IBMultiCombo
Developer’s Guide

IBMultiCombo Developer's Guide

First Edition/ July 20, 2011
Author/ Jeon Sanggu
Editor/ Jeong Gwangcheon
Publisher/ (주) 아이비 리더스
Inno Biz Leaders Co., Ltd.
152-879 1501, Hanhwa Biz Metro 1-cha, Guro-3 dong, Guro-gu, Seoul, Korea
Tel / 82-2-2621-2080
Fax / 82-2-2621-2088

Edited by Jeon Sanggu, printed by C&G

※ For technical consultation and inquiry about the contents, please contact IB Readers Ltd.
http://www.ibleaders.co.kr

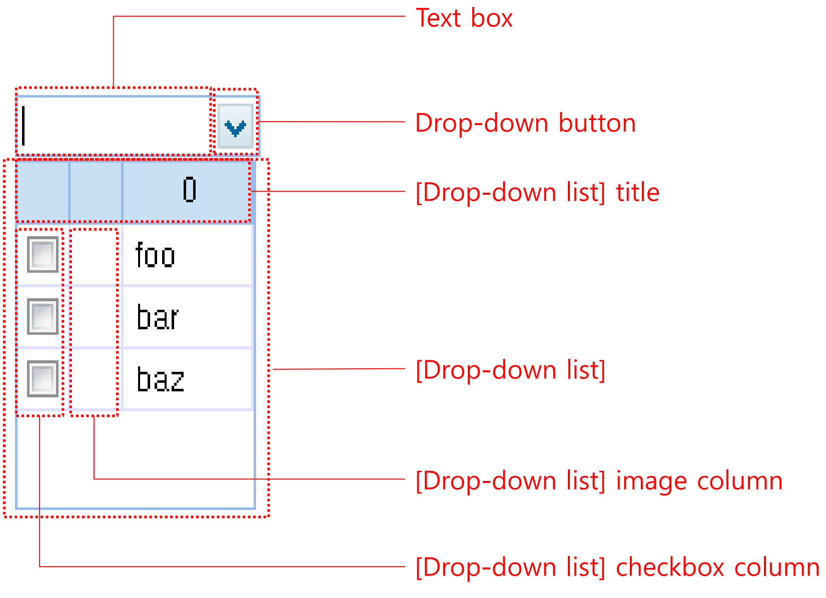
Table of contents

Chapter 1. IBMultiCombo overview

1. IBMultiCombo overview

1.1 Definition of IBMultiCombo

- A combo component that supports cross-browsing and cross-platform.
- No separate development tool is required and functions are implemented using general JavaScript. As a result, web developers (ASP.Net, JSP, PHP, etc.) can implement functions with easy understanding and application.
- A component that makes the Select tag function of the HTML stronger and more flexible.
- Providing stronger combo functions while resolving the limitations of the HTML.
- Providing an easy development environment for developers, and convenient operation methods to operators.
- IBMultiCombo has the following basic structure.



1.2 Major functions of IBMultiCombo

- **Page design convenience function**

  If the text in Option is too lengthy for the HTML Select tag, it is not suitable for page design because the control itself appears wide. IBMultiCombo provides page design conveniently by setting control width differently from the expanded width of the combo item.

- **Multi-column function**

  An item can be configured with both a single column and multiple columns.

- **Multi-selection function**

  Both single selection and multiple selection are supported.

- **Title display function**

  A title is provided for a single column or multiple columns, so that the information of the pertinent item can be checked.

- **Image display function**

  An image column is separately provided, while displaying text.

- **Function of applying default layout style sheet**

  A style sheet for common design works according to the HTML standard

Chapter 2. Setting the development environment of IBMultiCombo

2. Setting the development environment of IBMultiCombo

2.1 Installing IBMultiCombo in a server

2.1.1 Deployment files and contents

To create a development environment using IBMultiCombo, copy and install the provided files as described below in a specific folder of the web server.

The location of the files is as follows.

\Combo\

\Combo\ibmulticombo.js

\Combo\ibmulticomboinfo.js

\Combo\ibmulticombo.cfg

\Combo\Main\ibmulticombo.css

\Combo\Main\c_combo2.gif

The below table describes the content of each file.

File name	Content/Roles
ibmulticombo.js	IBMultiCombo engine js file
ibmulticomboinfo.js	The file that contains functions needed to initialize the combo component, and js functions that can be conveniently used for development (to be updated continuously).
ibmulticombo.cfg	The file used to define attribute values that will be globally set for all combo components.
ibmulticombo.css	Default StyleSheet file of the combo component
c_combo2.gif	Default button image of the combo component

Chapter 3. Development procedure

3. Development procedure

3.1 Setting a JS file link in the web page

Set a link in the js file to create an IBMultiCombo object and perform various functions in the created object.

```
<script type="text/javascript" src="/Combo/ibmulticombo.js"></script>
<script type="text/javascript" src="/Combo/ibmulticomboinfo.js"></script>
```

The ibmulticombo.js file among the above two files is the IBMultiCombo engine js file. And, the ibmulticomboinfo.js file contains the js functions that can be conveniently used for development, and **globally** (**window**) used with the name value of **createIBMultiCombo**.

3.2 Adding an IBMultiCombo object to the screen

Add an IBMultiCombo object to the screen after adding the js file.

3.2.1 Using createIBMultiCombo

Create ibmulticombo using the createIBMultiCombo(comboid,width,height,cols) function. The cols parameter is used to set the number of columns. Set the value greater than 2 to use multiple columns. Create an ibmulticombo element at the location that called the createIBMultiCombo function.

```
<body>
  <table class='combo_table'>
    <tr>
      <td>
        <script language="javascript">
          // Create an IBMultiCombo object
          createIBMultiCombo("myCombo", "200px", "22px", 1);
        </script>
      </td>
    </tr>
  </table>
</body>
```

3.2.2 Using createIBMultiCombo 2

Pass the ID of the element to use and setting value to the createIBMultiCombo function to create a multicombo object.

```
<script>
window.onload = function() {
      // Create an IBMultiCombo object at a certain time point when the DOM content
is completely loaded.
      createIBMultiCombo ('myCombo', {
                width: 100,        // width
                height: 22,        // height
                cols: 2,           // Number of columns
                showCol: 0,        // The index of the column that will be displayed on the
text box when an item is selected.
                data: [            // Items of Drop-down List
                         {text: "D1",code:"01"},
                         {text: "D2|D12",code:"02"},
                         {text: "D3",code:"03"}],
                multiSelect: true,     // Check box column use status
                maxSelect: 2,          // Number of maximum check box column selection
                useImage: true,        // Image column use status
                title: "title1|title2",    // Title text (Use a column separator '|'.)
                titleVisible: true     // Title display status
      });
      myCombo.Init();             // ibmulticombo initialization
};
</script>
…
<input type="text"   id="myCombo" />
```

3.3 Setting initial items

To create an ibmulticombo object using the createIBMultiCombo function, define the function that sets the initial items of IBMultiCombo in the Onload event of the Body tag, as described below.

```
<Body onload=”name of the function() to set default attributes">
```

The initial items are set in the Onload event of the Body tag because the available functions of ibmulticombo can be used when page loading is completed and the ibmulticombo object is created. To use the ibmulticombo function before creating an object, a JavaScript error will occur.

The following table shows the processing items needed to set the initial items, when creating ibmulticombo using createIBMultiCombo.

Basic code procedure	Function provided by IBMultiCombo	Number of calls
IBMultiCombo Initialize	Init Method	1

```
IBMultiCombo object.Init(ShowCol);
```

e.g.,) IBMultiCombo object.Init(1);

Parameter	Type	Remark
ShowCol	Integer	Column index that will be displayed in text, when using multiple columns.

The **ShowCol** parameter sets the index of the column that should be displayed as text in the text box, when an actual item is selected, if the number of columns set in the createIBMultiCombo() function is more than 2.

For example, if the combo item is set to “한국|Korea”, set the ShowCol value to 1 to display “Korea”.

3.4 Implementing functions

3.4.1 Adding an item

```
IBMultiCombo object.InsertItem(Index, Text, Code);
```

Use this function to add a multicombo item.

e.g.,) For a single column, IBMultiCombo object.InsertItem(0, “employee", "01");

For multiple columns, IBMultiCombo object.InsertItem(0, “employee|development team", "01");

Parameter	Type	Remark
Index	Integer	Row index of the item to add
Text	String	Text of the item to add (value displayed on the list)
Code	String	Code value of the item to add

The **Index** parameter sets the row index of the item to add. The row index begins with 0, and added to the last row, if -1 is set.

The **Text** parameter sets the text value of the item to add. A value to be displayed on the list when the drop down list is expanded. This value will also be displayed in the Edit box when an item is selected. If the number of columns is set to more than 1, multiple text values can be set using a “|” separator. If the number of columns is 2, set like “employee/development team”. If it is 3, set like “employee/development team/business division”.

The **Code** parameter sets the code value of the item to add. As the value is unique, the value must not be duplicated.

3.4.2 Adding an item using a server page

- **XML format**

IBMultiCombo object.GetItemXml(Url, [Param], Callback);

A request is made using the Get method, and an item is added using the XML data that is configured by calling a server page with this function.

e.g.,) IBMultiCombo object.GetItemXml("ComboItem.jsp", "id=abc&no=1");

Parameter	Type	Remark
Url	String	The page URL that sets the combo item with the XML data.
Param	String	Query string for the search condition (e.g., id=abc&no=1)
Callback	Function	Callback function after adding an item

The **Url** parameter set the URL of the server page that sets the combo item.

The **Param** parameter sets the search condition by combining the condition with "=" and "&" in a format of "condition name 1=value 1&condition name 2=value 2".

The **Callback** parameter sets the function to call, after adding an item.

- **JSON format**

IBMultiCombo object.GetItemJson(Url, [Param], Callback);

A request is made using the Get method, and an item is added using the JSON format data that is configured by calling a server page with this function.

e.g.,) IBMultiCombo object.GetItemJson("ComboItem.jsp", "id=abc&no=1");

Parameter	Type	Remark
Url	Integer	The page URL that sets the combo item with the JSON format data.
Param	String	Query string for the search condition (e.g., id=abc&no=1)
Callback	Function	Callback function after adding an item

The **Url** parameter set the URL of the server page that sets the combo item.

The **Param** parameter sets the search condition by combining the condition with "=" and "&" in a format of "condition name 1=value 1&condition name 2=value 2".

The **Callback** parameter sets the function to call, after adding an item.

3.4.3 Adding an item using a JavaScript string

- **XML format**

IBMultiCombo object.GetItemXmlString(XmlStr);

This function is used to add an item using a string composed of the XML data.

This function adds one combo item at a time.

```
e.g.,) var sItem = "<COMBO>";
    sItem += " <DATA>";
    sItem += " <ITEM CODE='001'>한국|Korea|Seoul</ITEM>";
    sItem += " <ITEM CODE ='002'>미국|USA|Washington DC</ITEM>";
    sItem += " <ITEM CODE ='003'>중국|China|Beijing</ITEM>";
    sItem += " </DATA>";
    sItem += "</COMBO>";
    myCombo.GetItemXmlString(sItem);
```

Parameter	Type	Remark
XmlStr	String	A string that sets the combo item with the XML data.

The **XmlStr** parameter can be set by setting the XML data using one string.

- **JSON format**

IBMultiCombo object.GetItemJsonString(JsonStr);

This function is used to add an item using a string composed of the JSON format data.

This function adds one combo item at a time.

```
e.g.,) var sItem = "{DATA : [\n";
    sItem += " {Code: ‘001’, Text: ‘한국|Korea|Seoul’},\n";
    sItem += " {Code: ‘002’, Text: ‘미국|USA|Washington DC’},\n";
    sItem += " ] }";
    myCombo.GetItemJsonString(sItem);
```

Parameter	Type	Remark
JsonStr	String	A string that sets the combo item with the JSON format data.

The JsonStr parameter can be set by setting the JSON format data using one string.

