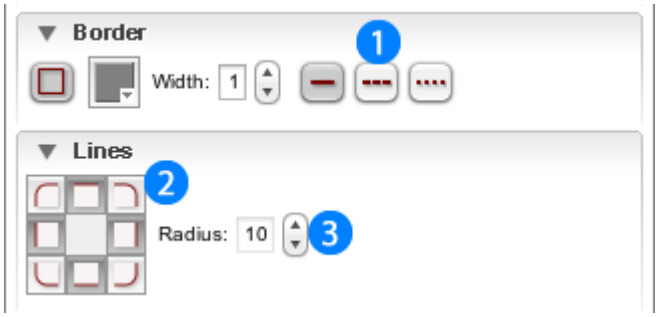
Rectangle is a basic element, besides in the Hand Drawing theme, its appearance doesn't change for other UI themes.

UI Theme:	Hand Drawn	Blueprint	Other UI Themes

Legend			

You can completely control the look of the rectangle. If you want to add text into the rectangle, and rounded corner is not necessary, you can consider using the Text Box.

Element Specific Facilities



1. Change the line type of border
2. Show/hide any border lines
3. Set the border radius (in pixels)

Element Events

Element Clicked, Element Double-Clicked, Element Right-Clicked, Element Initialized, Element Hidden, Mouse Over, Mouse Out, Mouse Move, Mouse Down, Mouse Up, Key Down, Key Up, Custom Event

Element Actions

Change Visibility, Change Location, Change Size

Element Properties

Id, X Coordinate, Y Coordinate, Width, Height, Visible, Note

Ellipse

Ellipse is a basic element, besides in the Hand Drawing theme, its appearance doesn't change for other UI themes.

UI Theme:	Hand Drawn	Blueprint	Other Themes
Legend			

Ellipse can be used to consist complex shape by conjoining it with other elements.

Element Specific Facilities

1. Border show / hide
2. Set border color
3. Border thickness (in pixels)
4. Border type

Remarks: the "dashed" and "dotted" line types are not supported in simulation, and they will be rendered as solid line.

Element Events

Element Clicked, Element Double-Clicked, Element Right-Clicked, Element Initialized, Element Hidden, Mouse Over, Mouse Out, Mouse Move, Mouse Down, Mouse Up, Key Down, Key Up, Custom Event

Element Actions

Change Visibility, Change Location, Change Size

Element Properties

Id, X Coordinate, Y Coordinate, Width, Height, Visible, Note

Triangle

Triangle is a basic element, besides in the Hand Drawing theme, its appearance doesn't change for other UI themes.

UI Theme:	Hand Drawn	Blueprint	Other Themes
Legend			

Element Specific Facilities

1. Show/hide border, set Border color and change border thickness (in pixels)
2. Set border line type
3. Input the angle (in degrees) and click the button to rotate the triangle

Remarks: the "dashed" and "dotted" line types are not supported in simulation, and they will be rendered as solid line.

Element Events

Element Clicked, Element Double-Clicked, Element Right-Clicked, Element Initialized, Element Hidden, Mouse Over, Mouse Out, Mouse Move, Mouse Down, Mouse Up, Key Down, Key Up, Custom Event

Element Actions

Change Visibility, Change Location, Change Size

Element Properties

Id, X Coordinate, Y Coordinate, Width, Height, Visible, Note

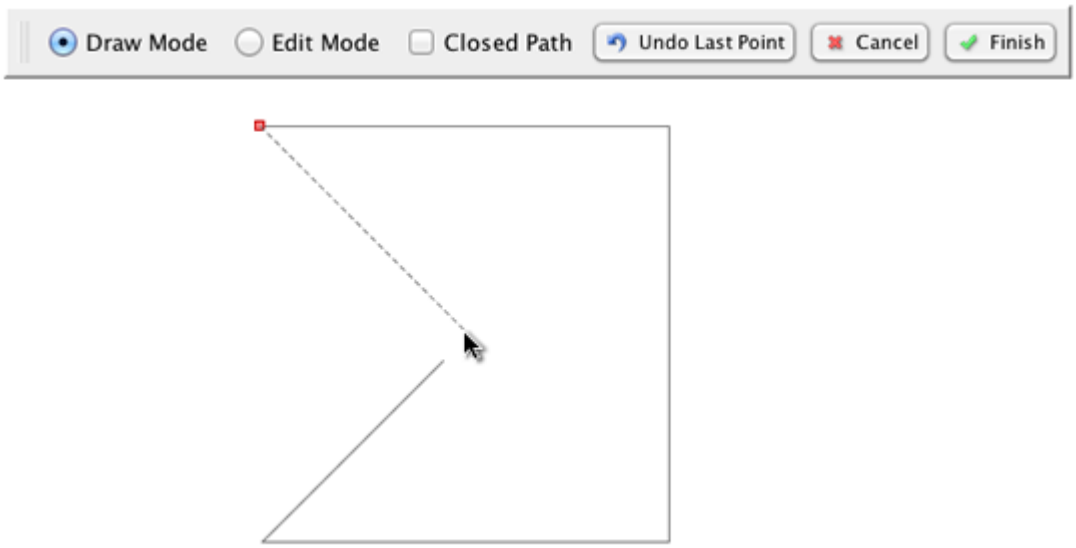
Polygon

Polygon is a basic element, besides in the Hand Drawing theme, its appearance doesn't change for other UI themes.

UI Theme:	Hand Drawn	Blueprint	Other Themes
Legend			

When you drag the Polygon button into the editing area, you start defining the polygon (point by point).

You can hold SHIFT key to draw horizontal, vertical or 45 degree lines:



You can click the "Finish" button to quit the polygon drawing, when you are satisfied with the shape.

If you choose "Edit Mode", you can edit any existing point with drag and drop.

If you select "Closed Path", the polygon will be closed automatically.

Element Specific Facilities

1. Enable/disable border rendering
2. Set Border color
3. Change border thickness (in pixels)
4. Set border line type

Remarks: the "dashed" and "dotted" line types are not supported in simulation, and they will be rendered as solid line.

Element Events

Element Clicked, Element Double-Clicked, Element Right-Clicked, Element Initialized, Element Hidden, Mouse Over, Mouse Out, Mouse Move, Mouse Down, Mouse Up, Key Down, Key Up, Custom Event

Element Actions

Change Visibility, Change Location, Change Size

Element Properties

Id, X Coordinate, Y Coordinate, Width, Height, Visible, Note

Placeholder

Placeholder is a basic element, besides in the Hand Drawing theme, its appearance doesn't change for other UI themes.

UI Theme:	Hand Drawn	Blueprint	Other Themes
Legend			

Placeholder can emulate not specific image or block in web site or software.

Element Specific Facilities

1. Set border color
2. Set border thickness (in pixels)
3. Border line type

Remarks: the "dashed" and "dotted" line types are not supported in simulation, and they will be rendered as solid line.

Element Events

Element Clicked, Element Double-Clicked, Element Right-Clicked, Element Initialized, Element Hidden, Mouse Over, Mouse Out, Mouse Move, Mouse Down, Mouse Up, Key Down, Key Up, Custom Event

Element Actions

Change Visibility, Change Location, Change Size

Element Properties

Id, X Coordinate, Y Coordinate, Width, Height, Visible, Note

Image Box

Image Box is a basic element, besides in the Hand Drawing theme, its appearance doesn't change for other UI themes.

UI Theme:	Hand Drawn	Blueprint	Other Themes

Legend			

Image Box is a container of image, it can reference an image in the images panel. You can change its image source from the tools panel, or double-clicking the image box to do the job.

Element Specific Facilities

By clicking the  button, you can choose the image source from an external image file or from the icon library.

Element Events

Element Clicked, Element Double-Clicked, Element Right-Clicked, Element Initialized, Element Hidden, Mouse Over, Mouse Out, Mouse Move, Mouse Down, Mouse Up, Key Down, Key Up, Custom Event

Element Actions

Change Visibility, Change Location, Change Size, Set Image Source

Element Properties

Id, X Coordinate, Y Coordinate, Width, Height, Visible, Note

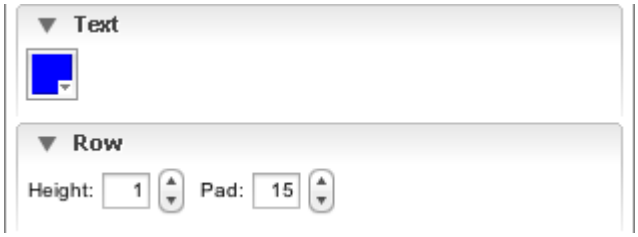
Mock Text

Mock Text is a basic element, besides in the Hand Drawing and Wireframe themes, its appearance doesn't change for other UI themes.

UI Theme:	Hand Drawn	Blueprint	Wireframe	Other Themes
Legend				

Mock Text can simulate a paragraph of text, without adding any detail about the content. You can regard it as a placeholder for text.

Element Specific Facilities



1. The text color
2. The row height (in pixels)
3. The padding between rows (in pixels)

Element Events

Element Clicked, Element Double-Clicked, Element Right-Clicked, Element Initialized, Element Hidden, Mouse Over, Mouse Out, Mouse Move, Mouse Down, Mouse Up, Key Down, Key Up, Custom Event

Element Actions

Change Visibility, Change Location, Change Size

Element Properties

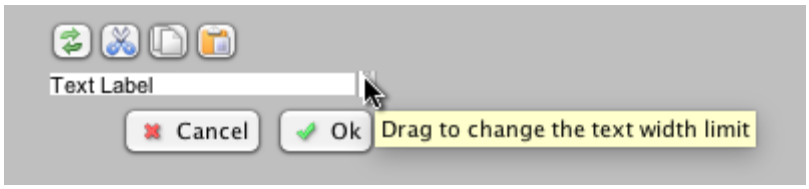
Id, X Coordinate, Y Coordinate, Width, Height, Visible, Note

Text Label (Text Box)

Text Label is a basic element. Its has the same look for all UI themes, except that it has a default "Comic Sans MS" font in the Hand Drawing theme.

UI Theme:	Hand Drawn	Blueprint	All Themes
Legend			

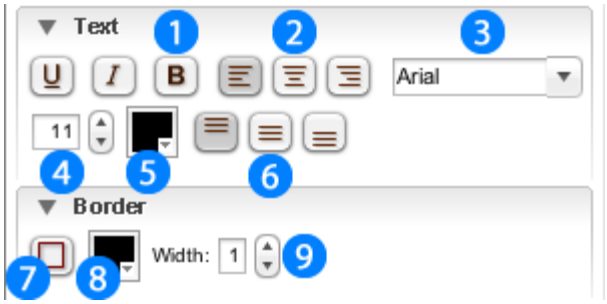
Text Box can simulate a paragraph of text, or a single row headline. Double clicking the Text Box element will bring up its content editor.



Dragging the handler on the right can change the preferred width, if the text can not be hold in a single row, it will wrap to next row automatically.

Remarks: adding HTML tags into text content is allowed. But it will not take effect until you run the simulation in web browser.

Element Specific Facilities



1. Specify text style (underline, italic and bold)
2. Horizontal align of the text (left, center or right)
3. Set font name
4. Set font size
5. Text color
6. Vertical align of the text (top, middle or bottom)
7. Show/hide border
8. Border color
9. Border width (in pixels)

Element Events

Element Clicked, Element Double-Clicked, Element Right-Clicked, Element Initialized, Element Hidden, Mouse Over, Mouse Out, Mouse Move, Mouse Down, Mouse Up, Key Down, Key Up, Custom Event

Element Actions

Change Visibility, Change Location, Change Size, Change State, Change Text, Change Opacity, Change Background Color, Change Text Color

Element Properties

Id, X Coordinate, Y Coordinate, Width, Height, Visible, Note

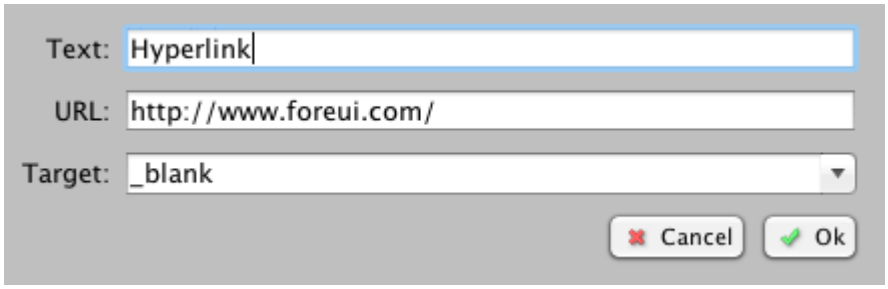
Current Value of Text Label, Current Length of Text

Hyperlink

Hyperlink is a basic element. Its has the same look for all UI themes, except that it has a default "Comic Sans MS" font in the Hand Drawing theme.

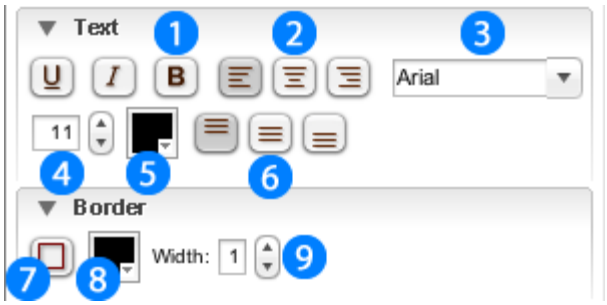
UI Theme:	Hand Drawn	Blueprint	All Themes
Legend			

Hyperlink can simulate a text link. Double clicking the Hyperlink element will bring up its content editor.



Remarks: if you leave the URL empty, the Hyperlink element will not bring you to any URL after you click on it.

Element Specific Facilities



1. Specify text style (underline, italic and bold)
2. Horizontal align of the text (left, center or right)
3. Set font name
4. Set font size
5. Text color
6. Vertical align of the text (top, middle or bottom)
7. Show/hide border
8. Border color
9. Border width (in pixels)

Element Events

Element Clicked, Element Double-Clicked, Element Right-Clicked, Element Initialized, Element Hidden, Mouse Over, Mouse Out, Mouse Move, Mouse Down, Mouse Up, Key Down, Key Up, Custom Event

Element Actions

Change Visibility, Change Location, Change Size, Change State, Change Text, Change Opacity, Change Background Color, Change Text Color

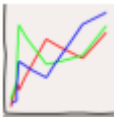
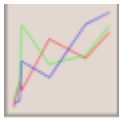
Element Properties

Id, X Coordinate, Y Coordinate, Width, Height, Visible, Note

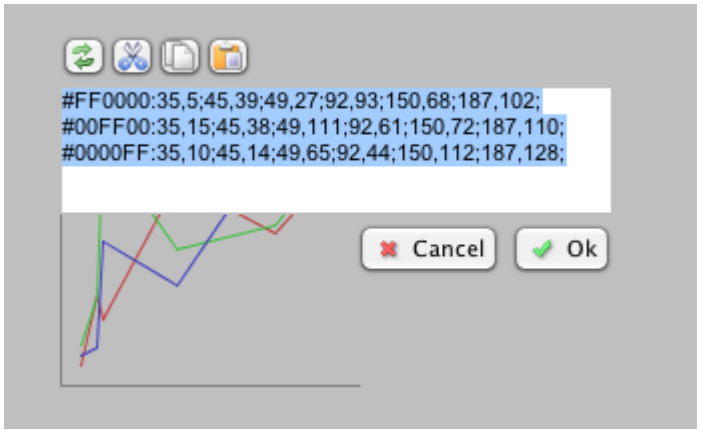
Current Value of Text Box

Line Chart

Line Chart is a basic element. It has thicker lines in the Hand Drawing theme, and has the same look in other UI themes.

UI Theme:	Hand Drawn	Other Themes
Legend		

Double clicking the line chart can edit the data for the lines.



Each row is a record, and its format is "Color:x1,y1;x2,y2;x3,y3;.....".

Element Specific Facilities

1. Show/hide any border of the chart
2. Border color
3. Set border width (in pixels)
4. 5. Input the lines count, and press the "Shuffle" button to generate the lines randomly.

Element Events

Element Clicked, Element Double-Clicked, Element Right-Clicked, Element Initialized, Element Hidden, Mouse Over, Mouse Out, Mouse Move, Mouse Down, Mouse Up, Key Down, Key Up, Custom Event

Element Actions

Change Visibility, Change Location, Change Size

Element Properties

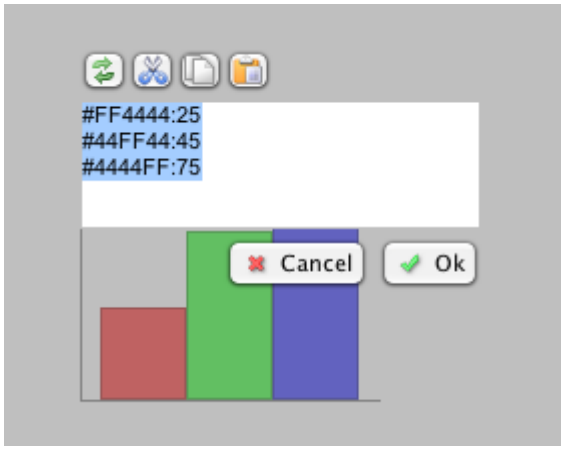
Id, X Coordinate, Y Coordinate, Width, Height, Visible, Note

Bar Chart

Bar Chart is a basic element. It has thicker borders in the Hand Drawing theme, and has the same look in other UI themes.

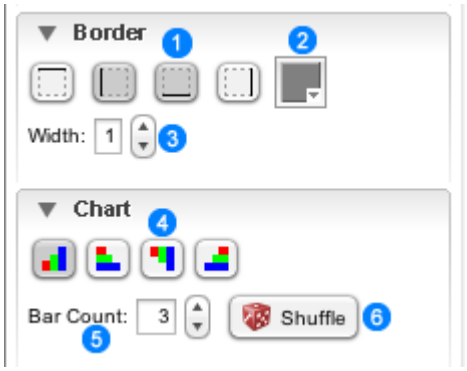
UI Theme:	Hand Drawn	Other Themes
Legend		

Double clicking the line chart can edit the data for the bars.



Each row is for a bar, and its format is "Color:Height".

Element Specific Facilities



1. Show/hide any border of the chart
2. Border color
3. Set border width (in pixels)
4. Set the orientation of the chart
5. 6. Input the bars count, and press the "Shuffle" button to generate the bars randomly.

Element Events

Element Clicked, Element Double-Clicked, Element Right-Clicked, Element Initialized, Element Hidden, Mouse Over, Mouse Out, Mouse Move, Mouse Down, Mouse Up, Key Down, Key Up, Custom Event

Element Actions

Change Visibility, Change Location, Change Size

Element Properties

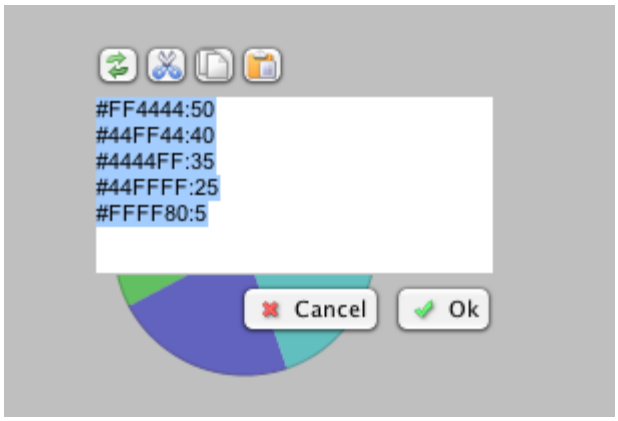
Id, X Coordinate, Y Coordinate, Width, Height, Visible, Note

Pie Chart

Pie Chart is a basic element. It has the same look in all UI themes.

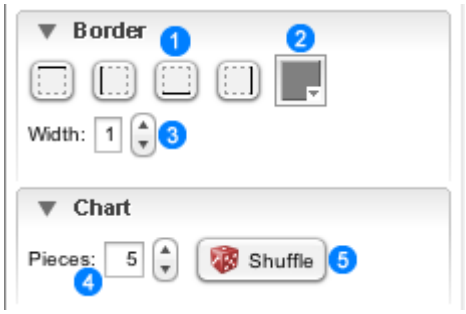
UI Theme:	All Themes
Legend	

Double clicking the line chart can edit the data for the pie chart.



Each row is for a piece, and its format is "Color:Value".

Element Specific Facilities



1. Show/hide any border of the chart
2. Border color
3. Set border width (in pixel)

4. 5. Input the pieces count, and press the "Shuffle" button to generate the chart randomly.

Element Events

Element Clicked, Element Double-Clicked, Element Right-Clicked, Element Initialized, Element Hidden, Mouse Over, Mouse Out, Mouse Move, Mouse Down, Mouse Up, Key Down, Key Up, Custom Event

Element Actions

Change Visibility, Change Location, Change Size

Element Properties

Id, X Coordinate, Y Coordinate, Width, Height, Visible, Note

iFrame

iFrame is a basic element. In the Hand Drawing theme, it has a thicker border, and it has the same look in other UI themes.

UI Theme:	Hand Drawn	Other Themes
Legend		

iFrame can simulate the iframe element in web page. Double clicking the iFrame element can change the URL of its content page. The iFrame element will be converted to a real iframe when you run HTML5 simulation.

Element Specific Facilities

1. Show/hide border
2. Border color
3. Set border width (in pixels)
4. Specify the policy for scrolling

Element Events

Element Clicked, Element Double-Clicked, Element Right-Clicked, Element Initialized, Element Hidden, Mouse Over, Mouse Out, Mouse Move, Mouse Down, Mouse Up, Key Down, Key Up, Custom Event

Element Actions

Change Visibility, Change Location, Change Size, Set IFrame Source URL

Element Properties

Id, X Coordinate, Y Coordinate, Width, Height, Visible, Note

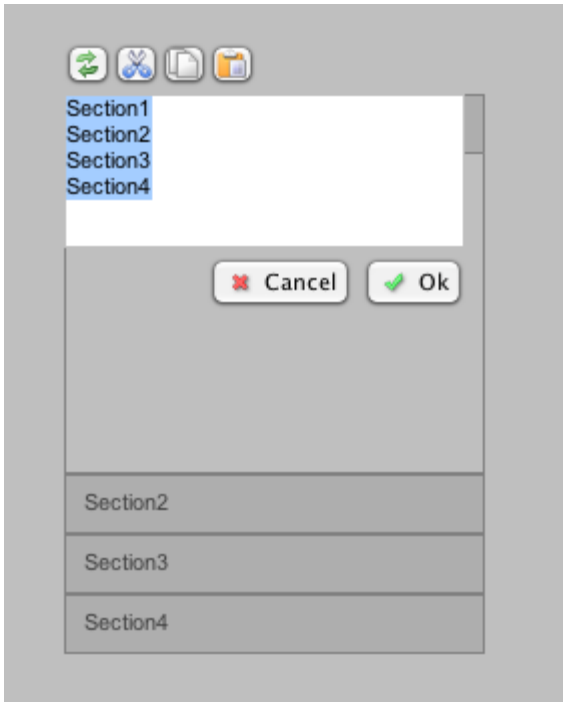
Accordion

Accordion is a basic element, besides in the Hand Drawing theme, its appearance doesn't change for other UI themes.

UI Theme:	Hand Drawn	Other Themes
Legend		

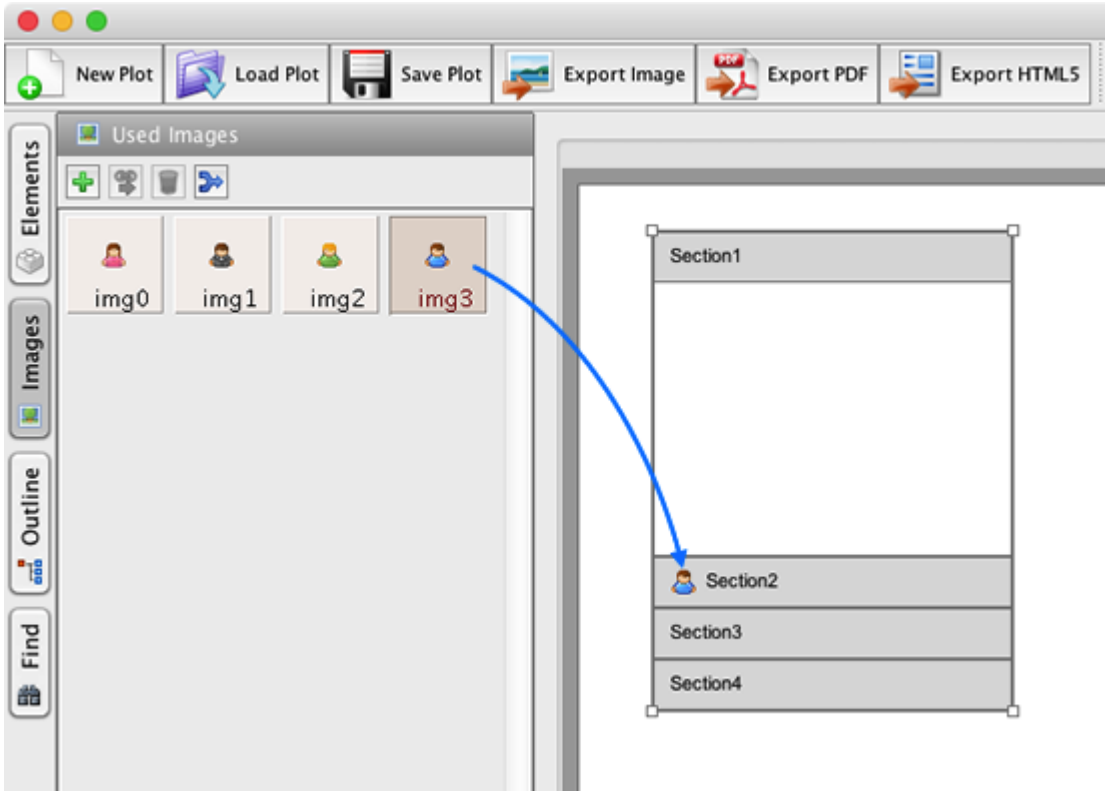
Accordion is a container element, each section of Accordion can accept elements. You can drag element into Accordion's section **with right mouse button hold or Ctrl key pressed**.

Double clicking the Accordion can edit the titles for all sections.

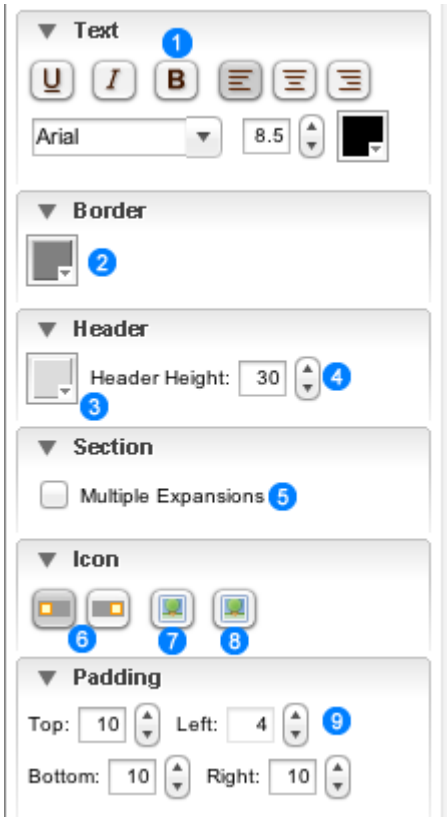


Each row is for one section. Adding more rows will add more sections into accordion as well. Removing certain rows will also remove the corresponding sections. If a section is removed from accordion and no other section can accept its embedded content, it will extract its embedded elements to plot before the deletion.

Accordion can accept images at its section headers. You can drag image from Images Panel to Accordion's section header directly, thus you can change the icon for each section header.



Element Specific Facilities



1. Text formatting
2. Border color
3. Header background color

4. Header height (in pixels)
5. Whether to allow multiple sections be expanded at a time
6. Placement of the icon in header (left or right)
7. Set icons for all collapsed sections
8. Set icons for all expended sections
9. Set padding for section area

Element Events

Element Clicked, Element Double-Clicked, Element Right-Clicked, Element Initialized, Element Hidden, Mouse Over, Mouse Out, Mouse Move, Mouse Down, Mouse Up, Key Down, Key Up, Custom Event, Section Expanded/Collapsed

Element Actions

Change Visibility, Change Location, Change Size, Set Section Title,Expand Accordion Section, Collapse Accordion Section (multi-expansion mode only)

Element Properties

Id, X Coordinate, Y Coordinate, Width, Height, Visible, Note

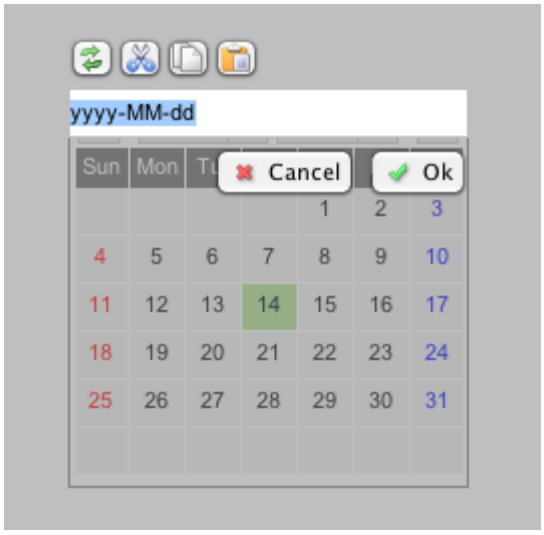
Accordion Section Count, Section Expand Flags, Index of First Expanded Section, Index of Section that is Recently Expanded, Index of Section that is Recently Collapsed, Section Titles as Array

Calendar

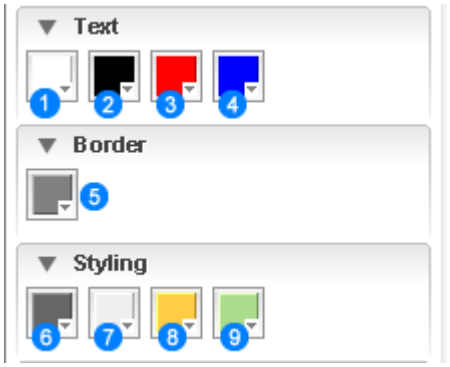
Calendar belongs to the "Widgets" category. It has thicker borders in the Hand Drawing theme.

UI Theme:	Hand Drawn	Other Themes
Legend		

Double clicking the Calendar element can change the date it is presenting. The date format is "yyyy-MM-dd".



Element Specific Facilities



1. Text color for header
2. Text color
3. Color for Sunday text
4. Color for Saturday text
5. Border color
6. Calendar header background color
7. Calendar button color
8. Calendar highlighted button color
9. Calendar selected button color

Element Events

Element Clicked, Element Double-Clicked, Element Right-Clicked, Element Initialized, Element Hidden, Mouse Over, Mouse Out, Mouse Move, Mouse Down, Mouse Up, Key Down, Key Up, Custom Event, Calendar Date Changed

Element Actions

Change Visibility, Change Location, Change Size, Set Selected Date in Calendar

Element Properties

Id, X Coordinate, Y Coordinate, Width, Height, Visible, Note

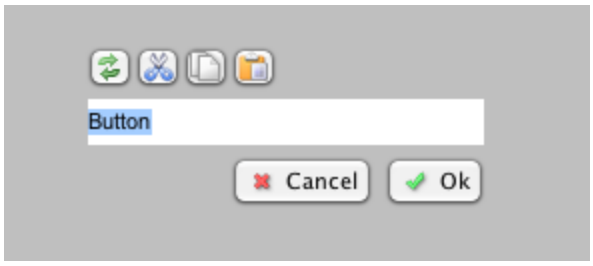
Current Selected Date in Calendar, Current Selected Year in Calendar, Current Selected Month in Calendar, Current Selected Day of Month in Calendar

Button

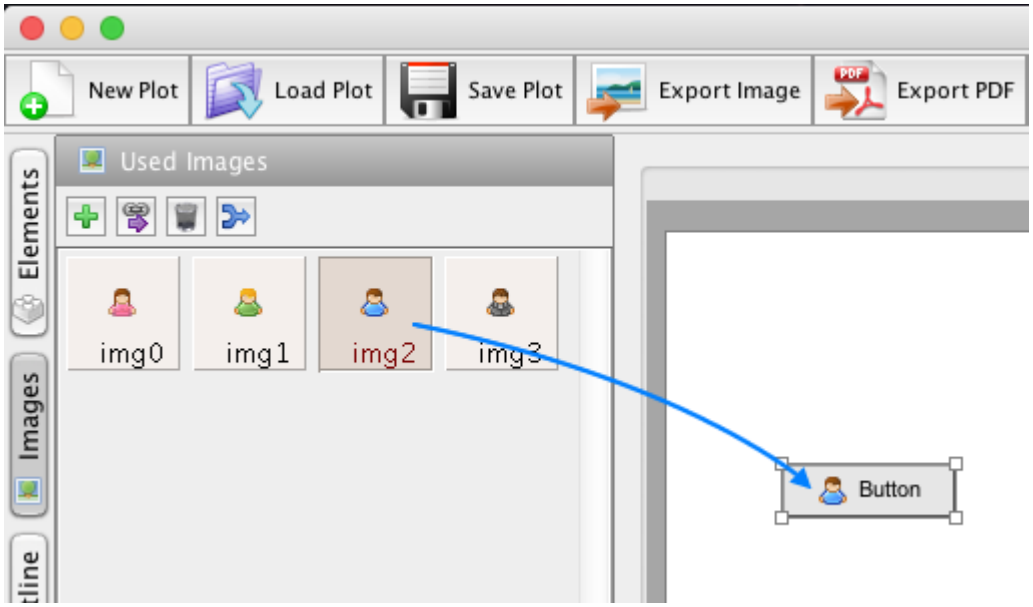
Button belongs to the "Widgets" category and it has different looks in various UI themes.

UI Theme:	Hand Drawn	Wire Frame	Windows XP	MAC OS X	Window 7
Legend					

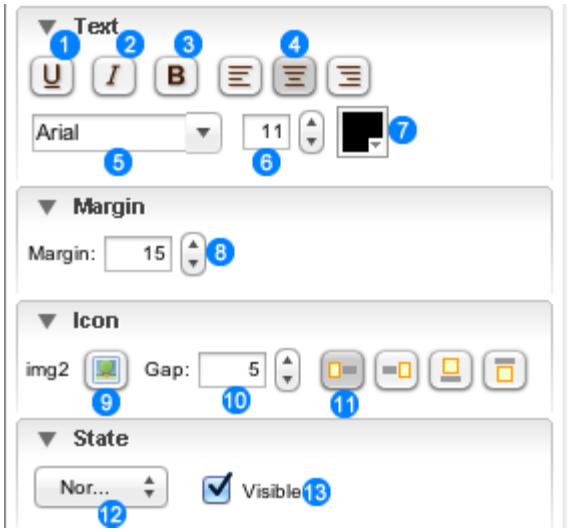
Double clicking the Button element can change its text label.



Button can accept image as its icon. You can change its icon from Tools Panel, or drag image from Images Panel directly into the button.



Element Specific Facilities



1. Underline the text
2. Italic the text
3. Bold the text
4. Align of the text
5. Font of the text
6. Text size
7. Text color
8. Text margin (in pixels)
9. Set icon on the button
10. Set the distance between the icon and the text (in pixels)
11. Set the icon's position with the text
12. 13.Set button state

Element Events

Element Clicked, Element Double-Clicked, Element Right-Clicked, Element Initialized, Element Hidden, Mouse Over, Mouse Out, Mouse Move, Mouse Down, Mouse Up, Key Down, Key Up, Custom Event

Element Actions

Change Visibility, Change Location, Change Size, Change State

Element Properties

Id, X Coordinate, Y Coordinate, Width, Height, Visible, Note

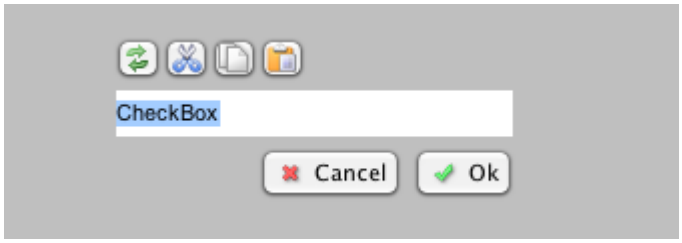
CheckBox

CheckBox belongs to the "Widgets" category and it has different look in various UI themes.

UI Theme:	Hand Drawn	Wire Frame	Windows XP	MAC OS X	Window 7

Legend					

Double clicking the CheckBox element can change its text label.



Element Specific Facilities

1. Font of the text label
2. Text size
3. Text color
4. 5. Set the state of the checkbox

Element Events

Element Clicked, Element Double-Clicked, Element Right-Clicked, Element Initialized, Element Hidden, Mouse Over, Mouse Out, Mouse Move, Mouse Down, Mouse Up, Key Down, Key Up, Custom Event

Element Actions

Change Visibility, Change Location, Change Size, Change State, Select/Unselect Check Box

Element Properties

Id, X Coordinate, Y Coordinate, Width, Height, Visible, Note

Whether the CheckBox is Selected

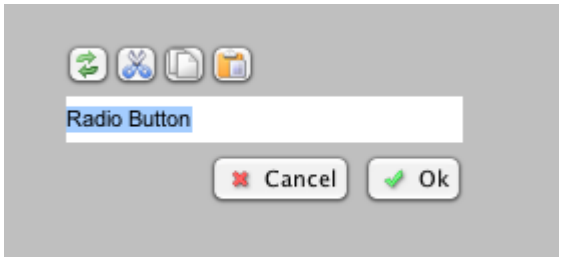
Radio Button

Radio Button belongs to the "Widgets" category and it has different look in various UI themes.

UI Theme:	Hand Drawn	Wire Frame	Windows XP	MAC OS X	Window 7

Legend					

Double clicking the element can change its text label.



If you wish to create a group of radio buttons, which only allows one radio button selected a time, you can use Radio Button Group element.

Element Specific Facilities

1. Font of the text label
2. Font size (in pixels)
3. Text color
4. 5. Change radio button's state

Element Events

Element Clicked, Element Double-Clicked, Element Right-Clicked, Element Initialized, Element Hidden, Mouse Over, Mouse Out, Mouse Move, Mouse Down, Mouse Up, Key Down, Key Up, Custom Event, Selection Changed

Element Actions

Change Visibility, Change Location, Change Size, Change State, Select/Unselect Radio Button

Element Properties

Id, X Coordinate, Y Coordinate, Width, Height, Visible, Note

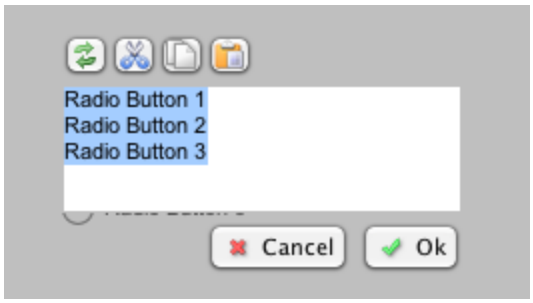
Whether the Radio Button is Selected

Radio Button Group

Radio Button Group belongs to the "Widgets" category and it has different look in various UI themes.

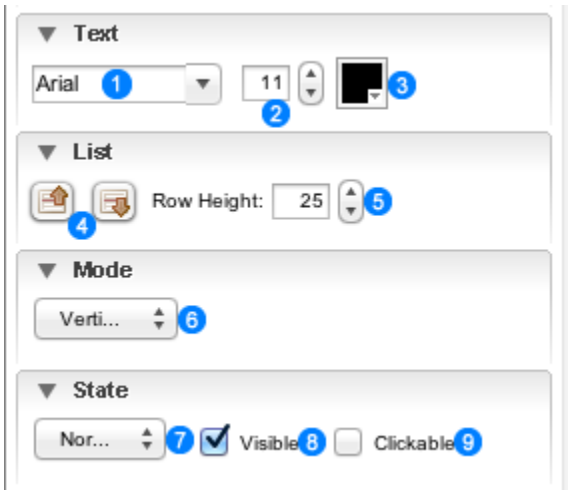
UI Theme:	Hand Drawn	Wire Frame	Windows XP	MAC OS X	Window 7
Legend					

Double clicking the element can change its content.



One row for one radio button. You can add/remove radio buttons by adding/removing rows here.

Element Specific Facilities



1. Font of the text label
2. Font size (in pixels)
3. Text color
4. Select previous or next radio button
5. Change row height
6. Layout mode for the radio button group
7. 8. 9. Change radio button group's state

Element Events

Element Clicked, Element Double-Clicked, Element Right-Clicked, Element Initialized, Element Hidden, Mouse Over, Mouse Out, Mouse Move, Mouse Down, Mouse Up, Key Down, Key Up, Custom Event, Selection Changed

Element Actions

Change Visibility, Change Location, Change Size, Change State, Set Selected Index

Element Properties

Id, X Coordinate, Y Coordinate, Width, Height, Visible, Note

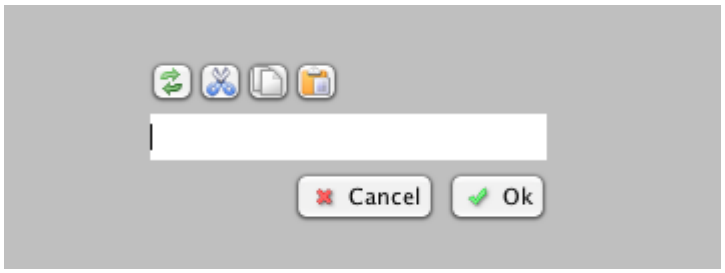
Index of Selected Radio Button, Text of Selected Radio Button

Text Edit Box

Text Edit Box belongs to the "Widgets" category and it has different look in various UI themes.

