COMBO BOX

msComboBox :

The ComboBox displays a list of values, allowing for making a single selection from that list. The component enables the user to enter custom values through the keyboard.

The Kendo UI for jQuery ComboBox enables you to render a series of two or more ComboBoxes in which each ComboBox is filtered according to the selected options in the previous ComboBox.

Some of the properties of ComboBox are mentioned below,

- ✓ Id*
- ✓ Savefield*
- ✓ Datasource*
- ✓ Text field*
- ✓ Value field*
- ✓ Template field*
- ✓ Cascade from
- ✓ Cascade from field

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

DATASOURCE*:

It displays the list of items which is mentioned in the combo box.

Example:

Datasrc - [{"code":"FRU", "description":"Fruits"}, {"code":"ANI", "description":"Animals"}]

TEXT FIELD*:

It displays the description part in the list of items.

Example:

Datasrc - [{"code":"FRU", **"description":"Fruits"**}, {"code":"ANI", "**description":"Animals"**}]

VALUE FIELD*:

It displays the code part, which is saved in the database.

Example:

Datasrc - [{**"code":"FRU"**, "description":"Fruits"}, {**"code":"ANI"**, "description":"Animals"}]

TEMPLATE FIELD*:

It displays the list in 3 different ways,

1. Text
2. Value
3. Value-Text

TEXT:

It displays only the text.

Example:

Text - Fruits

VALUE:

It displays only the code.

Example:

Value - FRU

VALUE - TEXT:

It displays both the text and value.

Example:

Value-Text : FRU-Fruits

Datasrc - [{**"code":"FRU"**, "description":"Fruits"}, {**"code":"ANI"**, "description":"Animals"}]

CASCADE FROM:

Cascading functionality by defining the cascadeFrom property, pointing to the Id of the parent ComboBox.

Thus, the data in each ComboBox is enabled and filtered based on the selection in the previous ComboBox.

CASCADE FROM FIELD:

The child ComboBox takes the following actions during initialization:

- ✓ Enables the cascading functionality only if the cascadeFrom property is set.
- ✓ Tries to find the parent object.
- ✓ Listens for changes in the parent value.
- ✓ Remains disabled if the parent has no value.

LOOP:

User can use this loop with in the ComboBox then, it should mention the loop id in a particular ComboBox'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)

ComboBox widget works within the loop.

PARENT GROUP:

ComboBox 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)

ComboBox widget will works within the parent group.

ACTION NAME:

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

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

Mandatory*:

User must select a picture. 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_Combo1]

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_Combo1]

Show:

This ComboBox 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_Combo1]

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_Combo1]

Enable:

It enables the ComboBox, 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_Combo1]

Disable:

It disables the ComboBox 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_Combo1]

Setvalue:

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

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

Syntax:

Setvalue = “”

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

Example:

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

APPLY [SetValue] ON [#config_test_Combo1] VALUE ["FRU"];

2. User can pass the values through variable,

Syntax:

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

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

Example:

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

APPLY [SetValue] ON [#config_test_Combo1] 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_Combo1] 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 ["combo1"];

Example:

```
FIELD_BEGIN [NAME = "config_test"]
RULE_BEGIN [NAME = "Initial Load", ORDER = "1"]
APPLY [Enable] ON [#config_test_1] VALUE ["combo1"];
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.