3.4.4 Multiple selection function

IBMultiCombo object.SetMultiSelect(Select);

This function is used to set whether multiple items will be selected or not.

e.g.,) IBMultiCombo object.SetMultiSelect(true);

Parameter	Type	Remark
Select	Boolean	Multi-selection status

The **Select** parameter sets whether multiple items will be selected or not.

If set to True, a check box column will be displayed at the leftmost of the drop down list, so that several items can be selected. If set to False, only a single item can be selected, and the check box column will not be displayed.

3.4.5 Title display function

IBMultiCombo object.SetTitleVisible(Visible);

This function is used to set whether the title row will be displayed or not.

e.g.,) IBMultiCombo object.SetTitleVisible(true);

Parameter	Type	Remark
Visible	Boolean	Title row display status

The **Visible** parameter sets whether the title row will be displayed on the drop down list or not.

If set to True, the title row will be displayed on the first row of the drop down list, and the title of each column will be displayed. If set to False, the title row will not be displayed.

For more information on the text value displayed on the title row, see the SetTitle function that will be described in the following section.

3.4.6 Setting a title

IBMultiCombo object.SetTitle(Title);

This function is used to set the text value that will be displayed on the title row.

e.g.,) For a single column, IBMultiCombo object.SetTitle(”job titlej");
For multiple columns, IBMultiCombo object.SetTitle(”job title|team|department");

Parameter	Type	Remark
Title	String	Text value to be displayed on the title row

The **Title** parameter sets the text value of the title row that will be displayed on the first row of the drop down list. Set each column of multiple columns, using a “|” separator.

3.4.7 Image display function

IBMultiCombo object.SetUseImage(Use);

This function is used to set whether the image column will be displayed or not.

e.g.,) IBMultiCombo object.SetUseImage(true);

Parameter	Type	Remark
Use	Boolean	Image column display status

The Use parameter sets whether the image column will be displayed on the drop down list or not.

If set to True, the image column will be displayed on the first column of the drop down list. If MultiSelect is used, the image column will be displayed next to the check box column. If set to False, the image column will not be displayed. For more information on image setting, see the ItemImage function that will be described in the following section.

3.4.8 Setting an image

IBMultiCombo object.ItemImage(Index, Url);

This function is used to set an image for the image column of a specific column.

e.g.,) IBMultiCombo object.ItemImage(0, "../Image/comboitem.gif");

Parameter	Type	Remark
Index	Integer	Row index to set an image
Url	String	Image path

The **Index** parameter sets the row index to set an image.

The row index begins with 0.

The **Url** parameter sets the path of the image to display on the image column.

3.4.9 Selecting an item

- Selection using a code value

IBMultiCombo object.SetSelectCode(Code, [Event]);

This function is used to select an item, using a code value.

e.g.,) If the OnChange event occurs, IBMultiCombo object.SetSelectCode("01", true);

If the OnChange doesn’t occur, IBMultiCombo object.SetSelectCode("01", false);

Parameter	Type	Remark
Code	String	Code value of the item to select
Event	Boolean	OnChange event occurrence status

The **Code** parameter sets the code value of the item to select. A “|” separator can be used for multiple selections, like “01|03”. If the Null value is set to -1 or “”, the selected item will be cleared.

The **Event** parameter sets whether the OnChange event will occur or not, when an item is selected.

If set to True, the OnChange event occurs. If set to False, the OnChange event will not occur. True is the default value. Therefore, if no parameter value is set, the OnChange event will occur.

➢ Selection using a row index

```
IBMultiCombo object.SetSelectIndex(Index, [Event]);
```

This function is used to select an item, using a row index.

e.g.,) If the OnChange event occurs, IBMultiCombo object.SetSelectIndex("0", true);

If the OnChange doesn’t occur, IBMultiCombo object.SetSelectIndex("0", false);

Parameter	Type	Remark
Index	String	Row index of the item to select
Event	Boolean	OnChange event occurrence status

The **Index** parameter sets the row index of the item to select. The row index begins with 0, and a “|” separator can be used for multiple selections, like “1|3”. If the Null value is set (-1 or “”, the selected item will be cleared.

The **Event** parameter sets whether the OnChange event will occur or not, when an item is selected.

If set to True, the OnChange event occurs. If set to False, the OnChange event will not occur. True is the default value. Therefore, if no parameter value is set, the OnChange event will occur.

- Selection using a text value

IBMultiCombo object.SetSelectText(Text, [Event]);

This function is used to select an item, using a text value.

e.g.,) If the OnChange event occurs, IBMultiCombo object.SetSelectText(”employee", true);

If the OnChange doesn’t occur, IBMultiCombo object.SetSelectText(”employee", false);

Parameter	Type	Remark
Text	String	Text value of the item to select
Event	Boolean	OnChange event occurrence status

The **Text** parameter sets the text value of the item to select. This parameter sets the text value of the column that will be displayed when an item is selected while multiple columns are used. A “|” separator can be used for multiple selections, like “employee|section manager” If the Null value is set to -1 or “”, the selected item will be cleared.

The **Event** parameter sets whether the OnChange event will occur or not, when an item is selected.

If set to True, the OnChange event occurs. If set to False, the OnChange event will not occur. True is the default value. Therefore, if no parameter value is set, the OnChange event will occur.

3.4.10 Using an event

function IBMultiCombo object_event name(parameter) { }

For a multicombo event, create a script function in a form of “name of the using object_event name” and describe the action that should be taken when the event occurs in the function.

If ibmulticombo is created using the createIBMultiCombo function

```
function IBMultiCombo object_OnSelect(Index, Text, Code) {
    // An event occurs when the item is selected.
    alert(”Selected item [Index] " + Index + " [Text] "+ Text + " [Code] "+ Code);
}
```

The above method can be used, and pass the callback function to create ibmulticombo

```
createIBMultiCombo(‘myCombo’, {
  …
  onSelect: function(index, text, code) {
    // An event occurs when an item is selected.
    alert(”Selected item [Index] " + Index + " [Text] "+ Text + " [Code] "+
Code);
  }
});
```

when passing the event callback function as a parameter.

3.5 Implementing combo item XML

The page in the URL, which is passed as a parameter, will be displayed, if the GetItemXml function is called.

At this time, the content of the page composed in an XML format will be read and displayed as data.

The page that includes the combo item data is composed of the following structure XML.

Description	XML composition	Remark
Beginning of a large category	<COMBO>	
Beginning of the data	<DATA>	
Beginning of the item	**<ITEM Code="001">**	* Code: Code value of the item
Text setting (for a single	**Korea**	

column)		
End of the item	**</ITEM>**	
Beginning of the item	**<ITEM Code="002">**	* Code: Code value of the item
Text setting (for multiple columns)	**한국|Korea|Seoul**	* For multiple columns, each item must be separated by the "|" character.
End of the item	**</ITEM>**	
End of the data	</DATA>	
End of the large category	</COMBO>	

Cautions for creating XML

- HTML related tags (html, body, input, etc.) or script related tags (JavaScript, alert, etc.) cannot be used in an XML file. If used, an XML error occurs.
- The following special characters cannot be used. Therefore, those characters should be converted before creating an XML file. If used without conversion, an XML occur will occur.

Special character	&	<	>
Converted character	&amp;	&lt;	&gt;

The following special characters should also be converted in the MSXML version earlier than 2.0.

Special character	‘	"
Converted character	&apos;	&quot;

An XML error will not occur if CDATA Section is used, even though special characters are not converted.

Example of the error	<TD>CEO **&** employee</TD>	XML error
Example of the converted character	<TD>CEO **&amp;** employee</TD>	CEO & employee
Example of CDATA	<TD>**<![CDATA[**CEO **&** employee**]]>**</TD>	CEO & employee

3.6 Implementing the combo item with JSON

The page in the URL, which is passed as a parameter, will be displayed, if the GetItemJson function is called.

At this time, the content of the page composed in a JSON format will be read and displayed as data.

The page that includes the combo item data is composed of the following structure JSON.

Description	JSON composition
Beginning of the large category and data	{ DATA : [
Beginning of the item	**{**
Code and text setting (for a single column)	**"Code": "001", "Text": "한국"**
End of the item	**},**
Beginning of the item	**{**
Code and text setting (for multiple columns)	**"Code": "002", "Text": "한국|Korea|Seoul"**
End of the item	**}**
End of the large	] }

category and data	

* For multiple columns, a “|” separator must be used to separate each column when setting text.

Chapter 4. Development practice

4. Development practice

4.1 Creating an IBMultiCombo object

Create ibmulticombo using createIBMultiCombo.

4.1.1 Using createIBMultiCombo

Call the function as shown below to set the column that will be used for output.

Obj.Init(ShowCol);

※ ShowCol: The parameter that sets the index of the column, which should be displayed as text in the Edit box, when an actual item is selected, if the number of columns is more than 2.

- **Creating an object and adding an item**

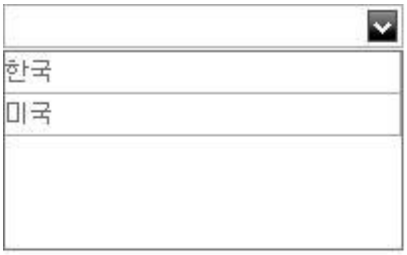
```
<HTML>
 <HEAD>
  <script type="text/javascript" src="/Combo/ibmulticombo.js"></script>
  <script type="text/javascript" src="/Combo/ibmulticomboinfo.js"></script>
 </HEAD>
  <script type="text/javascript">
   // Call the PageLoad function when the Body element is Onload.
   function PageLoad() {
      // Initialize the multicombo
      myCombo.Init(0);
      // Add an item
      myCombo.InsertItem(0,"한국","01") ;
      myCombo.InsertItem(1,"미국","02") ;
   }
   // OnFocus event
   function myCombo_OnFocus(){
   }
   // OnBlur event
   function myCombo_OnBlur(){
    }
  </script>
  <body onload="PageLoad()">
  <table class='combo_table'>
    <tr>
      <td>
        <script language="javascript">
          // Create an IBMultiCombo object
          createIBMultiCombo("myCombo", "200px", "22px", 1);
        </script>
      </td>
    </tr>
  </table>
  </body>
</HTML>
```

4.1.2 Using createIBMultiCombo 2

- **Creating an object and adding an item**

```
<html>
<head>
<script type="text/javascript" src="/Combo/ibmulticombo.js"></script>
<script type="text/javascript" src="/Combo/ibmulticomboinfo.js"></script>
<script>
  window.onload = function() {
      // When the element (input element of which ID is myCombo) used by
ibmulticombo is
      // completed loaded.
      createIBMultiCombo('myCombo', {
          width: 100,          // width
          height: 22,          // height
          cols: 1,             // Number of columns
          showCol: 0,          // The index of the column that will be displayed
on the text box when an item is selected.
            data: [                // Items of the drop down list
                 {text: “Korea", code:"01"},
                 {text: “U.S.", code:"02"}],
            onFocus: function() {},   // onFocus event
            onBlur: function(){}      // onBlur event
     });
     myCombo.Init();           // Ibmulticombo initialization
   };
</script>
</head>
<body>
   <input type=”text” id=”myCombo” />
</body>
</html>
```

You can see that the following shape object will be created, if you create and execute the source code as shown above.



4.2 Implementing various functions

Implement the creation of various function buttons and jobs that should be processed by function buttons, using the JavaScript function.