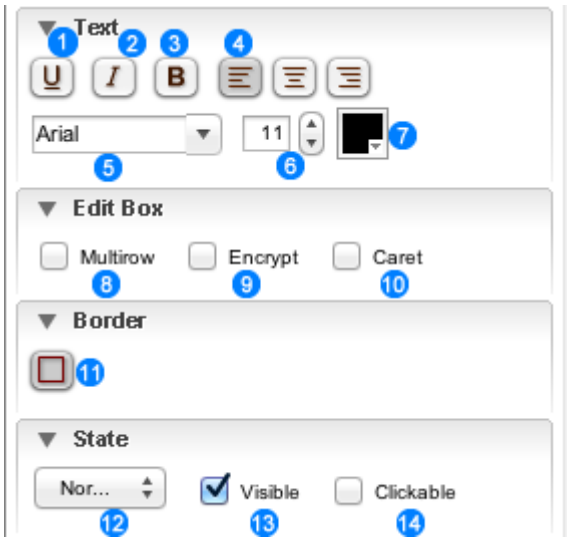
UI Theme:	Hand Drawn	Wire Frame	Windows XP	MAC OS X	Window 7
Legend					

Text Edit Box element can accept user's input, and can serve as input field, multi line text editing area and password field. You can double click the Text Edit Box element to change its default value.



If you need a Text Edit Box with custom look, you can hide its border and place image or other elements under it.

Element Specific Facilities



1. Underline the text

2. Make text italic
3. Use bold font
4. Text alignment
5. Font of the text
6. Font size
7. Text color
8. Multiple row mode
9. Display as password
10. Show caret in edit box
11. Show / hide border
12. 13.14. Change state for text edit box

Element Events

Element Clicked, Element Double-Clicked, Element Right-Clicked, Element Initialized, Element Hidden, Mouse Over, Mouse Out, Mouse Move, Mouse Down, Mouse Up, Key Down, Key Up, Custom Event, Value Changed, Focus Gain, Focus Lost

Element Actions

Change Visibility, Change Location, Change Size, Change State, Set Value of TextEditBox, Set Focus

Element Properties

Id, X Coordinate, Y Coordinate, Width, Height, Visible, Note

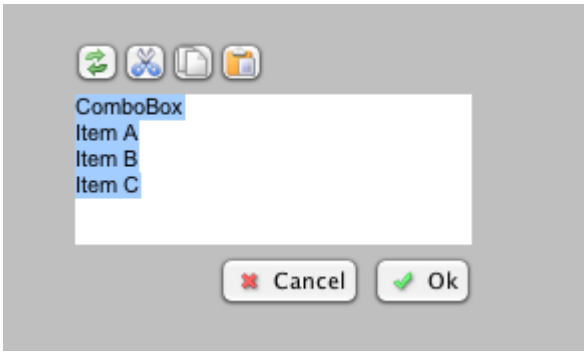
Current Value of TextEditBox, Current Text Length

ComboBox

ComboBox belongs to the "Widgets" category and it has different look in various UI themes.

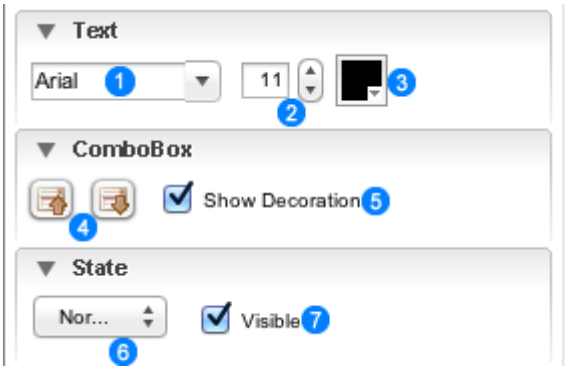
UI Theme:	Hand Drawn	Wire Frame	Windows XP	MAC OS X	Window 7
Legend					

The content of drop down list of ComboBox can be modified by double clicking the element.



If you need a ComboBox with custom look, you can hide its decoration and place image or other elements under it.

Element Specific Facilities



1. Font of the text
2. Font size
3. Text color
4. Select previous/next item
5. Show / hide decoration
6. 7. Change the state of the ComboBox

Element Events

Element Clicked, Element Double-Clicked, Element Right-Clicked, Element Initialized, Element Hidden, Mouse Over, Mouse Out, Mouse Move, Mouse Down, Mouse Up, Key Down, Key Up, Custom Event, Selection Changed

Element Actions

Change Visibility, Change Location, Change Size, Change State, Set Selected Index

Element Properties

Id, X Coordinate, Y Coordinate, Width, Height, Visible, Note

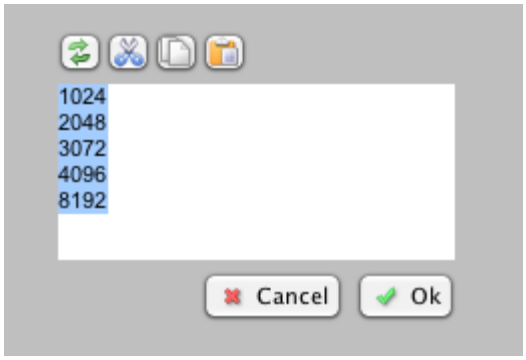
Index of the Selected Item, Text of the Selected Item

Stepper (Spinner)

Stepper (or spinner) belongs to the "Widgets" category and it has different look in various UI themes.

UI Theme:	Hand Drawn	Wire Frame	Windows XP	MAC OS X	Window 7
Legend					

Stepper allows you to specify a set of values for user to choose. Double clicking the Stepper element can change the candidate values.



Stepper is like a list that only show the current selected row, and its values can be strings or numbers.

Element Specific Facilities

1. Font of the text
2. Font size
3. Text color
4. Step value up/down
5. 6. 7. Change state for Stepper

Element Events

Element Clicked, Element Double-Clicked, Element Right-Clicked, Element Initialized, Element Hidden, Mouse Over, Mouse Out, Mouse Move, Mouse Down, Mouse Up, Key Down, Key Up, Custom Event, Value Changed, Focus Gain, Focus Lost

Element Actions

Change Visibility, Change Location, Change Size, Change State, Set Focus, Set Spinner Value

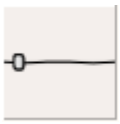
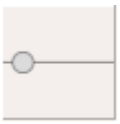
Element Properties

Id, X Coordinate, Y Coordinate, Width, Height, Visible, Note

Current Value of Stepper

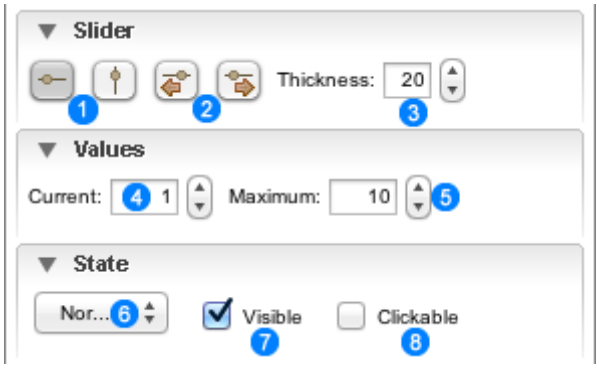
Slider

Slider belongs to the "Widgets" category and it has different look in various UI themes.

UI Theme:	Hand Drawn	Wire Frame	Windows XP	MAC OS X	Window 7
Legend					

You can assign the maximum and current value of the slider in tools panel.

Element Specific Facilities



1. Slider direction (horizontal or vertical)
2. Slide backward/forward
3. Slider thickness (in pixels)
4. Set slider current value
5. Set slider maximum value
6. 7. 8. Change state of slider

Element Events

Element Clicked, Element Double-Clicked, Element Right-Clicked, Element Initialized, Element Hidden, Mouse Over, Mouse Out, Mouse Move, Mouse Down, Mouse Up, Key Down, Key Up, Custom Event, Value Changed

Element Actions

Change Visibility, Change Location, Change Size, Change State, Set Slider Value

Element Properties

Id, X Coordinate, Y Coordinate, Width, Height, Visible, Note

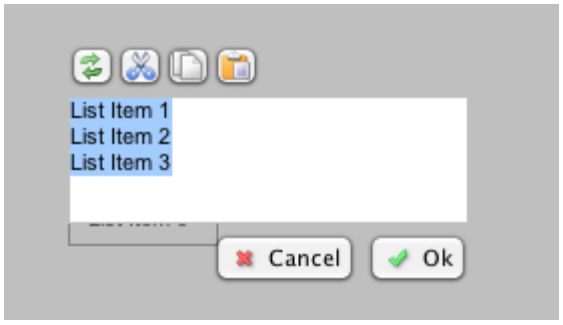
Current Value of Slider, Max value of Slider

List

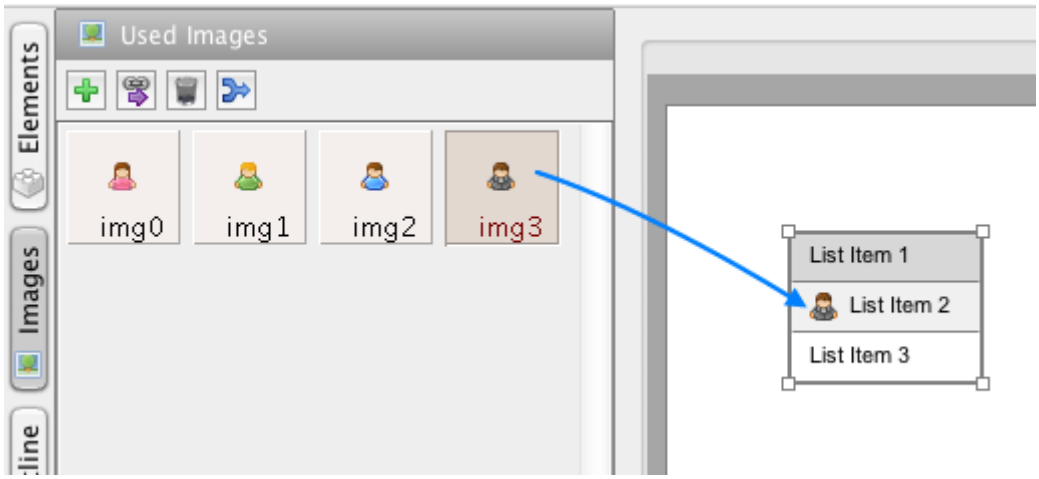
List belongs to the "Widgets" category and it has different look in various UI themes.

UI Theme:	Hand Drawn	Wire Frame	Windows XP	MAC OS X	Window 7
Legend					

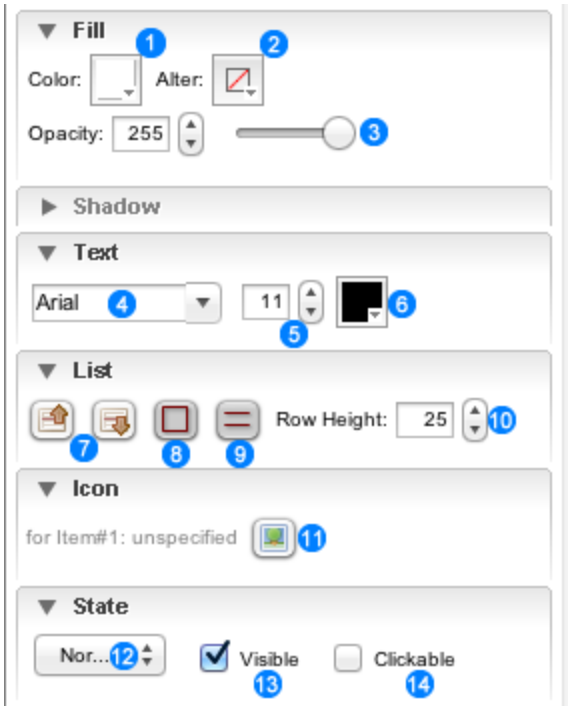
Double clicking the List element can change the content for each item.



List can accept image from Images Panel as its icon at each item. You can change its icon at any item from the Tools Panel, or directly drag the image into the list item.



Element Specific Facilities



1. Background color
2. Alternate row background color
3. Set fill opacity with input number/slider
4. Font of the text
5. Text size (in pixels)
6. Text color
7. Select previous/next item
8. Show / hide the border of the list
9. Show / hide the horizontal lines (this option has no effect on high-fidelity UI themes)
10. Set row height
11. Set icon for item
12. 15. Change state for the list

Element Events

Element Clicked, Element Double-Clicked, Element Right-Clicked, Element Initialized, Element Hidden, Mouse Over, Mouse Out, Mouse Move, Mouse Down, Mouse Up, Key Down, Key Up, Custom Event, Selection Changed

Element Actions

Change Visibility, Change Location, Change Size, Change State, Set Selected Index

Element Properties

Id, X Coordinate, Y Coordinate, Width, Height, Visible, Note

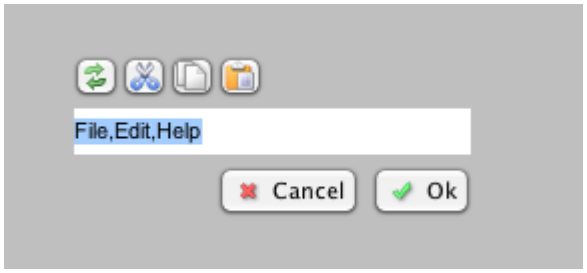
Index of the Selected Item, Text of the Selected Item

Menu Bar

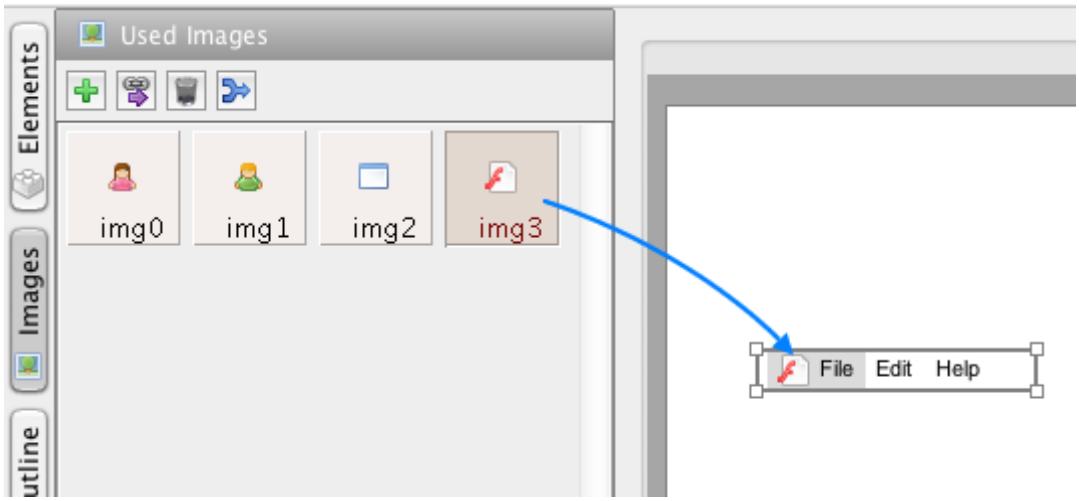
Menu Bar belongs to the "Widgets" category and it has different look in various UI themes.

UI Theme:	Hand Drawn	Wire Frame	Windows XP	MAC OS X	Window 7
Legend					

Double clicking the Menu Bar element can change the title for each menu.



Menu Bar can accept image from Images Panel as its icon at each menu. You can change its icon from the Tools Panel, or directly drag the image into the menu.



Element Specific Facilities

1. Font of the text
2. Text size (in pixels)
3. Text color
4. Select previous/next menu
5. Set icon for the selected menu or set a same icon for all menus of the bar when selected none of it
6. 7. 8. Change state for the menu bar

Element Events

Element Clicked, Element Double-Clicked, Element Right-Clicked, Element Initialized, Element Hidden, Mouse Over, Mouse Out, Mouse Move, Mouse Down, Mouse Up, Key Down, Key Up, Custom Event, Selection Changed

Element Actions

Change Visibility, Change Location, Change Size, Change State

Element Properties

Id, X Coordinate, Y Coordinate, Width, Height, Visible, Note

Index of the Selected Item

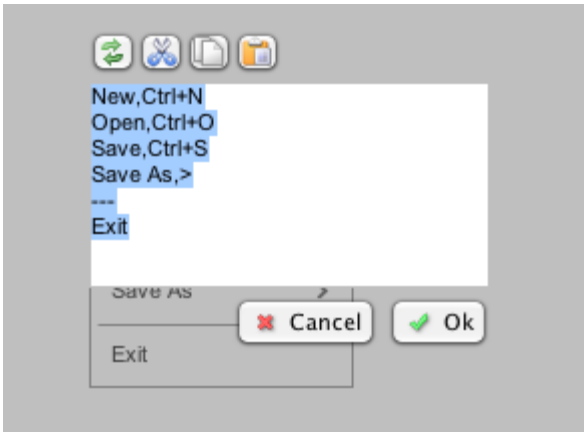
Menu

Menu belongs to the "Widgets" category and it has different look in various UI themes.

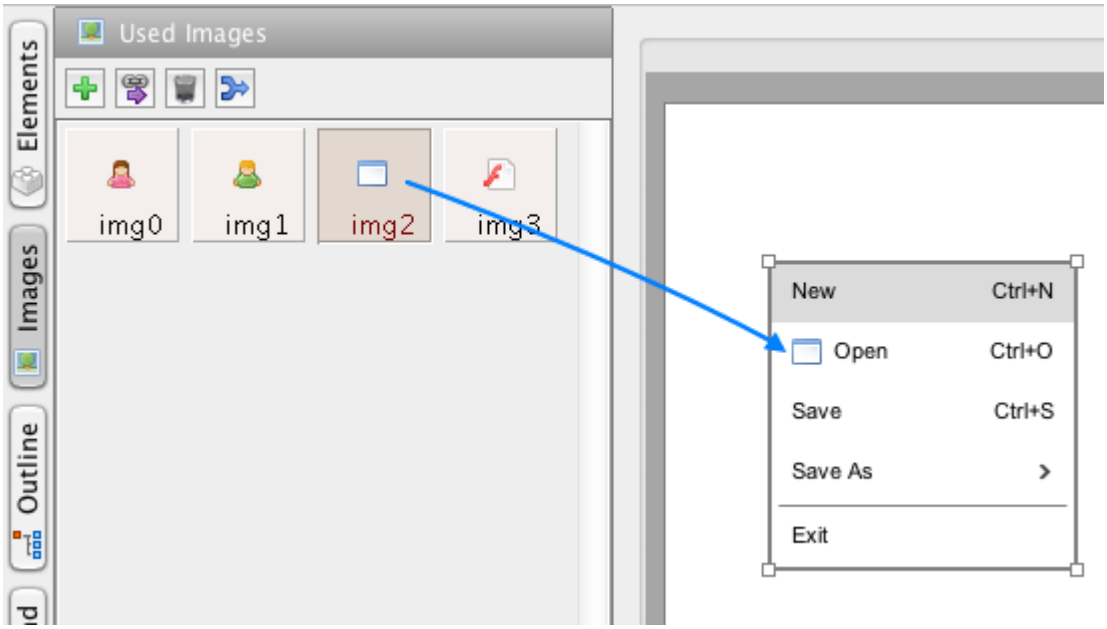
UI Theme:	Hand Drawn	Wire Frame	Windows XP	MAC OS X	Window 7
Legend					

Double clicking the Menu element can change the text for each menu item. Each row is for a menu item. You can add the hot key tips after the menu item text, use a comma to separate the item text and hot key

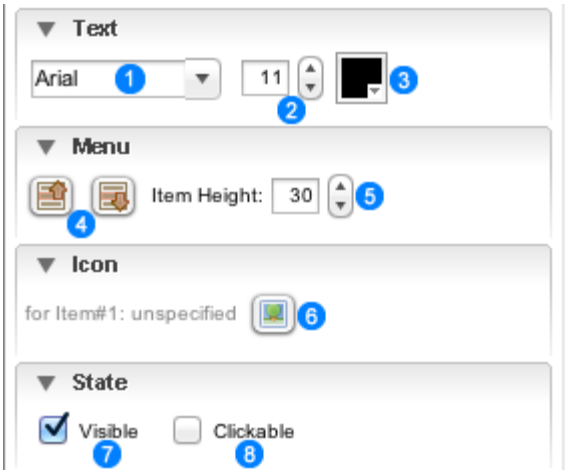
tips. If you need a separator within the menu, use 3 or more minus signs or equal signs in a row. If you like to make an item to be expansible, just append ",>" after the item text.



Menu can accept image from Images Panel as its icon at each menu item. You can change its icon from the Tools Panel, or directly drag the image into the menu item.



Element Specific Facilities



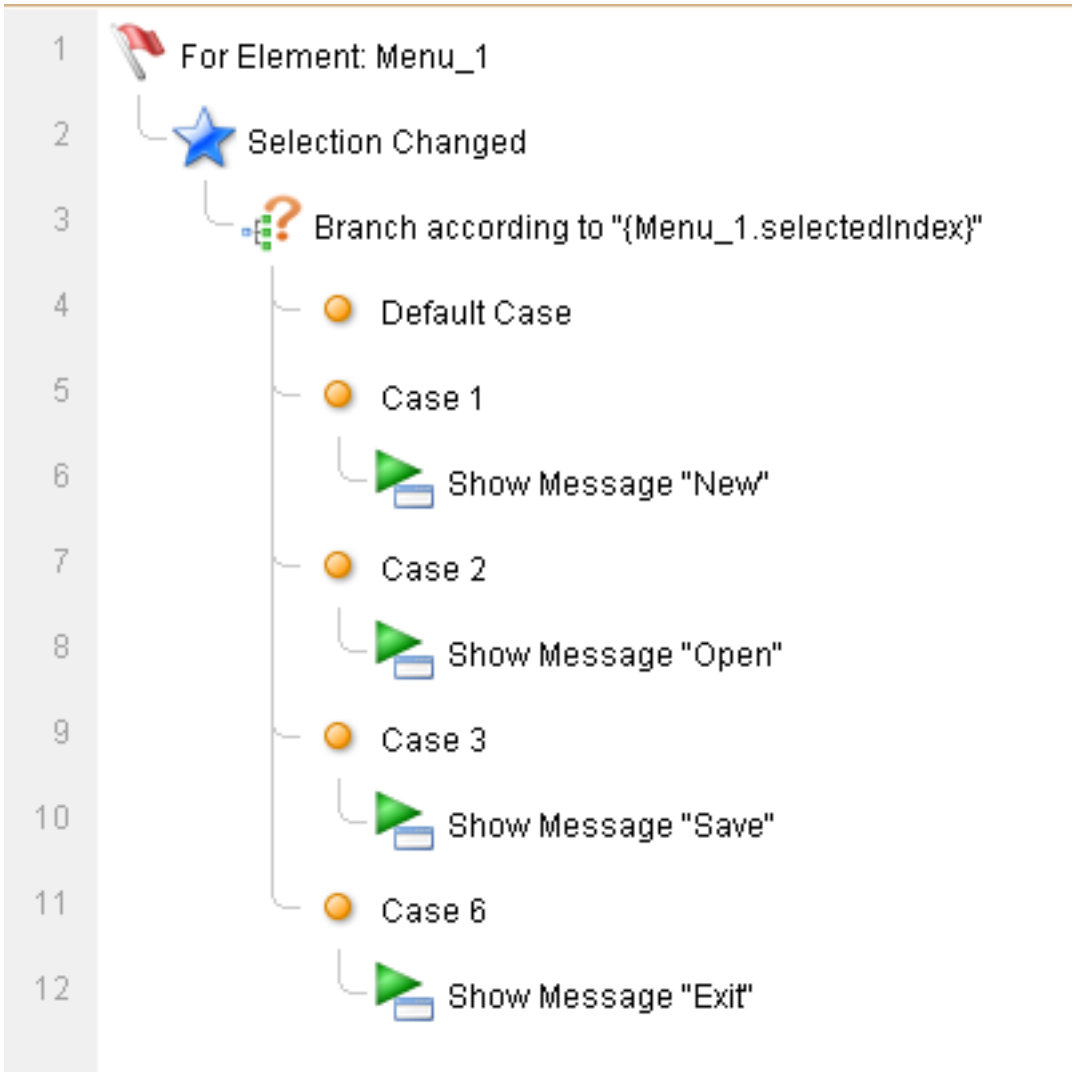
1. Font of the text

2. Text size (in pixels)
3. Text color
4. Select previous/next menu item
5. Set menu item height
6. Set icon for the selected menu item
7. 8. Set the state of menu

Behavior Definition

When you select any item in the menu, a Selection Changed event will be triggered. So handling this event is the correct way to define the behavior for menu element.

Use the facilities in behavior editor to define the behavior for the menu like this:



The selectedIndex property of the Menu element will be an integer that identify which item is selected.

Element Events

Element Clicked, Element Double-Clicked, Element Right-Clicked, Element Initialized, Element Hidden, Mouse Over, Mouse Out, Mouse Move, Mouse Down, Mouse Up, Key Down, Key Up, Custom Event, Selection Changed

Element Actions

Change Visibility, Change Location, Change Size

Element Properties

Id, X Coordinate, Y Coordinate, Width, Height, Visible, Note

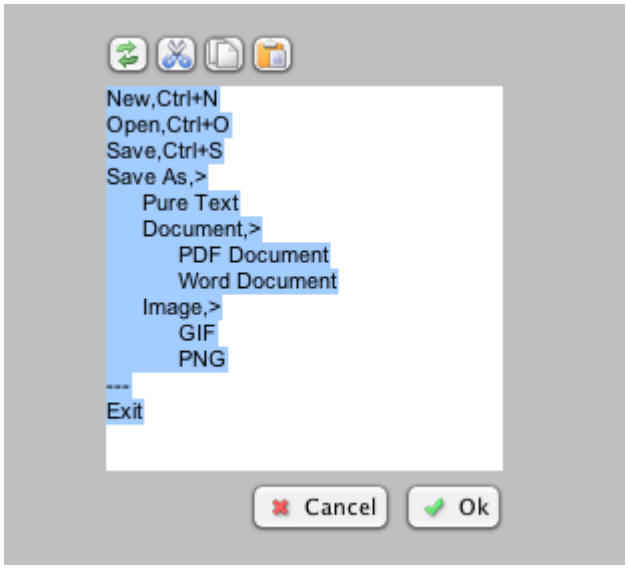
Index of the Selected Item

Multilevel Menu

Multilevel Menu belongs to the "Widgets" category and it has different look in various UI themes.

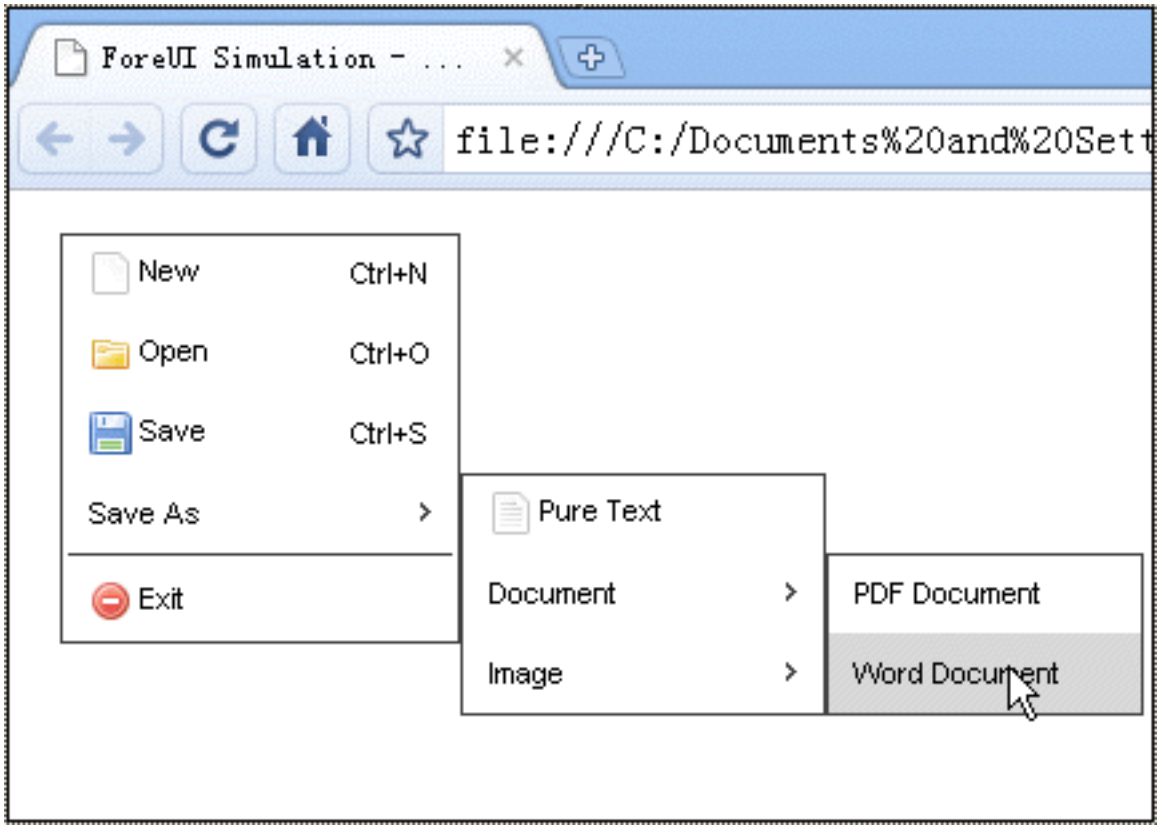
UI Theme:	Hand Drawn	Wire Frame	Windows XP	MAC OS X	Window 7
Legend					

Double clicking the Multilevel Menu element can change the text for each menu item.

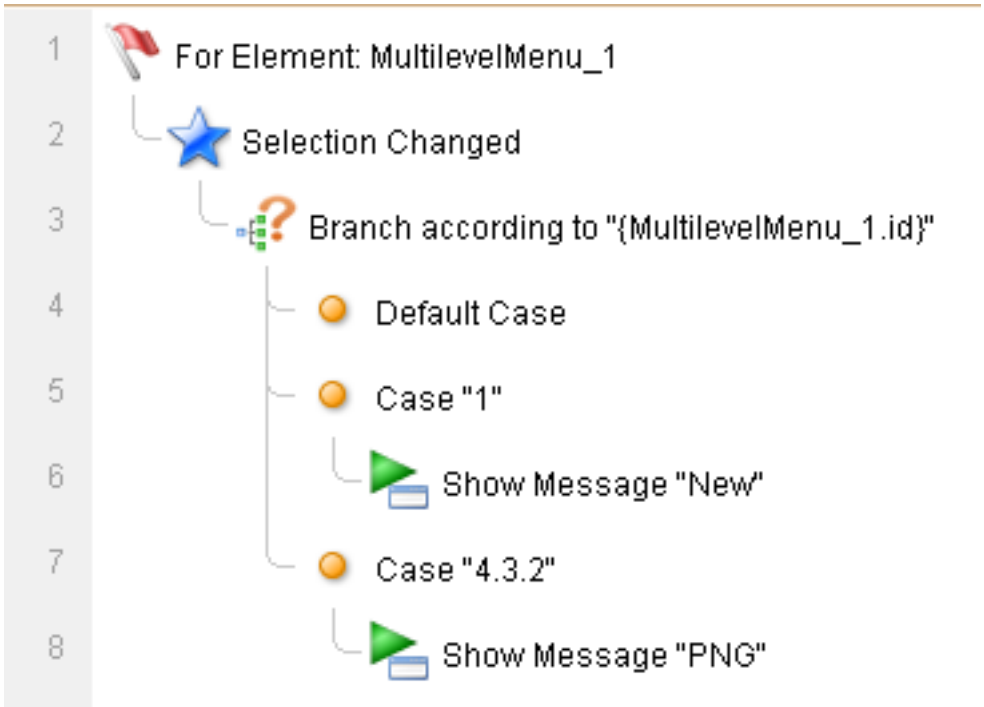


Similar with the Menu element, each row is for a menu item. You can append a comma plus hot key tips after the item text, or use 3 minus or equal signs to get a separator, or append ",>" after the item text to mark the item as expansible. The difference is that you can use **Tab** here to make a multilevel structure of menus.

If you run the HTML5 simulation, the Multilevel Menu element will be converted to a real, functional multilevel menu.



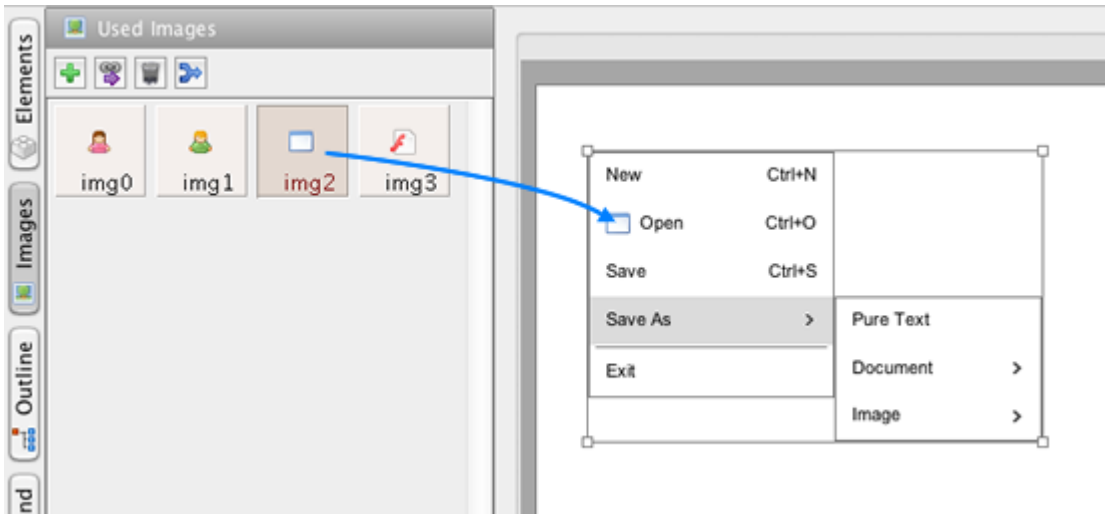
When you choose an item from the multilevel menu, the Selection Changed event will be triggered, you can handle this event can check its selection identification, which will be a string like “a.b.c”. Here is an example:



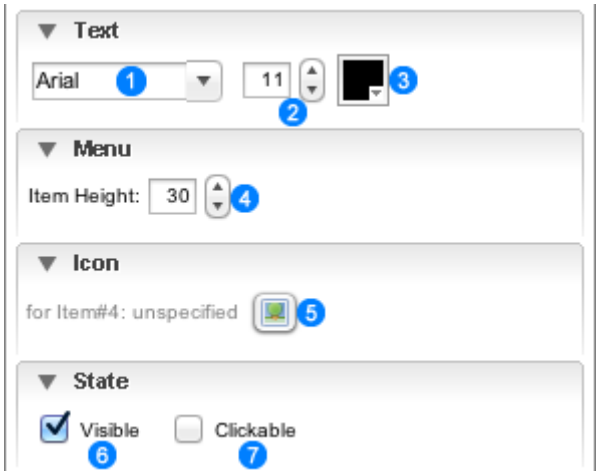
In the example above, the case “1″ means user selected the first item in the level-1 menu; Case “4.3.2″ means user click the fourth item in the level-1 menu, then click the third item in a level-2 menu, and finally select the second item in the level-3 menu.

Remarks: The identification is not a number but a string, so you see the quote marks for each case.

Multilevel Menu can accept image from Images Panel as its icon at each menu item. You can change its icon from the Tools Panel, or directly drag the image into the menu item.



Element Specific Facilities



1. Font of the text
2. Text size (in pixels)
3. Text color
4. Set item height (in pixel)
5. Set icon for the selected item
6. 7. Set multilevel menu state

Element Events

Element Clicked, Element Double-Clicked, Element Right-Clicked, Element Initialized, Element Hidden, Mouse Over, Mouse Out, Mouse Move, Mouse Down, Mouse Up, Key Down, Key Up, Custom Event, Selection Changed

Element Actions

Change Visibility, Change Location, Change Size

Element Properties

Id, X Coordinate, Y Coordinate, Width, Height, Visible, Note

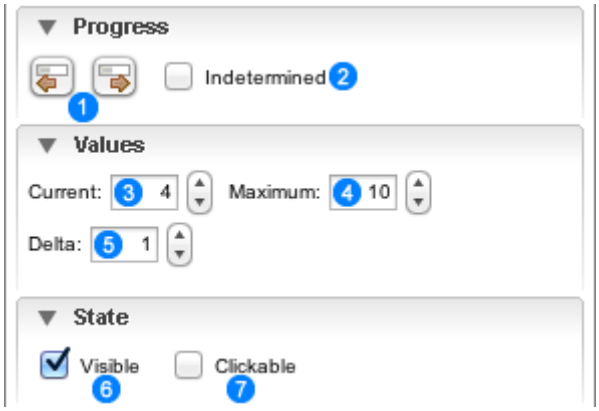
The Identification of the Selection

Progress Bar

Progress Bar belongs to the "Widgets" category and it has different look in various UI themes.

UI Theme:	Hand Drawn	Wire Frame	Windows XP	MAC OS X	Window 7
Legend					

Element Specific Facilities



1. Decrease/increase current progress
2. Undetermined mode
3. Set current progress value
4. Set maximum progress value
5. Set decrease/increase step value
6. 7. Set progress bar state

Element Events

Element Clicked, Element Double-Clicked, Element Right-Clicked, Element Initialized, Element Hidden, Mouse Over, Mouse Out, Mouse Move, Mouse Down, Mouse Up, Key Down, Key Up, Custom Event, Value Changed

Element Actions

Change Visibility, Change Location, Change Size, Set Progress Value

Element Properties

Id, X Coordinate, Y Coordinate, Width, Height, Visible, Note

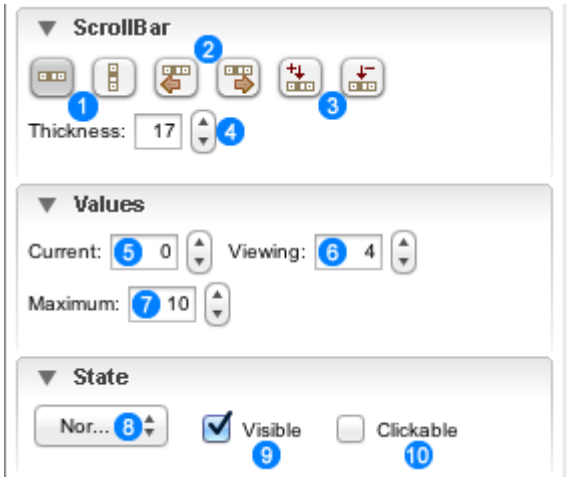
Current Progress Value, Max Progress Value

Scroll Bar

Scroll Bar belongs to the "Widgets" category and it has different look in various UI themes.

UI Theme:	Hand Drawn	Wire Frame	Windows XP	MAC OS X	Window 7
Legend					

Element Specific Facilities



1. Scroll bar direction (horizontal or vertical)
2. Scroll backward/forward
3. Make it more/less scrollable
4. Scroll bar thickness (in pixels)
5. Set scroll location
6. Set viewing range
7. Set maximum range
8. 9. 10. Change scroll bar's state

Element Events

Element Clicked, Element Double-Clicked, Element Right-Clicked, Element Initialized, Element Hidden, Mouse Over, Mouse Out, Mouse Move, Mouse Down, Mouse Up, Key Down, Key Up, Custom Event, Value Changed

Element Actions

Change Visibility, Change Location, Change Size, Change State

Element Properties

Id, X Coordinate, Y Coordinate, Width, Height, Visible, Note

Current Value of Scroll Bar, Max Value of Scroll Bar, View Size of Scroll Bar

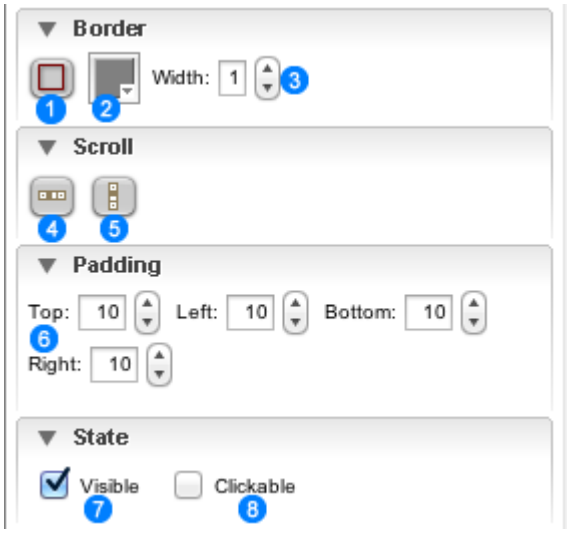
Scrollable Container

Scrollable Container belongs to the "Widgets" category and it has different look in various UI themes.

UI Theme:	Hand Drawn	Wire Frame	Windows XP	MAC OS X	Window 7
Legend					

Scrollable Container is a container element that can place other elements into its scrolling view. You content will really become scrollable when you launch the HTML5 simulation.

