

## FLIP SWITCH

### msFlipswitch :

The jQuery Switch is also known as a toggle button and allows you to select between two exclusive choices.

Switch in a basic scenario created from an input element, displaying **Yes** and **No** options.

Properties of flipswitch are mentioned below,

- ✓ Id\*
- ✓ Savefield\*
- ✓ Loop

### ID\*:

It specifies the input id of a particular widget. Each and every widget must have an UNIQUE id. Id should accept only the numbers, alphabets and underscore.

It does not allow to type special characters in an ID and not allow to type negative integers.

Mandatory : TRUE

#### Example:

id - Test\_123 (Will work)

id - Test-@@ (Will not accept as a widget id)

id = -123 or -test (Will not accept as a widget id)

### SAVEFIELD\*:

This savefield property accept only the boolean values like, True or False. It is a mandatory field.

**True** - All the values are saved in database

**False** - Values are not saved in database

### LOOP:

User can use this loop with in the flipswitch then, it should mention the loop id in a particular flipswitch's loop property.

User can mention the loop in ui and rule,

**Syntax:**

**UI --> loop** (Loop id is unique and user defined)

**RULE --> screenid\_widgetid of loop** (Like this mentioned in rule)

Flipswitch widget works within the loop.

**PARENT GROUP:**

Flipswitch widget is also used within the parent group, but should follow some rules like parent group id is properly mentioned in ui and rule.

User can mention the parent group in ui and rule,

**Syntax:**

**UI --> pgroup** (Parent group id is unique and user defined)

**RULE --> screenid\_parentgroupid** (Like this mentioned in rule)

Flipswitch widget will work within the parent group.

**ACTION NAME:**

Actions available for the flipswitch widget in framework 2.0 are mentioned below,

- Mandatory
- Optional
- Show
- Hide
- Enable
- Disable
- Setval

**Mandatory\*:**

User cannot skip this field, if they can try to skip means cannot allowed to submit the form.

Mandatory field is mentioned as red asterisk symbol.

**Syntax:**

APPLY [Mandatory] ON [#screenid\_widgetid]

**Example:**

APPLY [Mandatory] ON [#config\_test\_flipswitch1]

**Optional:**

User can select or they can skip a particular widget, Which is allowed to submit a form.

**Syntax:**

APPLY [Optional] ON [#screenid\_widgetid]

**Example:**

APPLY [Optional] ON [#config\_test\_flipswitch1]

**Show:**

This flipswitch will show/display to the user, so that the user can select the option from the list.

**Syntax:**

APPLY [Show] ON [#screenid\_widgetid]

**Example:**

APPLY [Show] ON [#config\_test\_flipswitch1]

**Hide:**

This rule is to hide a mentioned widget. So it is not visible to the user.

**Syntax:**

APPLY [Hide] ON [#screenid\_widgetid]

**Example:**

APPLY [Hide] ON [#config\_test\_flipswitch1]

**Enable:**

It enables the flipswitch, so the user can able to select the option from the list and an enable action is used to activate the widget from disabled state.

**Syntax:**

APPLY [Enable] ON [#screenid\_widgetid]

**Example:**

APPLY [Enable] ON [#config\_test\_flipswitch1]

## Disable:

It disables the flipswitch widget. The user cannot select option from the list in this disable rule, and they appear blurred.

### Syntax:

```
APPLY [Disable] ON [#screenid_widgetid]
```

### Example:

```
APPLY [Disable] ON [#config_test_flipswitch1]
```

## Setvalue:

1. The user can set the value to "static," which means that when the form is initially opened, it displays the mentioned value in a specific flipswitch.

But this is not constant, user can change/add the value by manually at run time.

### Syntax:

**Setvalue = ""**

```
APPLY [SetValue] ON [#screenid_widgetid] VALUE [""];
```

### Example:

**Setvalue = "1"** (So, this value is reflected in the initial page of the form)

```
APPLY [SetValue] ON [#config_test_flipswitch1] VALUE ["1"];
```

2. User can pass the values through variable,

### Syntax:

```
APPLY [SetValue] ON [$tempvariable] VALUE [""];
```

```
APPLY [SetValue] ON [#screenid_widgetid] VALUE [$tempvariable];
```

### Example:

```
APPLY [SetValue] ON [$test] VALUE ["1"];
```

```
APPLY [SetValue] ON [#config_test_flipswitch1] VALUE [$test];
```

3. User can get values from the another widget,

**Syntax:**

```
APPLY [SetValue] ON [#screenid_widgetid] VALUE [#screenid_widgetid];
```

**Example:**

```
APPLY [SetValue] ON [#config_test_flipswitch1] VALUE [#config_test_Combo2];
```

## EVENT NAME:

Events in rule files are,

- ✓ Load
- ✓ Change

## Load:

This event is should display all the values while loading on an initial page.

**Syntax:**

```
APPLY [Enable] ON [#screenid_widgetid] VALUE ["flipswitch1"];
```

**Example:**

```
FIELD_BEGIN [NAME = "config_test"]  
RULE_BEGIN [NAME = "Initial Load", ORDER = "1"]  
APPLY [Enable] ON [#config_test_1] VALUE ["flipswitch1"];  
RULE_END  
FIELD_END
```

## Change:

Change event is triggered when the value of the widget is changed by the user.

**Syntax:**

```
IF ((#screenid_widgetid != ""))  
BEGIN  
APPLY [Hide] ON [#screenid_widgetid];  
END  
ELSE  
BEGIN  
APPLY [Show] ON [#screenid_widgetid];  
END
```

**Example:**

```
FIELD_BEGIN [NAME = "config_test_1"]  
RULE_BEGIN [NAME = "condition1", ORDER = "1"]  
IF ((#config_test_1 != ""))  
BEGIN  
  APPLY [Hide] ON [#config_test_2];  
END  
ELSE  
BEGIN  
  APPLY [Show] ON [#config_test_2];  
END  
RULE_END  
FIELD_END
```

**ORIENTATION:**

In an application displays the form in 2 ways. They are,

- ✓ Vertical orientation
- ✓ Horizontal orientation

**Vertical orientation:**

An application shows in portrait mode.

**Horizontal orientation:**

An application shows in landscape mode.