4.2.1 Using createIBMultiCombo

Implement the item insertion function as follows.

```
function doAction(sAction) {
   switch(sAction) {
      case "Insert":              // Insert an item
         myCombo.InsertItem(2,"영국","03") ;
         break;
   }
}
```

```
<!— Item insertion input button -->
<input type="button" value="입력" onClick="doAction('Insert')">
```

4.3 Sample

4.3.1 Adding a large quantity of items

```
<html>
<head>
<script type="text/javascript" src="/Combo/ibmulticombo.js"></script>
<script type="text/javascript" src="/Combo/ibmulticomboinfo.js"></script>
<script>
   window.onload = function() {
         // When the element (input element of which ID is myCombo) used by ibmulticombo is
         // completed loaded.
         var data = [];
         for (var i = 0; i < 1000; i++) {
            data.push({text: "foo|222|bar"+i,code:"03"});
         }
         createIBMultiCombo('myCombo', {
               width: 100,          // width
               height: 22,          // height
               cols: 3,             // Number of columns
               showCol: 0,          // The index of the column that will be displayed on the text box when an item is selected.
```

```
                data: data,
                // Define the column width of the drop down list when creating an
ibmulticombo object.
                // Manually control width using the SetColWidth method.
                initColWidth: [40, 40, 60]
        });
          myCombo.Init();             // Ibmulticombo initialization
   };

</script>
</head>
<body>
   <input type=”text” id=”myCombo” />
</body>
</html>}
```

4.3.2 Creating a representative string by combing each column’s text.

```
<html>
<head>
<script type="text/javascript" src="/Combo/ibmulticombo.js"></script>
<script type="text/javascript" src="/Combo/ibmulticomboinfo.js"></script>
<script>
   window.onload = function() {
          // When the element (input element of which ID is myCombo) used by
ibmulticombo is
          // completed loaded.
          var data = [];
          for (var i = 0; i < 1000; i++) {
             data.push({text: "foo|222|bar"+i,code:"03"});
          }
          createIBMultiCombo(‘myCombo’, {
                width: 100,           // width
```

```
            height: 22,          // height
            cols: 3,             // Number of columns
            showCol: 0,          // The index of the column that will be displayed on
the text box when an item is selected.
            data: data,
            // A '$' character indicates a column and the following number
indicates the column index.
            // The representative string is used to display the item, which was
selected by combining the column of the selected row, or
            // to search an item (excluding multiSelect).
            delegateTextExp: "$2 / $1",   // Combining the 3rd column text and
2nd column text of the selected item.
            // Define the column width of the drop down list when creating an
ibmulticombo object.
            initColWidth: [40, 40, 60]
     });
     myCombo.Init();           // Ibmulticombo initialization
  };
</script>
</head>
<body>
  <input type="text" id="myCombo" />
</body>
</html>}
```

4.3.3 Entering the HTML string that wraps ibmulticombo 1

```
<script>
  …
</script>
</head>
<body>
  <script>
```

```
          // Enter the HTML string that wraps multicombo in the 6th parameter,
          // when calling the createIBMultiCombo function.
          createIBMultiCombo('myCombo', 200, 22, 1, 0, '<div></div>');
    </script>
</body>
```

4.3.4 Entering the HTML string that wraps ibmulticombo 2

```
<script>
    …
</script>
</head>
<body>
    <script>
          // To append an ibmulticombo element in a specific part of wrappinghtml,
          // enter the '{{maindiv}}' string.
          createIBMultiCombo('myCombo', 200, 22, 1, 0,
                '<div><div></div><div></div>{{maindiv}}</div>'
          );
    </script>
</body>
```

4.3.5 Entering the HTML string that wraps ibmulticombo 3

```
<script>
       window.onload = function() {
          // When the element (input element of which ID is myCombo) used by
ibmulticombo is
          // completed loaded.
            createIBMultiCombo('myCombo', {
                width: 100,            // width
                height: 22,            // height
                cols: 3,               // Number of columns
                showCol: 0,        // Index of the representative column
                wrappingHtml: ‘<div></div>’
             });
             myCombo.Init();            // Ibmulticombo initialization
       };
</script>
</head>
<body>
     <input type=”text” id=”myCombo” />
</body>
```

Chapter 5. Data structure

5. Data structure

IBMultiCombo supports the XML format and JSON format when adding an combo item.

5.1 XML

The following section shows the entire structure of the XML document that is read using the GetItemXml and GetItemXmlString function.

```
<?xml version="1.0" encoding="UTF-8" ?>
<COMBO>
    <!-- Mandatory item. DATA element -->
    <DATA>
        <!-- Mandatory item. ITEM element: Sets the code and text of the combo item -->
        <ITEM CODE=”code value”>text value</ITEM>
    </DATA>
</COMBO>
```

The following XML elements are used in the above XML document.

Element name	Required?	Description
COMBO	Required	Root element of the data
DATA	Required	Start element of the data
ITEM	Required	Data code and text setting element

5.1.1 COMBO element

The COMBO element means the root of the data.

This element is the root element of the XML document, and only one element must be used under basic XML definition. The <DATA> element used in this element means the entire data of the combo item and only one element should be used.

- **Attribute**

Attribute	Description
None	

- **Example**

```
<?xml version="1.0" encoding="UTF-8" ?>
<COMBO>
    <DATA>
        <ITEM CODE="001">한국</ITEM>
        <ITEM CODE="002">미국</ITEM>
        <ITEM CODE="003">중국</ITEM>
    </DATA>
</COMBO>
```

- **Supported version**

  7.0.0.15

5.1.2 DATA element

This element means the data area, and generally used inside the <COMBO> element to display the data inside the element as a combo item.

- **Attribute**

Attribute	Description
None	

- **Example**

```
<?xml version="1.0" encoding="UTF-8" ?>
<COMBO>
    <DATA>
        <ITEM CODE="001">한국</ITEM>
        <ITEM CODE="002">미국</ITEM>
        <ITEM CODE="003">중국</ITEM>
    </DATA>
</COMBO>
```

- **Supported version**

7.0.0.15

5.1.3 ITEM element

This element means each combo item of the data.

This element is used under the <DATA> element repetitively as many times as the number of combo items. The text value of the combo item is set as the value of this element. For multiple columns, concatenate each string with a column separator “|” to set the element value.

If the element is not available in the <DATA> element, it will be deemed that no combo item is available.

The CODE attribute sets the code value of the combo item.

Be careful that this value is not duplicated, because this value is the unique value of the combo item.

- **Attribute**

Attribute	Description
CODE	Code value of the combo item

- **Example**

```
<!-- For a single column -->
<?xml version="1.0" encoding="UTF-8" ?>
<COMBO>
      <DATA>
            <ITEM CODE="001">한국</ITEM>
            <ITEM CODE="002">미국</ITEM>
            <ITEM CODE="003">중국</ITEM>
      </DATA>
</COMBO>

<!-- For multiple columns -->
<?xml version="1.0" encoding="UTF-8" ?>
<COMBO>
```

```
        <DATA>
                <ITEM CODE="001">한국|Korea|Seoul</ITEM>
                <ITEM CODE="002">미국|USA|Washington DC</ITEM>
                <ITEM CODE="003">중국|China|Beijing</ITEM>
        </DATA>
</COMBO>
```

- **Supported version**

  7.0.0.15

5.2 JSON

The following section shows the entire structure of the JSON document that is read using the GetItemJson and GetItemJsonString function.

```
{
        // Mandatory item. DATA element
        DATA : [
                // Mandatory item. Combo item code and text setting
                {Code: “code value", Text: “text value"},
        ]
}
```

The following JSON elements are used in the above JSON document.

Element name	Required?	Description
DATA	Required	Start element of the data

5.2.1 DATA element

This element means the data area, and displays the data inside the element as a combo item.

The element value is composed of an array, and each array element means each combo item.

The combo item has the structure of Key:Value, and is set by mapping with Code and Text Key.

The Code Key sets the code value of the combo item.
Be careful that this value is not duplicated, because this value is the unique value of the combo item.

The Text Key sets the text value of the combo item.
For multiple columns, concatenate each string with a column separator “|” to set the element value.

- **Key**

Key	Description
Code	Code value of the combo item
Text	Text value of the combo item

- **Example**

```
// For a single column
{
        DATA : [
                {Code: "001", Text: "Korea"},
                {Code: "002", Text: "미국"},
                {Code: "003", Text: "중국"}
        ]
}
```

```
// For multiple columns
{
        DATA : [
                {Code: "001", Text: "한국|Korea|Seoul"},
                {Code: "002", Text: "미국|USA|Washington DC"},
                {Code: "003", Text: "중국|China|Beijing"}
        ]
}
```

- **Supported version**

  7.0.0.15

Chapter 6. IBMultiCombo Config

6. IBMultiCombo Config

6.1 Setting Config

To set the name of the license file, set it in the ibmulticombo.cfg file as shown below. The file format is XML.

```
<COMBO>
          <Cfg
                    LicenseFileName="license"
          />
</COMBO>
```

6.2 Property List

<table>
<tr><th>Property name</th><th>Description</th></tr>
<tr><td>LicenseFileName</td><td>Set the name of the license file (Default : "license")
- Supported version: 7.0.0.18</td></tr>
<tr><td>SaveNameMode</td><td>Set whether the default setting [Control ID]_code will be used or [Control ID] will be used as the name of the combo code that is sent when submitted. (Default: “0")
<table>
<tr><th>Setting value</th><th>Description</th></tr>
<tr><td>0</td><td>[Control ID]_code is used when submitted.</td></tr>
<tr><td>1</td><td>[Control ID] is used when submitted.</td></tr>
</table>
- Supported version: 7.0.0.20</td></tr>
<tr><td>UseSelectCount</td><td>Set whether the text value will be displayed or “X item selected” will be displayed as the selected item when selecting multiple items.
<table>
<tr><th>Setting value</th><th>Description</th></tr>
<tr><td>0</td><td>Display the text value (default).</td></tr>
</table></td></tr>
</table>

<table>
<tr><td></td><td><table><tr><td>1</td><td>Display “X item selected”.</td></tr></table>- Supported version: 7.0.0.34</td></tr>
<tr><td>EventArgsMode</td><td>Set whether the current multicombo object will be passed to the first parameter of all multicombo event handlers.<table><tr><th>Setting value</th><th>Description</th></tr><tr><td>0</td><td>Don’t pass the current object as the first parameter (default)</td></tr><tr><td>1</td><td>Pass the current object as the first parameter</td></tr></table>- Supported version: 7.0.0.46</td></tr>
<tr><td>Traditional</td><td>Set the same calling sequence of the event callback is applied, which was applied to the previous version (before v43).<table><tr><th>Setting value</th><th>Description</th></tr><tr><td>0</td><td>Don’t run the previous version (before v43). (Default)</td></tr><tr><td>1</td><td>Run the previous version (before v43).</td></tr></table>- Supported version: 7.0.0.52</td></tr>
<tr><td>MultiSeparator</td><td>Set MultiSeparator globally, which is a separator for selected value output. (default: “|”)
- Supported version: 7.0.0.56</td></tr>
<tr><td>SyncRequest</td><td>Set the sync and async request status when calling GetItemJson and GetItemXml.<table><tr><th>Setting value</th><th>Description</th></tr><tr><td>0</td><td>async request (Default)</td></tr><tr><td>1</td><td>sync request</td></tr></table>- Supported version: 7.0.0.57</td></tr>
<tr><td>UseAutoComplete</td><td>Set whether autoComplete will be used or not.</td></tr>
</table>

<table>
<tr>
<td></td>
<td>
<table>
<tr><th>Setting value</th><th>Description</th></tr>
<tr><td>0</td><td>Don’t use autoComplete (default).</td></tr>
<tr><td>1</td><td>Use autoComplete.</td></tr>
</table>
- Supported version: 7.0.0.61
</td>
</tr>
<tr>
<td>BatchOutput</td>
<td>
Set the sequence of the selected item.
<table>
<tr><th>Setting value</th><th>Description</th></tr>
<tr><td>0</td><td>Set the text of the last selected item to the last,
when displayed (default).</td></tr>
<tr><td>1</td><td>Set the text of the selected item to be displayed in order of the item inserted into the drop down list.</td></tr>
</table>
- Supported version: 7.0.0.61
</td>
</tr>
<tr>
<td>NarrowDDListShowBtn</td>
<td>
Set narrowDDListShowBtn
<table>
<tr><th>Setting value</th><th>Description</th></tr>
<tr><td>0</td><td>Use all multicombo areas as a button that toggles “show/hide” drop down list. (default)</td></tr>
<tr><td>1</td><td>Use the right button image of the multicombo
as a button that toggles “show/hide” drop down list.</td></tr>
</table>
- Supported version: 7.0.0.62
</td>
</tr>
</table>

Chapter 7. IBMultiCombo Events

7. IBMultiCombo Events

7.1 Using an event

Events are provided for the user to add or change the function in IBMultiCombo.

```
function object ID_event name(parameter, …) { } or function() {}
```

For example, if the IBMultiCombo ObjectID is “myCombo”and the function for the OnChange event should be handled, the function can be written as follows.

```
function myCombo_OnChange(OldIndex, OldText, OldCode, NewIndex, NewText,
NewCode) {
    alert(The value is changed from OldText + “to “ + NewText + “.");
}

Or

createIBMultiCombo(‘myCombo’, {
  …,
  onChange: function(oldIndex, oldText, oldCode, newIndex, newText, newCode) {
     alert(The value is changed from oldText + “to ” + newText + “.”);
  }
});
```

IBMultiCombo provides the following events.