Element Specific Facilities



1. Show/hide border
2. Border color
3. Border width (in pixels)
4. Show/hide horizontal scroll bar
5. Show/hide vertical scroll bar
6. Set paddings of scrolling view
7. 8. Set scrollable container state

Element Events

Element Clicked, Element Double-Clicked, Element Right-Clicked, Element Initialized, Element Hidden, Mouse Over, Mouse Out, Mouse Move, Mouse Down, Mouse Up, Key Down, Key Up, Custom Event

Element Actions

Change Visibility, Change Location, Change Size

Element Properties

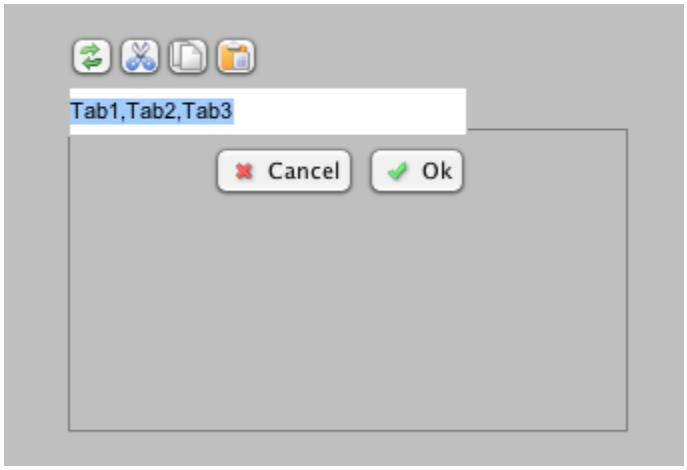
Id, X Coordinate, Y Coordinate, Width, Height, Visible, Note

Tabs

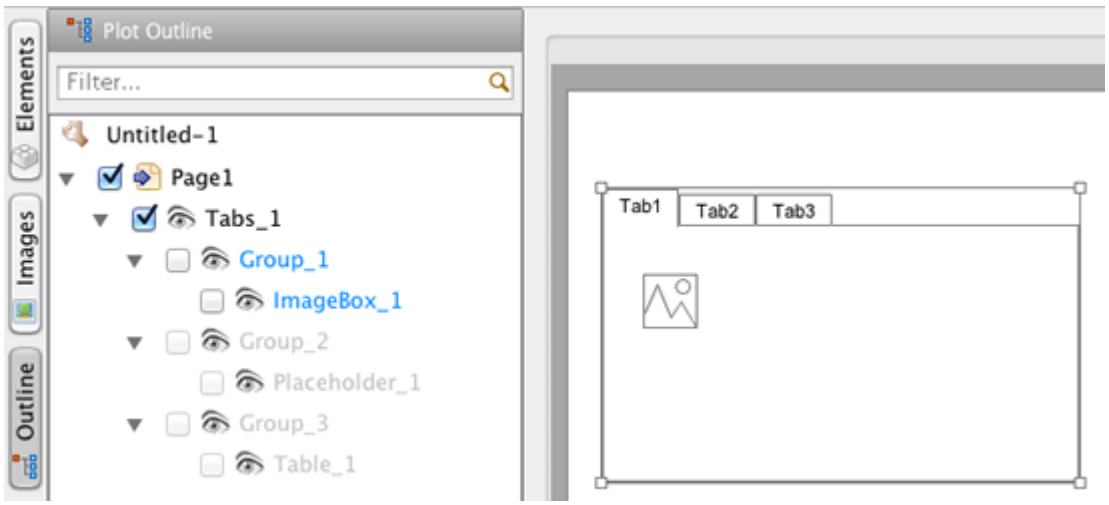
Tabs belongs to the "Widgets" category and it has different look in various UI themes.

UI Theme:	Hand Drawn	Wire Frame	Windows XP	MAC OS X	Window 7
Legend					

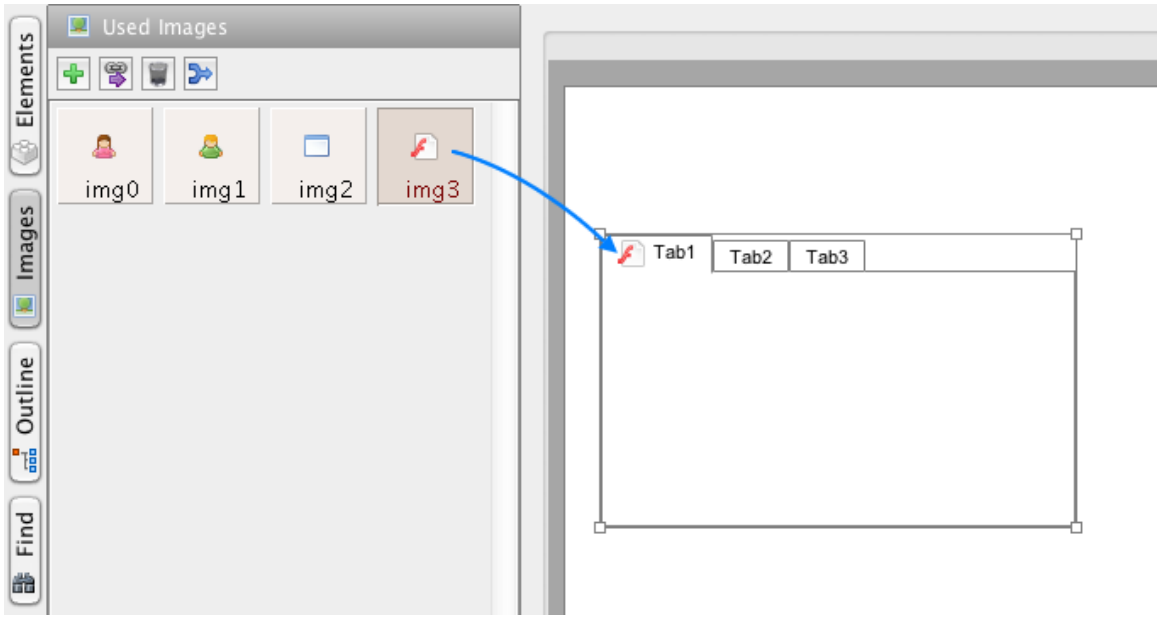
Double clicking the Tabs element can change the titles of its tabs. The titles are separated with comma, if you need to use comma in tab title, use '\' to escape it.



Tabs is a container element, and each tab can serve as a container. The elements that embedded in inactive tabs will be hidden (show in gray in the Outline View).



Tabs can accept image from Images Panel as the icon for each tab. You can change its icon from the Tools Panel, or directly drag the image into the tab.



Element Specific Facilities

1. Font of the tab titles
2. Font size
3. Text color
4. Select previous/next tab
5. Show tabs on top/bottom
6. Set tab height (in pixels)
7. Set icon for selected tab

8. Set container padding (in pixels)

9. 10. 11. Change state for tabs

Element Events

Element Clicked, Element Double-Clicked, Element Right-Clicked, Element Initialized, Element Hidden, Mouse Over, Mouse Out, Mouse Move, Mouse Down, Mouse Up, Key Down, Key Up, Custom Event, Selection Changed

Element Actions

Change Visibility, Change Location, Change Size, Change State, Set Selected Index

Element Properties

Id, X Coordinate, Y Coordinate, Width, Height, Visible, Note

Index of the Selected Tab

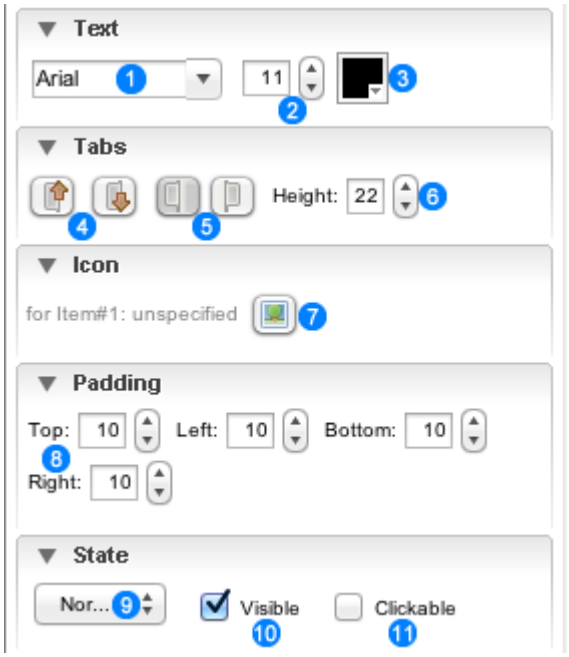
Vertical Tabs

Vertical Tabs belongs to the "Widgets" category and it has different look in various UI themes.

UI Theme:	Hand Drawn	Wire Frame	Windows XP	MAC OS X	Window 7
Legend					

Vertical Tabs is quite similar with the Tabs element, but its tabs are placed on left or right side.

Element Specific Facilities



1. Font of the tab titles

2. Font size
3. Text color
4. Select upper/lower tab
5. Show tabs on left/right
6. Set tab height (in pixels)
7. Set icon for selected tab
8. Set container padding (in pixels)
9. 10. 11. Change state for vertical tabs

Element Events

Element Clicked, Element Double-Clicked, Element Right-Clicked, Element Initialized, Element Hidden, Mouse Over, Mouse Out, Mouse Move, Mouse Down, Mouse Up, Key Down, Key Up, Custom Event, Selection Changed

Element Actions

Change Visibility, Change Location, Change Size, Change State, Set Selected Index

Element Properties

Id, X Coordinate, Y Coordinate, Width, Height, Visible, Note

Index of the Selected Tab

Table

Table belongs to the "Widgets" category and it has different look in various UI themes.

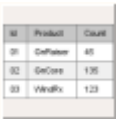
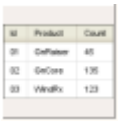
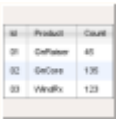
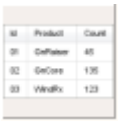
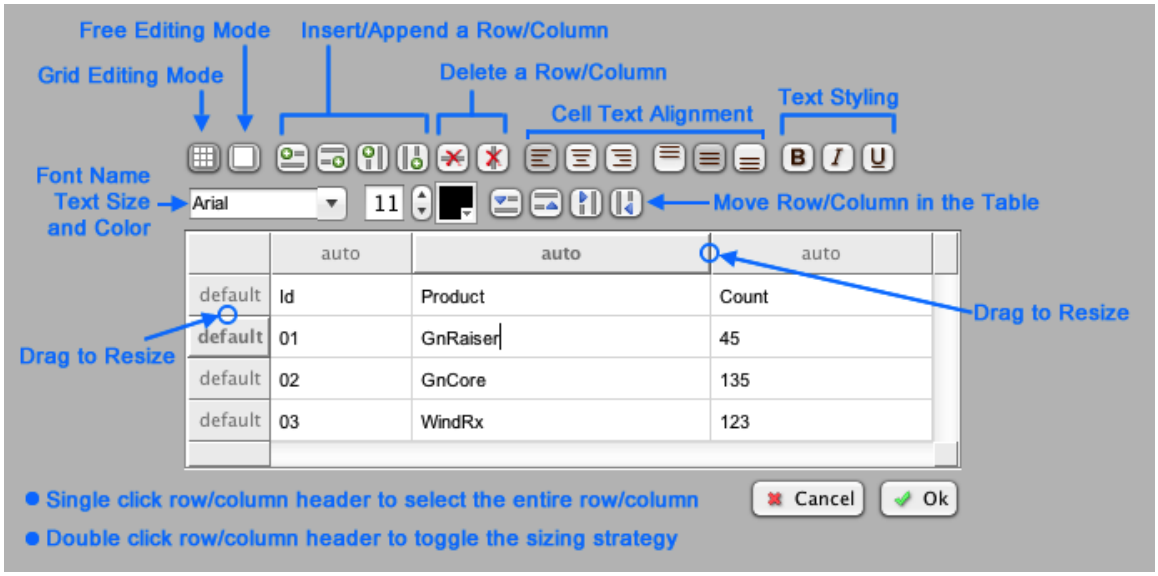
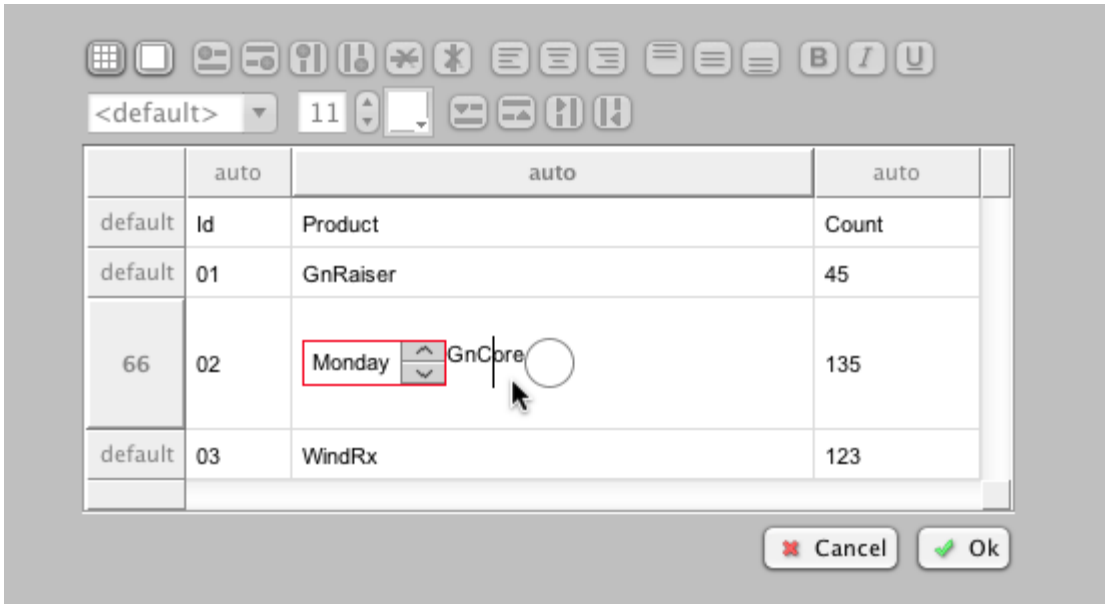
UI Theme:	Hand Drawn	Wire Frame	Windows XP	MAC OS X	Window 7
Legend					

Table can store data with multiple rows and columns. To change the data of table, just double click the Table element. There are two editing modes, the default one is to edit data grid by grid. In this editing mode, you can insert/append/remove row or column, specify the text alignment for certain table cell, or adjust the row height or column width by dragging the header border.



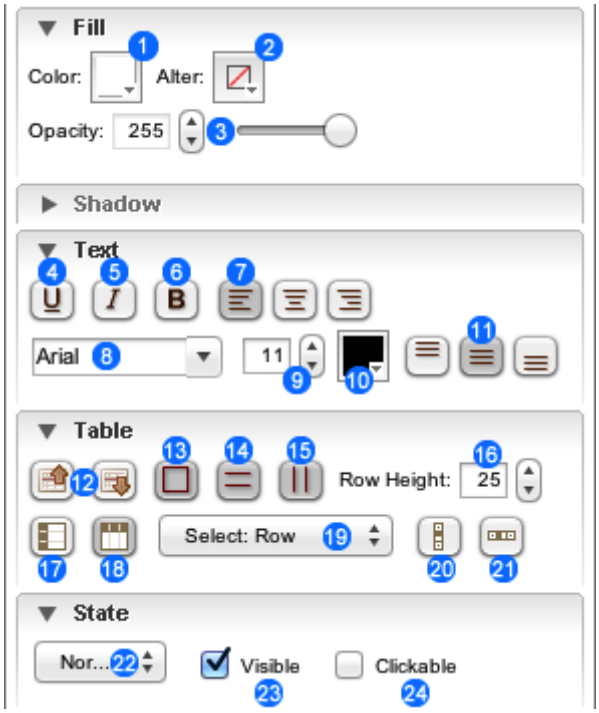
If you click the  button on left top corner, you will enter the free editing mode, which allows you to edit the data for the entire table. In this mode, each row in the editing area is for one row in the table. Data for different columns are separated with comma. If you need to use comma in table cell content, use '\' to escape it. You can click the  button to switch back to the grid editing mode.

Table is a container element, and each cell of Table can serve as a container. You can embed elements into any cell of the table. The embedded element will be place at the left inside the cell, and you can drag to relocate it within the table cell.



An (invisible) Table element can be used as data repository. You can use properties to get data, and use actions to set/insert/append/delete data.

Element Specific Facilities



1. Background color
2. Alternate row background color
3. Set fill opacity with number/slider
4. Underline text
5. Make text italic
6. Use bold text

7. Default text horizontal align (can be overrode by table cell's alignment)
8. Font of the text
9. Font size
10. Text color
11. Default text vertical align (can be overrode by table cell's alignment)
12. Select previous/next row
13. Table outer frame show / hide
14. Table horizontal lines show / hide
15. Table vertical lines show / hide
16. Default row height (in pixels, can be overrode by table cell's height)
17. Row header show / hide
18. Column header show / hide
19. Change select mode
20. Show / hide vertical scroll bar
21. Show / hide horizontal scroll bar
22. 23.24. Change state for the table

Element Events

Element Clicked, Element Double-Clicked, Element Right-Clicked, Element Initialized, Element Hidden, Mouse Over, Mouse Out, Mouse Move, Mouse Down, Mouse Up, Key Down, Key Up, Custom Event, Selection Changed

Element Actions

Change Visibility, Change Location, Change Size, Change State, Set Selected Index, Set Selected Column, Set Table Cell Value, Append New Row to Table, Insert New Row to Table, Delete Row from Table

Element Properties

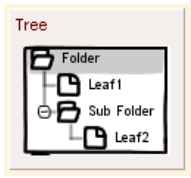
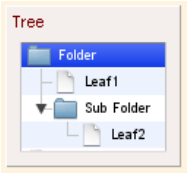
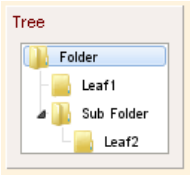
Id, X Coordinate, Y Coordinate, Width, Height, Visible, Note

Index of the Selected Table Row, Index of the Selected Table Column, Table Cell Values as Array[row][column], Table Row Count(Header excluded), Table Column Count

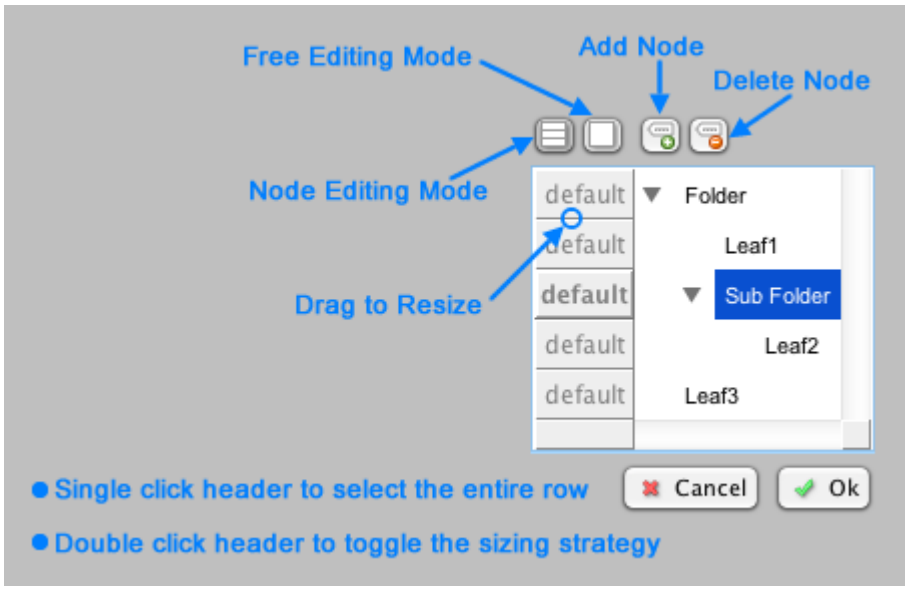
Tree

Tree belongs to the "Widgets" category and it has different look in various UI themes.

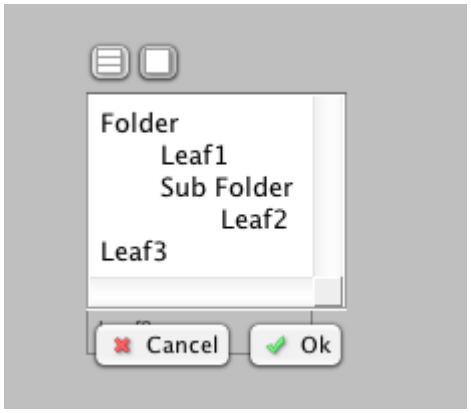
UI Theme:	Hand Drawn	Wire Frame	Windows XP	MAC OS X	Window 7

Legend					

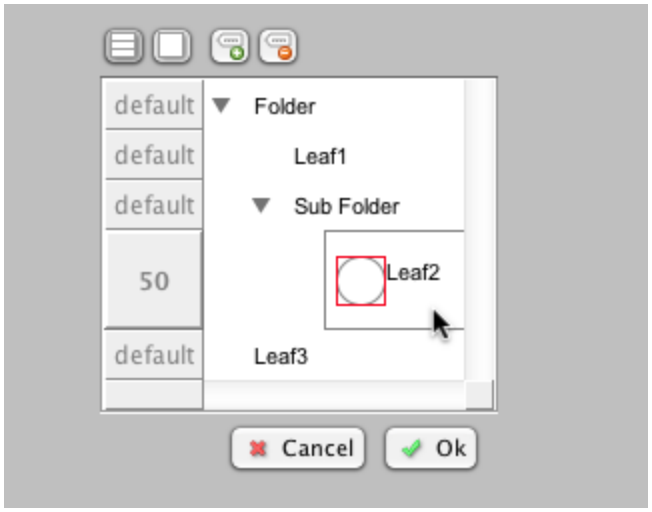
Tree can store data with hierarchy. To change the data in tree, just double click the Tree element. There are two editing modes, the default one is to edit data node by node. In this editing mode, you can add/remove node, change node height or expand/collapse certain nodes.



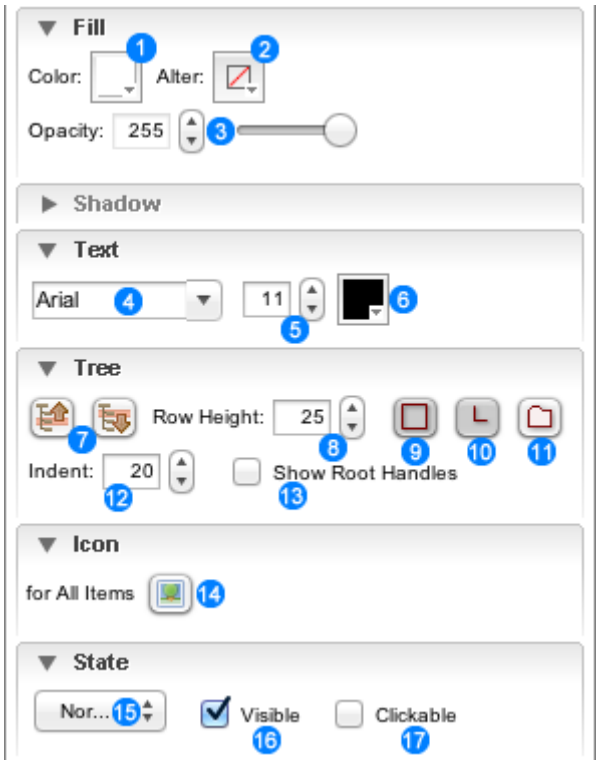
If you click the  button on left top corner, you will enter the second editing mode, which allows you to edit the data for the entire tree. In this mode, each row in the editing area is for one node in the tree. You can click the  button to switch back to the node editing mode.



Tree is a container element, and each node of Tree can serve as a container. You can embed elements into any node of the tree. The embedded element will be place at the left inside the node, and you can drag to relocate it within the tree node.



Element Specific Facilities



1. Background color
2. Alternate row background color
3. Set fill opacity with number/slider
4. Font of the text
5. Font size
6. Text color
7. Select previous/next note
8. Set note height (in pixels)
9. Show/hide tree border

10. Show/hide indent lines
11. Show/hide tree default icons
12. Set indent for each level (in pixels)
13. Whether to paint lines and small collapse/expand buttons for root nodes
14. Change icon for selected node
15. 16. 17. Change state for the tree

Element Events

Element Clicked, Element Double-Clicked, Element Right-Clicked, Element Initialized, Element Hidden, Mouse Over, Mouse Out, Mouse Move, Mouse Down, Mouse Up, Key Down, Key Up, Custom Event, Selection Changed,

Element Actions

Change Visibility, Change Location, Change Size, Change State, Set Selected Index, Set Tree Node Value

Element Properties

Id, X Coordinate, Y Coordinate, Width, Height, Visible, Note

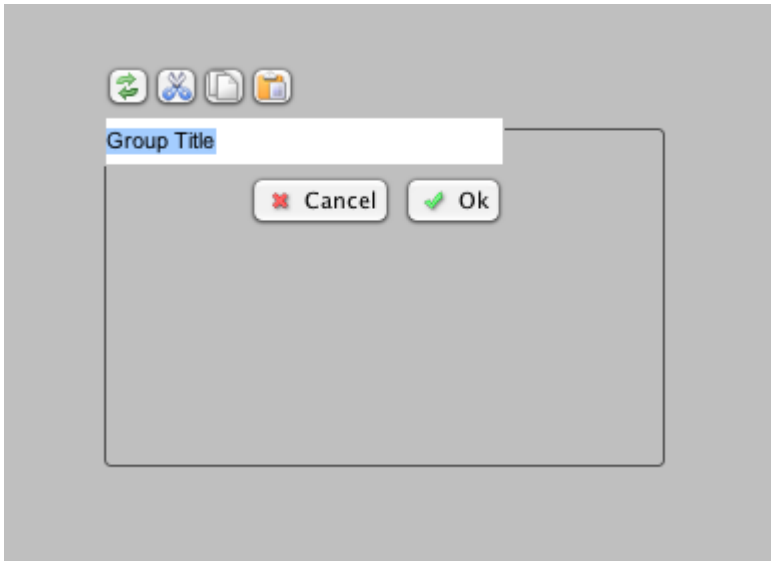
Index of the Selected Tree Item, Text of the Selected Item

Group Frame

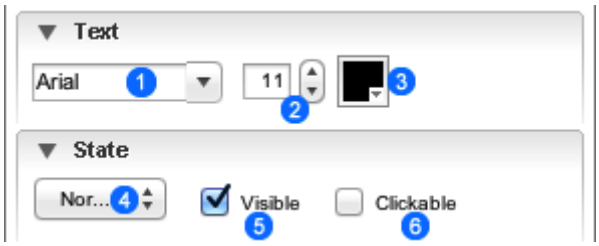
Group Frame belongs to the "Widgets" category and it has different look in various UI themes.

UI Theme:	Hand Drawn	Wire Frame	Windows XP	MAC OS X	Window 7
Legend					

Double clicking the Group Frame element can change its title text.



Element Specific Facilities



1. Font of the text
2. Text size
3. Text color
4. 5. 6. Change state of the group frame

Element Events

Element Clicked, Element Double-Clicked, Element Right-Clicked, Element Initialized, Element Hidden, Mouse Over, Mouse Out, Mouse Move, Mouse Down, Mouse Up, Key Down, Key Up, Custom Event

Element Actions

Change Visibility, Change Location, Change State, Change Size

Element Properties

Id, X Coordinate, Y Coordinate, Width, Height, Visible, Note

Window

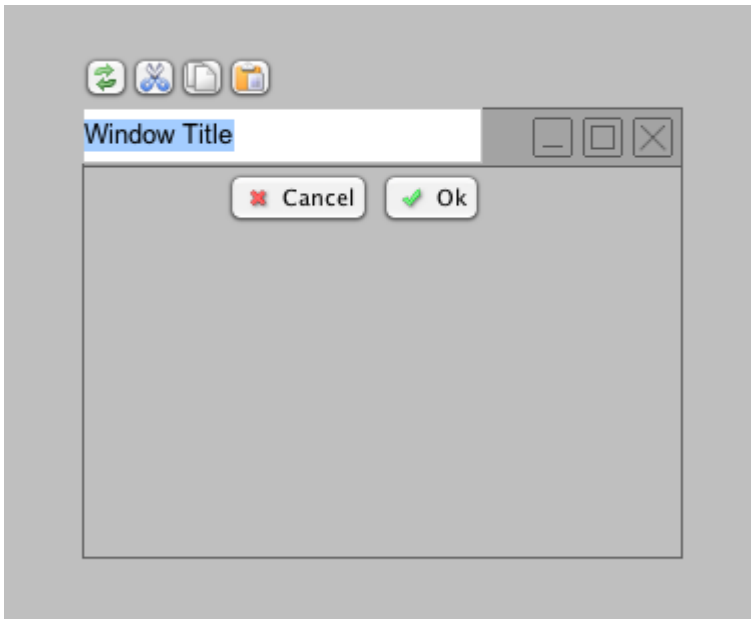
Window belongs to the "Widgets" category and it has different look in various UI themes.

UI Theme:	Hand Drawn	Wire Frame	Windows XP	MAC OS X	Window 7

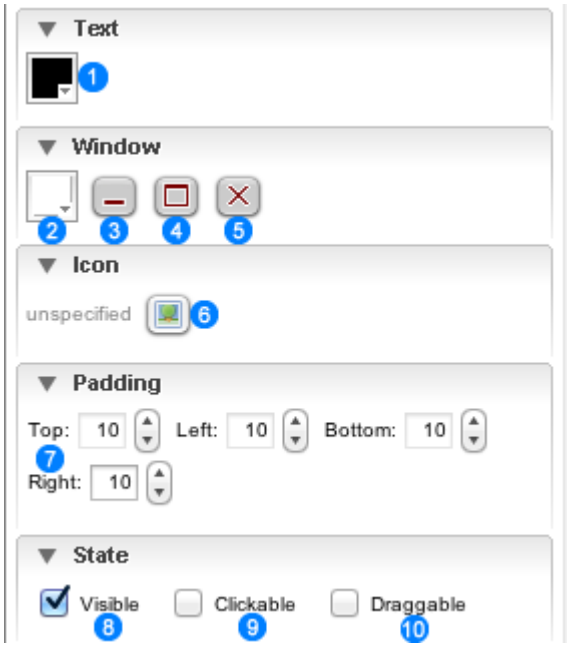
Legend					

Window element is a container element. If you turn the "Draggable" option on, the Window element can be moved by dragging its title bar in HTML5 simulation.

By double clicking the Window element, you can edit its title:



Element Specific Facilities



1. Text color
2. Client area color
3. Show / hide the minimize button

4. Show / hide the maximize button
5. Show / hide the close button
6. Set icon in the window title bar
7. Set container padding (in pixels)
8. Visible / invisible in simulation
9. If selected, the cursor will change to hand shape when hovering on the element
10. Allow / disallow dragging in simulation

Element Events

Element Clicked, Element Double-Clicked, Element Right-Clicked, Element Initialized, Mouse Over, Mouse Out, Mouse Move, Mouse Down, Mouse Up, Key Down, Key Up, Custom Event, Window Closed

Element Actions

Change Visibility, Change Location, Change Size, Set Window Title

Element Properties

Id, X Coordinate, Y Coordinate, Width, Height, Visible, Note

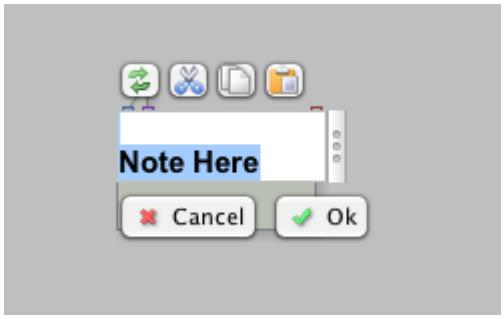
Balloon

Balloon belongs to the "Annotation" category, besides in the Hand Drawing and Wireframe themes, its appearance doesn't change for other UI themes.

UI Theme:	Hand Drawn	Wireframe	Other Themes
Legend			

Balloon element can be used to make comments on your plot.

By double clicking the Balloon element, you can edit its text inside:



Element Specific Facilities

1. Underline the text
2. Make text italic
3. Use bold font
4. Text alignment
5. Font of the text
6. Font size
7. Text color
8. Placement of the antenna (top/left/bottom/right)
9. 10. Change the state of balloon

Element Events

Element Clicked, Element Double-Clicked, Element Right-Clicked, Element Initialized, Element Hidden, Mouse Over, Mouse Out, Mouse Move, Mouse Down, Mouse Up, Key Down, Key Up, Custom Event

Element Actions

Change Visibility, Change Location, Change Size

Element Properties

Id, X Coordinate, Y Coordinate, Width, Height, Visible, Note

Cursor

Cursor belongs to the "Annotation" category and it has different look in various UI themes.

UI Theme:	Hand Drawn	Wire Frame	Windows XP	MAC OS X	Window 7
Legend					

Cursor element can simulate the mouse cursor on your plot. It is useful to simulate human's manipulation in your prototype.

Element Specific Facilities

1. Change the cursor style
2. 3. Change the state of cursor

Element Events

Element Clicked, Element Double-Clicked, Element Right-Clicked, Element Initialized, Element Hidden, Mouse Over, Mouse Out, Mouse Move, Mouse Down, Mouse Up, Key Down, Key Up, Custom Event

Element Actions

Change Visibility, Change Location, Change Size

Element Properties

Id, X Coordinate, Y Coordinate, Width, Height, Visible, Note

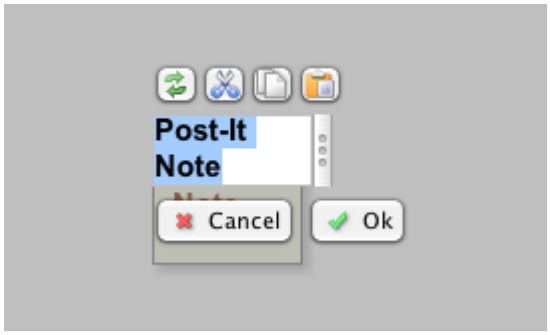
Post-It

Post-It belongs to the "Annotation" category, besides in the Hand Drawing and Wireframe themes, its appearance doesn't change for other UI themes.

UI Theme:	Hand Drawn	Wireframe	Other Themes
Legend			

Post-It element can be used to put some description on page or element.

By double clicking the Post-It element, you can edit its text inside:



Element Specific Facilities

1. Underline the text
2. Make text italic
3. Use bold font
4. Text alignment
5. Font of the text
6. Font size
7. Text color
8. 9. Change state of Post-It

Element Events

Element Clicked, Element Double-Clicked, Element Right-Clicked, Element Initialized, Element Hidden, Mouse Over, Mouse Out, Mouse Move, Mouse Down, Mouse Up, Key Down, Key Up, Custom Event

Element Actions

Change Visibility, Change Location, Change Size

Element Properties

Id, X Coordinate, Y Coordinate, Width, Height, Visible, Note

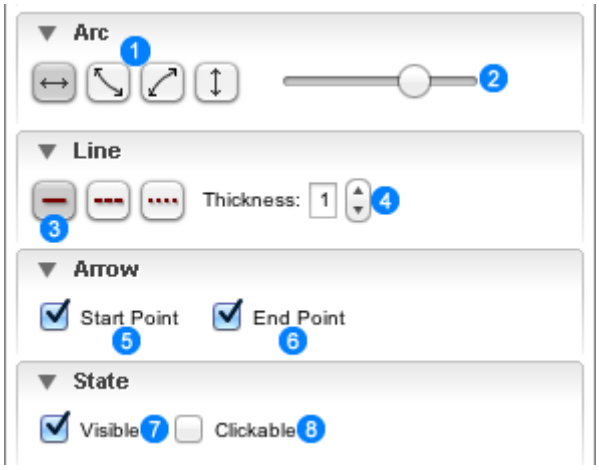
Arrow Line

Arrow Line belongs to the "Annotation" category, besides in the Hand Drawing theme, its appearance doesn't change for other UI themes.

UI Theme:	Hand Drawn	Other Themes
Legend		

Arrow Line can be used to present the relationship between elements.

Element Specific Facilities



1. Set arrow line direction
2. Set arc radian
3. Set line type
4. Set thickness (in pixels)
5. Show / hide the arrowhead on start point
6. Show / hide the arrowhead end point
7. 8. Change the state of Arrow Line

Remarks: the "dashed" and "dotted" line types are not supported in simulation, and they will be rendered as solid line.

Element Events

Element Clicked, Element Double-Clicked, Element Right-Clicked, Element Initialized, Element Hidden, Mouse Over, Mouse Out, Mouse Move, Mouse Down, Mouse Up, Key Down, Key Up, Custom Event

Element Actions

Change Visibility, Change Location, Change Size

Element Properties

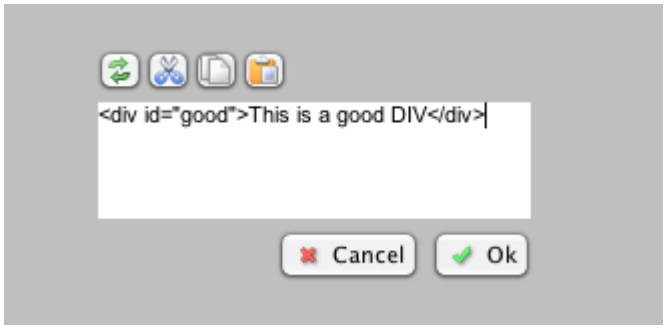
Id, X Coordinate, Y Coordinate, Width, Height, Visible, Note

Html

Html is a basic element that can allows you to embed raw HTML code into the simulation.

You can find the "Html" element in the elements pane, and add it into your page. It looks like:

Different than other elements, Html element is not visible in HTML5 simulation and exported image/PDF, instead it inserts its content (HTML code) into the HTML5 simulation. Here is an example of Html content:



When you export your plot into HTML5 simulation, the DIV with "good" id will be included in the simulation as well. The actual location of the inserted code depends on the location of the Html element. If the Html element is embedded into a cell of a Table element, the inserted code will appear within the <td> tag in the table.

Remarks: Embedding inappropriate HTML code into your plot may get unexpected result in the HTML5 simulation. It is better not to do so, unless you have to, and you really know what you are doing.

Css

Css is a basic element that can allows you to embed raw CSS code into the simulation.

You can find the "Css" element in the elements pane, and add it into your page. It looks like:

Different than other elements, Css element is not visible in HTML5 simulation and exported image/PDF, instead it inserts its content (CSS code) into the HTML5 simulation. Here is an example of Css content:

When you export your plot into HTML5 simulation, the CSS block will be inserted into the simulation as well.

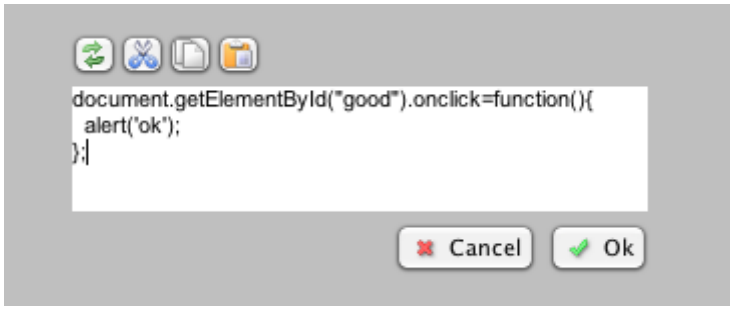
Remarks: Embedding inappropriate CSS code into your plot may get unexpected result in the HTML5 simulation. It is better not to do so, unless you have to, and you really know what you are doing.

Script

Script is a basic element that can allows you to embed raw Javascript code into the simulation.

You can find the "Script" element in the elements pane, and add it into your page. It looks like:

Different than other elements, Script element is not visible in HTML5 simulation and exported image/PDF, instead it inserts its content (Javascript code) into the HTML5 simulation. Here is an example of Script content:



When you export your plot into HTML5 simulation, the Javascript block will be inserted into the simulation as well.

Remarks: Embedding inappropriate Javascript code into your plot may get unexpected result in the HTML5 simulation. It is better not to do so, unless you have to, and you really know what you are doing.

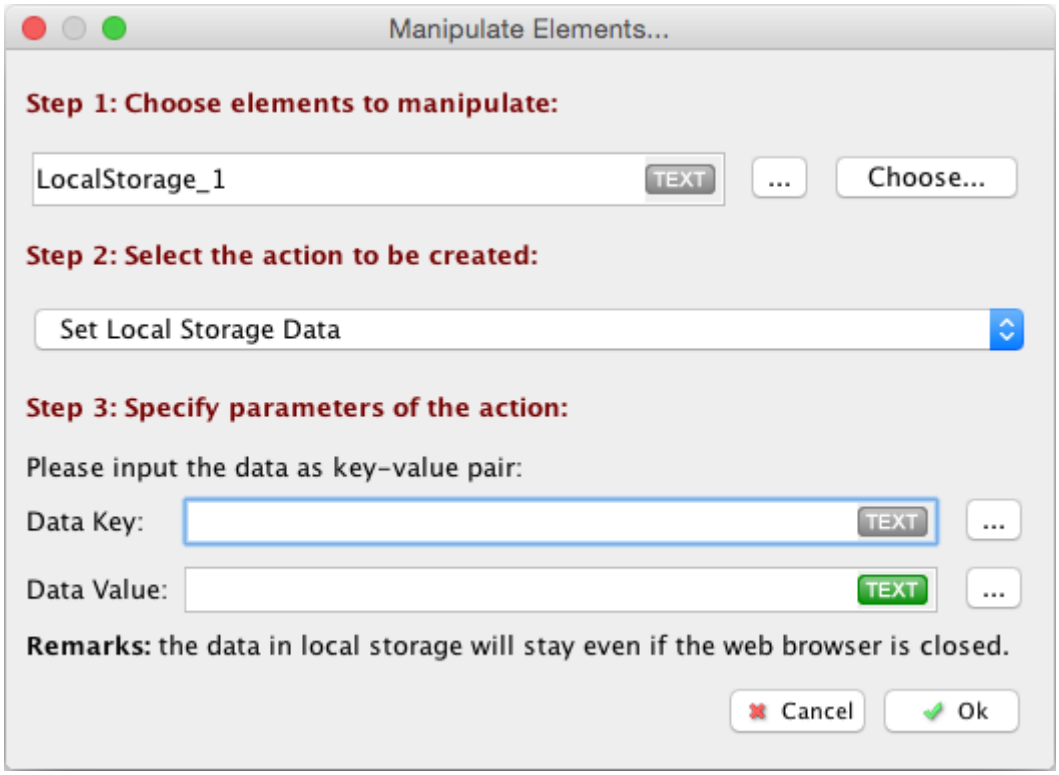
Local Storage

Local Storage is a basic element that can allows you to put value into, and get value from HTML5 local storage. You can find the "Local Storage" element in the elements pane, and add it into the editing area. It looks like:

Just like the Html, Css and Script elements, Local Storage element is not visible in HTML5 simulation and exported image/PDF. It is only visible in the editing mode.

