# TEXT BOX

## msTextbox :

The TextBox provides a set of default API configuration options that can be set during its initialization such as value, placeholder, and so on.

The TextBox provides a set of options that allow you to customize the appearance of the widget.

Some of the properties of text box are mentioned below,

- ✓ Id*
- ✓ Savefield
- ✓ Maximum length
- ✓ Minimum length
- ✓ Type

## 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

## MAXIMUM LENGTH:

To set the maximum length for a particular text box widget. So that the user can type up to the maximum length characters

User cannot allow to type more than the maximum length.

**Example:**

**Maximum limit - 10** (So that, the user can type up to the maximum length)

## MINIMUM LENGTH:

To set the minimum length of a particular text box widget. So that the user can type the characters at least it should meet the minimum length.

**Example:**

**Minimum length - 2** (User can meet the minimum requirements (i.e) user can type at least 2 characters)

## TYPE:

There are two types of type property,

1. Text
2. E-Mail

## Text:

Text property should accept the alpha numeric values.

**Example:**

**Text - TEST**

## E-Mail:

Email property accept the proper email id alone.

**Example:**

**E-Mail - abc@gmail.com**

## LOOP:

User can use this loop with in the text box then, it should mention the loop id in a particular text area’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)

Text box widget works within the loop.

## PARENT GROUP:

Text box 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)

Text box widget will works within the parent group.

## ACTION NAME:

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

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

## Mandatory*:

User must type the characters. They 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_textbox1]

## 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_textbox1]

## Show:

This widget will show/display to the user, so that the user can type the characters.

**Syntax:**

APPLY [Show] ON [#screenid_widgetid]

**Example:**

APPLY [Show] ON [#config_test_textbox1]

## 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_textbox1]

## Enable:

It enables the text box, so the user can type the characters 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_textbox1]

## Disable:

It disables the text box widget.

**Syntax:**

APPLY [Disable] ON [#screenid_widgetid]

**Example:**

APPLY [Disable] ON [#config_test_textbox1]

## Setmaximum:

i) It allows to type up to the maximum length, which is mentioned in the setmaximum rule, using static.

**Syntax:**

APPLY [SetMaximum] ON [#screenid_widgetid] VALUE ["10"];

**Example:**

APPLY [SetMaximum] ON [#config_test_1] VALUE ["10"];

ii) It allows to type up to the maximum length, which is mentioned in the setmaximum rule, send the values through variable.

**Syntax:**

APPLY [SetValue] ON [$tempvariable] VALUE ["10"];

APPLY [SetMaximum] ON [#screenid_widgetid] VALUE [$tempvariable];

**Example:**

APPLY [SetValue] ON [$test] VALUE ["10"];

APPLY [SetMaximum] ON [#config_test_textbox1] VALUE [$test];

iii) It allows to type up to the maximum length, which is mentioned in the setmaximum rule, send the values through an another widget.

**Syntax:**

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

**Example:**

APPLY [SetValue] ON [#config_test_textbox1] VALUE [#config_test_textbox2];

## Setminimum:

i) User can select at least 2 characters in the text area, which is mentioned in the setminimum in rule, through static.

**Syntax:**

APPLY [SetMinimum] ON [#screenid_widgetid] VALUE ["2"];

**Example:**

APPLY [SetMinimum] ON [#config_test_1] VALUE ["2"];

ii) User can select at least 2 characters in the text area, which is mentioned in the setminimum in rule, through variable.

**Syntax:**

APPLY [SetValue] ON [$tempvariable] VALUE ["2"];

APPLY [SetMinimum] ON [#screenid_widgetid] VALUE [$tempvariable];

**Example:**

APPLY [SetValue] ON [$test] VALUE ["2"];

APPLY [SetMinimum] ON [#config_test_textbox1] VALUE [$test];

iii) User can select at least 2 characters in the text area, which is mentioned in the setminimum in rule, through an another widget.

**Syntax:**

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

**Example:**

APPLY [SetValue] ON [#config_test_textbox1] VALUE [#config_test_textbox2];

## Setvalue:

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

**Syntax:**

**Setvalue = “Text”**

**APPLY [SetValue] ON [#screenid_widgetid] VALUE ["text"];**

**Example:**

**Setvalue = “Text”** (So, this value is reflected in the initial page of the form)

**APPLY [SetValue] ON [#config_test_textbox1] VALUE ["text"];**

2. User can pass the values through variable,

**Syntax:**

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

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

**Example:**

APPLY [SetValue] ON [$test] VALUE ["text"];

APPLY [SetValue] ON [#config_test_textbox1] 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_textbox1] VALUE [#config_test_textbox2];

## 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 [SetMinimum] ON [#screenid_widgetid] VALUE ["2"];**

**APPLY [SetMaximum] ON [#screenid_widgetid] VALUE ["10"];**

**Example:**

```
FIELD_BEGIN [NAME = "config_test"]
RULE_BEGIN [NAME = "Initial Load", ORDER = "1"]
APPLY [SetMinimum] ON [#config_test_1] VALUE ["2"];
APPLY [SetMaximum] ON [#config_test_1] VALUE ["10"];
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.