7.2 Event List

OnBlur Event

- **Function**

  An event occurs when a focus is lost.

- **Syntax**

Syntax	**function object ID_OnBlur**() { } or **function**(){}

- **Parameters**

Parameter	Type	Description
None		

- **Example**

```
// An event occurs when a focus is lost.
function myCombo_OnBlur() {
        Alert(”The focus is lost!");
}


Or


// Pass the callback when creating ibmulticombo.
createIBMultiCombo(‘myCombo’, {
    …,
    onBlur: function() {
        Alert(”The focus is lost.”);
    }
});
```

- **Supported version**

  7.0.0.0

OnChange Event

- **Function**

An event occurs when the selected item is changed.

An event will not occur, if the selected item is selected again.

- **Syntax**

Syntax	**function object ID_OnChange**(OldIndex, OldText, OldCode, NewIndex, NewText, NewCode) { } or **function**(){}

- **Parameters**

Parameter	Type	Description
OldIndex	String	Old index value
OldText	String	Old text value
OldCode	String	Old code value
NewIndex	String	New index value
NewText	String	New text value
NewCode	String	New code value

- **Example**

```
// An event occurs when the selected item is changed.
function myCombo_OnChange(OldIndex, OldText, OldCode, NewIndex,
NewText, NewCode) {
      alert(The value is changed from OldText + “to “ + NewText + “.");
}
Or
// Pass the callback when creating ibmulticombo.
createIBMultiCombo(‘myCombo’, {
   onChange:function(oldIndex, oldText, oldCode, newIndex, newText,
newCode) { alert(The value is changed from oldText + “to ” + newText + “.”);}});
```

- **Supported version**

7.0.0.0OnCheckClick Event

- **Function**

  An event occur when the check box is clicked, if multiple selection is used.

- **Syntax**

Syntax	**function object ID_OnCheckClick**(Index, Code) { }
	or **function**(){}

- **Parameters**

Parameter	Type	Description
Index	Integer	Index value of the clicked item
Code	String	Code value of the clicked item
Checked	Boolean	The value that specifies whether checked or not.

- **Example**

```
// An event occurs when the check box is clicked.
function myCombo_OnCheckClick(Index, Code, Checked) {
      alert("[Index:]" + Index + "[ Code:]"+ Code + “[Checked:]” + Checked);
}


Or


// Pass the callback when creating ibmulticombo.
createIBMultiCombo(‘myCombo’, {
   …,
   onCheckClick: function(Index, Code, Checked) {
      alert("[Index:]" + Index + "[ Code:]"+ Code + “[Checked:]” + Checked);
   }
});
```

- **Supported version**

  7.0.0.0

OnClear Event

Function

// An event occurs when all items are deleted.

An event occurs when all items are deleted using the RemoveAll function or the DeleteItem function.

Syntax

Syntax	**function object ID_OnClear**() { } or **function**(){}

Parameters

Parameter	Type	Description
None		

Example

```
// An event occurs when all items are deleted.
function myCombo_OnClear() {
      Alert(”All items are deleted.”);
}

Or

// Pass the callback when creating ibmulticombo.
createIBMultiCombo(‘myCombo’, {
   …,
   onClear: function() {
      alert(”All items are deleted.”);
   }
});
```

Supported version

7.0.0.0

OnFocus Event

Function

An event occurs when a focus is obtained.

Syntax

Syntax	**function object ID_OnFocus**() { } or **function**(){}

Parameters

Parameter	Type	Description
None		

Example

```
// An event occurs when a focus is obtained.
function myCombo_OnFocus() {
      Alert(”The focus is obtained!");
}

Or

// Pass the callback when creating ibmulticombo.
createIBMultiCombo(‘myCombo’, {
   …,
   onFocus: function() {
      Alert(”The focus is obtained!");
   }
});
```

Supported version

7.0.0.0

OnKeyDown Event

Function

An event occurs when a key on the keyboard is pressed.

Syntax

Syntax	**function object ID_OnKeyDown**(KeyCode, Shift, Ctrl) { }
	or **function**(){}

Parameters

Parameter	Type	Description
KeyCode	Integer	ASCII code value
Shift	Boolean	true: Shift key is pressed./ false: Shift key is not pressed.
Ctrl	Boolean	true: Ctrl key is pressed. /false: Ctrl key is not pressed.

Example

```
// An event ocurs if the key is pressed.
function myCombo_OnKeyDown(KeyCode, Shift, Ctrl) {
      alert(”A keyboard key is pressed. “);
}


Or


createIBMultiCombo(‘myCombo’, {      // Pass the callback when creating ibmulticombo.
   …,
   onKeyDown: function(keycode, shift, ctrl) {
      alert(”A keyboard key is pressed. “);
   }


});
```

- **Supported version**

  7.0.0.0

OnMultiCheckError Event

- **Function**

  This event occurs when the setting value is an item that doesn't exist, if multiple selections are made with a "|" separator using the script, and MultiSelect is performed. If the number of items that don't exist is more than 1, items are concatenated with a "|" separator and passed.

- **Syntax**

Syntax	**function object ID_OnMultiCheckError**(ErrorItem) { }
	or **function**(){}

- **Parameters**

Parameter	Type	Description
ErrorItem	String	List of items that don't exist If the number of items that don't exist is more than 1, items are concatenated with a "|" separator and received.

- **Example**

```
function myCombo_OnMultiCheckError(ErrorItem) {
      alert("Error Item :" + ErrorItem);
}

Or

createIBMultiCombo('myCombo', {      // Pass the callback when creating ibmulticombo.
   …,
   onMultiCheckError: function(errorItem) {
      alert("잘못된 아이템 :" + errorItem);
   }
});
```

- **Supported version**

  7.0.0.0

OnSelect Event

- **Function**

  An event occurs when the item is selected.

  Unlike the OnChange event, an event occurs even though the same value is selected.

- **Syntax**

Syntax	**function object ID_OnSelect**(Index, Text, Code) { }
	or **function**(){}

- **Parameters**

Parameter	Type	Description
Index	Integer	Index value of the selected item
Text	String	Text value of the selected item
Code	String	Code value of the selected item

- **Example**

```
// An event occurs when the item is selected.
function myCombo_OnSelect(Index, Text, Code) {
      alert("[Index:]" + Index + "[ Text:]"+ Text + "[ Code:]"+ Code);
}


Or
createIBMultiCombo(‘myCombo’, {      // Pass the callback when creating
ibmulticombo.
   …,
   onSelect: function(index, text, code) {
     alert("[Index:]" + index + "[ Text:]"+ text + "[ Code:]"+ code);
   }
});
```

- **Supported version**

  7.0.0.0

Chapter 8. IBMultiCombo Methods

8. IBMultiCombo Methods

8.1 Using a method

The following syntax can be used to use the method provided by IBMultiCombo.

Syntax	// When creating a multicombo object using createIBMultiCombo Object ID.method name();

IBMultiCombo provides the following functions.

8.2 Method List

AllowEvent4CheckAll Method

- **Function**

  Set when an event will occur or not, when all multicombo items are selected.

- **Syntax**

Syntax	Get	Obj.**GetAllowEvent4CheckAll**()
	Set	Obj.**SetAllowEvent4CheckAll**(isAllow)

- **Info**

Return	String,　Setting value (for Get Method)		
Parameter	Type	Required?	Description
isAllow	Boolean	Required	Set whether an event will be allowed or not. true: An event occurs.

- **Example**

```
// An event occurs for each data when Select All is clicked.
myCombo. SetAllowEvent4CheckAll (true);
// Check the event occurrence status of the control.
alert(myCombo.GetAllowEvent4CheckAll());
```

- **Supported version**

  7.0.0.98

BackColor Method

- **Function**

  Check or set the background color of the control. Set the setting value with the web color string.

  The value must be set with the web color string.

- **Syntax**

Syntax	Get	Obj.**GetBackColor**()
	Set	Obj.**SetBackColor**(Color)

- **Info**

Return	String, Setting value (for Get Method)		
Parameter	Type	Required?	Description
Color	String	Required	Color value WebColor (ex:#FF0000)

- **Example**

```
// Set the background color of the control to #FF0000.
myCombo.SetBackColor("#FF0000");
// Get the background color of the control.
alert(myCombo.GetBackColor());
```

- **Supported version**

  7.0.0.0

BatchOutput Method

Function

Set the direction of the string that is displayed on the element, which displays the string of the item that is selected when using multiSelect.

Syntax

Syntax	Get	Obj.**GetBatchOutput**()
	Set	Obj.**SetBatchOutput**(BatchOutput)

Info

Return	String, Setting value (for Get Method)		
Parameter	Type	Required?	Description
BatchOutput	Boolean	Required	The text of the selected item will be displayed on the output element in order of the items that are inserted into the drop down list. (default: false)

Example

```
// The selected items are displayed in order of insertion into the drop down list.
myCombo. SetBatchOutput( true );
// Get the name of the combo code.
alert(myCombo. GetBatchOutput());
```

Supported version

7.0.0.61

ColBackColor Method

➢ Function

Check or set the background color of a specific column. The Col parameter means the index number of the column. The check box column and the image column are excluded from the index, and begins with 0 after the check box column and the image column.

The value must be set with a web color string.

➢ Syntax

Syntax	Get	Obj.**GetColBackColor**(Col)
	Set	Obj.**SetColBackColor**(Col, Color)

➢ Info

Return	String, setting value (for Get Method)		
Parameter	Type	Required?	Description
Col	Integer	Required	Index value of a specific column
Color	String	Required	Color value WebColor (e.g., #FF0000)

➢ Example

```
// Set the column corresponding to the index No. 1 to the #FF0000 color.
myCombo.SetColBackColor(1, "#FF0000");
// Get the color of the column corresponding to the index No. 2.
alert(myCombo.GetColBackColor(2));
```

➢ Supported version

7.0.0.0

ColCount Method

- **Function**

  Return the number of columns.

- **Syntax**

Syntax	Obj.**ColCount**()

- **Info**

Return	Integer, Number of columns		
Parameter	Type	Required?	Description
None			

- **Example**

```
// Get the number of columns.
alert(”number of columns:" + myCombo.ColCount());
```

- **Supported version**

  7.0.0.0

ColFontColor Method

- **Function**

  Check or set the font color of a specific column. The Col parameter means the index number of the column. The check box column and the image column are excluded from the index, and begins with 0 after the check box column and the image column.

  The value must be set with a web color string.

- **Syntax**

Syntax	Get	Obj.**GetColFontColor**(Col)
	Set	Obj.**SetColFontColor**(Col, Color)

- **Info**

Return	String, Setting value (for Get Method)		
Parameter	Type	Required?	Description
Col	Integer	Required	Index value of a specific column
Color	String	Required	Color value WebColor (e.g., #FF0000)

- **Example**

```
// Set the font color of the column corresponding to the index No. 1 to the #FF0000 color.
myCombo.SetColFontColor(1, "#FF0000");
// Get the font color of the column corresponding to the index No. 2.
alert(myCombo.GetColFontColor(2));
```

- **Supported version**

  7.0.0.0

ComboBtnImage Method

- **Function**

  Use to specify the image of the combo drop down button with a specific image. Use the 17*19 image size.