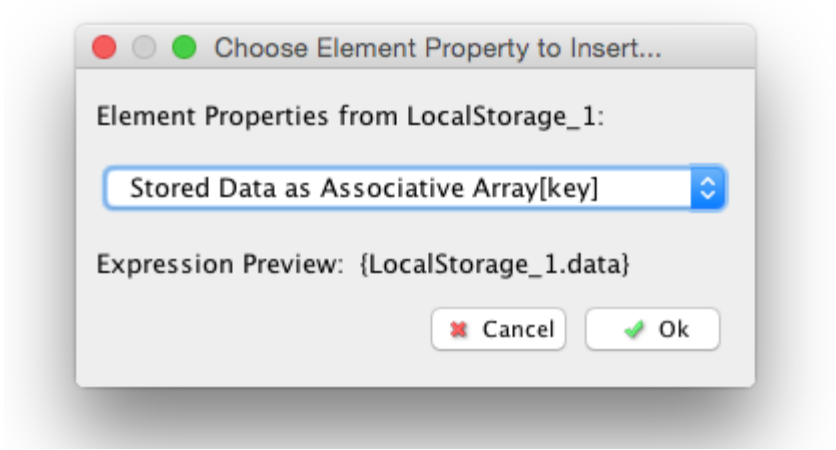
Although you can directly use the local storage in HTML5 (by using the Script element). This element can still bring you some benefits.

One advantage of using this Local Storage element is that it provides a namespace for each instance of this element. So you can use this element like a database table, and different instances of this element can store values with the same key. In the “Manipulate Elements…” window, you can choose the “Set Local Storage Data” action and then specify the data as a key-value pair.

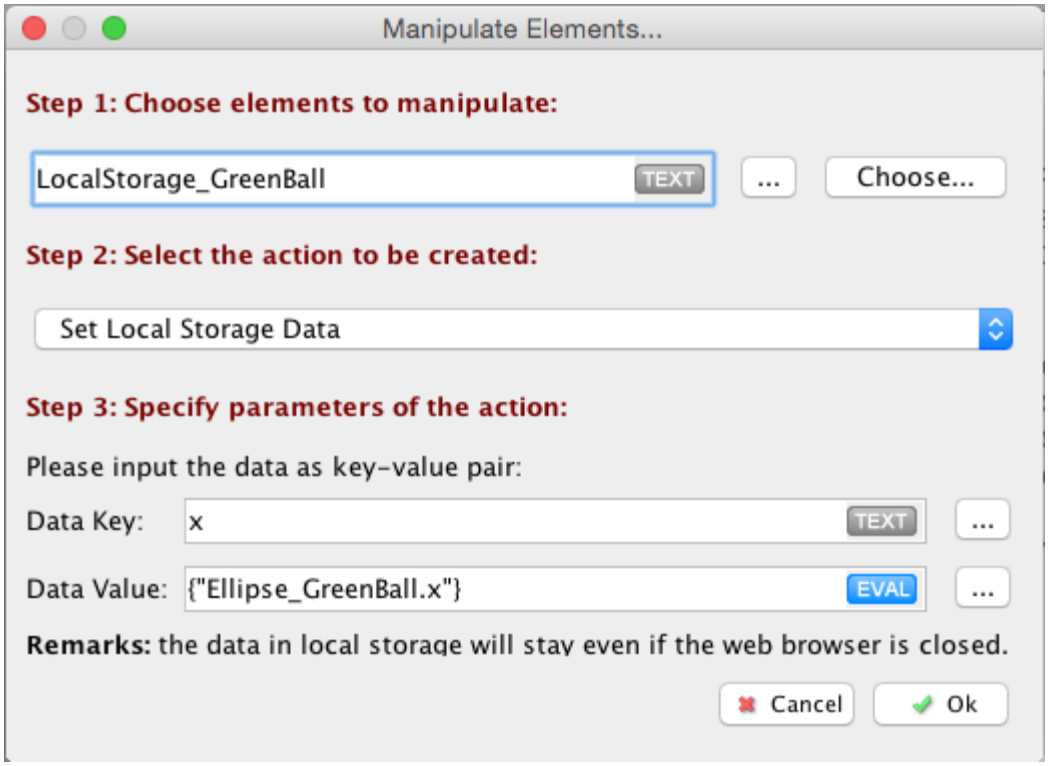


If you directly use the local storage in HTML5, it can only store data as text string. The local storage element in ForeUI however, can store any data type supported by ForeUI, which are number, string, array and object. You don't need to worry about how they get stored, ForeUI will take care of that for you.

You can retrieve the data as an element property (associative array type), and can insert it into any expression field.



For example, you store an integer value into local storage with key "x":



Then you can retrieve it like this:

The data type will still be integer (not changed).

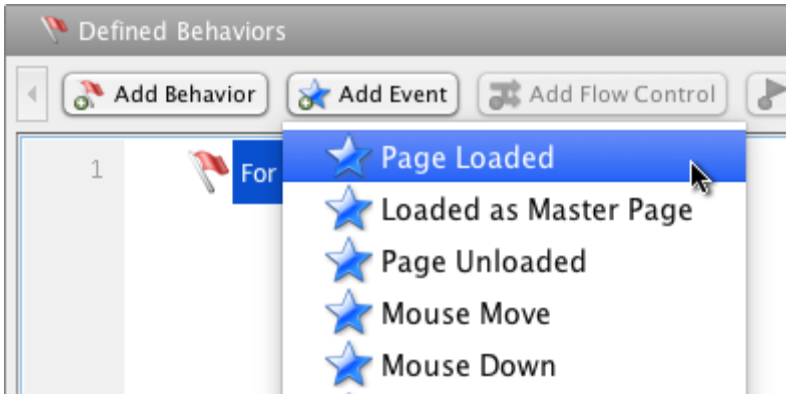
What can we do with it?

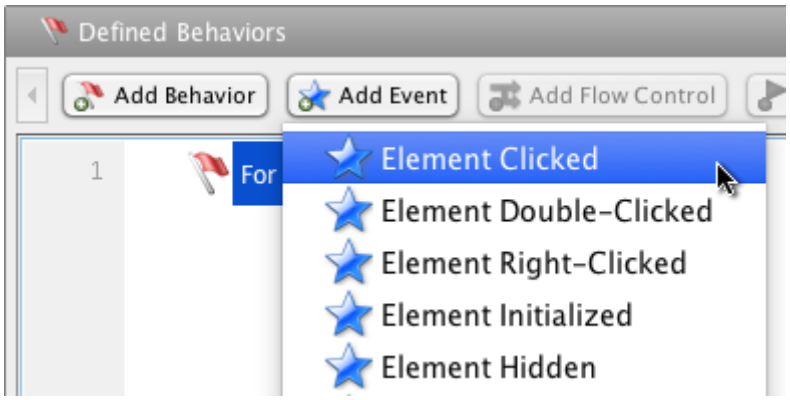
The data stored in local storage will be persisted and will not go away even if you close the we browser. This could be used to save the progress of the HTML5 game locally, or it could be used as a way for communications between ForeUI simulations.

6.2 Events

Event is actually the entrance of the event handler. Once an event is triggered, its handler(s) will be called. That's how ForeUI's simulation works.

After selecting the owner of behavior (pages or elements), you can add event for it:





You can also read:

Define Element's Behavior

Define Page's Behavior

Events for Page and All Elements

Some events are commonly available for page and all elements:

Global Mouse Move, Global Mouse Down, Global Mouse Up, Key Down, Key Up, Custom Event

Events for Page Only

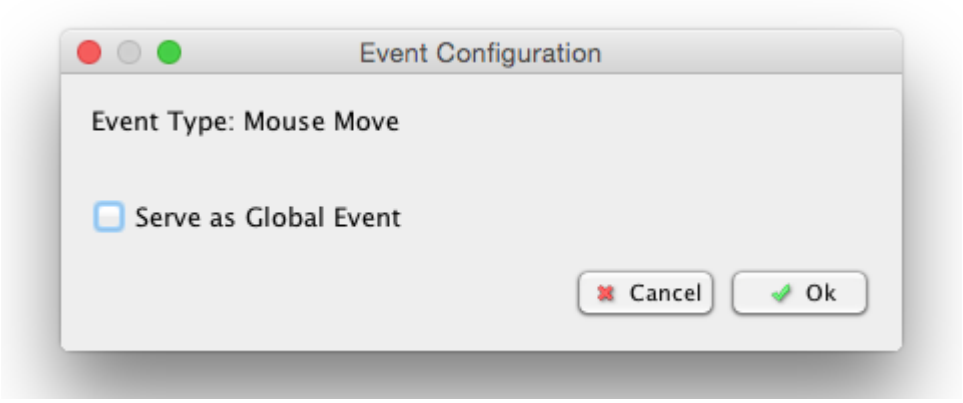
Please read this section: Events for Page

Events for Element Only

Please read this section: Events for Element

Mouse Move

This event can serve as global event or element event. When you add this event, you can decide if it should serve as a global event.



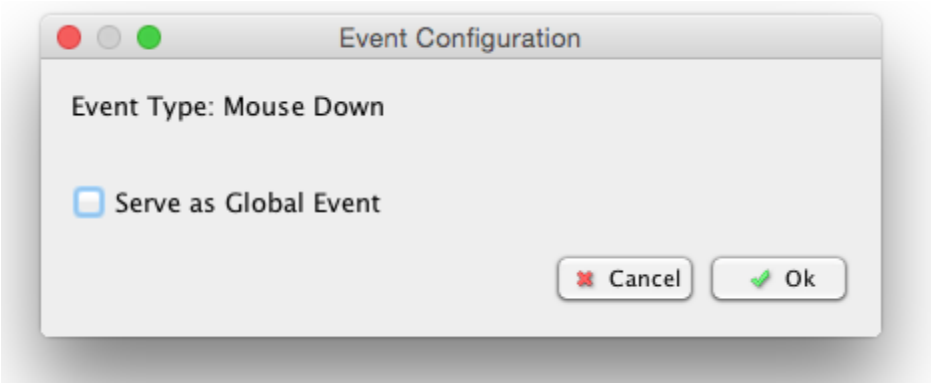
If the event is a global event, it will be triggered when mouse is moved, no matter where the mouse cursor is.

If the event is an element event (non-global), it will be triggered only when the mouse cursor is moving over the element that owns this event.

This event is available for page and all elements.

Mouse Down

This event can serve as global event or element event. When you add this event, you can decide if it should serve as a global event.



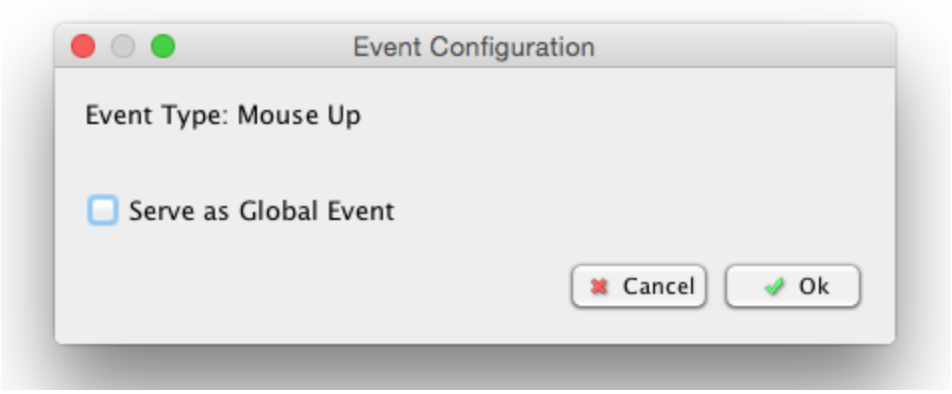
If the event is a global event, it will be triggered when mouse button is pressed down, no matter where the mouse cursor is.

If the event is an element event (non-global), it will be triggered only when the mouse is pressed down on the element that owns this event.

This event is available for page and all elements.

Mouse Up

This event can serve as global event or element event. When you add this event, you can decide if it should serve as a global event.



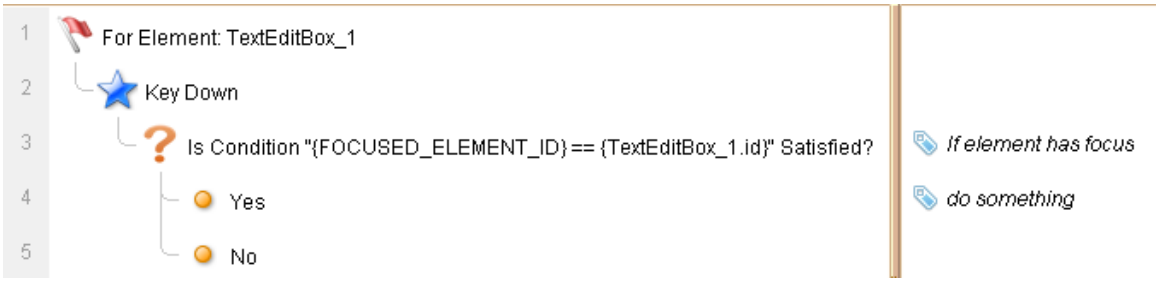
If the event is a global event, it will be triggered when mouse button is released, no matter where the mouse cursor is.

If the event is an element event (non-global), it will be triggered only when the mouse is released over the element that owns this event.

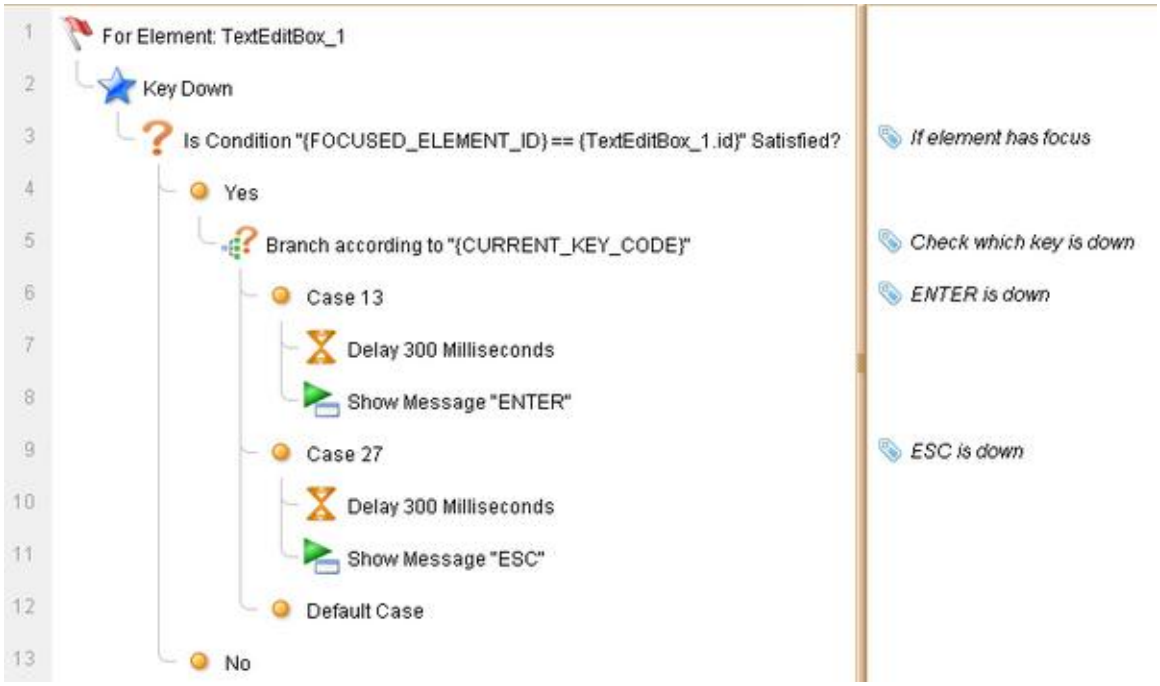
This event is available for page and all elements.

Key Down

This event will be triggered when any key on keyboard is pressed down. Please note that it is a **global event** (although its name has no "global" prefix). You can define its handler in any element, but you may need to check the focus owner before actually do something. The figure below shows how:



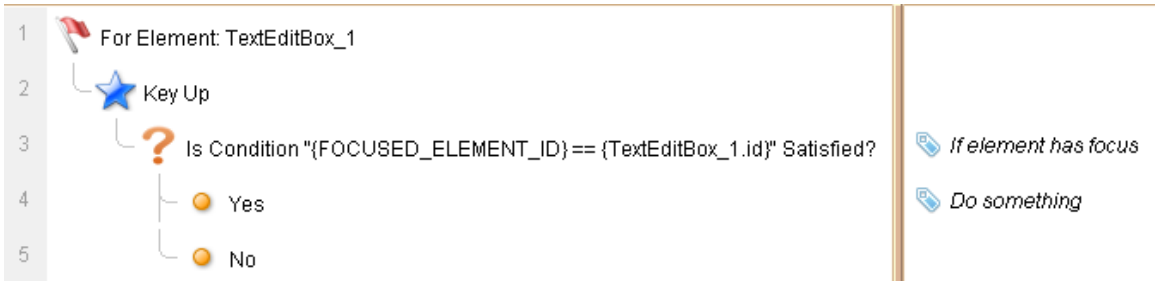
If you need to know which key is pressed in the event handler, you could use the "CURRENT_KEY_CODE" system property. The figure below shows how:



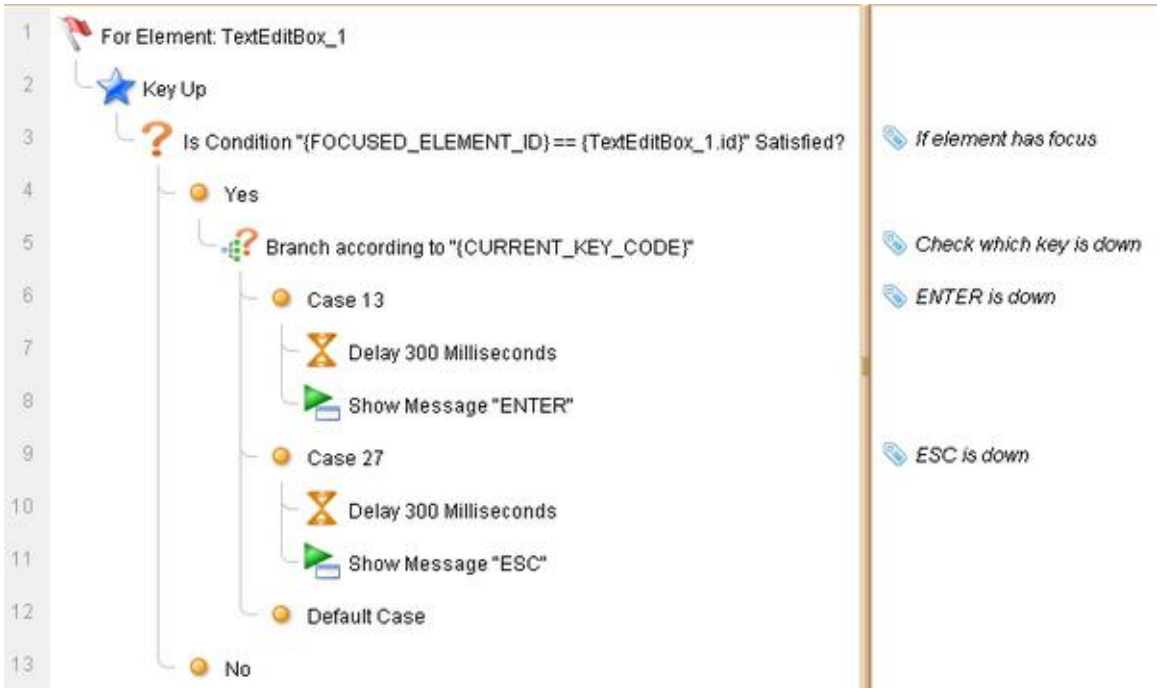
This event is available for page and all elements.

Key Up

This event will be triggered when any key on keyboard is released. Please note that it is a **global event** (although its name has no "global" prefix). You can define its handler in any element, but you may need to check the focus before actually do something. The figure below shows how:



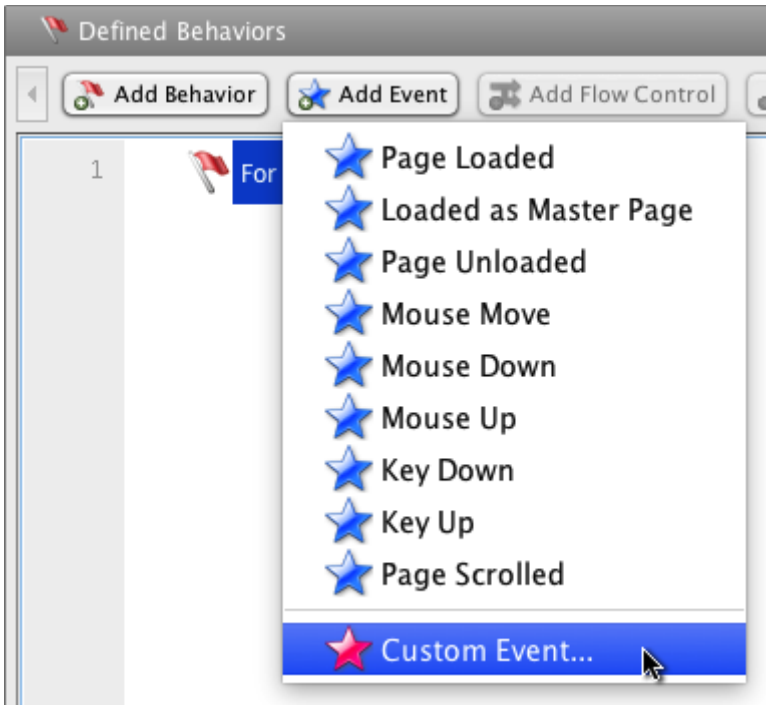
If you need to know which key is released in the event handler, you could use the "CURRENT_KEY_CODE" system property. The figure below shows how:



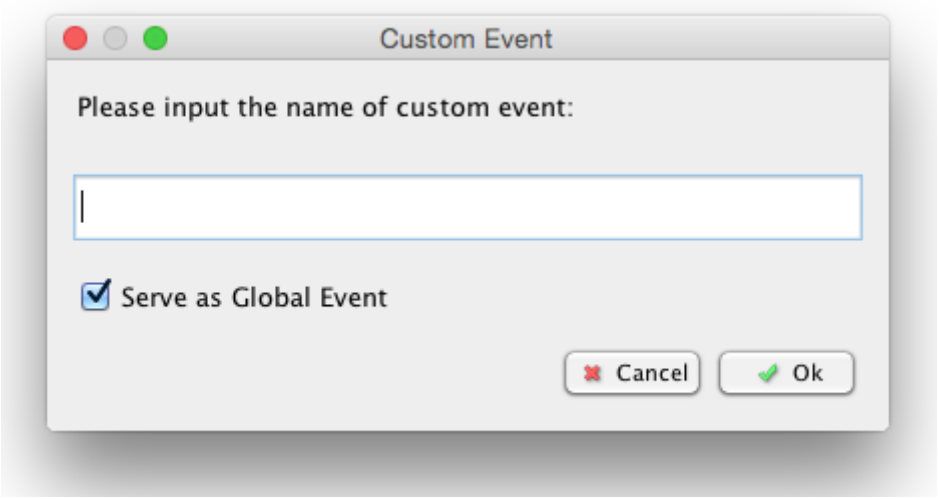
This event is available for page and all elements.

Custom Event

You can define custom event with just an event name, then create its event handler like other events.



Then you can specify the name of the custom event:

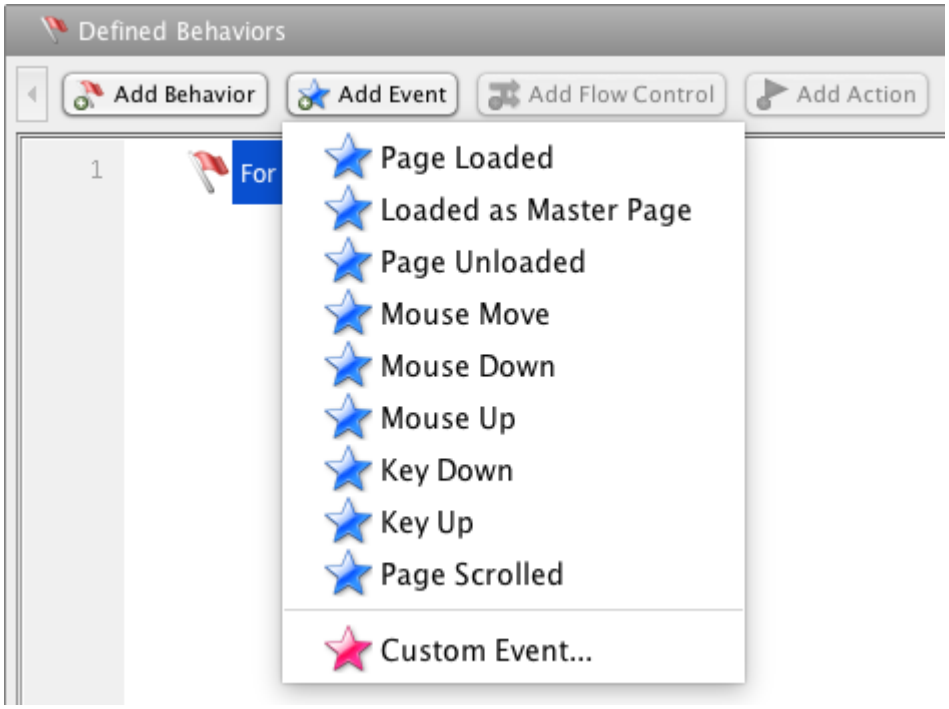


Here is an [example of how to use custom event](#).

Custom event can be defined for page or any element.

Events for Page Only

Some events are for page only, which means they will only be listed when you define behavior for page instead of element.



So far there are four events are for page only:

[Page Unloaded](#)

[Page Loaded](#)

[Loaded as Master Page](#)

[Page Scrolled](#)

Page Loaded

This event is triggered when the page is about to show. It is usually used to do some initialize work before the page is showing up.

For example, if you switch page from Page 1 to Page 2, the following events will occur successively:

1. "Page Unloaded" on Page 1
2. "Page Loaded" on Page 2

Remarks: this event could be triggered multiple times if you switch pages again and again.

This event is available for page only.

Page Unloaded

This event is triggered when the page is about to hide. It is usually used to do some clean up work before leaving the page.

For example, if you switch page from Page 1 to Page 2, the following events will occur successively:

1. "Page Unloaded" on Page 1
2. "Page Loaded" on Page 2

Remarks: this event could be triggered multiple times if you switch pages again and again.

This event is available for page only.

This event is supported since V3.90

Loaded as Master Page

This event is triggered when the page is loaded as master page. For example, page A use page B as master page. If page A is about to show (switching to page A), the "Loaded as Master Page" event is triggered on page B.

Remarks: this event is triggered before the Page Loaded event.

For example, during the loading of Page 1, which has Page 2 as master page, the following events will occur successively:

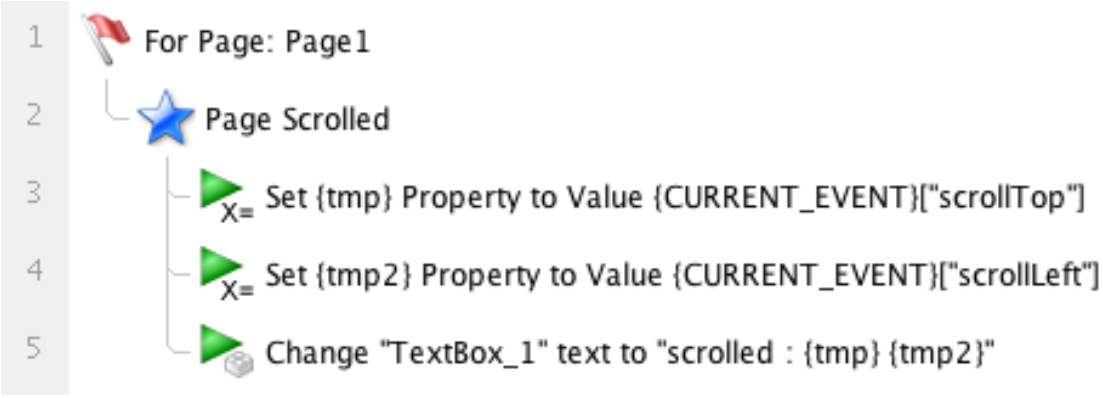
1. "Loaded as Master Page" on Page 2
2. "Page Loaded" on Page 1

Remarks: this event could be triggered multiple times if you switch pages again and again.

This event is available for page only.

Page Scrolled

This event is triggered when the page is scrolled. In the handler of this event, you could access the CURRENT_EVENT system property to retrieve the "scrollTop" and "scrollLeft" attributes, which represent how many pixels are scrolled vertically and horizontally. Here is an example:



Remarks: this event could be triggered multiple times if you scroll page again and again.

This event is available for page only.

Events for Element Only

Some events are for element only, which means they will only be listed when you define behavior for element instead of page.

Element Clicked

Element Double-Clicked

Element Right-Clicked

Element Initialized

Element Hidden

Mouse Over

Mouse Out

Section Expanded/Collapsed

Calendar Date Changed

Selection Changed

Value Changed

Focus Gain

Focus Lost

Window Closed

Element Clicked

This event will be triggered when the element is clicked with left mouse button.

This event is available for all elements.

Element Double-Clicked

This event will be triggered when the element is double clicked with left mouse button.

This event is available for all elements.

Element Right-Clicked

This event will be triggered when the element is clicked with right mouse button.

Remarks: If you handled this event and run the simulation, the context menu of browser will not be displayed when you right click on the element.

This event is available for all elements.

Element Initialized

This event will be triggered when the element is initialized. This event will be triggered only once.

This event is available for all elements.

Element Hidden

This event will be triggered when the element is hidden. This event can be triggered multiple times.

Remarks: in Window element, the "Element Hidden" event is renamed to "Window Closed".

This event is available for all elements.

Mouse Over

This event will be triggered when the mouse cursor hover on the element.

This event is available for all elements.

Mouse Out

This event will be triggered when the mouse cursor moved out of the element's region.

This event is available for all elements.

Section Expanded/Collapsed

This event will be triggered when any section of Accordion element is expanded or collapsed.

This event is available for:

Accordion

Calendar Date Changed

This event will be triggered when the date of Calendar element is changed.

This event is available for:

Calendar

Selection Changed

This event will be triggered when the element's current selection is changed.

This event is available for:

Radio Button, Radio Button Group, ComboBox, List, Menu Bar, Menu, Multilevel Menu, Tabs, Vertical Tabs, Table, Tree

Value Changed

This event will be triggered when the element's current value is changed.

This event is available for:

Text Edit Box, Stepper, Slider, Progress Bar

Focus Gain

This event will be triggered when the Text Edit Box element received the input focus.

This event is available for:

Text Edit Box, Stepper

Focus Lost

This event will be triggered when the Text Edit Box element lost its input focus.

This event is available for:

Text Edit Box, Stepper

Window Closed

This event will be triggered when the Window element is closed (hidden). It is an alias of the Element Hidden event.

This event is available for:

Window

Element Selected

This event will be triggered when the checkbox or radio button element is selected.

This event is available for:

CheckBox, RadioButton

Element Unselected

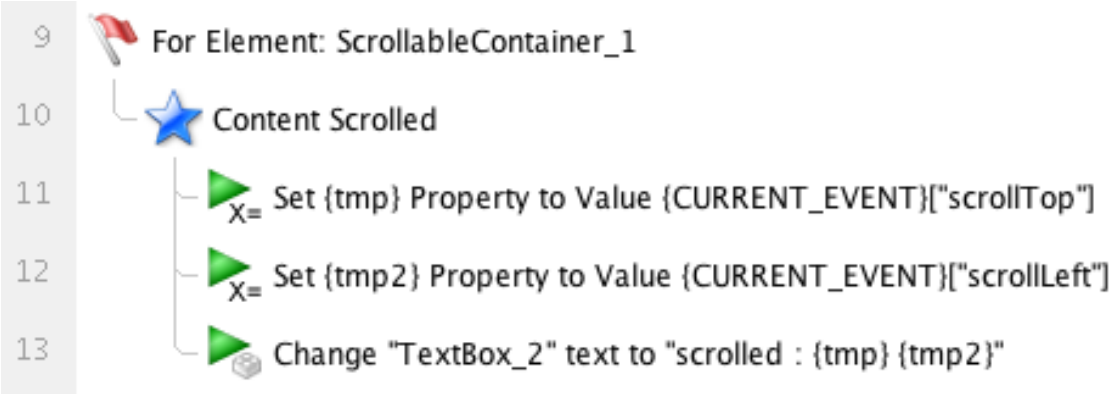
This event will be triggered when the checkbox or radio button element is unselected.

This event is available for:

CheckBox, RadioButton

Content Scrolled

This event will be triggered when the content of ScrollableContainer is scrolled. In the handler of this event, you could access the CURRENT_EVENT system property to retrieve the "scrollTop" and "scrollLeft" attributes, which represent how many pixels are scrolled vertically and horizontally. Here is an example:



This event is available for:

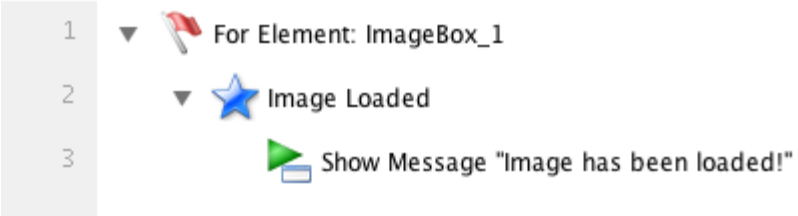
Scrollable Container

Image Loaded

This event will be triggered when the the image is loaded by the browser.

If the source of image has not been set, this event will not be triggered.

By handling this event, you can make sure some tasks will not be executed until the image has been loaded by the browser.



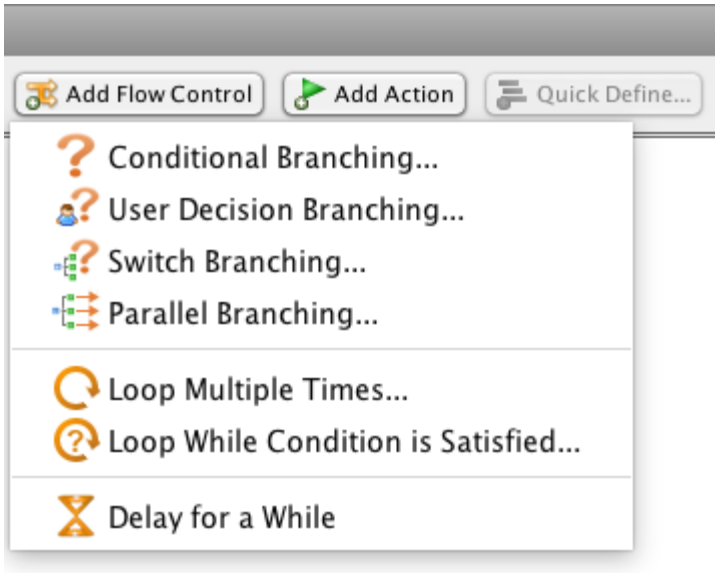
This event is available for:

Image Box

6.3 Flow Controls

Flow control can change the execution route in event handler. It could be a kind of branching, loop or delay.

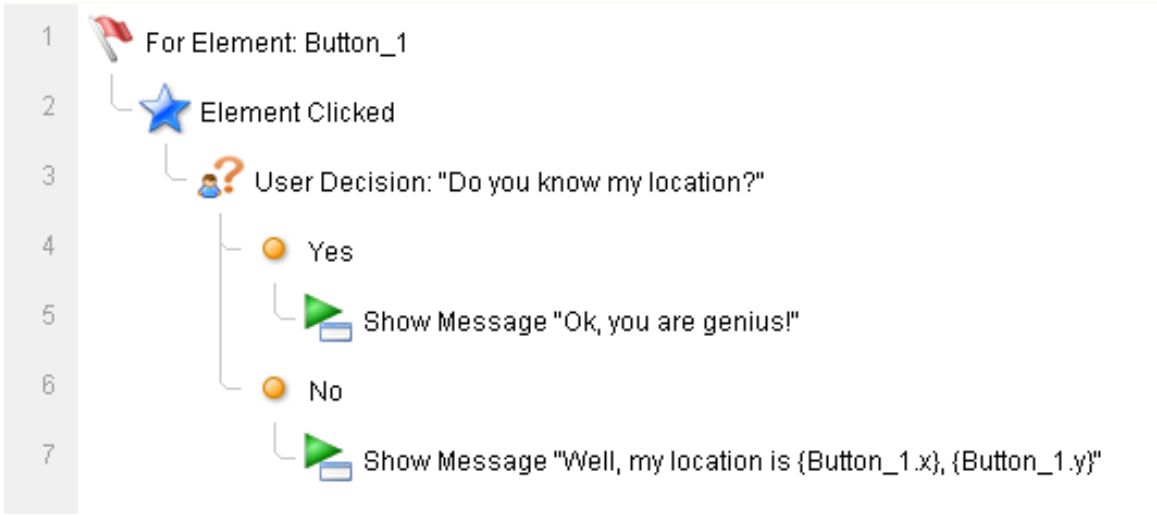
To add a flow control, you will need to select an event in the behavior editor, or an output route of another flow control, and then click the "Add Flow Control" button in the toolbar.



Branching

Branching has one input route and two or more output routes, it will direct user to different routes, according to user's decision, or condition status.

Branching can be nested (place branching under another branching).

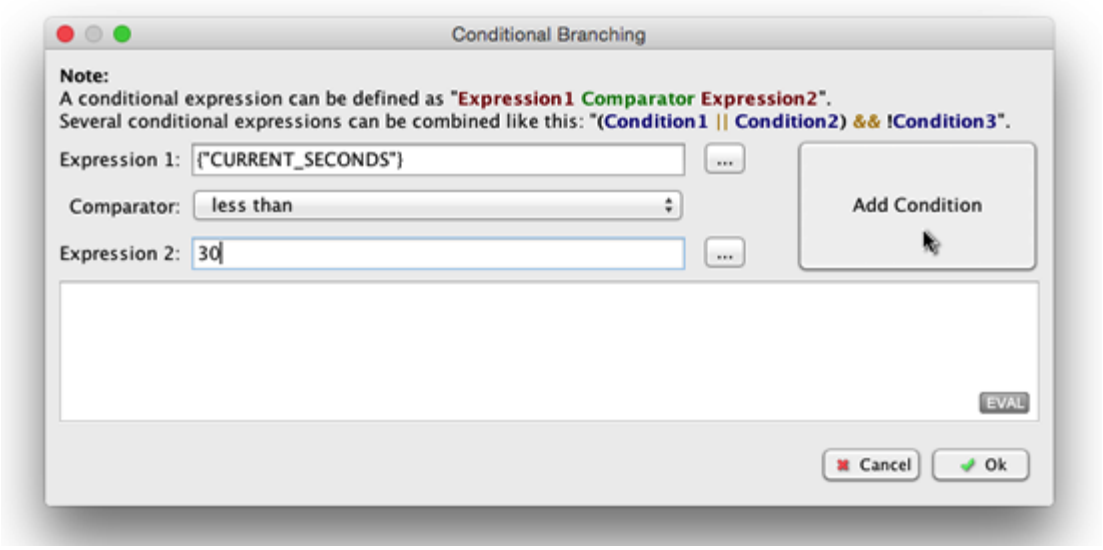


- Conditional Branching
- User Decision Branching
- Switch Branching

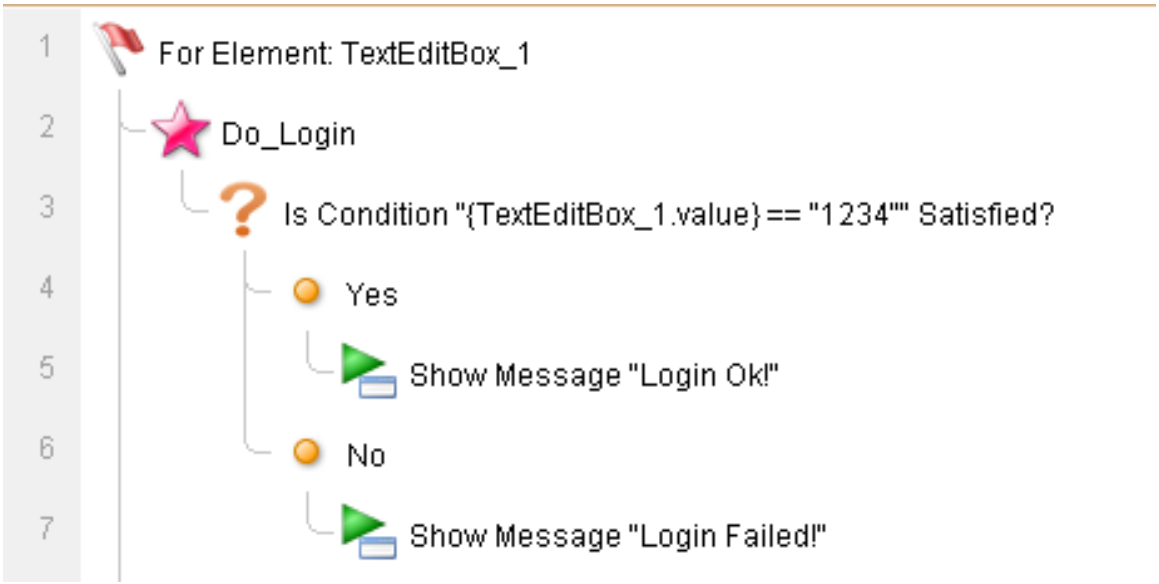
Conditional Branching

Conditional Branching has one input route and two output routes, it will check the value of an expression, and decide which route to go.