- **Syntax**

Syntax	Obj.**ComboBtnImage**(Url)

- **Info**

Return	None		
Parameter	Type	Required?	Description
Url	String	Required	Image path of the combo button

- **Example**

```
// Change the button image of the combo drop down.
myCombo.ComboBtnImage("../image/combo2.gif");
```

- **Supported version**

  7.0.0.0

DeleteItem Method

- **Function**

  Delete the specific item of the combo with the index or code value.

  The second column of the control will be deleted, if 1 is inserted into the row index of the item to delete.

- **Syntax**

Syntax	Obj.**DeleteItem**(Index, isCode)

- **Info**

Return	None		
Parameter	Type	Required?	Description
Index	Integer	Required	Row index or code of a specific item

- **Example**

```
// Delete the item of the row 1.
myCombo.DeleteItem(0);

// Delete the row that has the “01” code value.
myCombo.DeleteItem(‘01’);
```

- **Supported version**

  7.0.0.0

DisabledBackColor Method

Function

Check or set the background color of the control, if Enable is set to False.

Make user to set with a web color string.

Syntax

Syntax	Get	Obj.**GetDisabledBackColor**()
	Set	Obj.**SetDisabledBackColor**(Color)

Info

Return	String, setting value (for Get Method)		
Parameter	Type	Required?	Description
Color	String	Required	Color value WebColor (e.g., #FF0000)

Example

```
//Set the background color to the #FF0000 color, if Enable is set to False.
myCombo.SetEnable(false);
myCombo.SetDisabledBackColor( "#FF0000");
// Get the background color if Enable is set to False.
alert(myCombo.GetDisabledBackColor());
```

Supported version

7.0.0.0

DisplayDropDownList Method

- **Function**

  Set whether the drop down list will be displayed in ibmulticombo or not.

- **Syntax**

Syntax	Obj. DisplayDropDownList (isDisplay, [isEvent])

- **Info**

Return	None		
Parameter	Type	Required ?	Description
isDisplay	Boolean	Required	Display status of the drop down list true: Display the drop down list. false: Don't display the drop down list.
Event	Boolean	Optional	OnFocus, OnBlur callback call status True: If IisDisplay isTrue, call onFocus callback. If isDisplay is False, call onBlur callback. false: Don't call callback (default).

- **Example**

```
myCombo. DisplayDropDownList (true);      // Display the drop down list.
myCombo.DisplayDropDownList(false);      // Don't display the drop down list.

// Call onFocus callback when displaying the drop down list.
myCombo.DisplayDropDownList(true, true);
// Call onBlur callback when not displaying the drop down list.
myCombo.DisplayDropDownList(false, true);
```

- **Supported version**

  7.0.0.44

DropDownOutLineColor Method

- **Function**

  Check or set the color of the outline in the drop down area of the combo.

  Make user to set with a web color string.

- **Syntax**

Syntax	Get	Obj.**GetDropDownOutLineColor**()
	Set	Obj.**SetDropDownOutLineColor**(Color)

- **Info**

Return	String, setting value (for Get Method)		
Parameter	Type	Required?	Description
Color	String	Required	Color value WebColor (e.g., #FF0000)

- **Example**

```
// Set the color to the #FF0000 color.
myCombo.SetDropDownOutLineColor("#FF0000");
// Get the color.
alert(myCombo.GetDropDownOutLineColor());
```

- **Supported version**

  7.0.0.0

DisposeCombo Method

- **Function**

  Delete the pertinent control using the method that deletes the combo control. (However, the method is available only when the combo control is created using wrappingHtml.)

- **Syntax**

Syntax	Obj. DisposeCombo()

- **Info**

Return	Boolean		
Parameter	Type	Required?	Description
None			

- **Example**

```
<script>
        window.onload = function() {
            createIBMultiCombo('myCombo', {
                    width: 100,             // width
                    height: 22,             // height
                    cols: 3,                // Number of columns
                    showCol: 0,         // Representative column index
                    wrappingHtml: '<div></div>'
                });
                myCombo.Init();             // ibmulticombo initialization
        };
function deleteCombo (){
                myCombo.DisposeCombo();
}
```

```
</script>
</head>
<body>
    <input type=”text” id=”myCombo” />
    <input type=”button” value=”Click" onclick="deleteCombo()">
</body>
```

- **Supported version**

  7.0.0.102

DropHeight Method

- **Function**

  Check or set the height of the combo’s drop down area.

  If DropHeight is set higher than the sum of each item’s height, the value set with DropHeight will be the sum of each item’s height.

  It must be set in numbers. A string is not supported.

- **Syntax**

Syntax	Get	Obj.**GetDropHeight**()
	Set	Obj.**SetDropHeight**(Value)

- **Info**

Return	Integer, setting value (for Get Method)		
Parameter	Type	Required?	Description
Value	Integer	Required	The height value to set (default: 105px)

- **Example**

```
// Set the height of the drop down area to 300.
myCombo.SetDropHeight(300);
// Get the height of the drop down area.
alert(myCombo.GetDropHeight());
```

- **Supported version**

  7.0.0.0

EditFontBold Method

- **Function**

  Check or set the bold font on the Edit window.

- **Syntax**

Syntax	Get	Obj.**GetEditFontBold**()
	Set	Obj.**SetEditFontBold**(Value)

- **Info**

Return	Boolean, setting value (for Get Method)		
Parameter	Type	Required?	Description
Value	Boolean	Required	Bold font (true/false)

- **Example**

```
// Set the bold font on the Edit window.
myCombo.SetEditFontBold(true);
// Get the bold font on the Edit window.
alert(myCombo.GetEditFontBold());
```

- **Supported version**

  7.0.0.0

Editable Method

➢ Function

Check or set whether the control can be edited or not.

If set to False, keyboard input is limited but the API can be set using the combo list drop down and script. The code and text value of the multicombo will be sent, when submitted.

➢ Syntax

Syntax	Get	Obj.**GetEditable**()
	Set	Obj.**SetEditable**(Value)

➢ Info

Return	Boolean, setting value (for Get Method)		
Parameter	Type	Required?	Description
Value	Boolean	Required	Editing status (True: Editable, false: Editing is not allowed)

➢ Example

```
// Set in such a way that editing is not allowed and only drop down is possible.
myCombo.SetEditable(false);
// Get the editable status.
alert(myCombo.GetEditable());
```

➢ Supported version

7.0.0.19

Enable Method

Function

Check or set whether the control can be use or not. If set to False, keyboard input and combo list drop down are limited but the API can be set using the script. The code and text value of the multicombo will not be sent, when submitted.

Syntax

Syntax	Get	Obj.**GetEnable**()
	Set	Obj.**SetEnable**(Value)

Info

Return	Boolean, Setting value (for Get Method)		
Parameter	Type	Required?	Description
Value	Boolean	Required	Control availability (true/false)

Example

```
// Set the control unavailable.
myCombo.SetEnable(false);
// Get control availability.
alert(myCombo.GetEnable());
```

Supported version

7.0.0.0

EnableAllCheckBtn Method

Function

Enable the button that selects all items of the drop down list from the check box column in the title area of the drop down list.

The button will be displayed only when TitleVisible and MultiSelect are set to True.

Syntax

Syntax	Get	Obj.**GetEnableAllCheckBtn**()
	Set	Obj.**SetEnableAllCheckBtn**(Value)

Info

Return	Boolean, Setting value (for Get Method)		
Parameter	Type	Required?	Description
Value	Boolean	Required	Enabling status of the all item selection button of the drop down list (default: false).

Example

```
myCombo.SetTitleVisible(true);
// Enable the all item selection button of the drop down list.
myCombo.Set EnableAllCheckBtn (true);
// Enabling status of the all item selection button of the drop down list.
alert(myCombo.Get EnableAllCheckBtn ());
```

Supported version

7.0.0.73

FindItem Method

- ### Function

Find an item that is same with a string to search in a specific column, and return the row index or code. Input the code value into the first parameter only and skip the remaining parameters, if searching with a code value.

Return -1, if search fails.

- ### Syntax

Syntax	Obj.**FindItem**(Text, Col, IsIndex)

- ### Info

Return	String, the index value or the code value of the found row.		
Parameter	Type	Required?	Description
Text	String	Required	A string to search
Col	Integer	Optional	A column index to search for a string (If a Boolean value is entered instead of an integer value, it is used as the parameter of the IsIndex.)
IsIndex	Boolean	Optional	true: Return the index value (default). False: Return the code value.

- ### Example

```
// Find the row number that is same with a specific character.
var nIdx = myCombo.FindItem("section manager", 0, true);
if( nIdx == -1) {
        // Return -1, if search fails.
        Alert(”Unable to find the same character.”);
} else {
        alert(nIdx);
}
var code = myCombo.FindItem(“code1”);      // Row index that has the code value.
```

- **Supported version**

  7.0.0.0

Focus Method

- **Function**

  Give a focus to the input element that displays the selected item of multicombo.

- **Syntax**

Syntax	Obj.**Focus**()

- **Info**

Return	None		
Parameter	Type	Required?	Description
None			

- **Example**

```
// Give a focus to the input element that displays the selected item of myCombo.
myCombo.Focus();
```

- **Supported version**

  7.0.0.58

FontColor Method

- **Function**

  Check or set the overall font color of the combo.

  The value must be set with a web color string.

- **Syntax**

Syntax	Get	Obj.**GetFontColor**()
	Set	Obj.**SetFontColor**(Color)

- **Info**

Return	String, Setting value (for Get Method)		
Parameter	Type	Required?	Description
Color	String	Required	Color value WebColor (e.g., #FF0000)

- **Example**

```
// Set the color to the #FF0000 color.
myCombo.SetFontColor("#FF0000");
// Get the color.
alert(myCombo.GetFontColor());
```

- **Supported version**

  7.0.0.0

FocusHighlightColor Method

- **Function**

  Check or set the border color of the combo when a focus is set on the combo.

- **Syntax**

Syntax	Get	Obj.**GetFocusHighlightColor** ()
	Set	Obj.**SetFocusHighlightColor**(Color)

- **Info**

Return	String, Setting value (for Get Method)		
Parameter	Type	Required?	Description
Color	String	Required	Color value WebColor (e.g., #FF0000)

- **Example**

```
// Set the border color of the combo when a focus is set on the combo.
myCombo. SetFocusHighlightColor ("#FF0000");
// Check the border color of the combo when a focus is set on the combo.
alert(myCombo. GetFocusHighlightColor ());
```

- **Supported version**

  7.0.0.73

FontName Method

- **Function**

  Check or set the overall font name of the combo. The mobile device doesn't support this method, and the default of each device will be used for display.

- **Syntax**

Syntax	Get	Obj.**GetFontName**()
	Set	Obj.**SetFontName**(Value)

- **Info**

Return	String,　Setting value (for Get Method)		
Parameter	Type	Required?	Description
Value	String	Required	Font name (e.g., Gulim font)

- **Example**

```
// Set the font name to Gulim.
myCombo.SetFontName("Gulim font");
// Get the font name.
alert(myCombo.GetFontName());
```

- **Supported version**

  7.0.0.0

FontSize Method

- **Function**

  Check or set the overall font size of the combo.

  If the font size is bigger than the multicombo size, the multicombo size also increases.

- **Syntax**

Syntax	Get	Obj.**GetFontSize**()
	Set	Obj.**SetFontSize**(Value)

- **Info**

Return	Integer, Setting value (for Get Method)		
Parameter	Type	Required?	Description
Value	Integer	Required	Font size

- **Example**

```
// Set the font size to 11.
myCombo.SetFontSize(11);
// Get the font size.
alert(myCombo.GetFontSize());
```

- **Supported version**

  7.0.0.22

GetIndexText Method

- **Function**
- Check the text value of a specific item.