After selecting the "Conditional Branching" item from the "Add Flow Control" menu, you will see the editor like this:



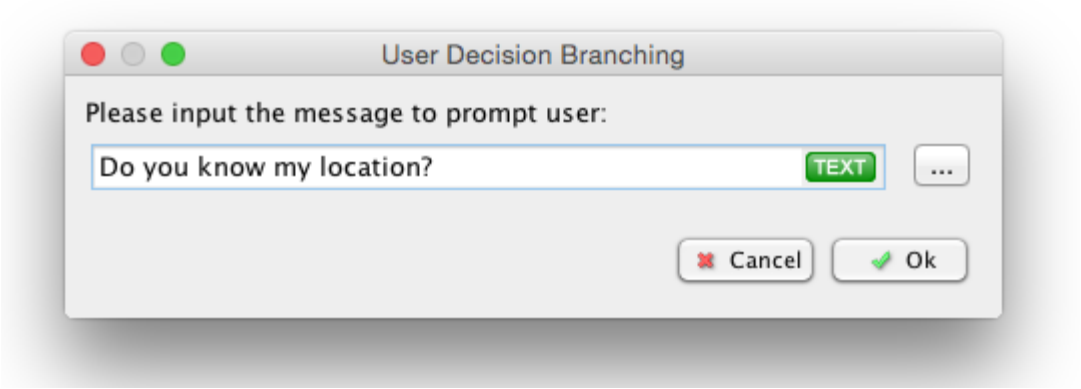
On the editor, you can pick some system/element properties to make the condition expression. This expression will be checked to decide which route will take effect.



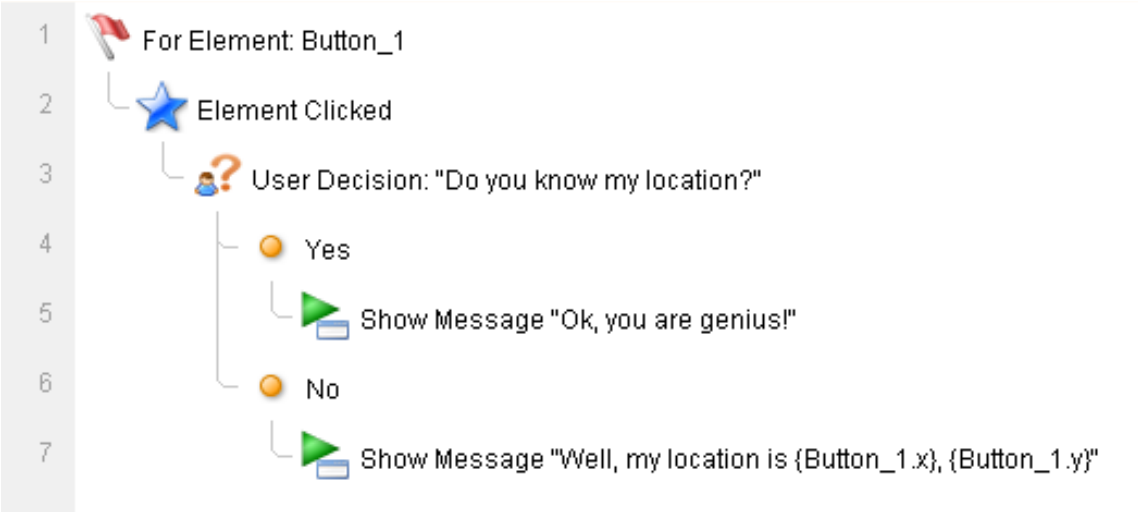
User Decision Branching

User Decision Branching has one input route and two output routes, it will ask user a question, and user's answer will decide which route to go.

After selecting the "User Decision Branching" item from the "Add Flow Control" menu, you will see the editor like this:



You can input the question to ask user. You can also use properties in the question too, just press the ... button and input property into the question. Below is an example that uses properties in the question.

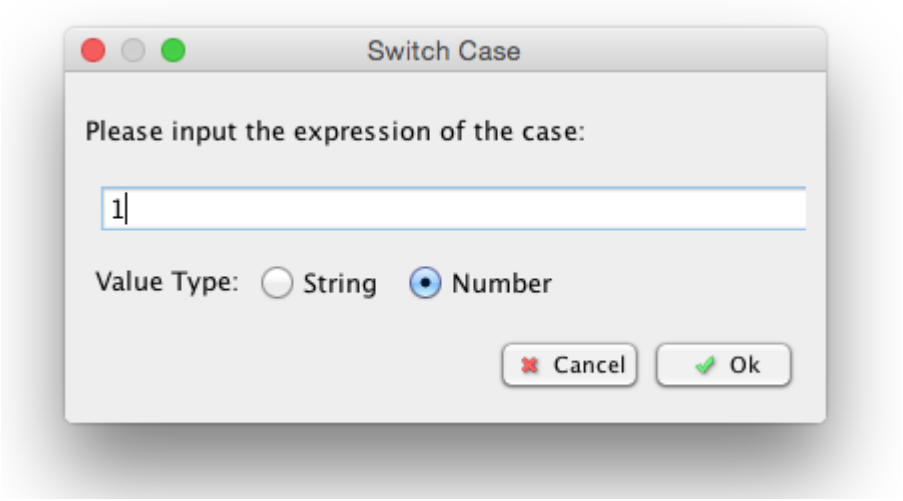


Switch Branching

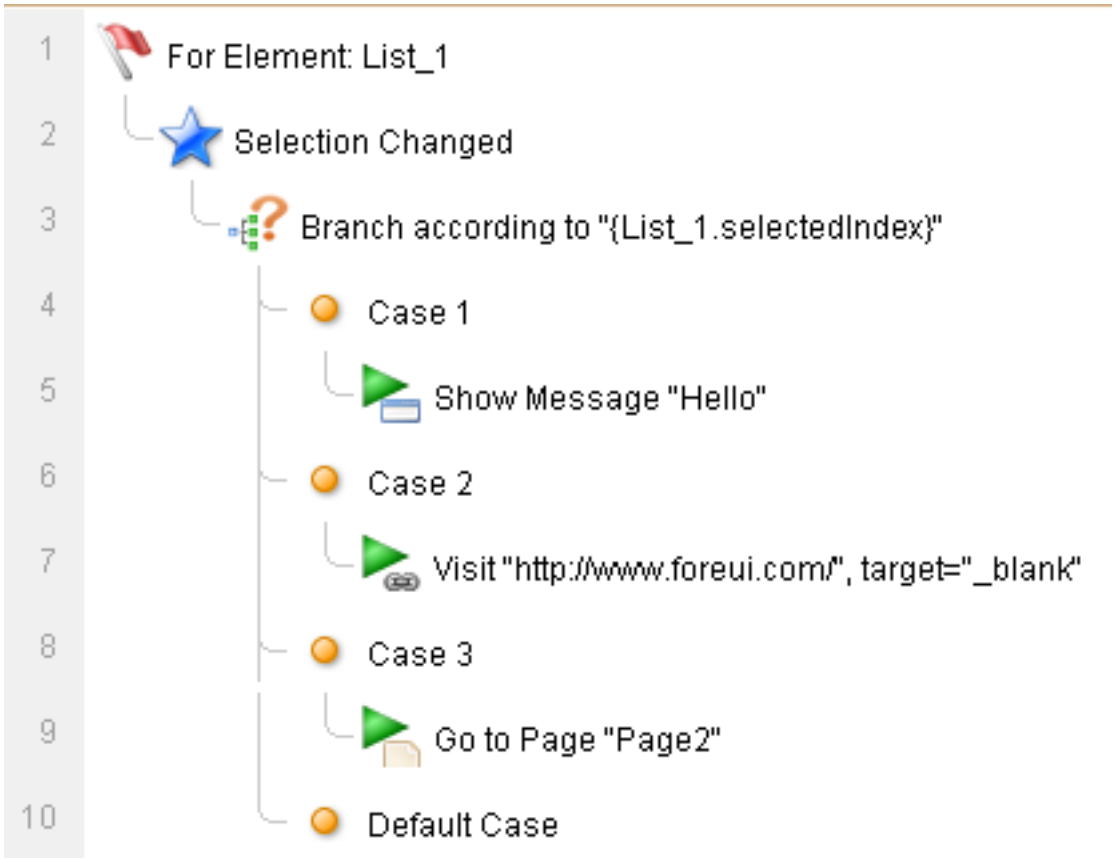
Switch Branching has one input route and multiple output routes, it will check the value of an expression, and decide which route to go.

After selecting the "Switch Branching" item from the "Add Flow Control" menu, you will see the editor like this:

You can define as many cases as you need, each case will be one output route. When adding case, you can insert property in the case expression, and you need to specify the value type of the expression (string or number).



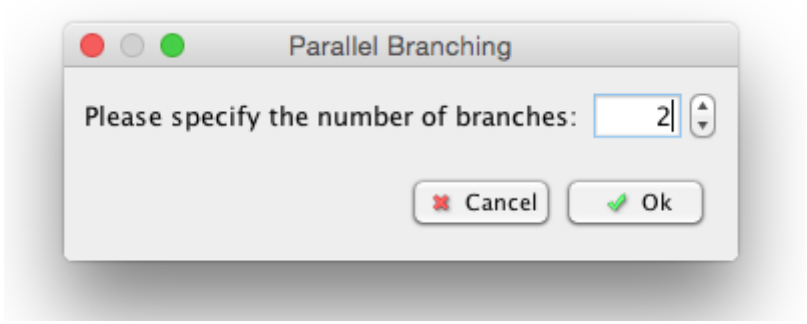
Here is an example of how switch branching is used:



Parallel Branching

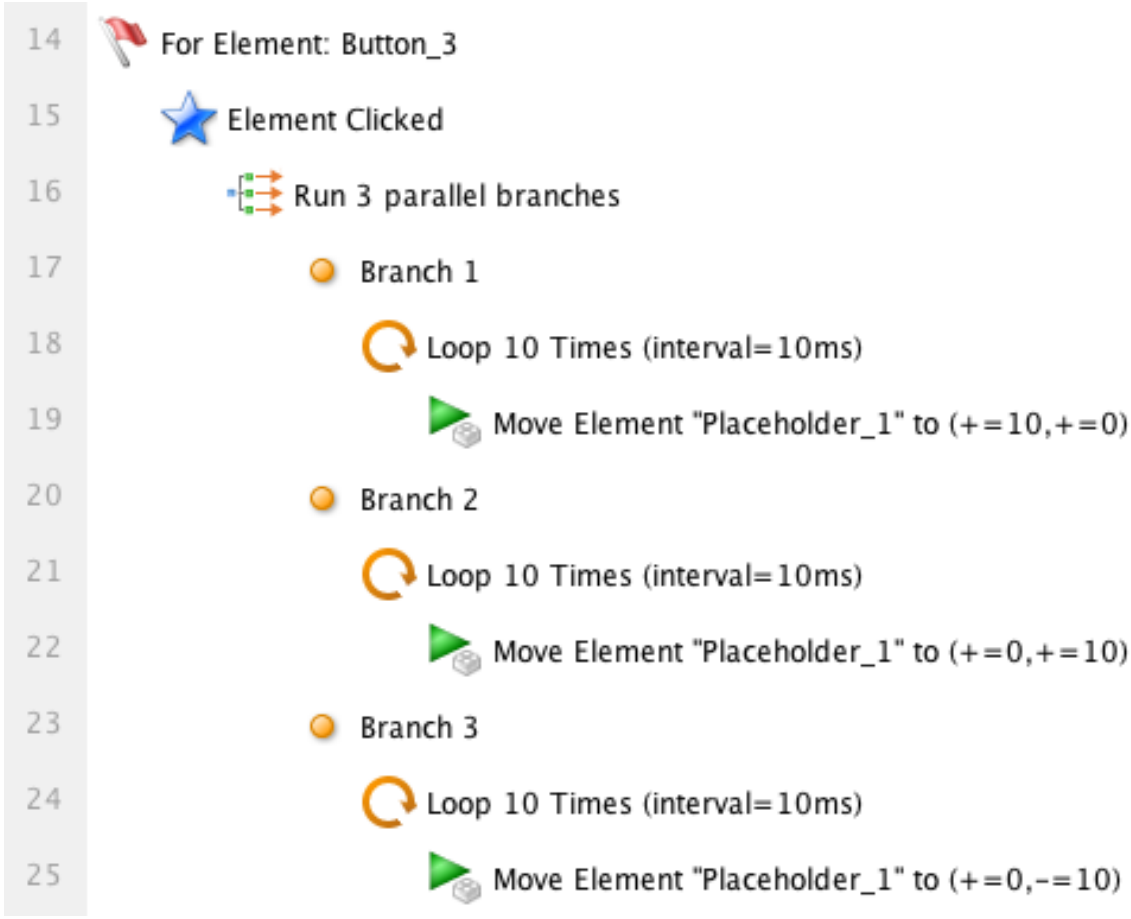
Parallel Branching has one input route and multiple output routes, it will run all routes (branches) in parallel.

After selecting the "Parallel Branching" item from the "Add Flow Control" menu, you will see the editor like this:



Here you can specify how many branches you will need. You can append more branches later, by double-clicking the item in the behavior editor.

Here is an example of how to define parallel tasks by using the Parallel Branching:



Here is a working example on ForeUI store. You can download it and learn how to use Parallel Branching from it.

Loop

Loop will repeat executing the actions under it, for certain count or condition. Loop can also be nested (place loop under another loop).

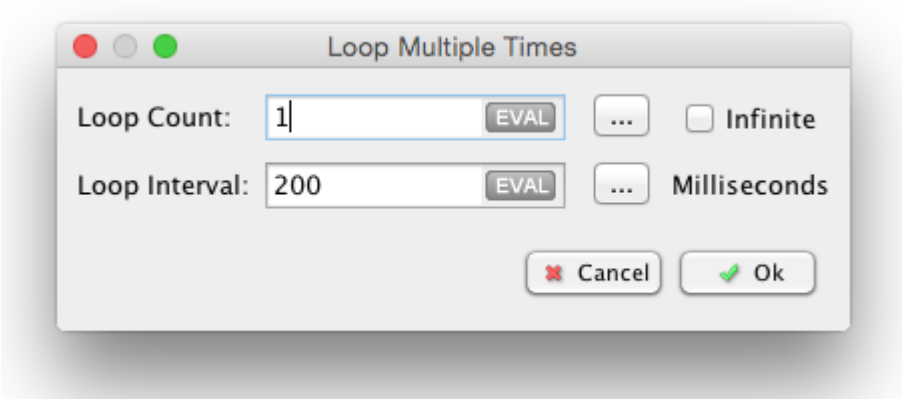
Here is an example of how to use loop.

- Loop Multiple Times
- Loop While Condition is Satisfied

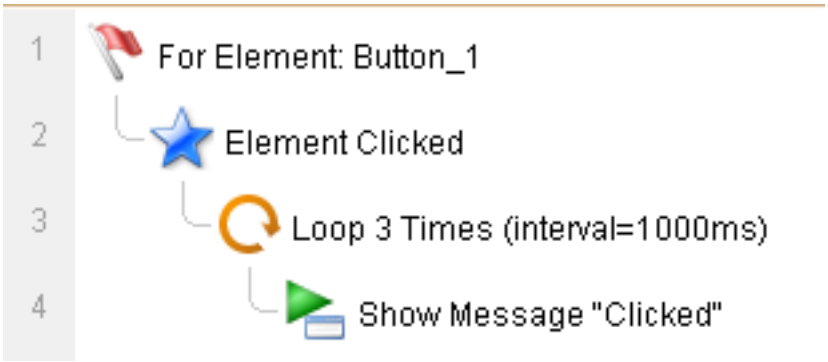
Loop Multiple Times

Loop Multiple Times will repeat the actions under it for certain circles.

After selecting the "Loop Multiple Times" item from the "Add Flow Control" menu, you will see the editor like this:



You can define the number of circles and the interval between circles.

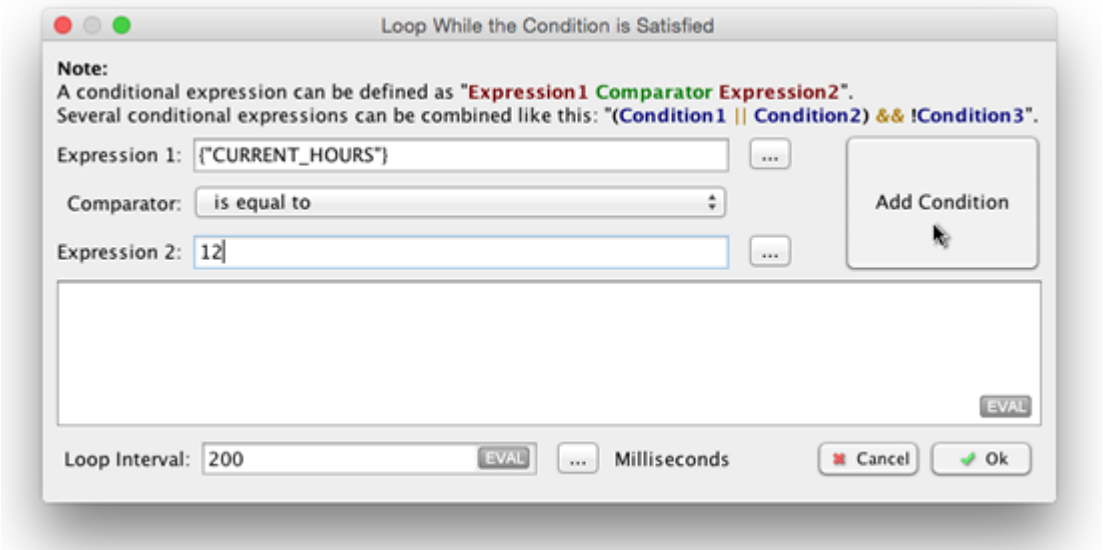


If you select the "Infinite" option, the loop will be infinite and continue forever (unless you close the browser window).

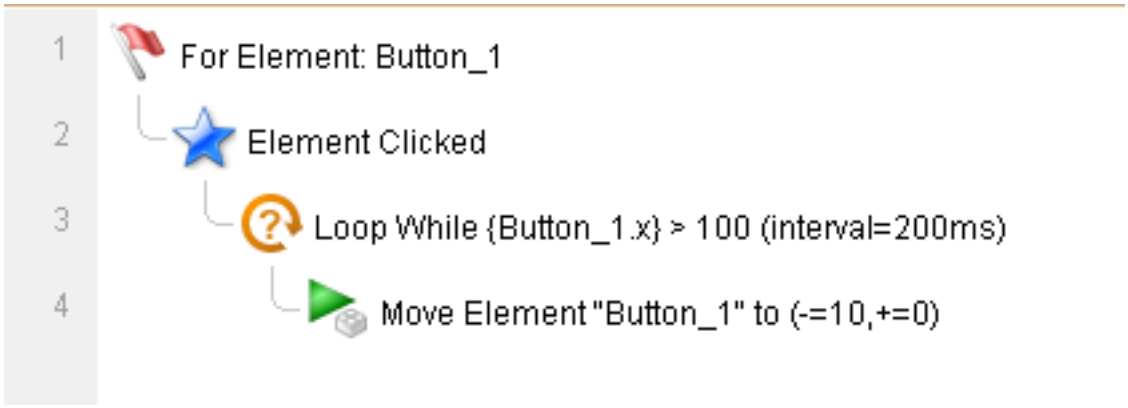
Loop While Condition is Satisfied

Loop Multiple Times will repeat the actions under it while the condition is satisfied.

After selecting the "Loop While Condition is Satisfied" item from the "Add Flow Control" menu, you will see the editor like this:

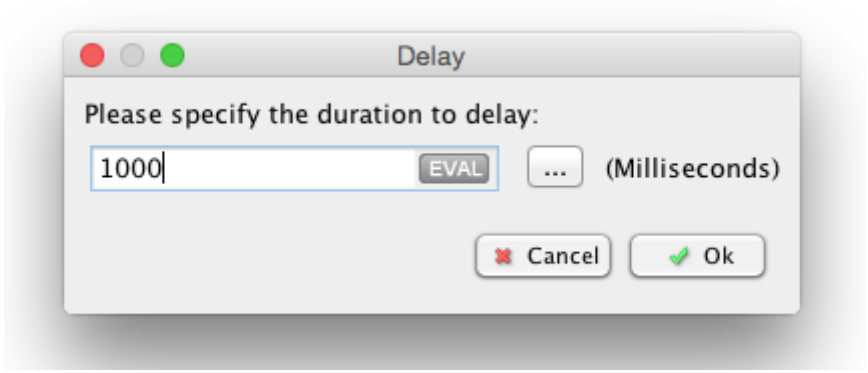


The editor is similar with that for Conditional Branching, the expression you created will be checked to decide whether the looping will continue.



Delay

The delay flow control will delay the subsequent action for specific time. Delay can be nested (place delay under another delay).



Delay can be placed under event or other flow controls. You can put some actions under the delay, or after the delay, these two ways are equivalent, you can choose any one you like:

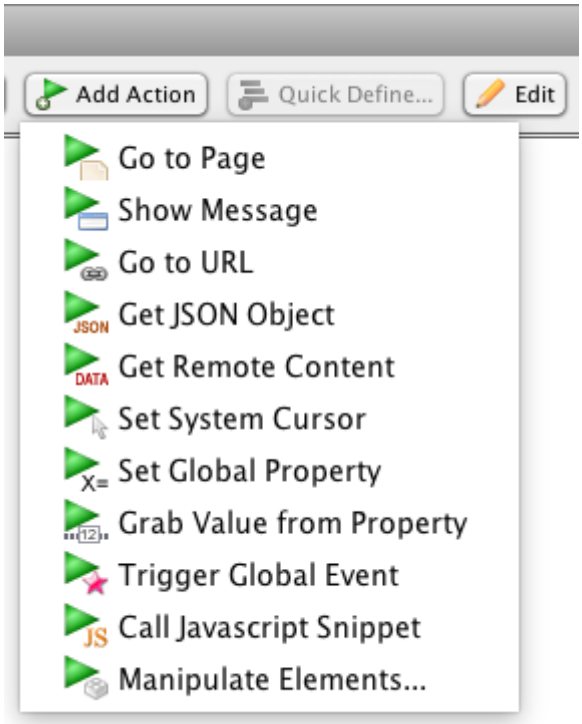
You can see the ... button on the right of input field, that means you can use properties to build your delay time.

Here is an example on how to use the delay.

6.4 Actions

In event handler, action is the actual part that really does something. By placing different actions under different events and flow controls, you can fully define the behavior of the prototype.

In the behavior editor, when you select an event or output route of flow control, the "Add Action" button will be enabled in the toolbar. You can click it to show the actions.



Go to Page

Show Message

Visit URL (Go to URL)

Get JSON Object

Get Remote Content

Set System Cursor

Set Global Property

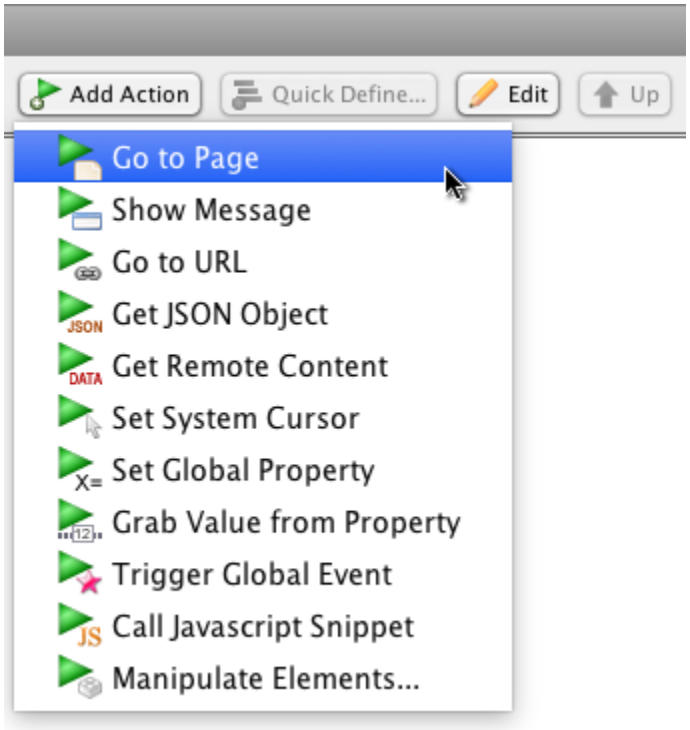
Grab Value from Property

Trigger Custom Event (Trigger Global Event)

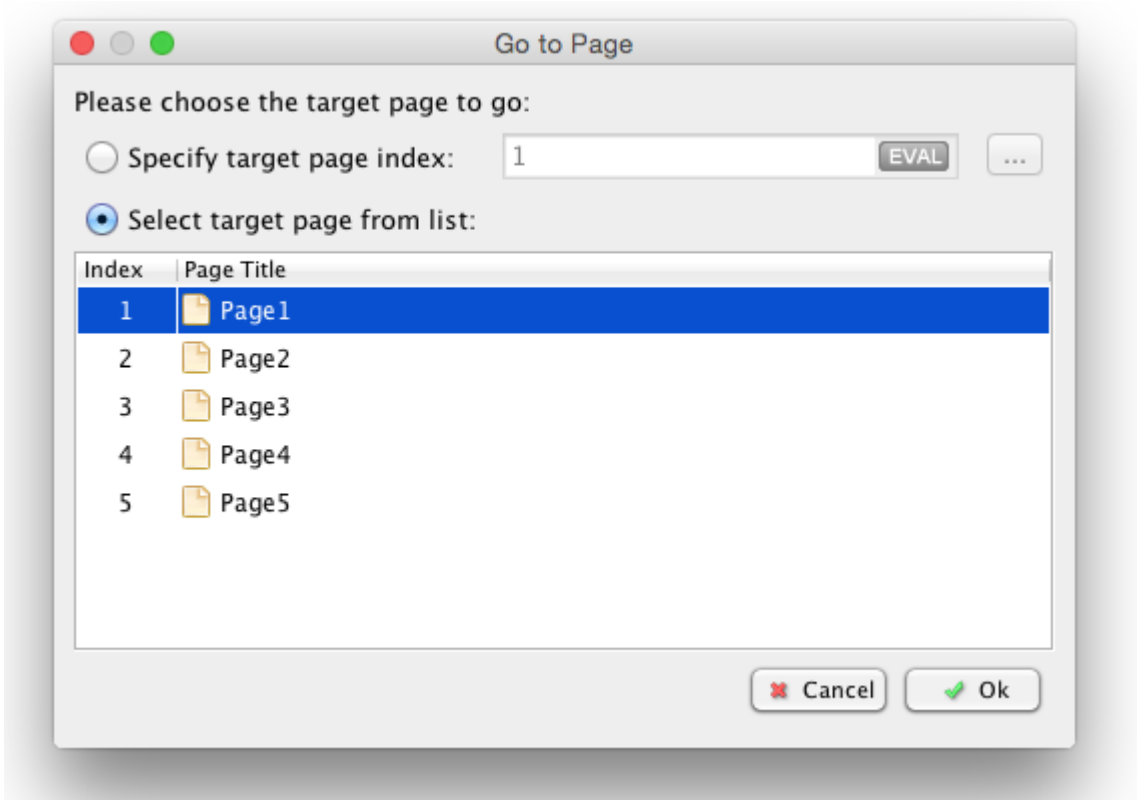
Manipulate Elements...

Go to Page

Go to Page action will switch the page to the one you specified, the content in current page will be hidden and those on the target page will then be displayed.



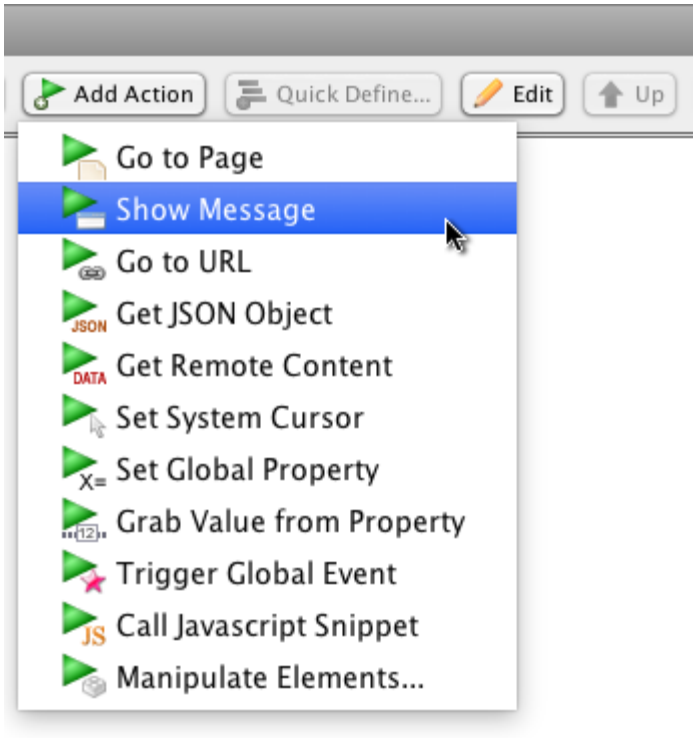
You can specify the target page index (starts with 1), or select the target page from the page list.

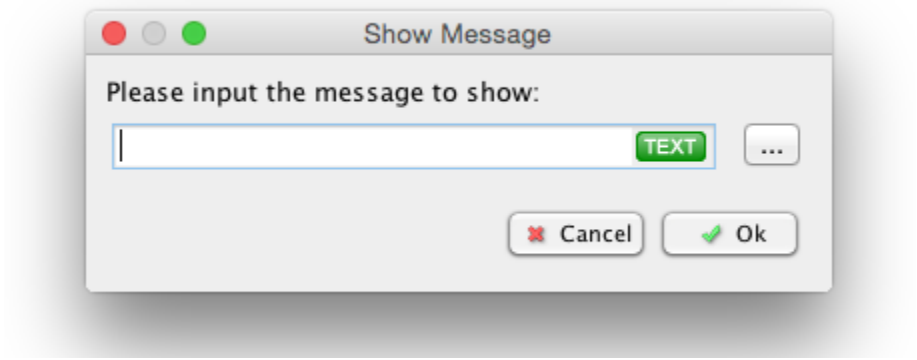


Once this action get performed, the HTML5 simulation goes to the target page, which means all content in the previous page will be hidden, and the content in target page will be displayed. Please notice this action will not trigger the Element Hidden event, instead the "Page Loaded" event will be triggered at the target page. If the target page has master page, the Loaded as Master Page event will also be triggered on the master page too.

Show Message

Show Message action will display a window on screen and prompt user the specified message.

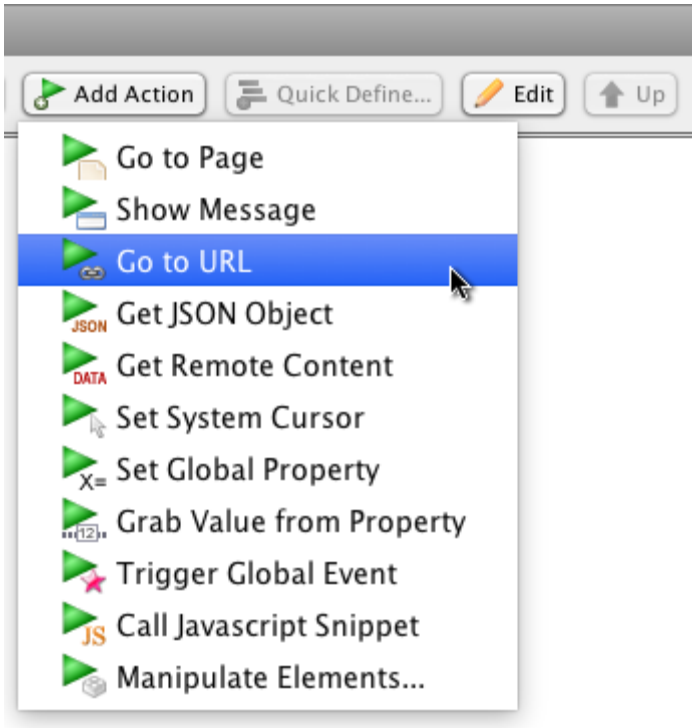


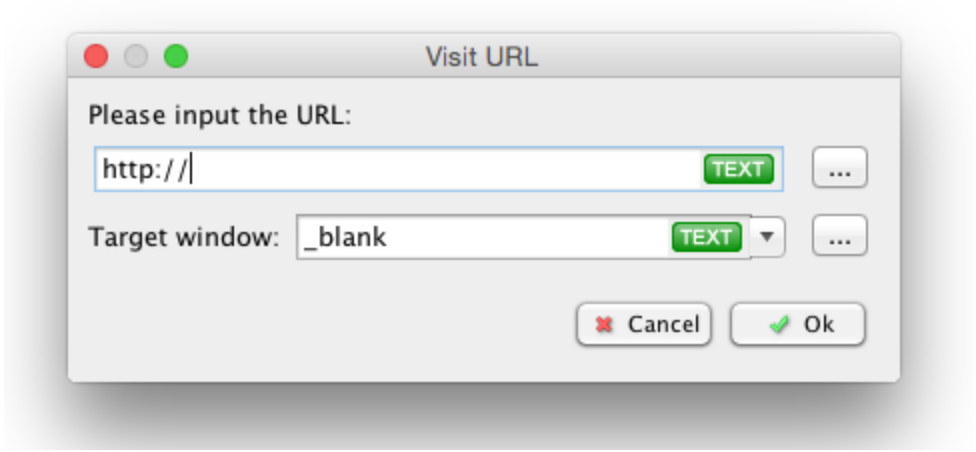


You can see the  button on the right of the message field, so you can place property into the message. It is also quite useful to output some property values for debugging.

Visit URL

Visit URL action will redirect browser to a specified URL.



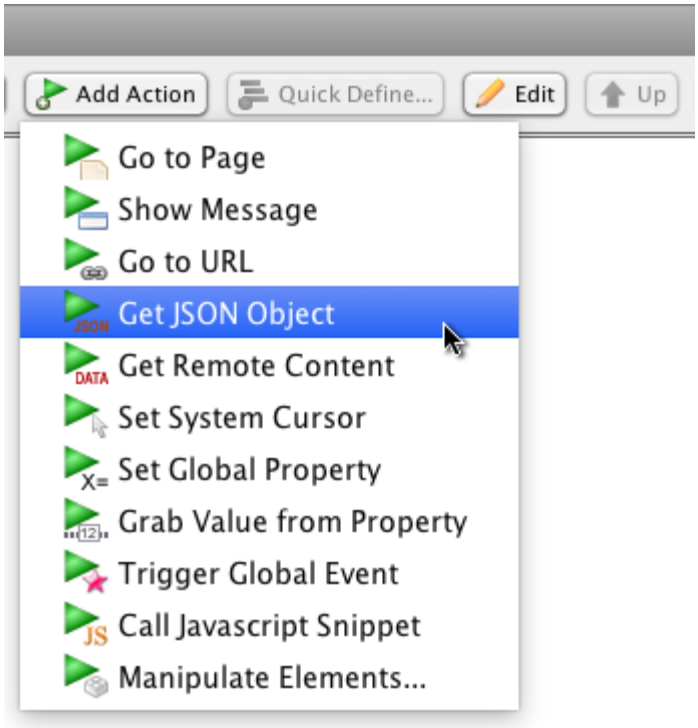


Both the URL and target window fields can accept properties.

Remarks: if the target window field is set to "_self", the current web page will be unloaded after the action is performed, and the upcoming logic will not be executed.

Get JSON Object

This action allows the HTML simulation to get JSON object from remote server via JSONP API, and store the JSON data to an object type property. That means ForeUI simulation can actually integrate with other web service via JSONP APIs, and we can even create some web apps with ForeUI!



When you add this action, you will be asked to provide some parameters, as shown below:

You will need to input the JSON or JSONP API here, and specify the name of property that will store the returned JSON data (as an object). There is an optional parameter that allows you to specify the custom event to trigger, when there is any error happened.

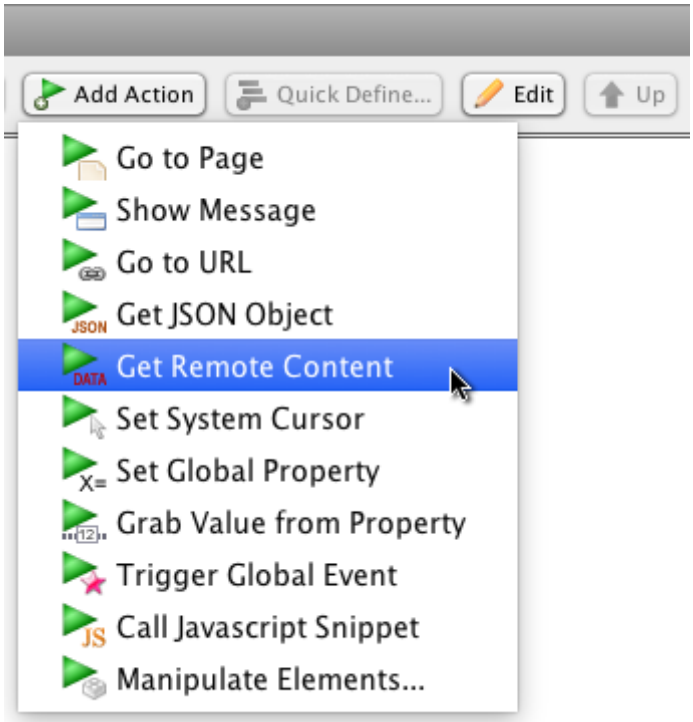
You should also set a timeout for the action. In case the remote server does not response in certain seconds, the custom event for error processing will also be triggered.

Remarks: if you input a JSON API here, you will need to use a JSONP proxy to make it work. Your HTML5 simulation must have internet access to access the proxy server.

You can find the detailed example for the usage of this action in this blog post.

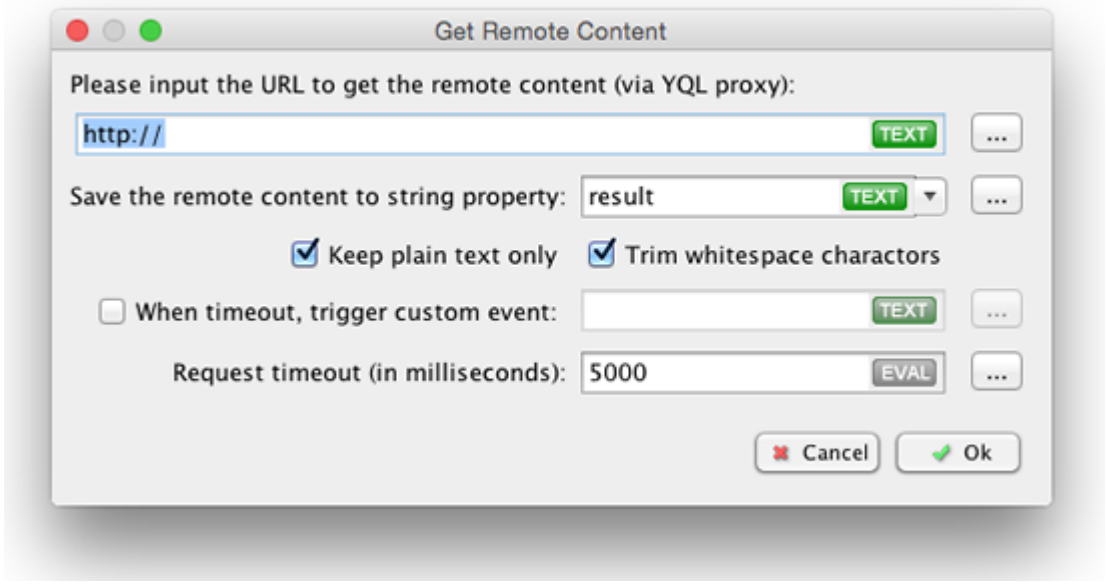
Get Remote Content

This action allows the HTML simulation to retrieve data from remote server via YQL proxy, and store the data to an string type property. That means ForeUI simulation can actually integrate with other web services, and we can even create some web apps with ForeUI!