- **Syntax**

Syntax	Obj. GetIndexText(Index_Code, Col)

- **Info**

Return	String, setting value		
Parameter	Type	Required?	Description
Index	Integer	Required	Index value
Col	Integer	Required	Column index

- **Example**

```
// Check the text of the first column of the first row.
alert(myCombo.GetIndexText(1, 1));

// Check the text of the first column of the first row.
alert(myCombo.GetIndexText(“1”, “1”));
```

- **Supported version**

7.0.0.0

GetItemCount Method

- **Function**

  Return the total number of columns.

- **Syntax**

Syntax	Obj.**GetItemCount**()

- **Info**

Return	Integer, number of items		
Parameter	Type	Required?	Description
None			

- **Example**

```
// Get the number of items.
Alert(”number of items:” + myCombo.GetItemCount());
```

- **Supported version**

  7.0.0.0

GetItemJson Method

- **Function**

  Make a Get type request, and connect a server page to read the JSON format data and set the combo item.

  The Param parameter sets the search condition by combining the condition with "=" and "&" in a format of "condition name 1=value 1&condition name 2=value 2".

- **Syntax**

Syntax	Obj.**GetItemJson**(Url, [Param, Callback, Clear, Begin])

- **Info**

Return	None		
Parameter	Type	Required?	Description
Url	String	Required	A page URL that sets the combo item with the JSON format data.
Param	String	Optional	Query string for the search condition (e.g., id=abc&no=1)
Callback	Function	Optional	The function called after making a request Provide the JSON data as a parameter to the function.
Clear	Blooean	Optional	Whether the existing item will be deleted or not, when reading items (default: true).
Begin	Integer	Optional	Row index of the item to add (default: Last row)
ReqHeader	Object	Optional	Setting value object of the HTTP request header Default={}

- **Example**

```
// Set the combo item by connecting the “Comboitem.jsp” file.
myCombo.GetItemJson("ComboItem.jsp",                    "id=abc&no=1",
function(json){console.log(json);});

// Set the HTTP request header.
myCombo.GetItemJson(   "ComboItem.jsp",   "id=abc&no=1",   function(xml)
{ console.log(xml); }, false, 1, {"Wait":"1","Sync":"1"} );
```

- **Supported version**

  7.0.0.15

GetItemJsonString Method

➢ Function

Set the combo item using the JSON format data string.

This function adds one combo item at a time.

➢ Syntax

Syntax	Obj.**GetItemJsonString**(JsonStr, [Clear, Begin])

➢ Info

Return	None		
Parameter	Type	Required?	Description
JsonStr	String, Object	Required	A string or object that composes the combo item in a JSON format.
Clear	Blooean	Optional	Whether the existing item will be deleted or not, when reading items (default: true).
Begin	Integer	Optional	Row index of the item to add (default: Last row)

➢ Example

```
// Set the combo item using the JSON format string.
var sItem = "{DATA : [\n";sItem += " {Code: '001', Text: '한국|Korea|Seoul'},\n";
sItem += " {Code: '002', Text: '미국|USA|Washington DC'},\n";
sItem += " {Code: '003', Text: '중국|China|Beijing'}\n";sItem += " ] }";
myCombo.GetItemJsonString(sItem);

var sItem = {jsonStr: {DATA : [{Code: '001', Text: '한국|Korea|Seoul'},{Code:
'002', Text: '미국|USA|Washington DC'},{Code: '003', Text:
'중국|China|Beijing'}] } };
myCombo.GetItemJsonString()
```

➢ Supported version

7.0.0.15

GetItemXml Method

- **Function**

  Make a Get type request, and connect a server page to read the XML data and set the combo item.

  The Param parameter sets the search condition by combining the condition with "=" and "&" in a format of "condition name 1=value 1&condition name 2=value 2".

- **Syntax**

Syntax	Obj.**GetItemXml**(Url, [Param, Callback, Clear, Begin])

- **Info**

Return	None		
Parameter	Type	Required?	Description
Url	String	Required	A page URL that sets the combo item with the XML data.
Param	String	Optional	Query string for the search condition (e.g., id=abc&no=1)
Callback	Function	Optional	The function called after making a request Provide the XML data as a parameter to the function.
Clear	Blooean	Optional	Whether the existing item will be deleted or not, when reading items (default: true).
Begin	Integer	Optional	Row index of the item to add (default: Last row)
ReqHeader	Object	Optional	Setting value object of the HTTP request header Default={}

- **Example**

```
// Set the combo item by connecting the “Comboitem.jsp” file.
myCombo.GetItemXml("ComboItem.jsp", "id=abc&no=1",
function(xml){console.log(xml);});

// Set the HTTP request header.
myCombo.GetItemXml( "ComboItem.jsp", "id=abc&no=1", function(xml)
{ console.log(xml); }, false, 1, {"Wait":"1","Sync":"1"} );
```

- **Supported version**

  7.0.0.15

GetItemXmlString Method

- **Function**

Set the combo item using the XML data string.

This function adds the combo item at a time.

- **Syntax**

Syntax	Obj.**GetItemXmlString**(XmlStr, [Clear, Begin])

- **Info**

Return	None		
Parameter	Type	Required?	Description
XmlStr	String	Required	A string that sets the combo item with the XML data.
Clear	Blooean	Optional	Whether the existing item will be deleted or not, when reading items (default: true).
Begin	Integer	Optional	Row index of the item to add (default: Last row)

- **Example**

```
// Set the combo item using the XML format string.
var sItem = "<COMBO>";
sItem += " <DATA>";
sItem += " <ITEM CODE='001'>한국|Korea|Seoul</ITEM>";
sItem += " <ITEM CODE ='002'>미국|USA|Washington DC</ITEM>";
sItem += " <ITEM CODE ='003'>중국|China|Beijing</ITEM>";
sItem += " </DATA>";
sItem += "</COMBO>";

myCombo.GetItemXmlString(sItem);
```

- **Supported version**

7.0.0.15

HoverBackColor Method

- **Function**

  Check or set the color that is applied to the item, when the mouse is hovered over the item on the list and when the combo is expanded.

  The value must be set with a web color string.

- **Syntax**

Syntax	Get	Obj.**GetHoverBackColor**()
	Set	Obj.**SetHoverBackColor**(Color)

- **Info**

Return	String, setting value (for Get Method)		
Parameter	Type	Required?	Description
Color	String	Required	Color value WebColor (e.g., #FF0000)

- **Example**

```
// Set the color to the #FF0000 color.
myCombo. SetHoverBackColor ("#FF0000");
// Get the color.
alert(myCombo. GetHoverBackColor ());
```

- **Supported version**

  7.0.0.68

HoverFontColor Method

- **Function**

  Check or set the font color that is applied to the item, when the mouse is hovered over the item on the list, when the combo is expanded.

  The value must be set with a web color string.

- **Syntax**

Syntax	Get	Obj.**GetHoverFontColor**()
	Set	Obj.**SetHoverFontColor**(Color)

- **Info**

Return	String, setting value (for Get Method)		
Parameter	Type	Required?	Description
Color	String	Required	Color value WebColor (e.g., #FF0000)

- **Example**

```
// Set the color to the #FF0000 color.
myCombo. SetHoverFontColor ("#FF0000");
// Get the color.
alert(myCombo. GetHoverFontColor ());
```

- **Supported version**

  7.0.0.68

ImeMode Method

- **Function**

  Set or check the IME that will be applied by default, when a focus is put on the combo input area. However, this function is available for Internet Explorer only.

  The setting value is as follows;

Setting value	Description
auto	Use the configured IME mode without change (default).
active	Set the IME mode to Korean when a focus is put (Korean/English switching)
inactive	Set the IME mode to English when a focus is put (Korean/English switching)
disabled	Only English input is allowed (no Korean/English switching)

- **Syntax**

Syntax	Get	Obj.**GetImeMode**()
	Set	Obj.**SetImeMode**(Mode)

- **Info**

Return	String, Setting value (for Get Method)		
Parameter	Type	Required?	Description
Mode	String	Required	IME setting value (default: “auto")

- **Example**

```
// Set Korean to be selected first when a focus is put.
myCombo.SetImeMode("active");
// Set to English only mode when a focus is put.
myCombo.SetImeMode("disabled");
```

- **Supported version**

  7.0.0.40

Init Method

Function

Specify the column to be used as a reference in the item. When the item of the column specified as a reference is selected, the text of that column will be displayed.

Syntax

Syntax	Obj.**Init**(showCol)

Info

Return	none		
Parameter	Type	Required?	Description
showCol	Integer	Optional	The column that is used as the output of the selected item. (default: 0)

Example

```
// Call the Init function for the OnLoad event of the Body.(default: 0)
myCombo.Init(0);
```

Supported version

7.0.0.0

InsertItem Method

- **Function**

Add an item. Set the text value using a “|” separator when using multicolumn.

Add to the last row, if the Index parameter is -1.

- **Syntax**

Syntax	Obj.**InsertItem**(Index, Text, Code);

- **Info**

Return	Integer, row index of the added item		
Parameter	Type	Required?	Description
Index	Integer	Required	Row index of the item to add
Text	String	Required	Text of the item to add
Code	String	Optional	Code of the item to add If not entered, the Index value is converted to a string and used as the code value.

- **Example**

```
// Add an item to the first row.
myCombo.InsertItem(0, “section manager", "01");
// Add an item to the last row.
myCombo.InsertItem(-1, “section manager", "01");
// Add an item if there are several columns (3 columns).
myCombo.InsertItem(0, “general manager|team manager|sales department", "01");

// Convert the index value to a string and use as the code value.
myCombo.InsertItem(0, “item 1”);
```

- **Supported version**

7.0.0.0

InsertBatchItem

- **Function**

  Pass the text and code as the property of the object, which is passed when calling InsertItem.

  Add to the last row, if the Index parameter is -1.

  Unlike InsertItem, InsertBatchItem doesn't modify the width of the column automatically when adding an item. Therefore, the width of the column should be entered when only InsertBatchItem is used.

- **Syntax**

Syntax	Obj.**InsertBatchItem**(index, data, colWidth);

- **Info**

Return	Integer, row index of the added item		
Parameter	Type	Required?	Description
index	Integer	Required	Row index of the item to add
data	Array	Required	Item to add
colWidth	Array	Optional	Column width of the drop down list

- **Example**

```
// Add an item to the first row.
myCombo.InsertBatchItem(0, [
   {text: “Foo|Bar”, code: “Baz”},
   {text: ”Foo|Bar”, code: “Baz”},
   {text: “Foo|Bar”, code: ”Baz”}]);

// Set the column width of the drop down list.
myCombo.InsertBatchItem(0, [
   {text: “Foo|Bar”, code: “Baz”},
   {text: ”Foo|Bar”, code: “Baz”},
   {text: “Foo|Bar”, code: ”Baz”}],
   [80, 80]);
```

- **Supported version**

  7.0.0.57

ItemCheck Method

- **Function**

  Select a specific item in the check box as an index or code value, or get the status of item selection, when MultiSelect is selected.

- **Syntax**

Syntax	Get	Obj.**GetItemCheck**(Index_Code)
	Set	Obj.**SetItemCheck**(Index_Code, Flag, Event)

- **Info**

Return	Boolean,　Setting value (for Get Method)		
Parameter	Type	Required?	Description
Index_Code	String	Required	Value of a specific index or code
Flag	Boolean	Required	Selection/Clear status
Event	Boolean	Optional	Event handler calling status (default: true)

- **Example**

```
myCombo.SetMultiSelect(true);
// Select the check box of the item corresponding to the “01” code.
myCombo.SetItemCheck("01", true);

// Select the check box of the item corresponding to the “02” code, and
// don’t call the event handler.
myCombo.SetItemCheck(“02”, true, false)

// Get the check box selection status of the item corresponding to the “02” code.
alert(myCombo.GetItemCheck("02"));
```

- **Supported version**

  7.0.0.0

ItemHeight Method

- **Function**

  Check or set the height of the item row that is displayed when expanding a combo item.

  It must be set in numbers. A string is not supported.

- **Syntax**

Syntax	Get	Obj.**GetItemHeight**()
	Set	Obj.**SetItemHeight**(Value)

- **Info**

Return	Integer, Setting value (for Get Method)		
Parameter	Type	Required?	Description
Value	Integer	Required	Set the height of the item row (default: 21px)

- **Example**

```
// Set item row height to 30.
myCombo.SetItemHeight(30);
// Get item row height.
alert(myCombo.GetItemHeight());
```

- **Supported version**

  7.0.0.0

ItemImage Method

- **Function**

  Set an image for the image column that is displayed when UseImage is set to True.

  Set an image for the image column of a specific row index, using the row index.

- **Syntax**

Syntax	Obj.**ItemImage**(Index, Url);

- **Info**

Return	None		
Parameter	Type	Required?	Description
Index	Integer	Required	Row index
Url	String	Required	Path of the image file

- **Example**

```
// Set an image.
myCombo.SetUseImage(true);
myCombo.ItemImage(0, "../Image/comboitem.gif");
```

- **Supported version**

  7.0.0.0

LineColor Method

Function

Check or set the color of the line inside the drop down area of the combo.

The value must be set with a web color string.

Syntax

Syntax	Get	Obj.**GetLineColor**()
	Set	Obj.**SetLineColor**(Color)

Info

Return	String, setting value (for Get Method)		
Parameter	Type	Required?	Description
Color	String	Required	Color value WebColor (e.g., #FF0000)

Example

```
// Set the color to the #FF0000 color.
myCombo.SetLineColor("#FF0000");
// Get the color.
alert(myCombo.GetLineColor());
```

Supported version

7.0.0.0

MaxLength Method

- **Function**

  Check or set the length of the combo’s input string.

- **Syntax**

Syntax	Get	Obj.**GetMaxLength**()
	Set	Obj.**SetMaxLength**(Value)

- **Info**

Return	Integer, Setting value (for Get Method)		
Parameter	Type	Required?	Description
Value	Integer	Required	Maximum length setting value of the string

- **Example**

```
// Set the maximum length of the string.
myCombo.SetMaxLength(10);
// Get the maximum length setting value of the string.
alert(myCombo.GetMaxLength());
```

- **Supported version**

  7.0.0.0

MaxSelect Method

- **Function**

  Check of set the number of items that can be selected to the maximum, when MultiSelect is True.

- **Syntax**

Syntax	Get	Obj.**GetMaxSelect**()
	Set	Obj.**SetMaxSelect**(Value)

- **Info**

Return	Integer, Setting value (for Get Method)		
Parameter	Type	Required?	Description
Value	Integer	Required	The number of maximum multi-selection

- **Example**

```
// Set the number of maximum multi-selection to 3.
myCombo.SetMultiSelect(true);
myCombo.SetMaxSelect(3);
// Get the number of maximum multi-selection.
alert(myCombo.GetMaxSelect());
```

- **Supported version**

  7.0.0.0

MultiSelect Method

Function

Check or set the multi-selection status of the item. For multi-selection, the check box column is displayed to the left side of the dropdown list, so that several items can be selected.

Syntax

Syntax	Get	Obj.**GetMultiSelect**()
	Set	Obj.**SetMultiSelect**(Value)

Info

Return	Boolean,　Setting value (for Get Method)		
Parameter	Type	Required?	Description
Value	Boolean	Required	Multi-selection status

Example

```
// Set the multi-selection status.
myCombo.SetMultiSelect(true);
// Get the multi-selection setting status.
alert(myCombo.GetMultiSelect());
```

Supported version

7.0.0.0

MultiSeparator Method

Function

Check and set the separator that separates the multiple selected data, when MultiSelect is used. The result of getting or setting SelectCode, SelectIndex, and SelectText differs, and the text displayed on the combo also differs, depending on the setting of this property.

Syntax

Syntax	Get	Obj.**GetMultiSeparator**()
	Set	Obj.**SetMultiSeparator**(Value)

Info

Return	String,　Setting value (for Get Method)		
Parameter	Type	Required?	Description
Value	String	Required	Multi-selection separator

Example

```
// Set “/” as a multi-selection separator.
myCombo.SetMultiSelect(true);
myCombo.SetMultiSeparator("/");
// Get the multi-selection separator.
alert(myCombo.GetMultiSeparator());
```

Supported version

7.0.0.0

NarrowDDListShowBtn Method

- **Function**

  Set the area that control the showing/hiding of the multicombo drop down list. If set to True, the showing/hiding of the drop down list can be toggled by clicking the button located at the right side of multicombo. If set to False, the showing/hiding of the drop down list can be toggled by clicking the entire multicombo area.

- **Syntax**

Syntax	Get	Obj.**GetNarrowDDListShowBtn** ()
	Set	Obj.**SetNarrowDDListShowBtn** (Value)

- **Info**

Return	Boolean, Setting value (for Get Method)		
Parameter	Type	Required?	Description
Value	Boolean	Required	Set the area used as a show/hide toggle button of the drop down list. (default: false)

- **Example**

```
// Control the show/hide toggling of the drop down list
// only when the right button of the multicolumn is clicked.
myCombo. SetNarrowDDListShowBtn (true);
// Get the show/hide area setting value of the drop down list.
alert(myCombo. GetNarrowDDListShowBtn ());
```

- **Supported version**

  7.0.0.62

OutLineColor Method

Function

Check or set the color of the control outline.

Make user to set with a web color string. An error occurs when set as an integer type (e.g., EE, 255)

Syntax

Syntax	Get	Obj.**GetOutLineColor**()
	Set	Obj.**SetOutLineColor**(Color)

Info

Return	String, Setting value (for Get Method)		
Parameter	Type	Required?	Description
Color	String	Required	Color value WebColor (e.g., #FF0000)

Example

```
// Set the color to the #FF0000 color.
myCombo.SetOutLineColor("#FF0000");
// Get the color.
alert(myCombo.GetOutLineColor());
```

Supported version

7.0.0.0

PlaceHolder4CheckAll Method

- **Function**

  Set or check the display string that will show instead of the text value or “X item selected” message, if CheckAll is selected from MultiSelect.

- **Syntax**

Syntax	Get	Obj.**GetPlaceHolder4CheckAll**()
	Set	Obj.**SetPlaceHolder4CheckAll**(Value)

- **Info**

Return	String, Setting value (for Get Method)		
Parameter	Type	Required?	Description
Value	String	Required	The string that will be displayed only when all items are selected.

- **Example**

```
// Display “All” when all items are selected.
myCombo.SetPlaceHolder4CheckAll(“All”);
```

- **Supported version**

  7.0.0.99

RemoveAll Method

Function

Remove all items.

Syntax

Syntax	Obj.**RemoveAll**();

Info

Return	None		
Parameter	Type	Required?	Description
None			

Example

```
// Remove all items.
myCombo.RemoveAll();
```

Supported version

7.0.0.0

SaveName Method

- **Function**

Set or check the name of the combo code that is sent, when submitted.

When submitted, the code and text of the item selected from multicombo will be sent, a query string is combined and sent like “[Control ID]_code=*code value of the selected item*&[Control ID]_text=*text value of the selected item*”. Use the pertinent method to rename [Control ID]_code at this time.

For example, if myCombo.SetSaveName("deptcode") is set, the code value is 02, and the item of which text is “sales department” is selected,
a query string is combined and sent like “deptcode=02&myCombo_text=sales department”.

- **Syntax**

Syntax	Get	Obj.**GetSaveName**()
	Set	Obj.**SetSaveName**(SaveName)

- **Info**

Return	String, Setting value (for Get Method)		
Parameter	Type	Required?	Description
SaveName	String	Required	The name of the combo code to use when sending a query. (Default value: [Control ID]_code)

- **Example**

```
// Set the combo code name to deptcode.
myCombo.SetSaveName("deptcode");
// Get the name of the combo code.
alert(myCombo.GetSaveName());
```

➢ Supported version

7.0.0.19

SelectBackColor Method

➢ Function

Check or set the color to display the item to select from the list when the combo is expanded, or that is applied when displaying the selected item.

The value must be set with a web color string.

➢ Syntax

Syntax	Get	Obj.**GetSelectBackColor**()
	Set	Obj.**SetSelectBackColor**(Color)

➢ Info

Return	String,　Setting value (for Get Method)		
Parameter	Type	Required?	Description
Color	String	Required	Color value WebColor (e.g., #FF0000)

➢ Example

```
// Set the color to the #FF0000 color.
myCombo.SetSelectBackColor("#FF0000");
// Get the color.
alert(myCombo.GetSelectBackColor());
```

➢ Supported version

7.0.0.0

SelectCode Method

- **Function**

  Check or set the code of the currently selected item. Set or check the multiple selection value if MultiSelect is selected, according to the MultiSeparator separator.

  (e.g., if the “01|02|03” code value is set to -1 or “”, the selected item is cleared.)

  If True is entered, select as many as the number of items that can be selected.

- **Syntax**

Syntax	Get	Obj.**GetSelectCode**()
	Set	Obj.**SetSelectCode**(Code, [Event])

- **Info**

Return	String, Setting value (for Get Method)		
Parameter	Type	Required?	Description
Code	String	Required	Code value of the item to select (e.g., “02")
Event	Boolean	Optional	OnSelect, OnChange Event occurrence status true: OnChange occurs (default). false: OnChange doesn’t occur.

- **Example**

```
myCombo.SetSelectCode("04", true);    // Select an item with the “04” code.
alert(myCombo.GetSelectCode());     //Check the code of the selected item.
.
// Set in such way that no item is selected.
myCombo.SetSelectCode( -1, true);
myCombo.SetSelectCode("", true);
.
myCombo.SetSelectCode(true);    // Select all items.
```

- **Supported version**

  7.0.0.0

SelectFontColor Method

- **Function**

  Check or set the color to display the item to select from the list when the combo is expanded, or that is applied when displaying the selected item.

  The value must be set with a web color string.

- **Syntax**

Syntax	Get	Obj.**GetSelectFontColor**()
	Set	Obj.**SetSelectFontColor**(Color)

- **Info**

Return	String, Setting value (for Get Method)		
Parameter	Type	Required?	Description
Color	String	Required	Color value WebColor (e.g., #FF0000)

- **Example**

```
// Set the font color to the #FF0000 color.
myCombo.SetSelectFontColor("#FF0000");
// Get the font color.
alert(myCombo.GetSelectFontColor());
```

- **Supported version**

  7.0.0.0

SelectIndex Method

- **Function**

Check or set the index of the currently selected item. Set or check the multiple selection value if MultiSelect is selected, according to the MultiSeparator separator.

(e.g., if the “1|2|3” index value is set to -1 or “”, the selected item is cleared.)

If True is entered, select as many as the number of items that can be selected.

- **Syntax**

Syntax	Get	Obj.**GetSelectIndex**()
	Set	Obj.**SetSelectIndex**(Index, [Event])

- **Info**

Return	String, Setting value (for Get Method)		
Parameter	Type	Required?	Description
Index	String	Required	Index of the item to select (e.g., “2")
Event	Boolean	Optional	OnSelect, OnChange Event occurrence status true : OnChange occurs (default). false : OnChange doesn’t occur.

- **Example**

```
myCombo.SetSelectIndex("4", true); // Select an item with the “4” index.

alert(myCombo.GetSelectIndex());     // Check the index of the selected item.

// Set in such way that no item is selected.
myCombo.SetSelectIndex(-1, true);
myCombo.SetSelectIndex("", true);
myCombo.SelectIndex(true);        // Select all items.
```

- **Supported version**

  7.0.0.0

SelectText Method

- **Function**

  Check or set the text of the currently selected item. Set or check the multiple selection value if MultiSelect is selected, according to the MultiSeparator separator.

  (e.g., if the “Seoul|Busan|Gwangju” text value is set to -1 or “”, the selected item is cleared.)

  If True is entered, select as many as the number of items that can be selected.