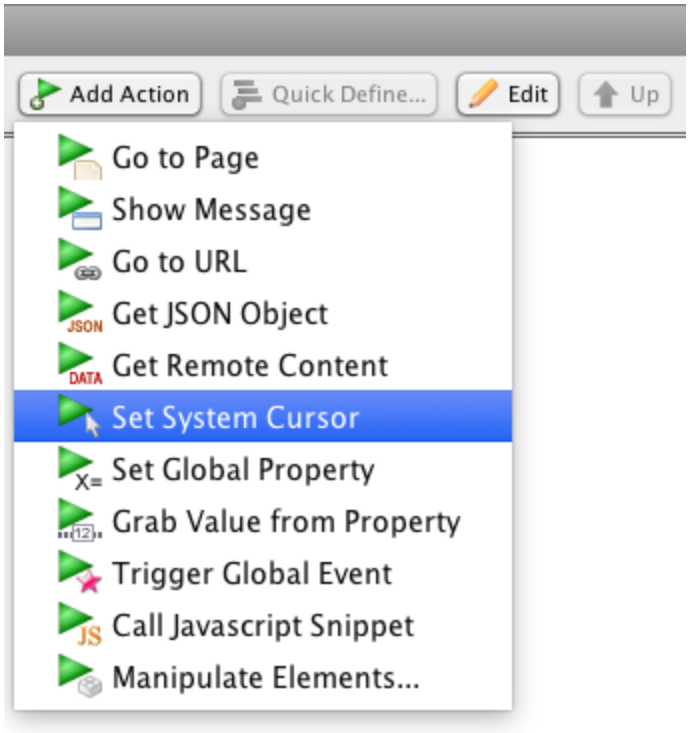
Comparing to the Get JSON Object action, this action doesn't require the remote server provides JSON-P API. So you can try to retrieve the content of much more web sites. However, this action also has its own limitation, which is its dependency on Yahoo's server, which acts as YQL proxy.

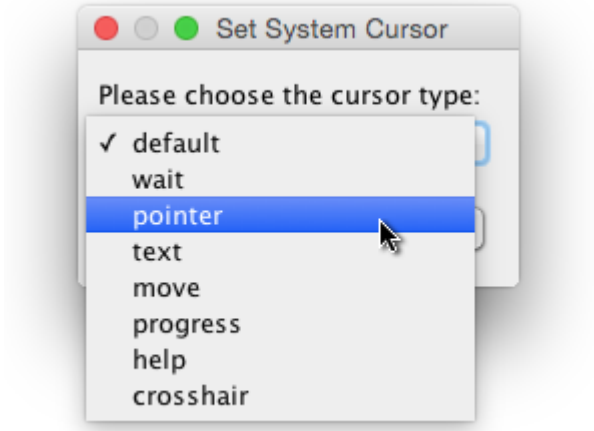
When you add this action, you will be asked to provide some parameters, as shown below:



Set System Cursor

Set System Cursor action can specify the style of cursor.

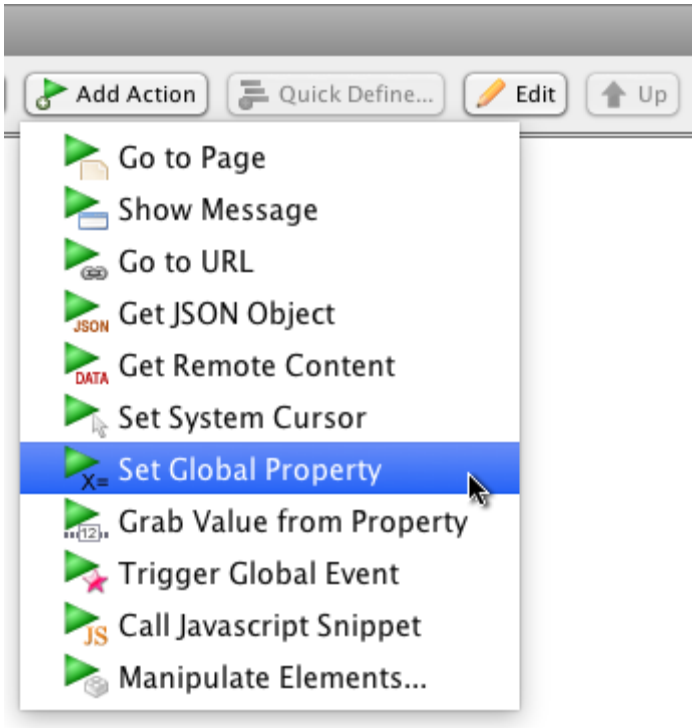


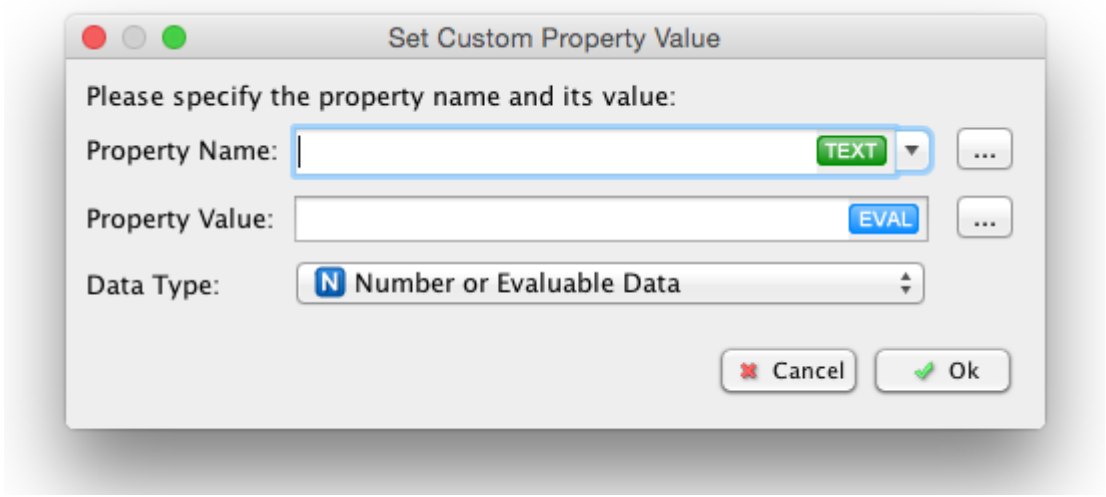


The name of cursor style is compatible with the cursor CSS attribute, and you can see more details here.

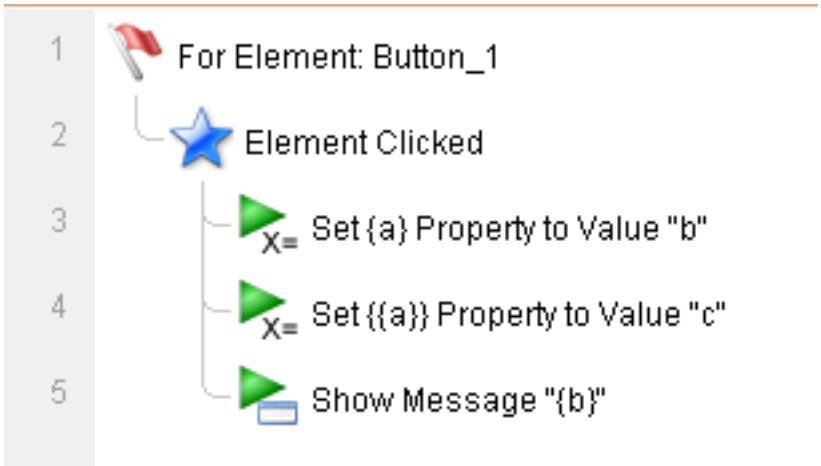
Set Global Property

Set Global Property action can modify or create user defined property. If the specified property does not exist before the action performed, the action will create one silently.





Both the property name and value fields can accept properties, that means you can use property in another property's name. Here is an example:

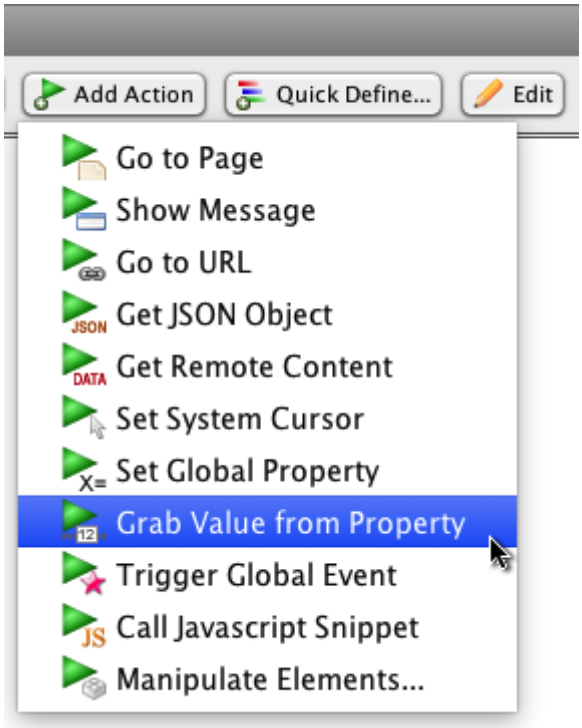


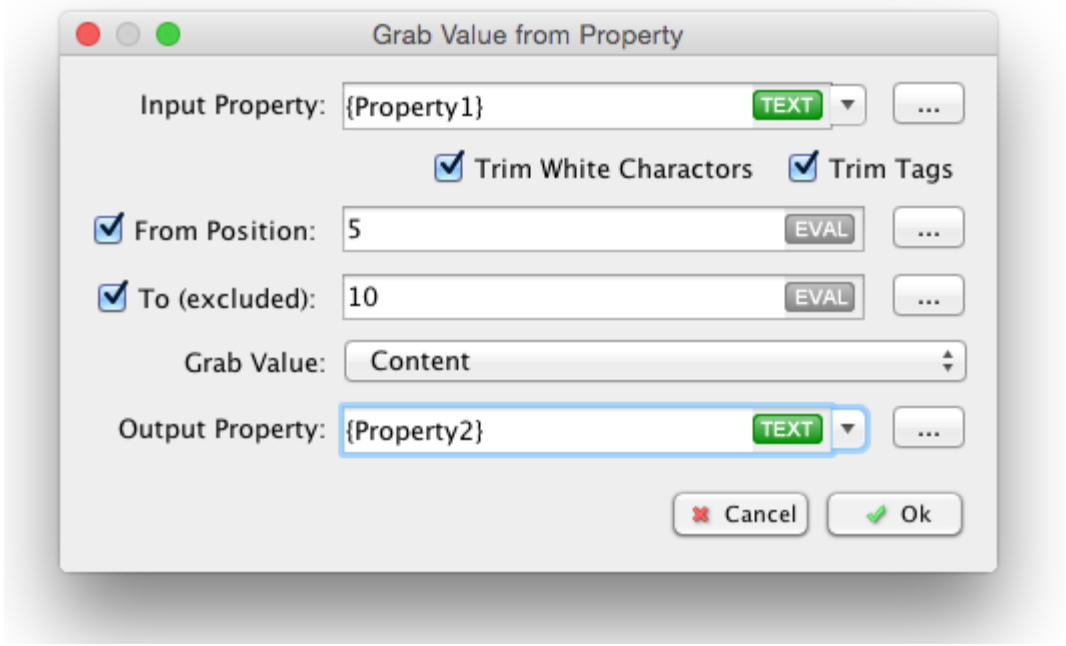
After clicking the button, you will see the value for property b is "c". Explanation: {a} = "b", {{a}}={b}="c", so {b}="c".

Please don't forget to choose the correct data type for the property.

Grab Value from Property

Grab Value from Property action allows you to specify a property as input string, and store the value you grabbed into a user defined property. If the specified output property does not exist before the action performed, the action will create one silently.



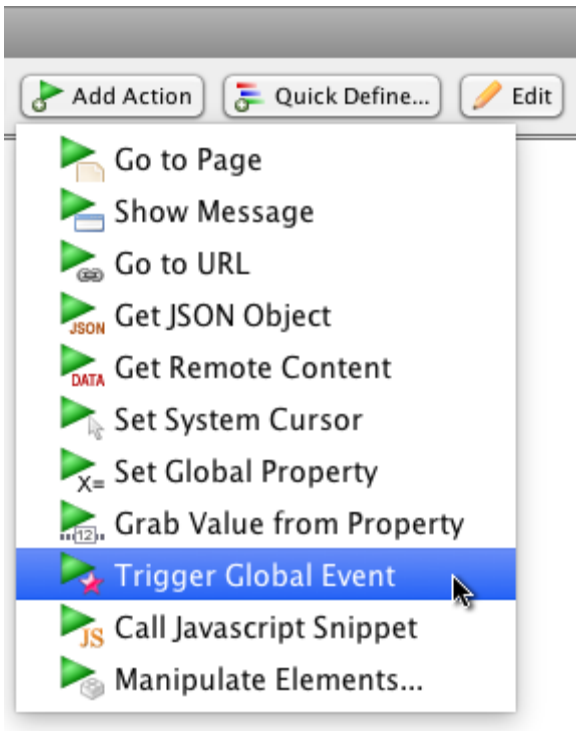


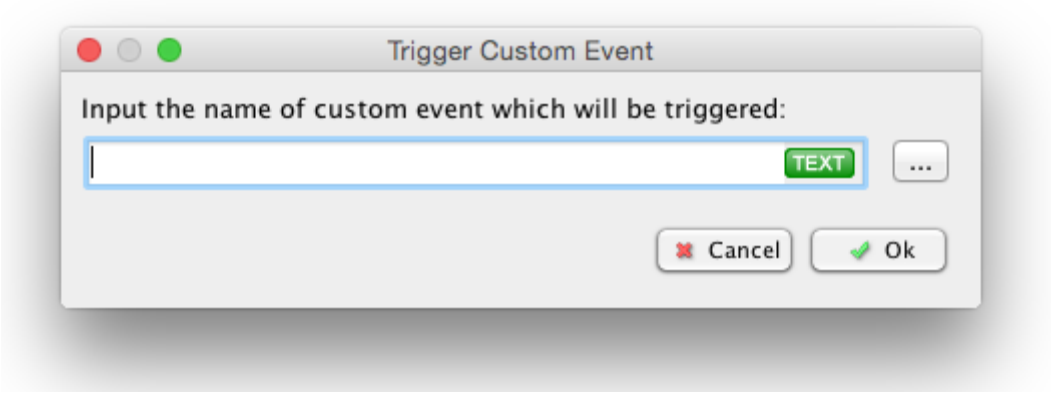
Here you can specify some parameters:

- Input Property: the expression of the input property, which can be system property, element property or user-defined property.
- Trim White Characters: when enabled, will trim all white characters from the input value, including spaces, tabs, returns, new lines etc.
- Trim Tags: when enabled, will trim all tags from the input value. Example: inputting "Hi" will get "Hi".
- From Position: when specified, will try to get the sub string from the given position. If this is not specified and "To" is specified, will try to get sub string from the beginning.
- To (excluded): when specified, will try to get the sub string from the start position to this position (excluded). If this is not specified and "From" is specified, will try to get sub string until the end.
- Grab Value: here you can choose what value will be stored in the output property, it could be "Content" or "Length of Content".
- Output Property: the expression of the output property, which must be a user-defined property.

Trigger Global Event

Trigger Global Event action will trigger the specific global custom event, thus the handlers for the global custom event will be called.

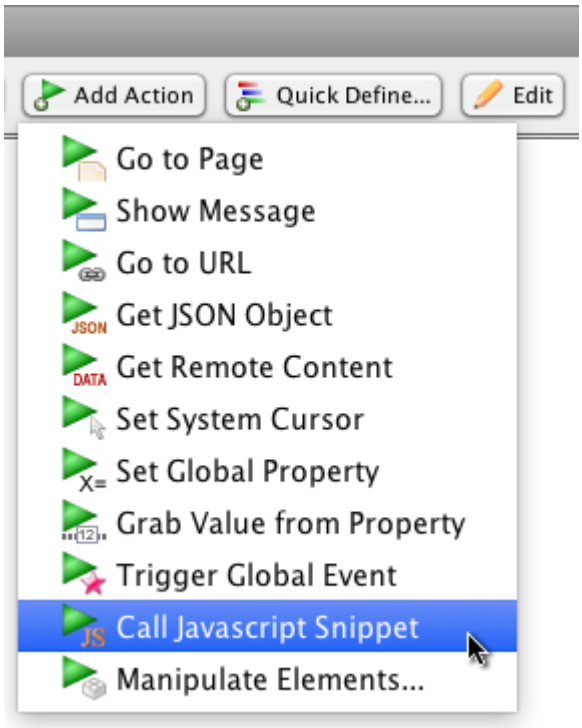


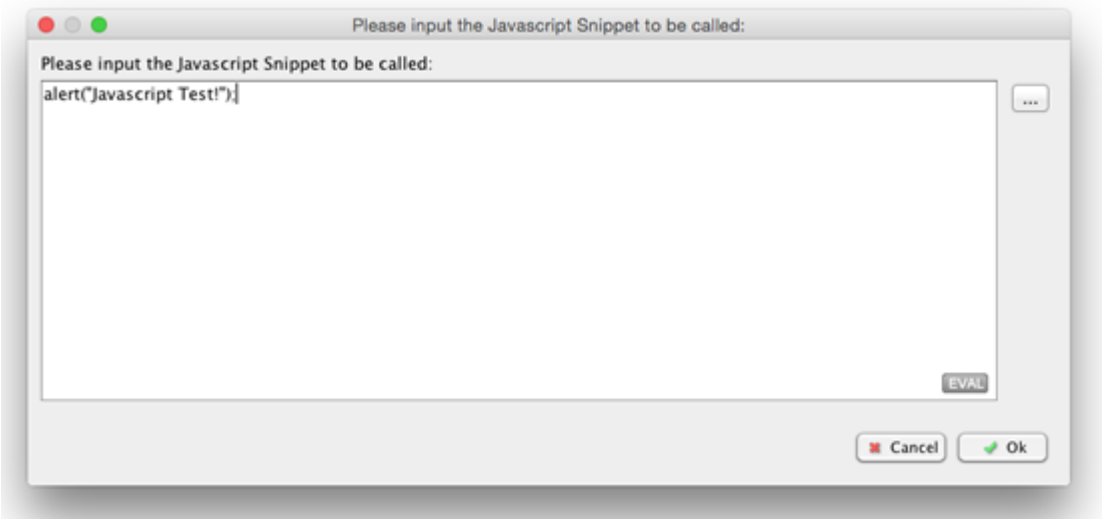


You can see the ... button on the right of the custom event name field, so you can use properties to build the custom event's name.

Call Javascript Snippet

This action allows you to input the Javascript snippet to be called. By using this action, you will be able to call any predefined Javascript function, thus integrate your ForeUI simulation with other application.





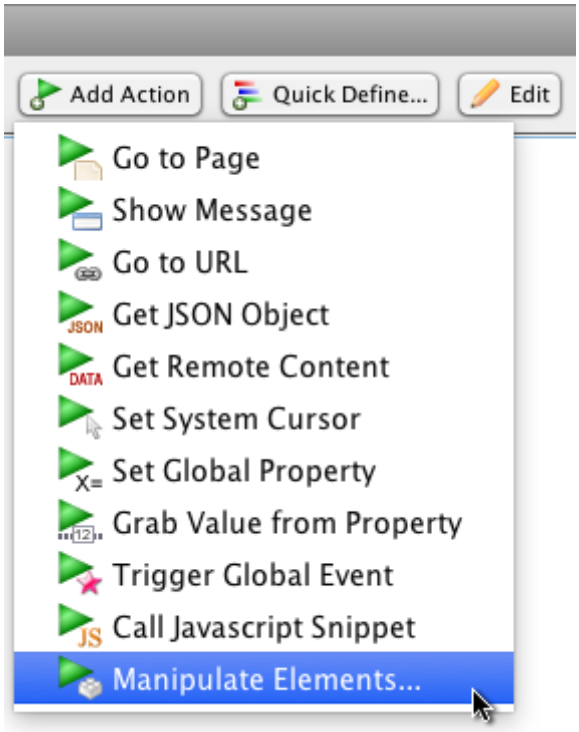
When inputting the Javascript code, please make sure NOT to use <script> tag to wrap the content.

You can insert properties in the Javascript, and they will be replaced by the actual values in the HTML5 simulation. The Javascript content only supports EVAL parsing mode.

Remarks: just like the Script element, this new action is prepared for advanced users. You should know what you are doing when using it, calling incorrect Javascript may bring strange behavior to your HTML5 simulation, and sometimes even halts the entire simulation.

Manipulate Elements...

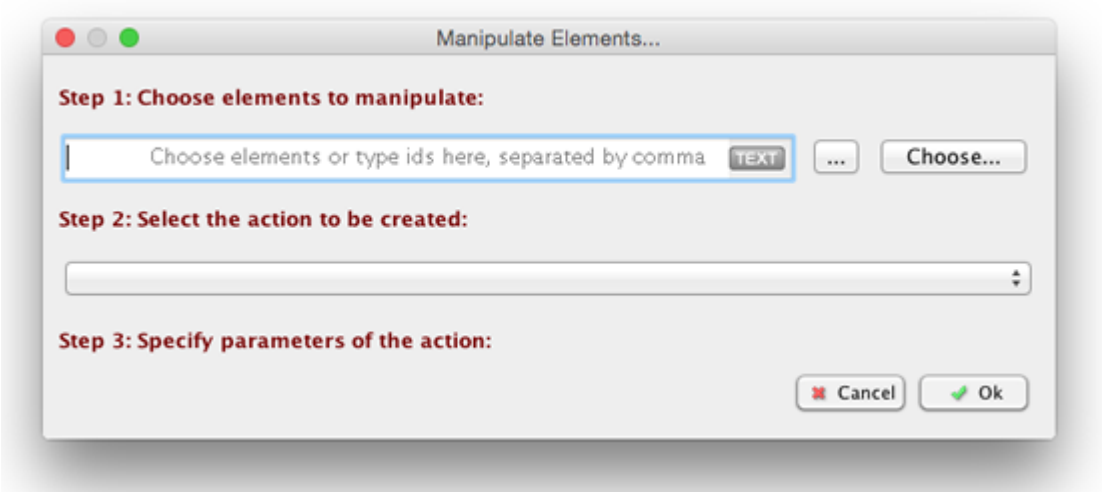
At the bottom of the popup menu for "Add Action" button, we can also find a menu item named "Manipulate Elements...".



This is not a single action, instead it is a set of actions that can operate on specify element, and change the element's appearance, status, attribute etc. Selecting the "Manipulate Elements..." menu item will bring out

the "Manipulate Elements..." window, which works as a wizard to create the action you need. The wizard allows you to:

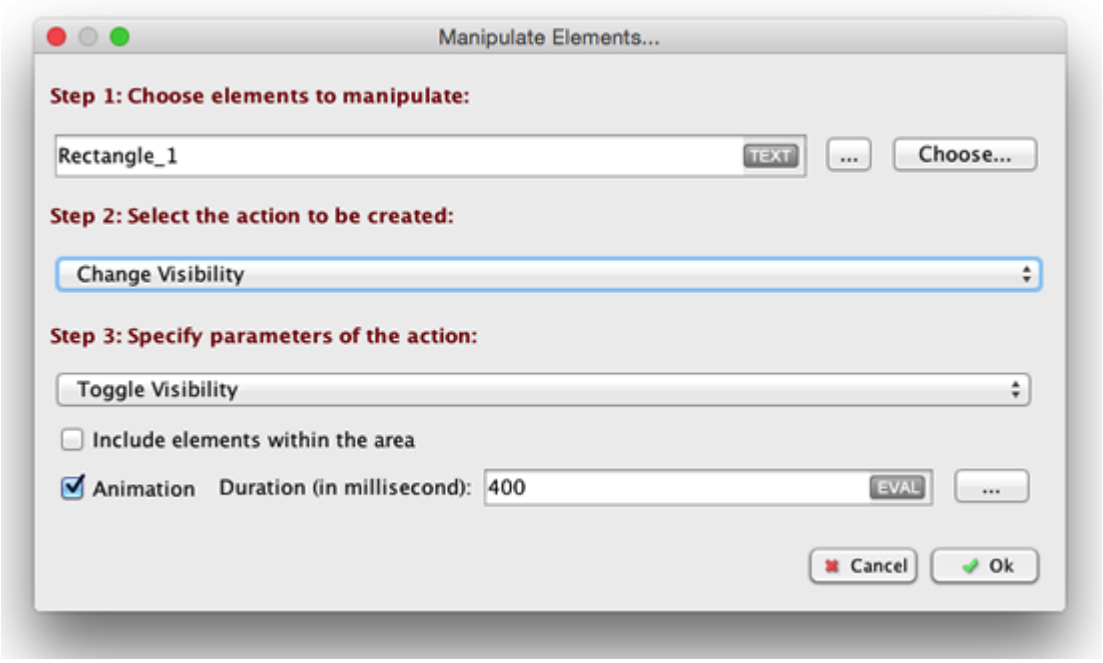
1. Choose the target element to operate on.
2. Select the action to be created.
3. Specify the parameters for the action.



Some actions are commonly available for all elements, such as Change Visibility and Change Location. So you can always see them on step 2, no matter what element you have selected in step 1.

Change Visibility

Change Visibility action can show/hide/toggle visibility for certain element.



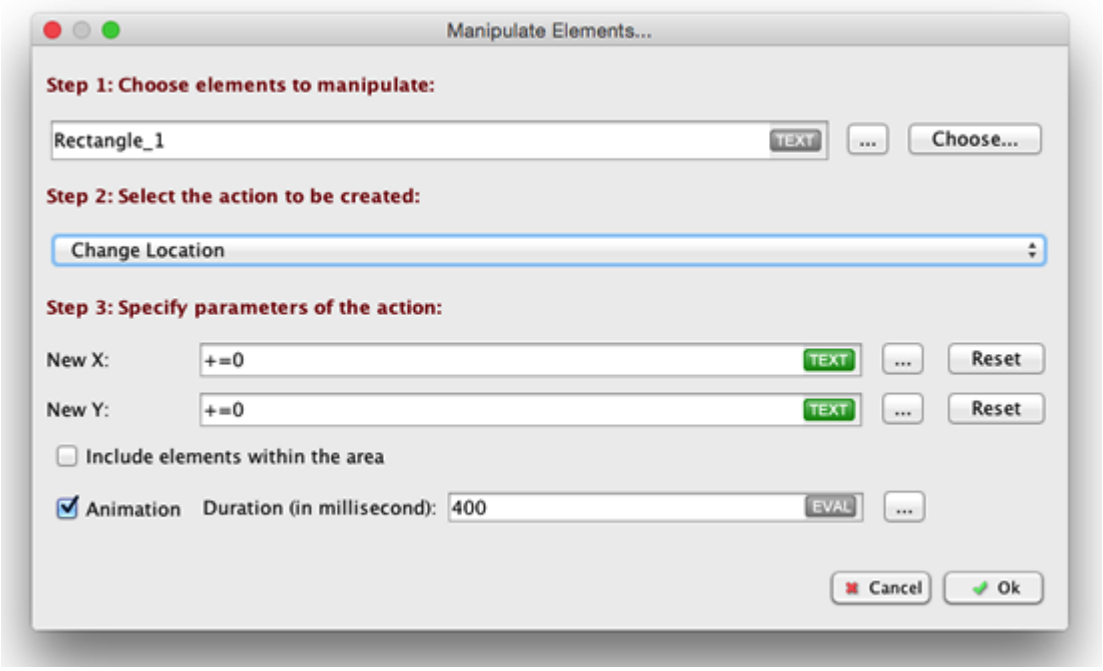
If you select the "Include elements within the area" option. All elements within the element's area will perform the same action, as long as those elements are not covered by the selected element. This will be useful if you wish to show/hide a batch of elements in an area.

At the bottom of the window you can also see an "Animation" option, turning on this option will easily apply animation on the visibility changing. You can also specify the duration of the animation (fade in/fade out).

This action is available for all elements.

Change Location

Change Location action can move certain element to specified position.



If the new value of coordinate starts with "+" or "-", that means it is a relative value. For example, the value of new x is "+=5" means moving the element right for 5 pixels; if the value of new x is "5", means moving the element to position that x=5.

You can use property value in the new x and/or y.

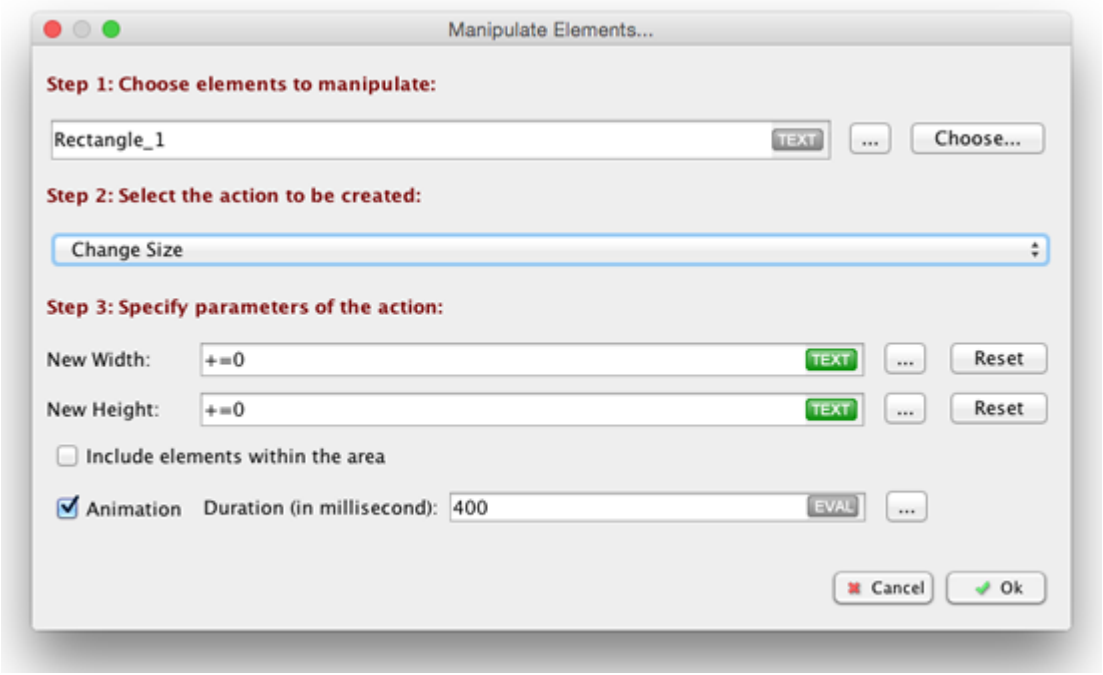
If you select the "Include elements within the area" option. All elements within the element's area will be move as well, as long as those elements are not covered by the selected element. This will be useful if you wish to move a batch of elements in an area.

At the bottom of the window you can also see an "Animation" option, turning on this option will easily apply animation on the moving. You can also specify the duration of the animation.

This action is available for all elements.

Change Size

Change Size action can change size of selected element.



If the new value of width/height starts with "+" or "-", that means it is a relative value. For example, the value of new width is "+=50" means horizontal enlarge the element for 50 pixels; if the value of new width is "50", means setting the element width to 50 pixels.

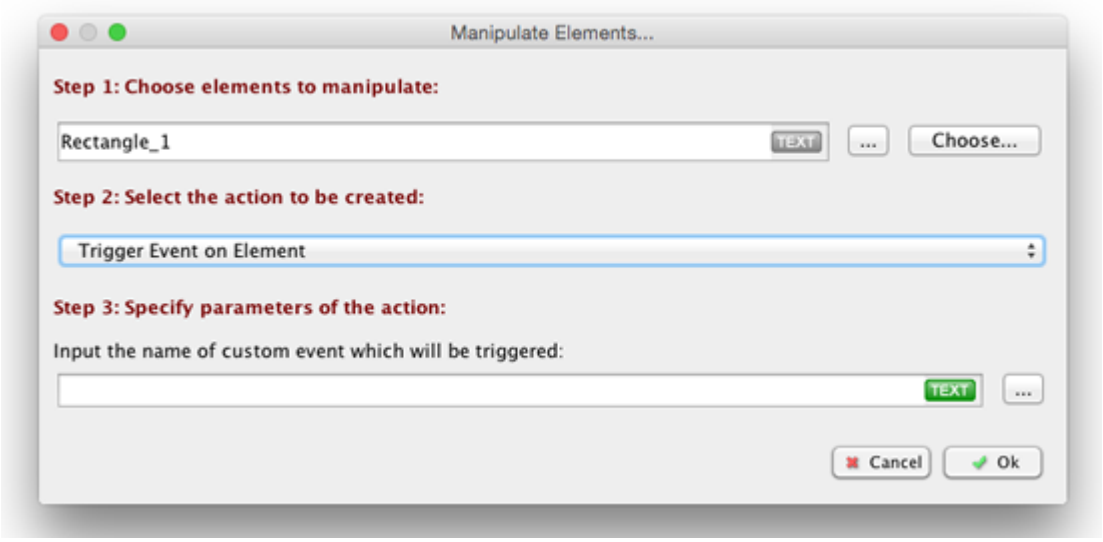
You can use property value in the new width and/or height.

At the bottom of the window you can also see an "Animation" option, turning on this option will easily apply animation on the resizing. You can also specify the duration of the animation.

This action is available for all elements.

Trigger Event on Element

This action is similar with Trigger Global Event, but it is for non-global custom event only. Only the non-global custom event that defined for specific element will be triggered.

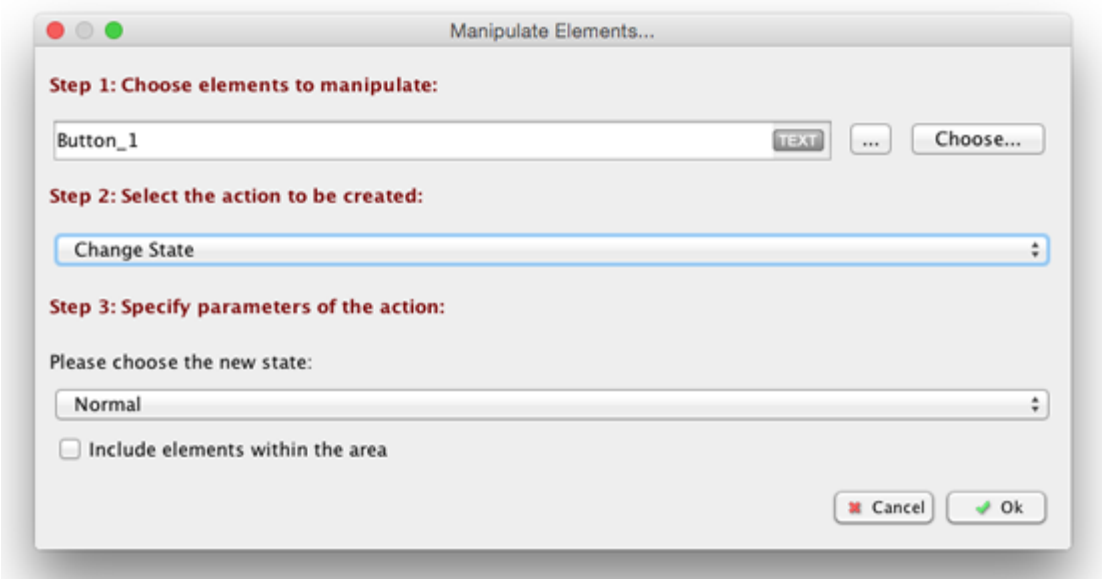


You can use property value in the event name field.

This action is available for all elements.

Change State

Change State action can change the state of one or more elements.



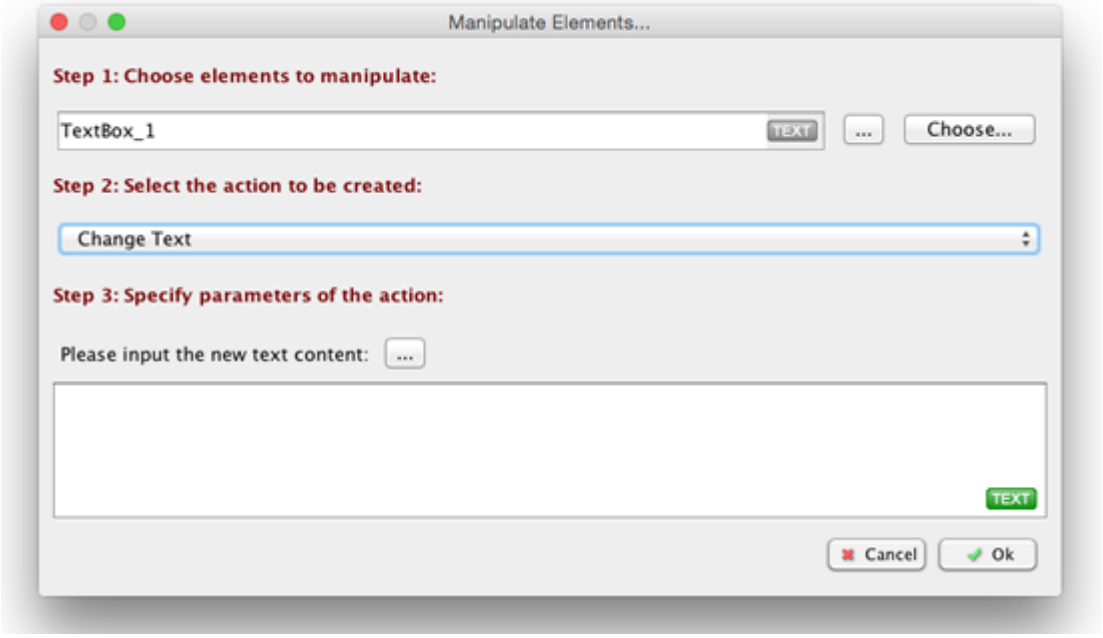
If you select the "Include elements within the area" option. All elements within the element's area will perform the same action, as long as those elements are not covered by the selected element. This will be useful if you wish to change state for a batch of elements in an area.

This action is available for these elements:

Button, CheckBox, ComboBox , Text Label, Hyperlink, Group Frame, List, Menu Bar, Radio Button, Radio Button Group, Scroll Bar, Slider, Stepper, Table, Tabs, Vertical Tabs, Text Edit Box, Tree, Accordion

Change Text

Change Text action can change text content of selected element.



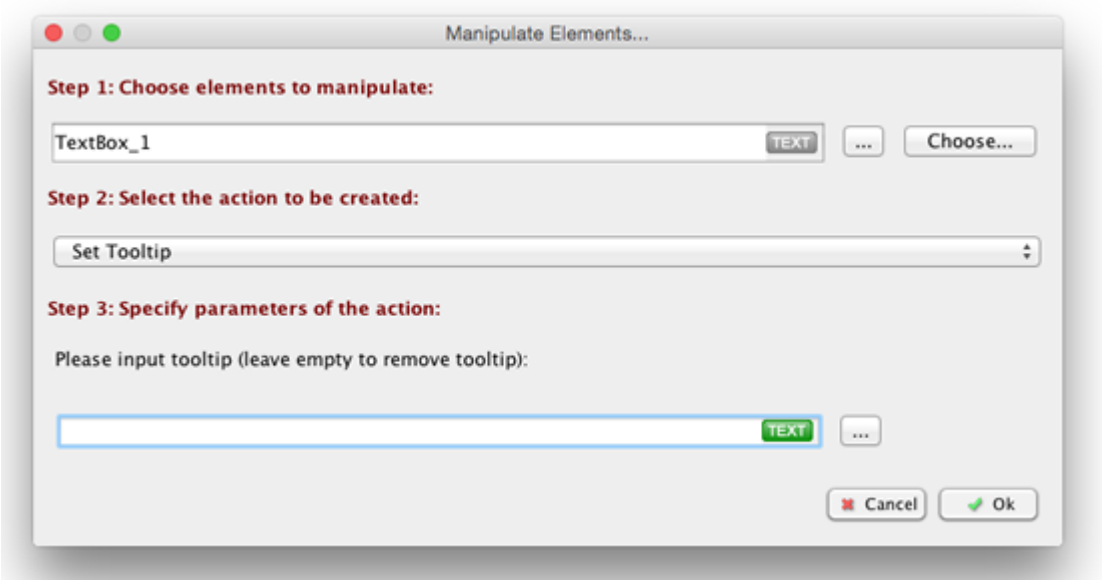
The new value can use property value inside.

This action is available for these elements:

Text Label, Hyperlink, Text Edit Box

Set Tooltip

Set Tooltip action can change tooltip of selected element(s).

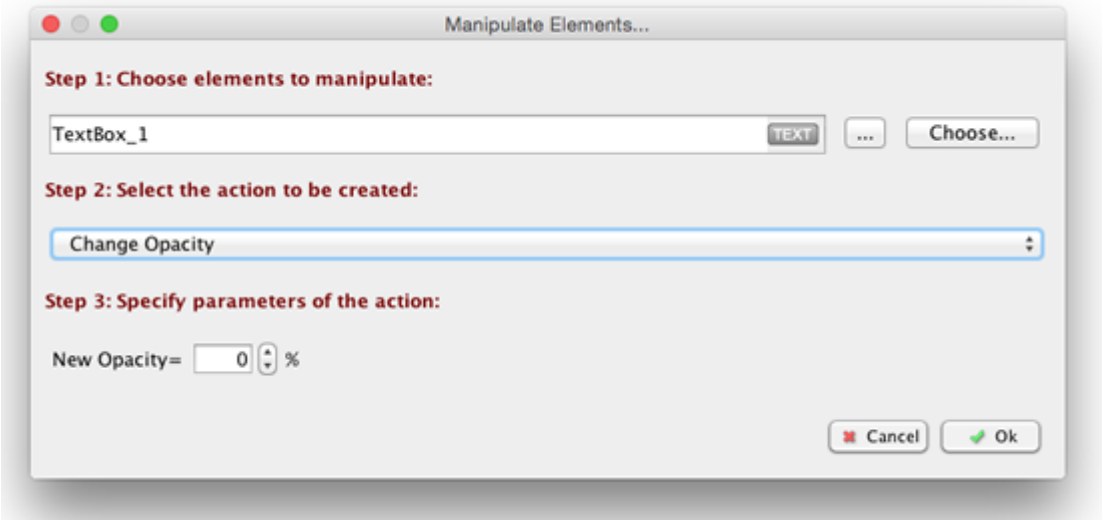


The new tooltip value can use property value inside.

This action is available for all elements, since V3.90.

Change Opacity

Change Opacity action can change the filling opacity of selected element.

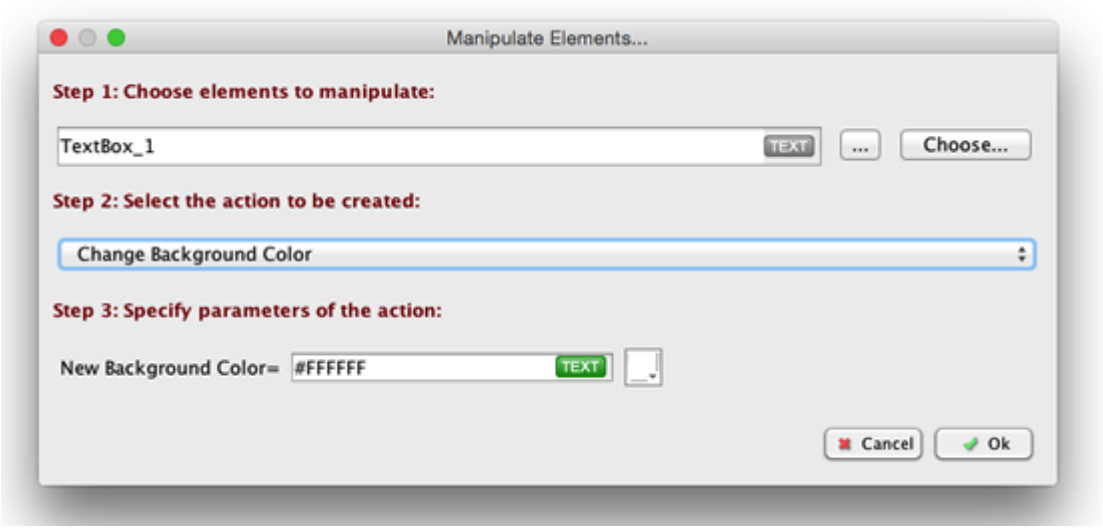


0% opacity means fully transparent (invisible), while 100% opacity means fully opaque.

This action is available for: Text Label, Hyperlink, Rectangle, Ellipse, Triangle, Polygon

Change Background Color

Change Background Color action can change the filling color of selected element.

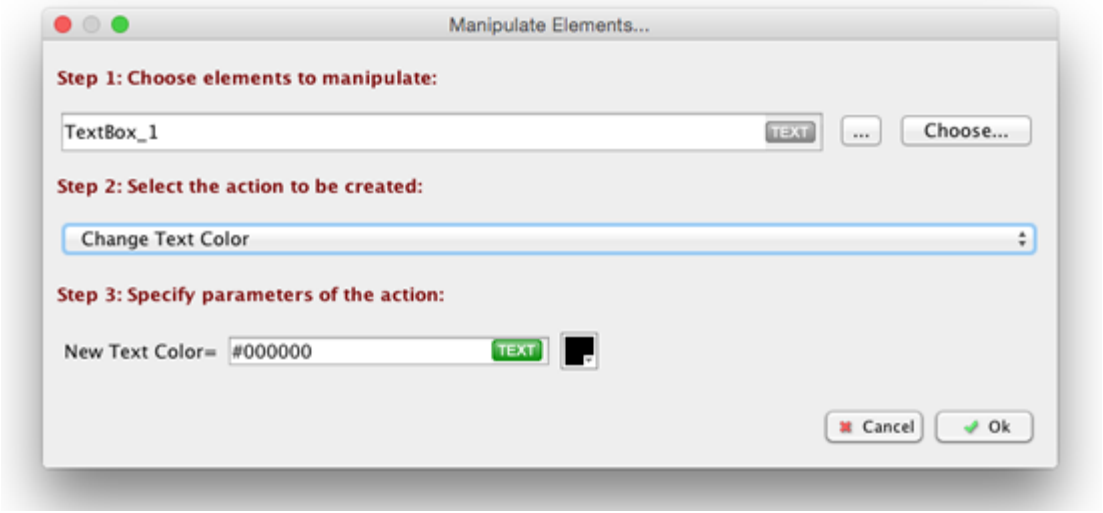


You can input the html color value in the field or click the color picker to pick a color.

This action is available for: Text Label, Hyperlink, Rectangle, Ellipse, Triangle, Polygon

Change Text Color

Change Text Color action can change the text color of selected element.

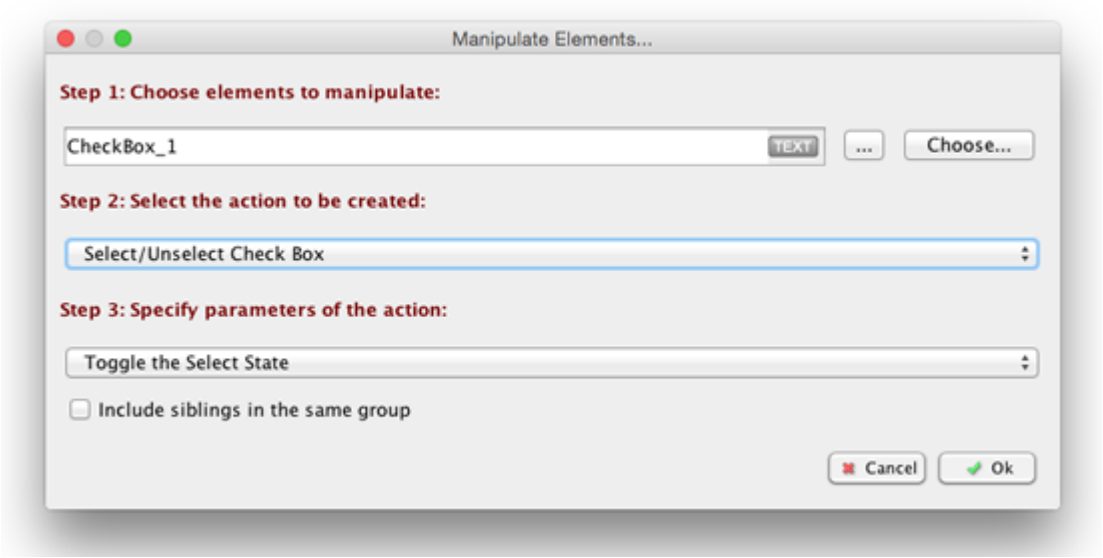


You can input the html color value in the field or click the color picker to pick a color.

This action is available for: Text Label, Hyperlink

Select/Unselect Check Box

Select/Unselect Check Box action can make CheckBox element checked or unchecked.

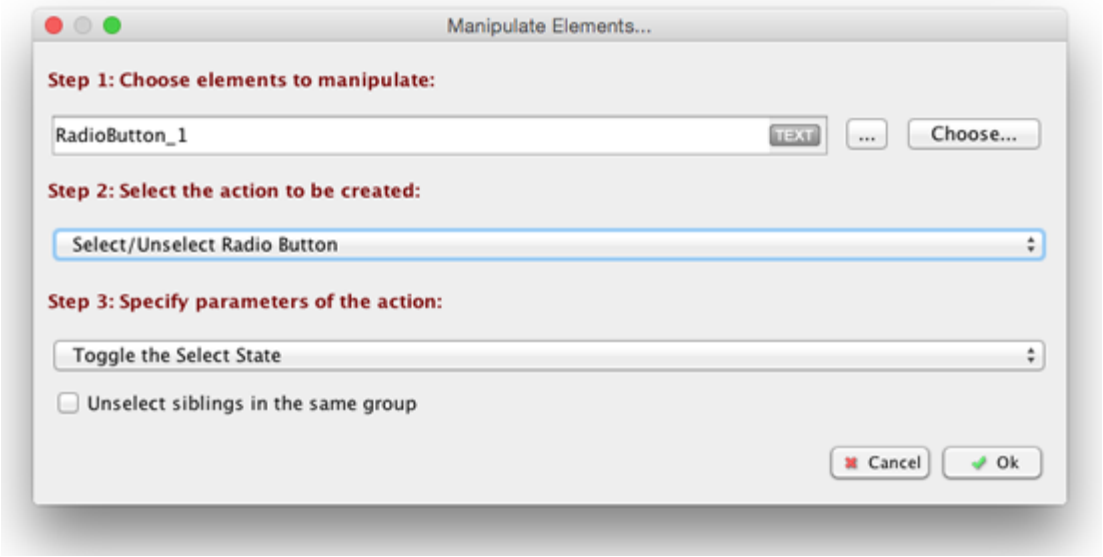


If you select the "Include siblings in the same group" option, all other CheckBox elements in the same Group will perform the same action.

This action is available for: CheckBox

Select/Unselect Radio Button

Select/Unselect Radio Button action can select or unselect the Radio Button element.

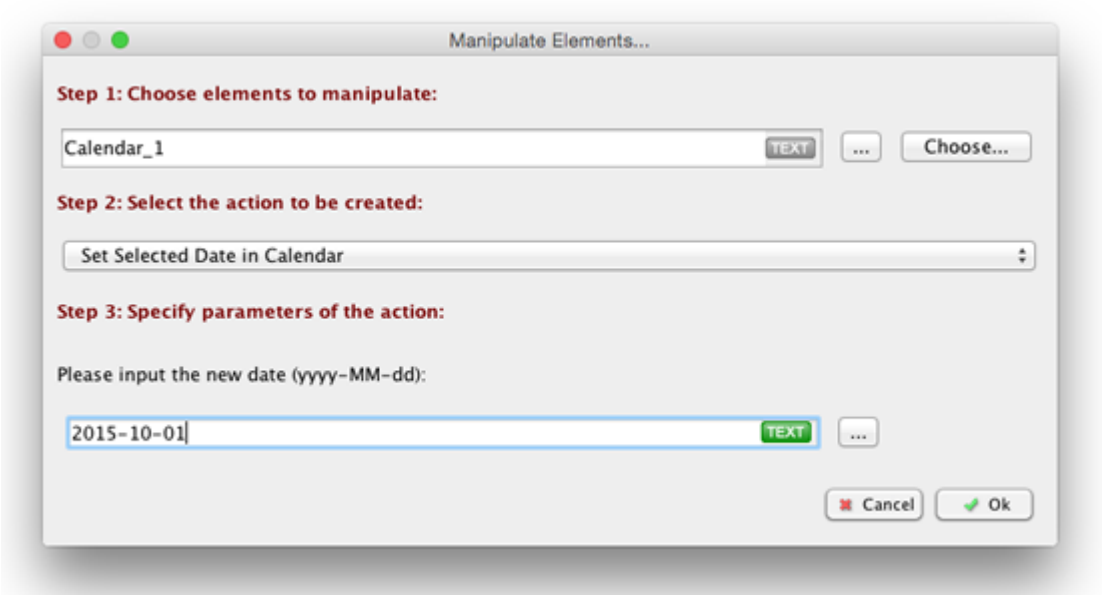


If you select the "Unselect siblings in the same group" option, all other Radio Button elements in the same Group will be unselected at a time.

This action is available for: Radio Button

Set Selected Date in Calendar

Set Selected Date in Calendar action can change data of Calendar element.

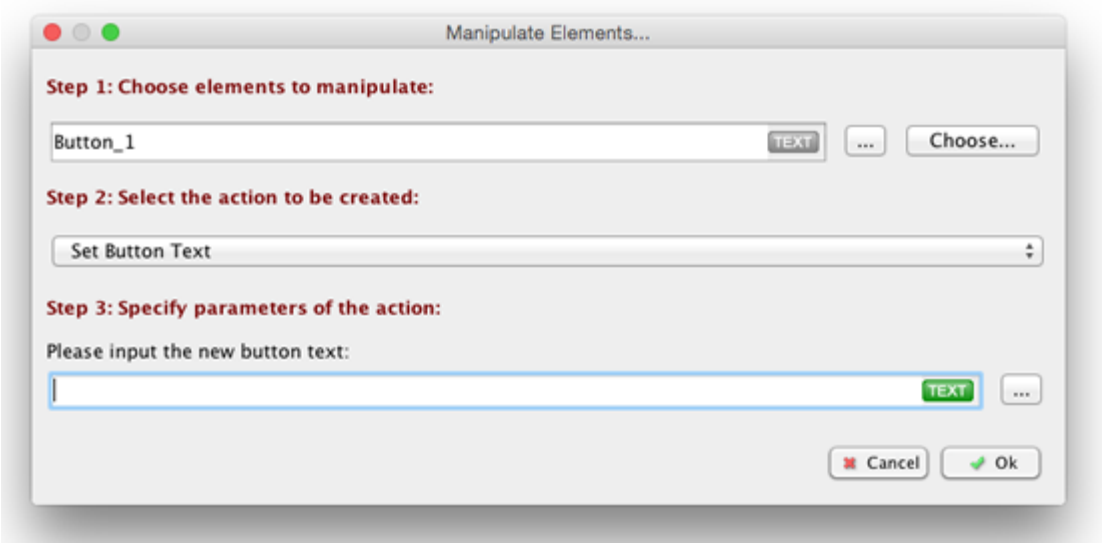


The new date should be "yyyy-MM-dd" format, you can use property values to dynamically generate the new date.

This action is available for: Calendar

Set Button Text

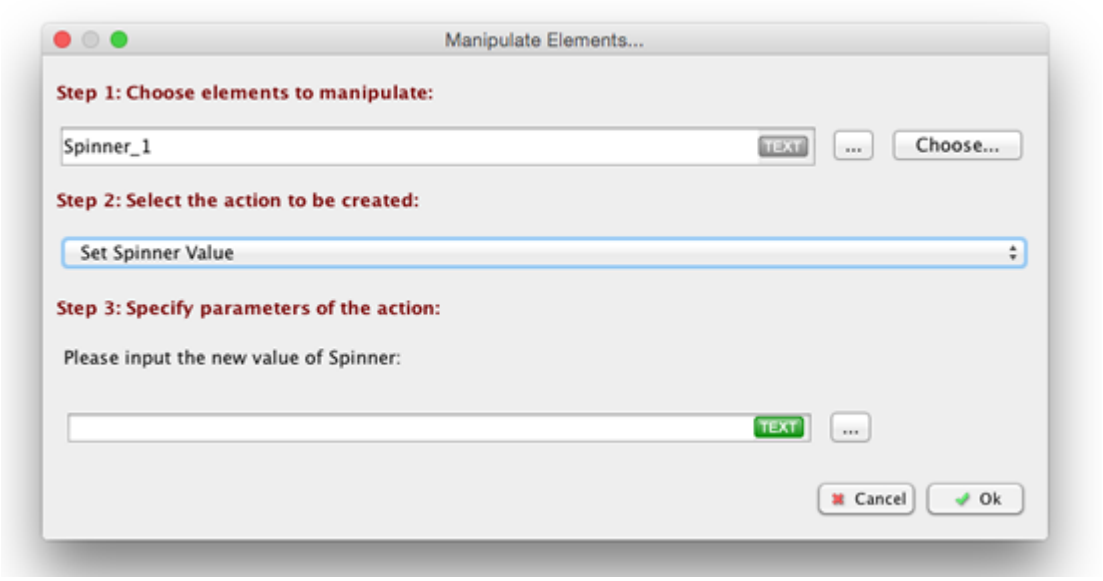
Set Button Text action can change the text label Button element.



This action is available for: Button

Set Stepper (Spinner) Value

Set Spinner Value action can change the current value of Stepper (Spinner) element.

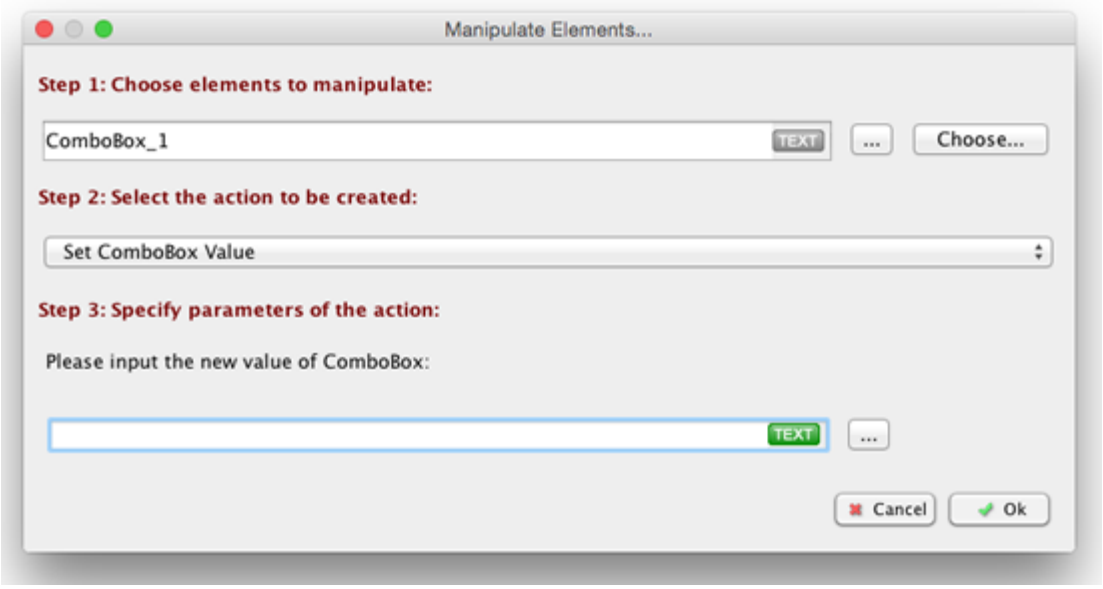


You can use property values to build the new value.

This action is available for: Stepper (Spinner)

Set ComboBox Value

Set ComboBox Value action can set the current value of ComboBox element.

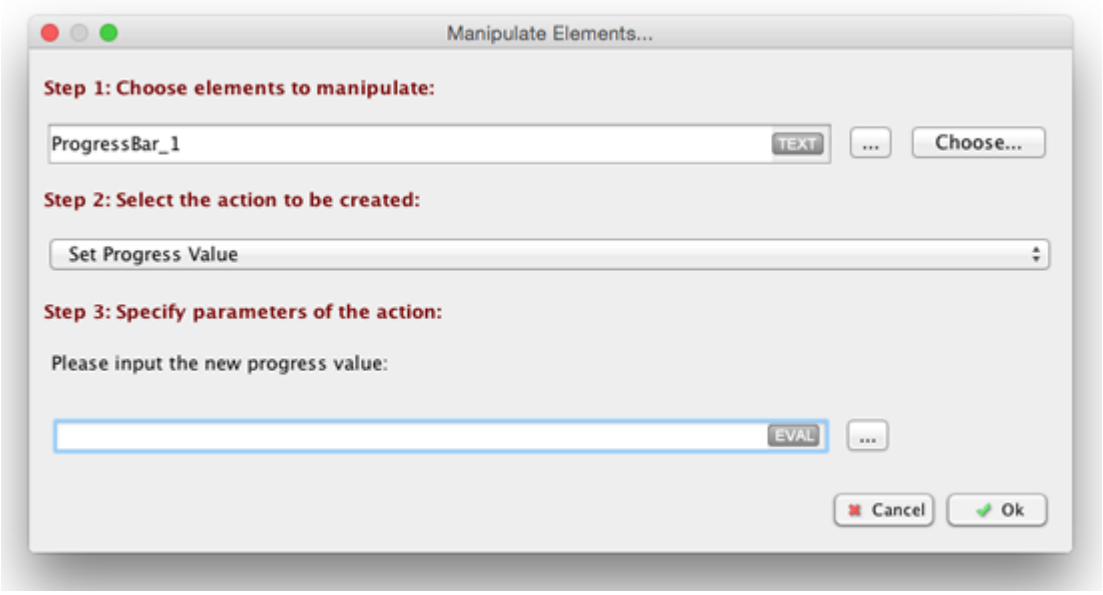


You can use property values to build the new value.

This action is available for: ComboBox

Set Progress Value

Set Progress Value action can change the current value of Progress Bar element.

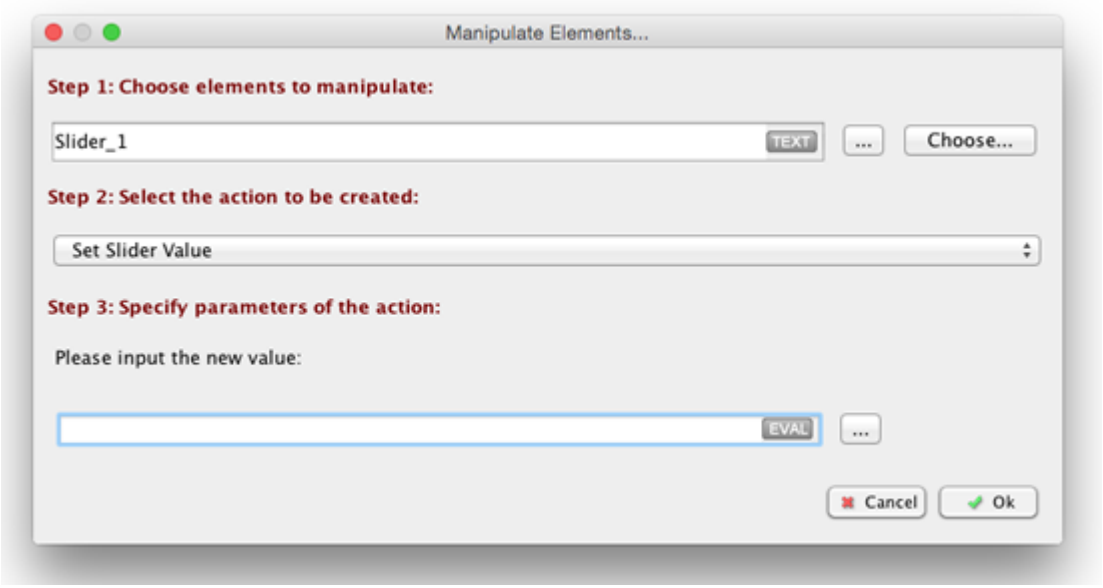


You can use property values to build the new value.

This action is available for: Progress Bar

Set Slider Value

Set Slider Value action can change the current value of Slider element.

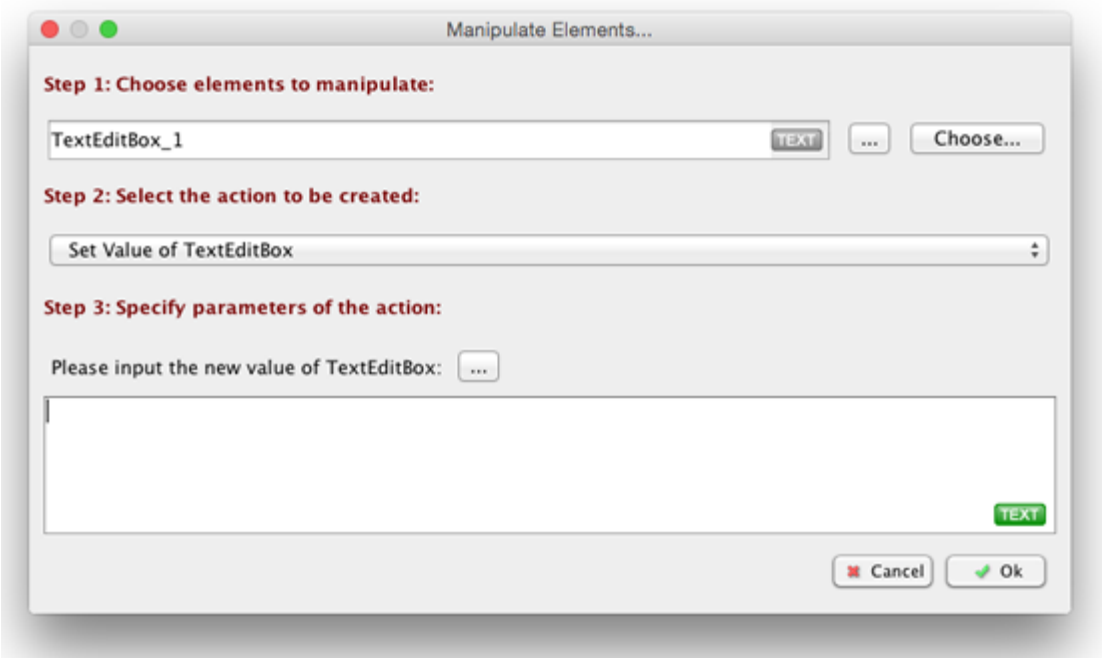


You can use property values to build the new value.

This action is available for: Slider

Set TextEditBox Value

Set Value of TextEditBox action can change the current value of Text Edit Box element.

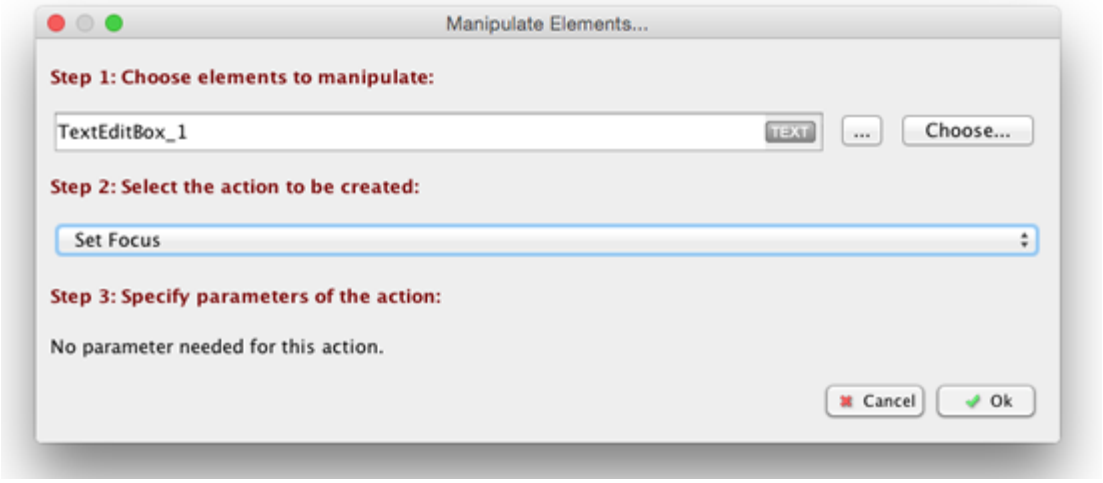


You can use property values to build the new value.

This action is available for: Text Edit Box

Set Focus

Set Focus action will set the current input focus on certain element. The previous focused element will trigger the Focus Lost event and the new focused element will trigger the Focus Gain event.

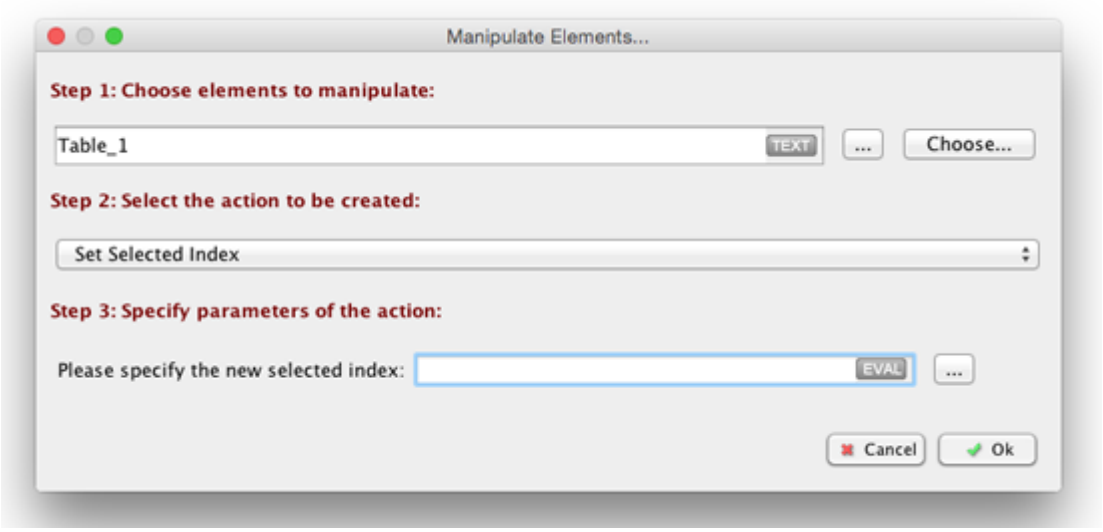


This action is available for these elements:

Stepper (Spinner), Text Edit Box

Set Selected Index

Set Selected Index action will change the current (row) selection of certain element. The Selection Changed event will be triggered after the action performed.



Remarks: the selected index starts from 1. The first item has the index 1. If you set the selected index to 0, the action will clear the selection.

When using this action on table element, please notice that:

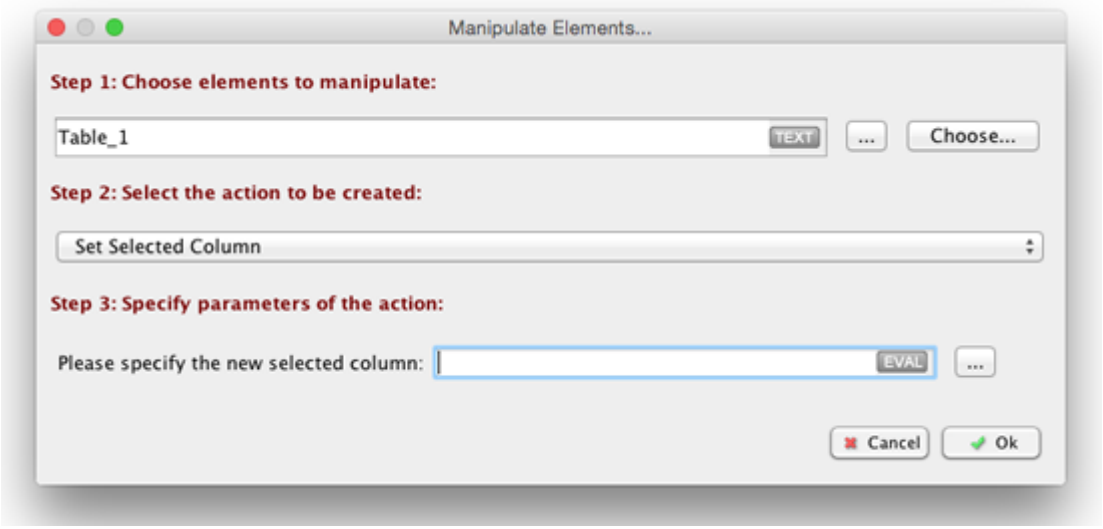
- If the table select mode is set to "Column", this action will not take effect.
- If the table select mode is set to "Cell" , the actual selected cell will be decided by "Selected Index" and "Selected Column" together.

This action is available for these elements:

Radio Button Group, ComboBox, List, Table, Tabs, Vertical Tabs, Tree

Set Selected Column

Set Selected Column action will change the current (column) selection of certain element. The Selection Changed event will be triggered after the action performed.



Remarks: the selected column starts from 1. The first item has the index 1. If you set the selected column to 0, the action will clear the selection.

When using this action on table element, please notice that:

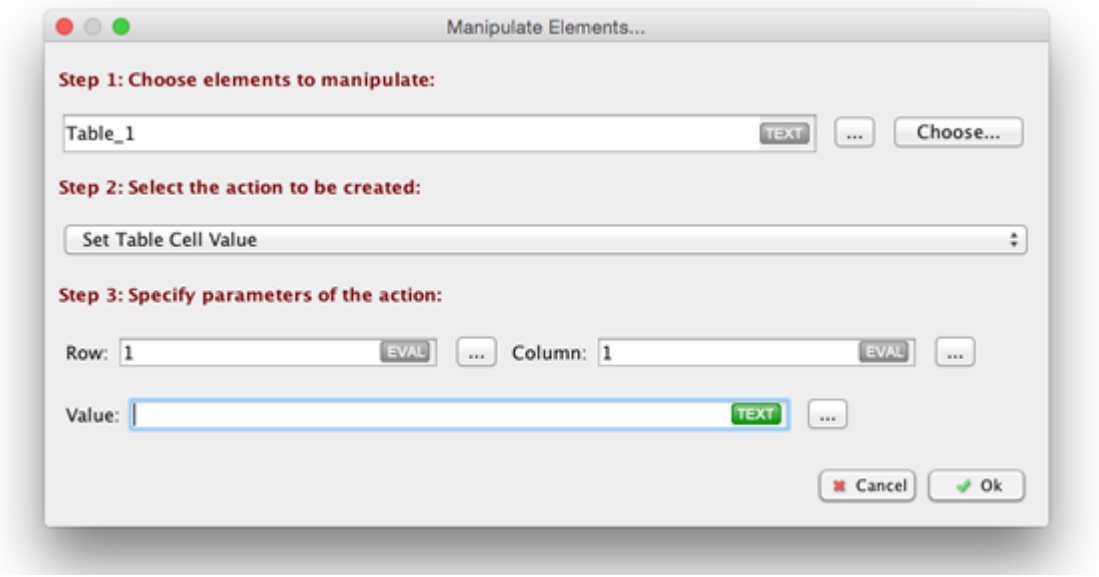
If the table select mode is set to "Row", this action will not take effect.

If the table select mode is set to "Cell" , the actual selected cell will be decided by "Selected Index" and "Selected Column" together.

This action is available for: Table

Set Table Cell Value

Set Table Cell Value action can change the value of certain cell in Table element.

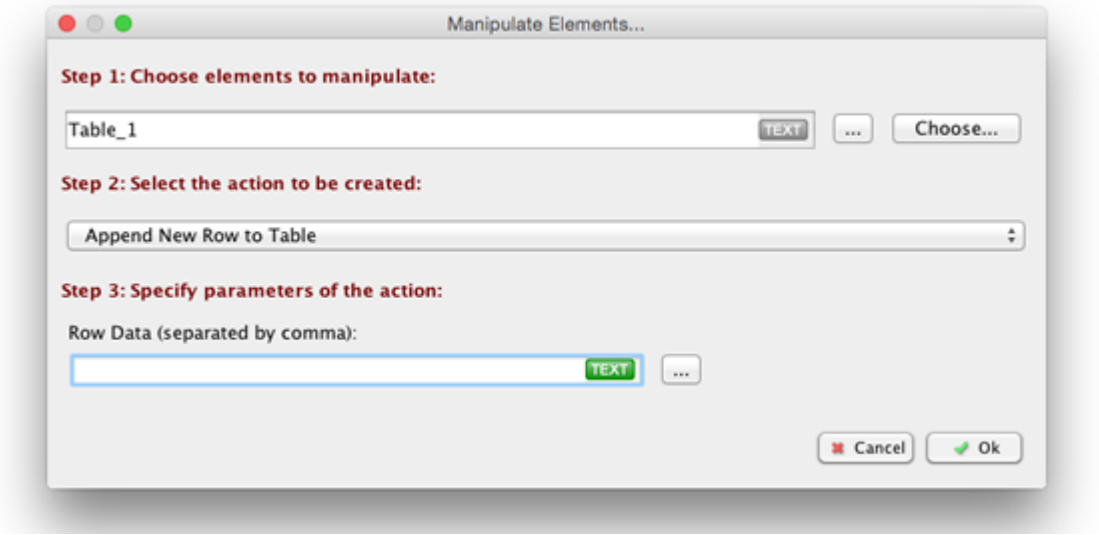


All parameters can accept property values.

This action is available for: Table

Append New Row to Table

Append New Row to Table action will add a new row at the bottom of Table element.



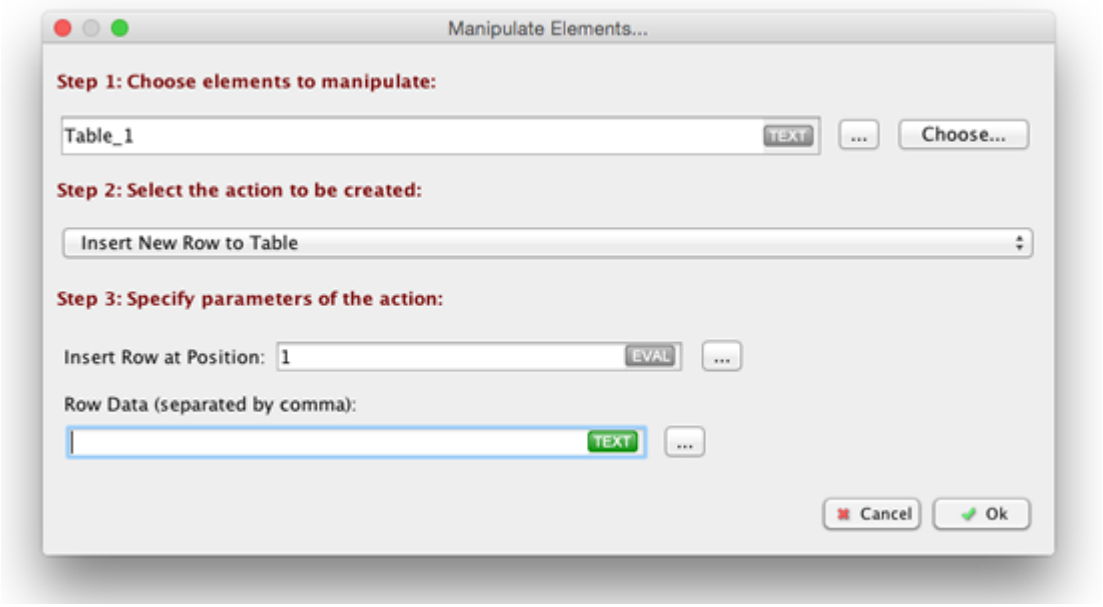
You will need to specify the data for the new row. If the table has multiple columns, you need to separate the data for each column with comma.

The new row data can accept property values. The values for each column should be separated by comma.

This action is available for: Table

Insert New Row to Table

Insert New Row to Table action will insert a new row into Table element, at specified position.



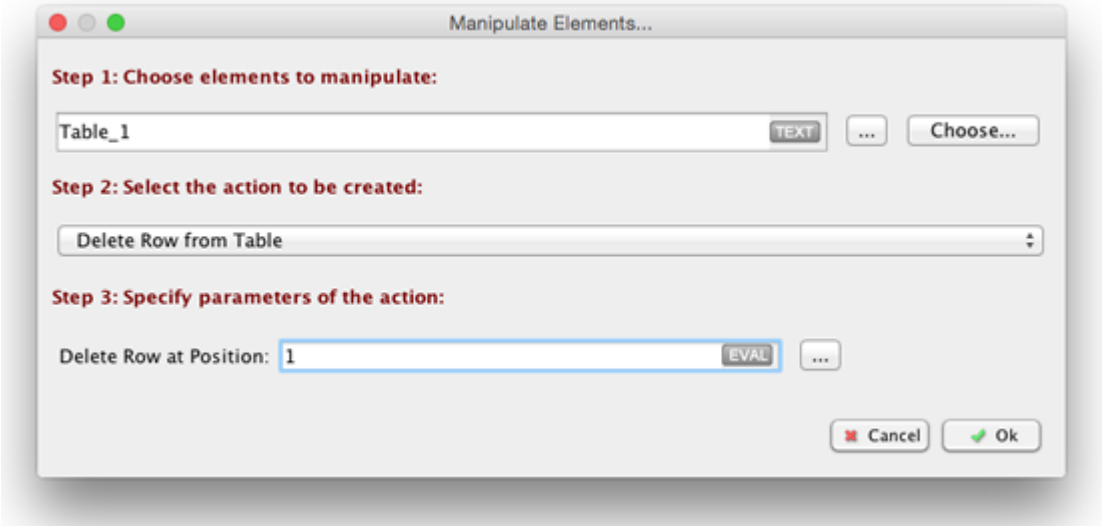
The new row will be inserted before the specified row position. The first row (exclude header) is position 1. You can also use property values to determine the position.

The new row data can accept property values. The values for each column should be separated by comma.

This action is available for: Table

Delete Row from Table

Delete Row from Table action will delete the specified row from Table element.

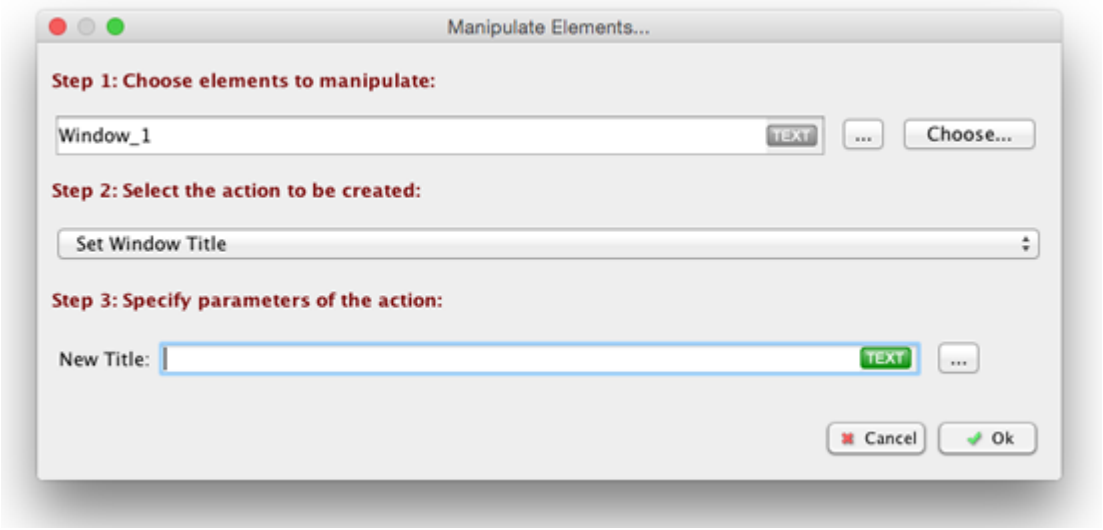


The row at specified row position will be deleted. The first row (exclude header) is position 1. You can also use property values to determine the position.

This action is available for: Table

Set Window Title

Set Window Title action will change the title of Window element.

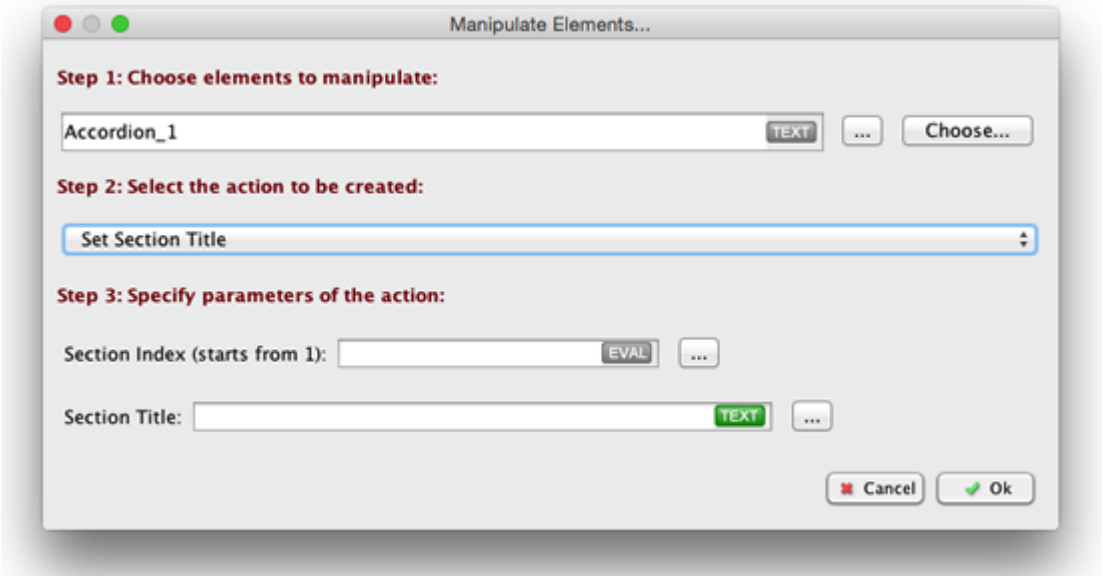


You can use property values to build the new title.

This action is available for: Window

Set Section Title

Set Section Title action can change the title of certain section in Accordion element.



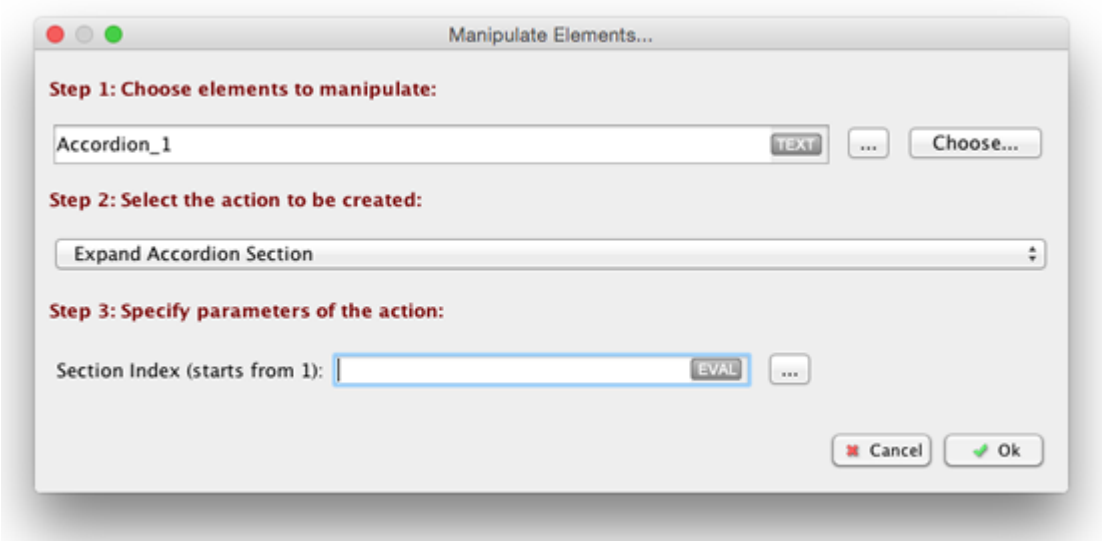
Both the section index and section title fields can accept property values.

Remarks: the section index starts from 1. The first section has the index 1.

This action is available for: Accordion

Expand Section

Expand Section action can expand the specified section of Accordion element.

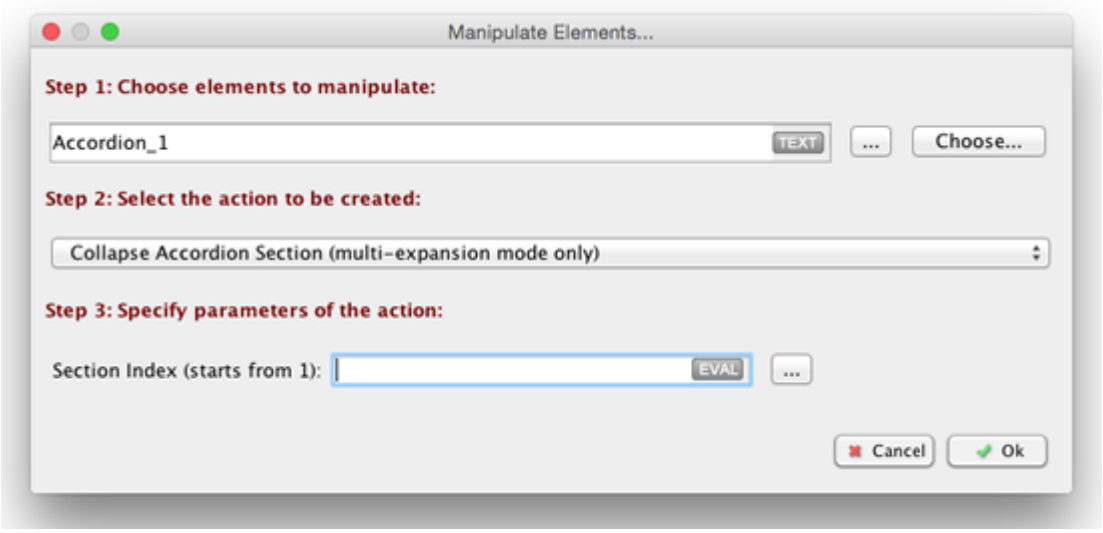


You can use property values to build the index of target section.

This action is available for: Accordion

Collapse Section

Collapse Section action can collapse the specified section of Accordion element.



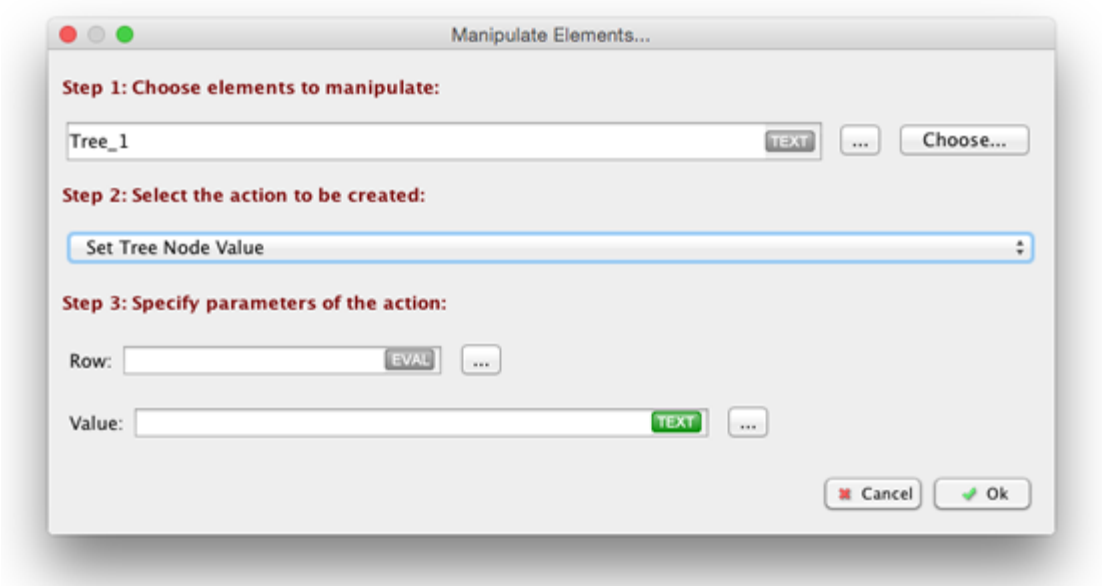
You can use property values to build the index of target section.

Remarks: This action will not take effect in single-expansion Accordion, since the Accordion does not know which section should be opened after closing the specified section.

This action is available for: Accordion

Set Tree Node Value

Set Tree Node Value action can change the node content of Tree element.



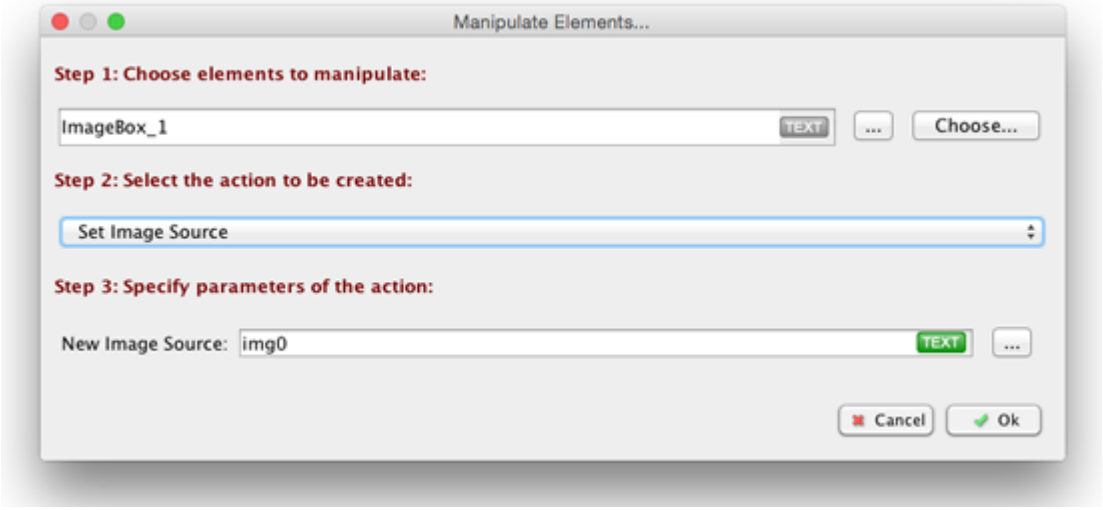
You can use property values to build the row index and the new value of tree node.

Remarks: Although the tree node can be collapsed, its row index will never change.

This action is available for: Tree

Set Image Source

Set Image Source action can change the image source of Image Box element.

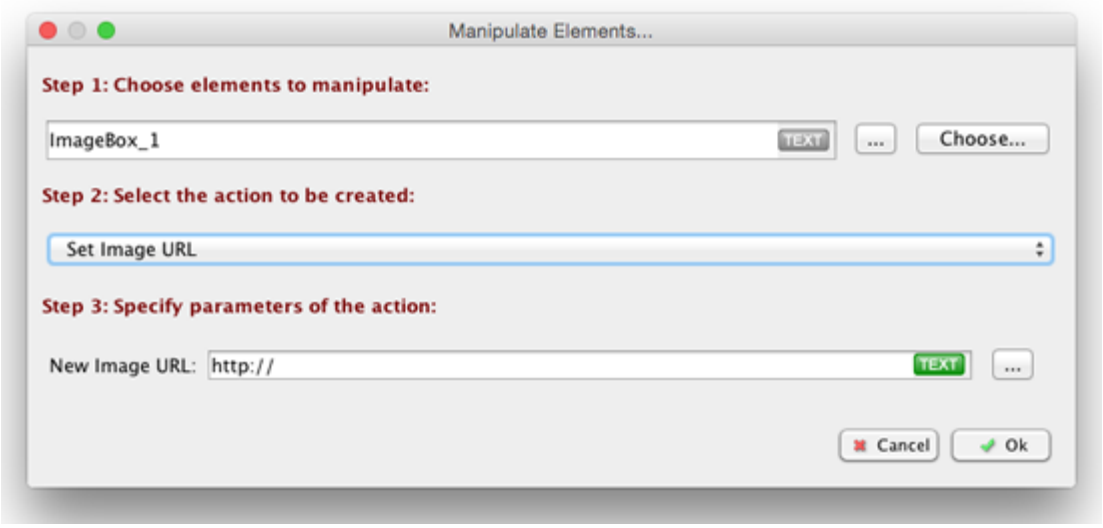


This action only allow specifying the image listed in images panel. If you want to specify an image on Internet, you should use the Set Image URL action.

This action is available for: Image Box

Set Image URL

Set Image URL action can change the image url of Image Box element.

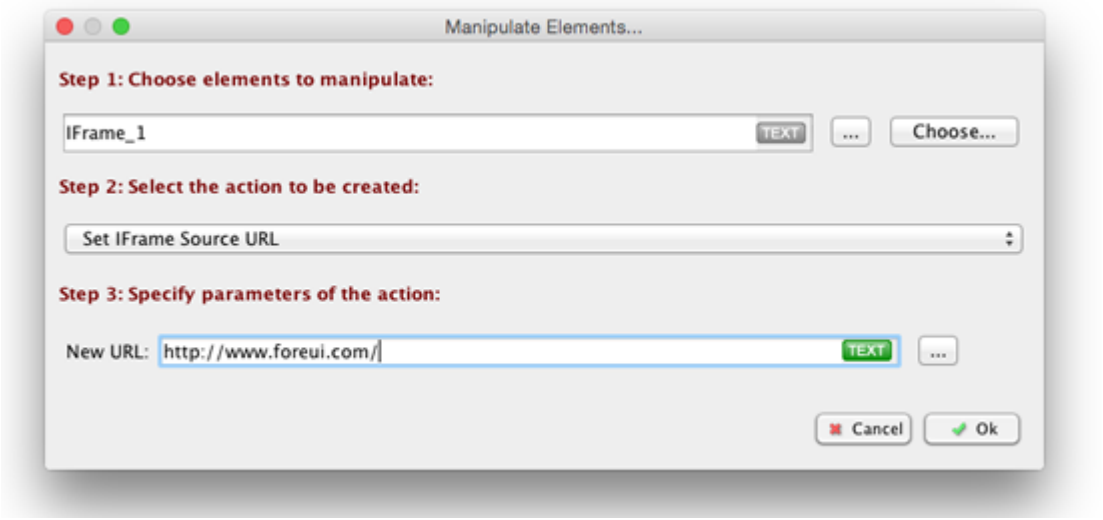


This action can specify the final source URL of the image, so it is possible to specify an image on the Internet. If you want to specify an image listed in the images panel, you should use Set Image Source action instead.

This action is available for: Image Box

Set IFrame Source URL

Set IFrame Source URL action can change the URL of selected iFrame element.



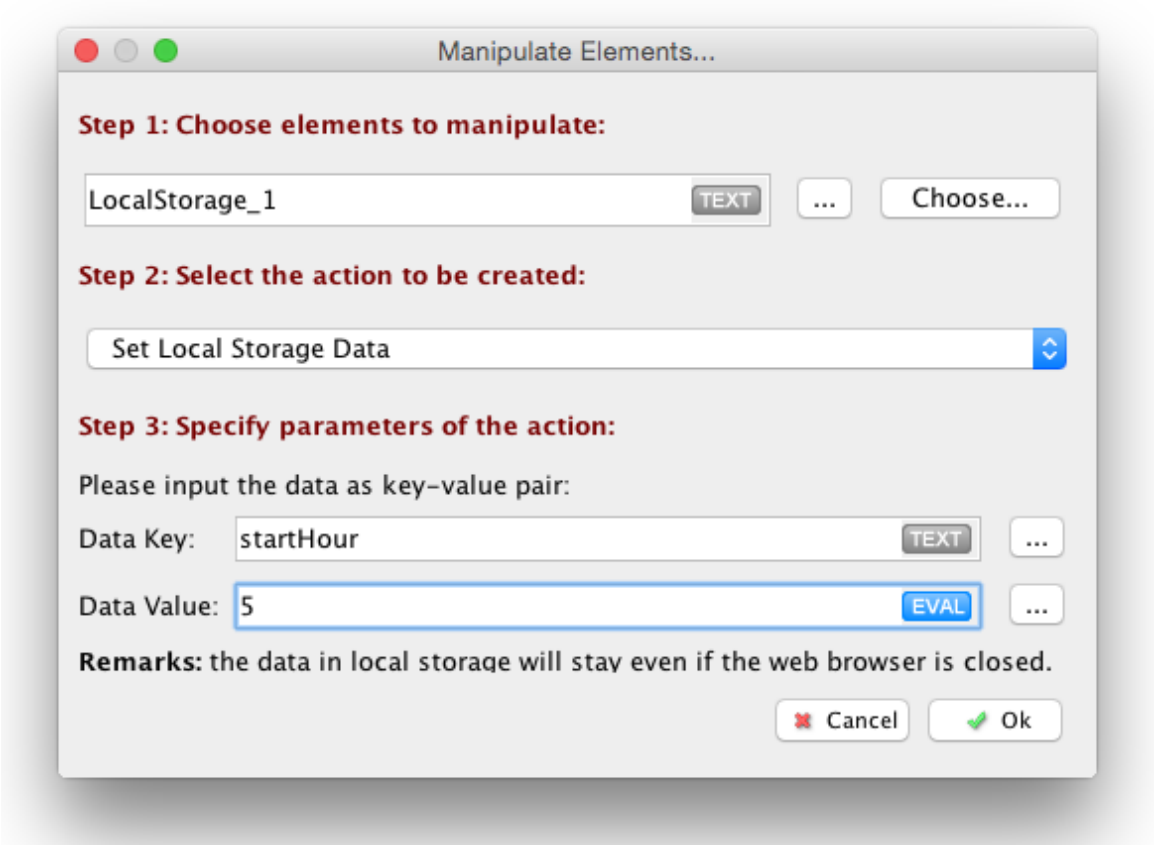
You can use property values to build the new URL value.

This action is available for: iFrame

Set Local Storage Data

Set Local Storage Data action can store data into Local Storage, with given key and value.

Since different local storage element has different namespace, you can use the same key for different data, as long as you store them in different local storage element.



You can use property values to build the key and value too. The key must be a text string, while the value could be number, text string, array or object.

This action is available for: Local Storage

6.5 Properties

Properties are some global variables that can be used in expression. You can learn more about property usage from this section: Use Expression.

There are three kinds of properties:

System Properties

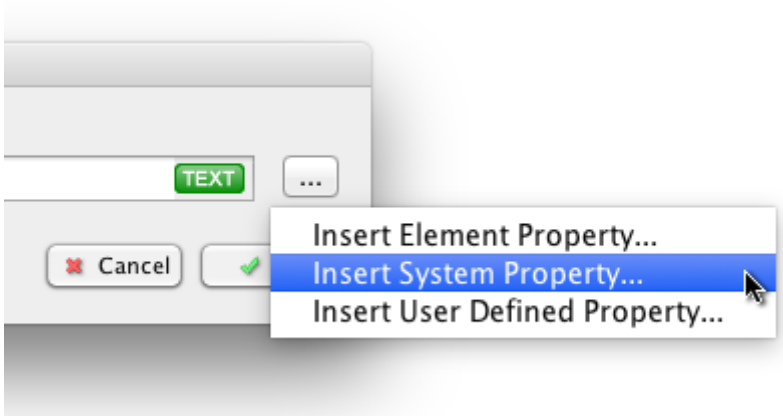
Element Properties

User Defined Properties

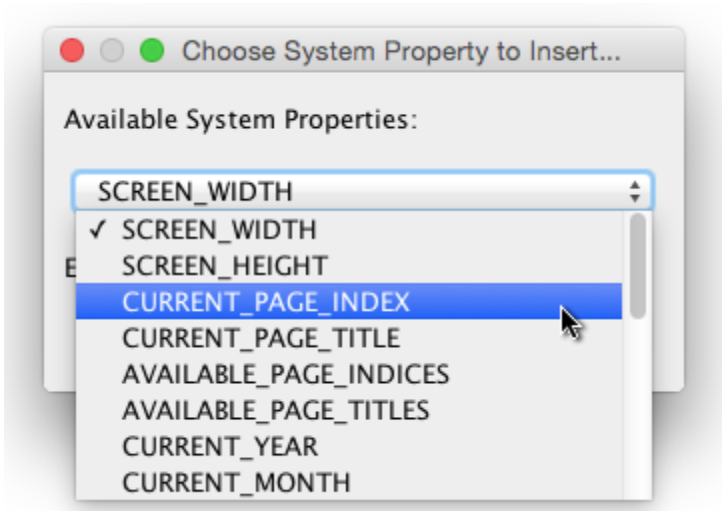
System Properties

ForeUI has some predefined system properties for you to get some information from the environment. The name of the system properties are all capital.

To insert a system property into expression, you can click the ... button beside the input field and choose "Insert System Property..." in popup menu.



Then you can choose the system property from a drop-down list:



Here are all available system properties in ForeUI (you can also review them in the System Property View).

Name	Type	Description	Comment
SCREEN_WIDTH	Number	The width of the screen	"Screen" means the page here
SCREEN_HEIGHT	Number	The height of the screen	"Screen" means the page here
CURRENT_PAGE_INDEX	Number	The index of the current page	Index starts with 1
CURRENT_PAGE_TITLE	String	The title of current page	

AVAILABLE_PAGE_INDICES	Array	Indices for available pages as array	All pages except those in the excluded folders
AVAILABLE_PAGE_TITLES	Array	Titles for available pages as array	All pages except those in the excluded folders
CURRENT_YEAR	Number	Current year	4 digits number. e.g. 2011
CURRENT_MONTH	Object	Current month	{"CURRENT_MONTH"}["full"] contains the full name of the month (e.g. January). {"CURRENT_MONTH"}["short"] contains the short name of the month (e.g. Jan). {"CURRENT_MONTH"}["number"] contains the numeric value of the month (1~12).
CURRENT_DAY	Number	Current day in the month	1-2 digits number
CURRENT_DAY_OF_WEEK	Object	Current day of the week	{"CURRENT_DAY_OF_WEEK"}["full"] contains the full name of the day (e.g. Monday). {"CURRENT_DAY_OF_WEEK"}["short"] contains the short name of the day (e.g. Mon). {"CURRENT_DAY_OF_WEEK"}["index"] contains the numeric index of the day (0~6, Sunday is 0).
CURRENT_HOURS	Number	Current hours in the day	1-2 digits number

CURRENT_MINUTES	Number	Current minutes in the hour	1-2 digits number
CURRENT_SECONDS	Number	Current seconds in the minute	1-2 digits number
CURRENT_TIMESTAMP	Number	Current timestamp	in milliseconds
CURRENT_KEY_CODE	Number	The key code of the current key event	Possible values are listed in Key Code Table
CTRL_KEY_STATE	Number	1 if CTRL is pressed, otherwise 0	
ALT_KEY_STATE	Number	1 if ALT is pressed, otherwise 0	
SHIFT_KEY_STATE	Number	1 if SHIFT is pressed, otherwise 0	
FOCUSED_ELEMENT_ID	String	The Id of focused element	
CURRENT_CURSOR_X	Number	The current X position of cursor	Relative to the page, in pixels
CURRENT_CURSOR_Y	Number	The current Y position of cursor	Relative to the page, in pixels
CURRENT_EVENT	Object	The current handling event.	Event object may contain different attributes, according to the event type.

You can use the value of system property in any place that supports expression. For example, the system property that represent the current cursor x coordinate looks like this in the expression (please notice the difference between TEXT and EVAL parsing modes):

TEXT **{CURRENT_CURSOR_X}** (current x coordinate of cursor, in TEXT parsing mode)

EVAL **{"CURRENT_CURSOR_X"}** (current x coordinate of cursor, in EVAL parsing mode)

If the system property is in Array type, You will need to access its member in this way:

TEXT **{AVAILABLE_PAGE_TITLES}[1]** (the title of first page, in TEXT parsing mode)

EVAL **{"AVAILABLE_PAGE_TITLES"}[1]** (the title of first page, in EVAL parsing mode)

If the system property is in Object type, You will need to access its member in this way:

TEXT **{CURRENT_DAY_OF_WEEK}["full"]** (full name of the current day of the week, in TEXT parsing mode)

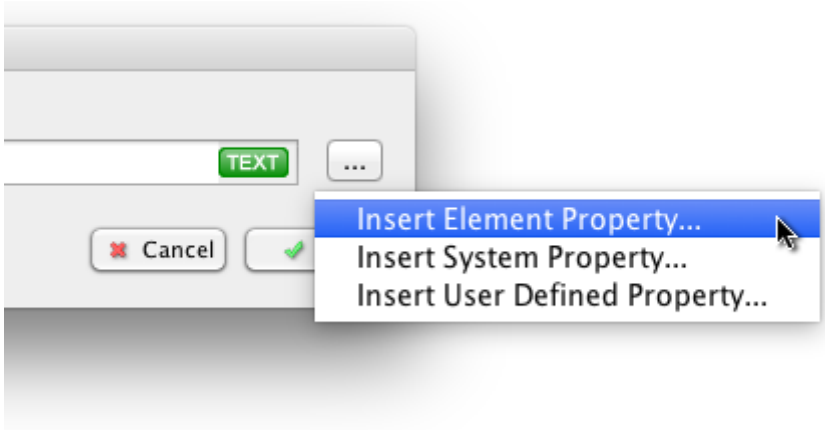
EVAL **{"CURRENT_DAY_OF_WEEK"}["full"]** (full name of the current day of the week, in EVAL parsing mode)

Remarks: system properties are read-only, you could **NOT** change their values with action.

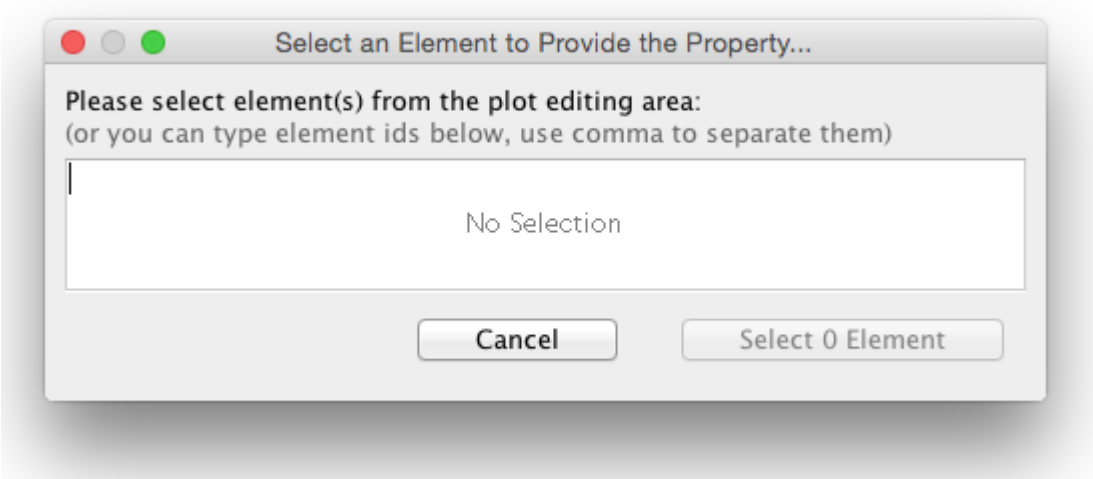
Element Properties

Every element has some properties. These properties are all read-only.

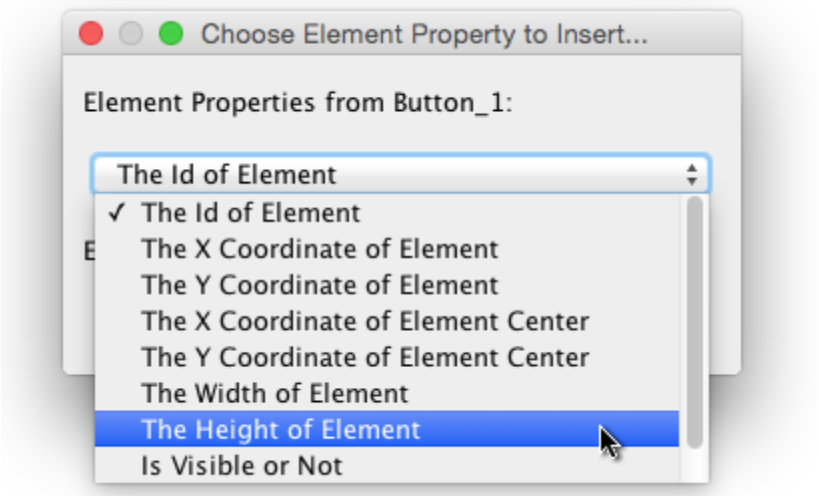
To insert an element property into expression, you can click the ... button beside the input field and choose "Insert Element Property..." in popup menu.



You will be asked to choose an element to provide the property. When the window below shows up, you can click on the editing area to choose element(s). You can also type element id in the input field.



After choosing the element, you can select the property from a drop-down list:



Some properties are commonly available for all elements, such as Id, X Coordinate, Y Coordinate, Width, Height, Visible and Note

Id

The Id property equals to the element's unique id. We can output it for debug purpose.

Format in expression: {Element.id}

Example: {Button_1.id} (will have the value "Button_1")

This property is available for all elements.

X Coordinate

The X Coordinate property is the left position of the element, relative to the page.

Format in expression: {Element.x}

Example: {Button_1.x}

This property is available for all elements.

Y Coordinate

The Y Coordinate property is the top position of the element, relative to the page.

Format in expression: {Element.y}

Example: {Button_1.y}

This property is available for all elements.

Center X Coordinate

The Center X Coordinate property is the left position of the element center, relative to the page.

Format in expression: {Element.cx}

Example: {Button_1.cx}

This property is available for all elements.

Center Y Coordinate

The Center Y Coordinate property is the top position of the element center, relative to the page.

Format in expression: {Element.cy}

Example: {Button_1.cy}

This property is available for all elements.

Width

The Width property is the horizontal size of the element.

Format in expression: {Element.width}

Example: {Button_1.width}

This property is available for all elements.

Height

The Height property is the vertical size of the element.

Format in expression: {Element.height}

Example: {Button_1.height}

This property is available for all elements.

Visible

The Visible property indicate whether the element is visible.

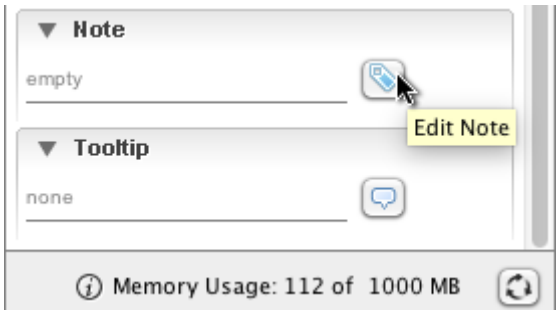
Format in expression: {Element.visible}

Example: {Button_1.visible}

This property is available for all elements.

Note

The Note property equals to the note you have input for the element.



Format in expression: {Element.note}

Example: {Button_1.note}

This property is available for all elements.

Current Value of Text Label/Stepper/Slider/Text Edit Box/Scroll Bar

This property is the current value of the element.

Format in expression: {Element.value}

Example: {TextEditBox_1.value}

This property is available for these elements:

Text Label, Stepper (Spinner), Slider, Text Edit Box, Scroll Bar

Max Value of Slider/Scroll Bar

This property is the max value of the element.

Format in expression: {Element.maxValue}

Example: {Slider_1.maxValue}

This property is available for these elements:

Slider, Scroll Bar

Current Progress Value

This property is the current value of the element.

Format in expression: {Element.currentValue}

Example: {ProgressBar_1.currentValue}

This property is available for these elements:

Progress Bar

Max Progress Value

This property is the max value of the element.

Format in expression: {Element.maxValue}

Example: {ProgressBar_1.maxValue}

This property is available for these elements:

Progress Bar

View Size of Scroll Bar

This property is the view size value of the element.

Format in expression: {Element.viewSize}

Example: {ScrollBar_1.viewSize}

This property is available for these elements:

Scroll Bar

Accordion Section Count

The Accordion Section Count property is the number of sections in Accordion element.

Format in expression: {Element.sectionCount}

Example: {Accordion_1.sectionCount}

This property is available for Accordion

Section Expanded Flags

The Section Expanded Flags property is an array that contains the status of section expansions in Accordion element. The status can be 0 (collapsed) or 1 (expanded).

To get the expand status for certain section, just append "{index}" statement after the property. The array index starts from 1, so the status for first section is {Element.sectionExpandedFlags}{1}.

Syntax:

{Element.sectionExpandedFlags}[index]

Example:

{"Accordion_1.sectionExpandedFlags"}[1] in EVAL mode, or {Accordion_1.sectionExpandedFlags}[1] in TEXT mode.

This property is available for Accordion

Index of First Expanded Section

This property represents the index of the first expanded Accordion element. The index starts from 1. If the property value is 0, means no section is expanded.

Format in expression: {Element.firstExpandedSection}

Example: {Accordion_1.firstExpandedSection}

This property is available for Accordion

Recent Expanded Section

The Recent Expanded Section property is the index of section that is recently expanded in Accordion element.

Format in expression: {Element.recentExpandedSection}

Example: {Accordion_1.recentExpandedSection}

This property is available for Accordion

Recent Collapsed Section

The Recent Collapsed Section property is the index of section that is recently collapsed in Accordion element.

Format in expression: {Element.recentCollapsedSection}

Example: {Accordion_1.recentCollapsedSection}

This property is available for Accordion

Section Titles as Array

The Section Titles as Array property is the titles of all sections in Accordion element.

Format in expression: {Element.sectionTitles}[Index]

Example: {Accordion_1.sectionTitles}[1]

This property is available for Accordion

Whether the CheckBox/Radio Button is Selected

This property indicates whether the CheckBox or Radio Button is selected.

Format in expression: {Element.selected}

Example: {RadioButton_1.selected}

This property is available for these elements:

CheckBox, Radio Button

Current Selected Date in Calendar

This property is a string that represent the current selected date in the Calendar element. The value will be in "yyyy-MM-dd" format.

Format in expression: {Element.dateString}

Example: {Calendar_1.dateString}

This property is available for Calendar

Current Selected Year/Month/Day of Month in Calendar

This property is a number that represent the current selected year/month/day of month in the Calendar element.

Format in expression: {Element.year} {Element.month} {Element.day}

Example: {Calendar_1.year} {Calendar_1.month} {Calendar_1.day}

This property is available for Calendar

Index of the Selected Item

This property is the index of current selected item in the element.

Format in expression: {Element.selectedIndex}

Example: {List_1.selectedIndex}

This property is available for these elements:

Radio Button Group, ComboBox, List, Menu Bar, Menu, Tabs, Vertical Tabs, Tree

Index of the Selected Table Row

This property is the index of current selected row in the table element.

Format in expression: {Element.selectedIndex}

Example: {Table_1.selectedIndex}

This property is available for these elements:

Table

Index of the Selected Table Column

This property is the index of current selected column in the table element.

Format in expression: {Element.selectedColumn}

Example: {Table_1.selectedColumn}

This property is available for these elements:

Table

Text of the Selected Item

This property is a string that represent the current selected item in the element.

Format in expression: {Element.selectedText}

Example: {ComboBox_1.selectedText}

This property is available for Radio Button Group, ComboBox, Tree, List

Text Length

This property is a integer that represent the current length of the text in the element.

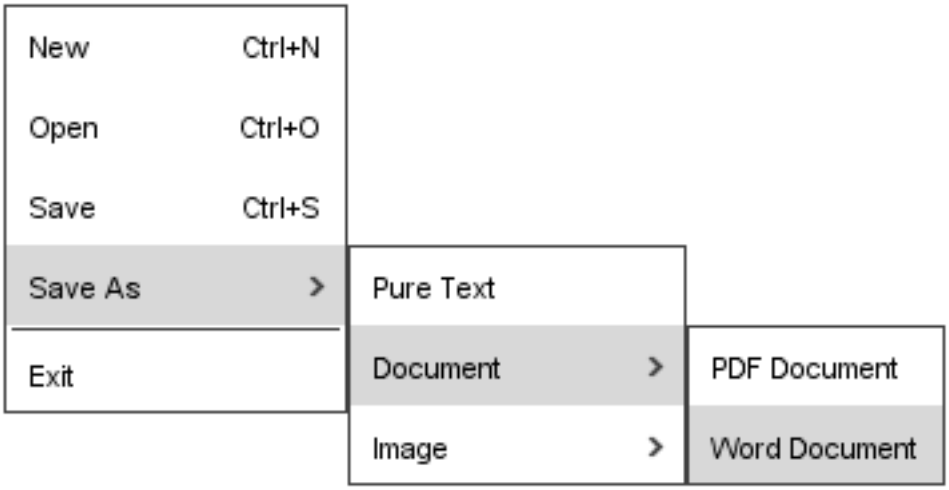
Format in expression: {Element.textLength}

Example: {TextBox_1.textLength}

This property is available for TextBox, TextEditBox

The Identification of the Selection

This property is a string as the unique identification of current selected item in the Multilevel Menu element. The identification will be in "a.b.c" format. For example, if you select the "Word Document" item in the menu showed below, the property value will be "4.2.2".



Format in expression: {Element.selectionIdentification}

Example: {MultilevelMenu_1.selectionIdentification}

This property is available for Multilevel Menu

Table Cell Values as Array[row][column]

The Table Cell Values property is a two dimensions array that contains the cell values of Table element. The array index starts from 1. The header row is not taken into account.

Format in expression: {Element.cellValues}[row][col]

Example: {Table_1.cellValues}[1][1] (This represent the value for the cell in first row, first column)

This property is available for Table

Table Row Count (Header excluded)

This property is the number of rows in the Table element. The header row is not taken into account.

Format in expression: {Element.rowCount}

Example: {Table_1.rowCount}

This property is available for Table

Table Column Count

This property is the number of columns in the Table element.

Format in expression: {Element.columnCount}

Example: {Table_1.columnCount}

This property is available for Table

Stored Data as Associative Array

This property is a associative array that contains the data stored in Local Storage element. The data type of value could be number, string, object or array, and the data type will be decided when you store the data into the local storage element.

To get specified data with given key, just append "[key]" statement after the property. The key is a text string and it will need to be quoted with double quote marks when the expression is in EVAL mode. For example, if you stored the data with key "sky" into LocalStorage_1 element, you can retrieve this value with {"LocalStorage_1.data"}["sky"].

Syntax:

{Element.data}[key]

Example:

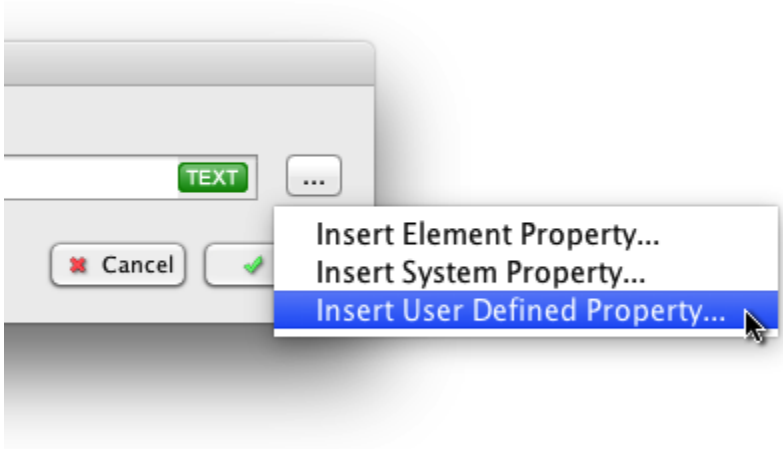
{"LocalStorage_1.data"}["key"] in EVAL mode, or {LocalStorage_1.data}[key] in TEXT mode.

This property is available for Local Storage

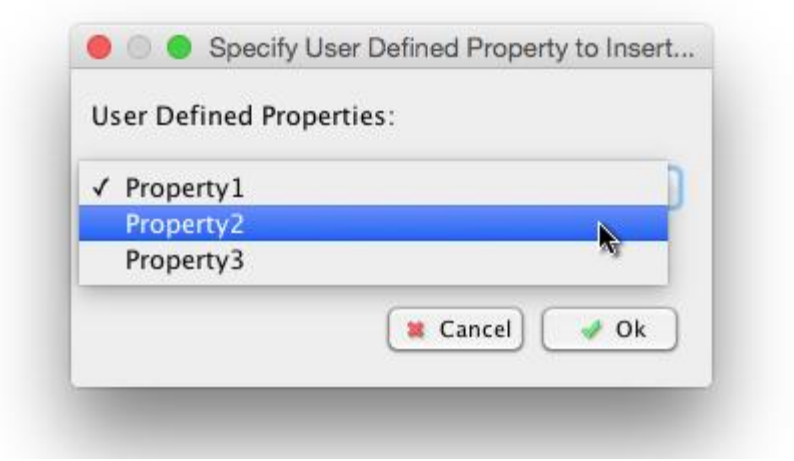
User Defined Properties

User defined properties are very helpful to store some simple data (a string or number). You can define your own properties in the Custom Property View, or define it dynamically via the Set Global Property action.

To insert an element property into expression, you can click the ... button beside the input field and choose "Insert User Defined Property..." in popup menu.



Then you can choose the user defined property from a drop-down list. You will see all previously defined properties here:



If you need to store data structure with 1 dimension (array) or 2 dimensions (table), you may consider storing them with an (invisible) Table element. You can use Set Table Cell Value, Append New Row to Table, Insert New Row to Table and Delete Row from Table actions to manage the data, and use the Table Row Count and Table Cell Values properties to retrieve your data.