  Set the text value of the column that will be displayed when an item is selected, if multicolumn is used.

- **Syntax**

Syntax	Get	Obj.**GetSelectText**()
	Set	Obj.**SetSelectText**(Text, [Event])

- **Info**

Return	String, setting value (for Get Method)		
Parameter	Type	Required?	Description
Text	String	Required	Text of the item to select (e.g., “Seoul”)
Event	Boolean	Optional	OnSelect, OnChange Event occurrence status true : OnChange occurs (default). false : OnChange doesn’t occur.

- **Example**

```
myCombo.SetSelectText("Seoul", true);     // Select an item with the “Seoul” text.
alert(myCombo.GetSelectText());           // Check the text of the selected item.
// Set in such way that no item is selected.
myCombo.SetSelectText(-1, true);
myCombo.SetSelectText("", true);
myCombo.SelectText(true);        // Select all items.
```

- **Supported version**

  7.0.0.0

SetColAlign Method

- **Function**

  Set the alignment of a specific column.

- **Syntax**

Syntax	Obj.**SetColAlign**(Col, Align);

- **Info**

Return	None		
Parameter	Type	Required?	Description
Col	Integer	Required	Column index
Align	String	Required	Alignment value 왼쪽 : "left" 가운데 : "center" 오른쪽 : "right"

- **Example**

```
// Set central alignment to the No. 1 column.
myCombo.SetColAlign(1, "center");
```

- **Supported version**

  7.0.0.0

SetColWidth Method

- **Function**

  Set the width of a specific column.

  If the width of a specific column is set by calling SetColWidth, the width of each column set inside multicombo is cleared, and column width entered by the user will be set.

- **Syntax**

Syntax	Obj.**SetColWidth**(Col, Width);

- **Info**

Return	None		
Parameter	Type	Required?	Description
Col	Integer	Required	Column index
Width	Integer	Required	Width value

- **Example**

```
// Set the width of the column No. 1 to 150.
myCombo.SetColWidth(1, 150);
```

- **Supported version**

  7.0.0.0

SetPrependIE7 Method

- **Function**

  Set whether prepend will be used or not for IE7.

- **Syntax**

Syntax	Obj. **SetPrependIE7** (isPrepend);

- **Info**

Return	None		
Parameter	Type	Required?	Description
isPrepend	Boolean	Required	For IE7, set the prepend status of the combo to solve the location change problem. However, the speed can be reduced if there is a lot of data. This method can be set globally.

- **Example**

```
// Set prepend.
myCombo. SetPrependIE7 (true);
```

- **Supported version**

  7.0.0.98

SetRequired Method

- **Function**

Set or clear the Required property, which is used to check mandatory input in the FormQueryString function.

If the Flag parameter is set to True, the Required property is set. If set to False, the Required property is cleared.

If the Required property is set, a warning window appears if there is no input value when mandatory input is checked by the FormQueryString function. The ReqValue parameter sets the name of the control to display in this case. The multicombo control ID is the default value.

If the multicombo control ID is myCombo, the following display messages are used according to the setting value.

Setting value	Display message
myCombo.SetRequired(true);	“myCombo”is a required field.
myCombo.SetRequired(true, "Department");	“Department”is a required field.
myCombo.SetRequired(false);	A warning message is not displayed, as Required check box is not selected.

- **Syntax**

Syntax	Obj.**SetRequired**(Flag, [ReqValue])

- **Info**

Return	None		
Parameter	Type	Required?	Description
Flag	Boolean	Required	Required property setting status
ReqValue	String	Optional	Name of the control to display when the Required check box is selected.

			(Default: multicombo control ID)

- **Example**

```
// Set the Required property.
myCombo.SetRequired(true);

// Set the Required property, and set the control name to display when the check box is selected to “Department”.
myCombo.SetRequired(true, "Department");

// Clear the Required property.
myCombo.SetRequired(false);
```

- **Supported version**

  7.0.0.19

SetSkip4CheckAll Method

- **Function**

  Set in such a way that only “Select All” can be set, when Select All is clicked.

- **Syntax**

Syntax	Obj. **SetSkip4CheckAll** (isSync);

- **Info**

Return	None		
Parameter	Type	Required?	Description
isSync	Boolean	Required	If set to True, only “Select All” can be clicked, when Select All is clicked. If set to False, all data can be selected.

- **Example**

```
// Set prepend.
myCombo. SetSkip4CheckAll (true);
```

- **Supported version**

  7.0.0.100

SyncRequest Method

Function

Set the sync and async request status when using GetItemJson and GetItemXml.

Syntax

Syntax	Get	Obj.**GetSyncRequest**()
	Set	Obj.**SetSyncRequest**(sync)

Info

Return	boolean, Setting value (for Get Method)		
Parameter	Type	Required?	Description
sync	Boolean	Required	Sync request status

Example

```
myCombo.SetSyncRequest(true);
console.log(“foo”);
myCombo.GetItemJson(“sample.json”, undefined, function() {
   console.log(“bar”);
});
console.log(“baz”);
// Display on the console in order of foo -> bar -> baz.

myCombo.SetSyncRequest(false);
console.log(“foo”);
myCombo.GetItemJson(“sample.json”, undefined, function() {
   console.log(“bar”);
});
console.log(“baz”);
// Display on the console in order of foo -> baz -> bar.
```

Supported version

7.0.0.57

Text Method

Function

Set or check the text value of a specific item.

Syntax

Syntax	Get	Obj.**GetText**(Index_Code, Col)
	Set	Obj.**SetText**(Index_Code, Col, Text)

Info

Return	String, setting value (for Get Method)		
Parameter	Type	Required?	Description
Index_Code	Integer/String	Required	Index or code value
Col	Integer	Required	Column index
Text	String	Required	Text to change

Example

```
// Set the text of the first column of the first row to “Seoul”.
myCombo.SetText(1, 1, "Seoul");
// Set the text of the first column of which code value is “02” to “Seoul”.
myCombo.SetText("02",1,"Seoul");
// Check the text of the first column of the first row.
alert(myCombo.GetText(1, 1));
```

Supported version

7.0.0.0

Theme Method

- **Function**

  Change the theme of the pertinent multicombo.

- **Syntax**

Syntax	Get	Obj.**GetTheme**()
	Set	Obj.**SetTheme**(path)

- **Info**

Return	Theme name (for Get Method)		
Parameter	Type	Required?	Description
path	String	Required	Input the path of the theme.

- **Example**

```
// Load “ibmulticombo.css” in the “Mono” directory, which is entered based on the “ibmulticombo.js” path,
// and change the theme.
myCombo.SetTheme(“Mono”);

// Load “ibmulticombo.css” in the “Theme/Mono” directory, which is entered based on the “ibmulticombo.js” path,
// and change the theme.
myCombo.SetTheme(“Theme/Mono”);

// Return the name of the theme that is used by myCombo.
myCombo.GetTheme();
```

- **Supported version**

  7.0.0.57

Title Method

Function

Set the text value of the title that is displayed when TitleVisible is set to True.

Syntax

Syntax	Get	None
	Set	Obj.**SetTitle**(Value)

Info

Return	None		
Parameter	Type	Required?	Description
Value	String	Required	Title to change (e.g., “country|capital|language")

Example

```
// Set a column composed of three columns.
myCombo.SetTitleVisible(true);
myCombo.SetTitle(”country|capital|language");
```

Supported version

7.0.0.0

TitleBackColor Method

Function

Set or check the background color of the title that is displayed when TitleVisible is set to True.

The value must be set with a web color string.

Syntax

Syntax	Get	Obj.**GetTitleBackColor**()
	Set	Obj.**SetTitleBackColor**(Color)

Info

Return	String, Setting value (for Get Method)		
Parameter	Type	Required?	Description
Color	String	Required	Color value WebColor (e.g., #FF0000)

Example

```
// Set the background color of the title.
myCombo.SetTitleVisible(true);
myCombo.SetTitleBackColor("#FF0000");
// Get the background color of the title.
alert(myCombo.GetTitleBackColor());
```

Supported version

7.0.0.0

TitleFontColor Method

- **Function**

  Set or check the font color of the title that is displayed when TitleVisible is set to True.

  The value must be set with a web color string.

- **Syntax**

Syntax	Get	Obj.**GetTitleFontColor**()
	Set	Obj.**SetTitleFontColor**(Color)

- **Info**

Return	String, Setting value (for Get Method)		
Parameter	Type	Required?	Description
Color	String	Required	Color value WebColor (e.g., #FF0000)

- **Example**

```
// Set the font color of the title.
myCombo.SetTitleVisible(true);
myCombo.SetTitleFontColor("#FF0000");
// Get the font color of the title.
alert(myCombo.GetTitleFontColor());
```

- **Supported version**

  7.0.0.0

TitleVisible Method

Function

Set or check whether the title row will be displayed or not.

Syntax

Syntax	Get	Obj.**GetTitleVisible**()
	Set	Obj.**SetTitleVisible**(Value)

Info

Return	Boolean,　Setting value (for Get Method)		
Parameter	Type	Required?	Description
Value	Boolean	Required	Title display status

Example

```
// Display the title.
myCombo.SetTitleVisible(true);
// Get the title display status.
alert(myCombo.GetTitleVisible());
```

Supported version

7.0.0.0

UnEditableBackColor Method

- **Function**

  Check or set the background color when the control cannot be edited (Editable = false).

  The value must be set with a web color string.

- **Syntax**

Syntax	Get	Obj.**GetUnEditableBackColor**()
	Set	Obj.**SetUnEditableBackColor**(Color)

- **Info**

Return	String, color value WebColor that is set (for Get Method)		
Parameter	Type	Required?	Description
Color	String	Required	Color value WebColor (e.g., #FF0000)

- **Example**

```
// Set the color value to “#FF0000” if the control cannot be edited.
myCombo.SetEditable(false);
myCombo.SetUnEditableBackColor("#FF0000");
// Get the color of “Uneditable”.
alert(myCombo.GetUnEditableBackColor());
```

- **Supported version**

  7.0.0.19

UnEditableFontColor Method

- **Function**

  Check or set the font color when the control cannot be edited (Editable = false).

  The value must be set with a web color string.

  The pertinent font color will be set in the combo input window only when the control cannot be edited (Editable = false).

- **Syntax**

Syntax	Get	Obj.**GetUnEditableFontColor**()
	Set	Obj.**SetUnEditableFontColor**(Color)

- **Info**

Return	String, color value WebColor that is set (for Get Method)		
Parameter	Type	Required?	Description
Color	String	Required	Color value WebColor (e.g., #FF0000)

- **Example**

```
// Set the font color value to “#FF0000” if the control cannot be edited.
myCombo.SetEditable(false);
myCombo.SetUnEditableFontColor("#FF0000");
// Get the font color of “Uneditable”.
alert(myCombo.GetUnEditableFontColor());
```

- **Supported version**

  7.0.0.19

UseAutoComplete Method

- **Function**

  Set or check whether automatic item search is available or not, by editing the combo item.

  There are several limitations when using this method. The limitations are as follows.

  1) Enabled for MultiSelect(false) only. (Multiple selection is also supported from 107 version.)

  2) If the same content is entered while entering the data in the Edit window, only the pertinent item will be displayed.

  3) A dotted line is displayed under the character when entering the Korean character in Firefox or Opera. In this case, it is regarded that a word is not composed and an event doesn't occur. Therefore, it cannot be compared.

  Therefore, complete a Korean word by moving the cursor using the left/right arrow key when entering Korean characters in Firefox or Opera.

  4) For multicolumn, the method is executed according to the data of the column that is displayed when an item is selected.

- **Syntax**

Syntax	Get	Obj.**GetUseAutoComplete**()
	Set	Obj.**SetUseAutoComplete**(Value,ignoreUpperCase)

- **Info**

Return	Boolean, Setting value (for Get Method)		
Parameter	Type	Required?	Description
Value	Boolean	Required	The status of automatic item search while editing
ignoreUpperCase	Boolean	Optional	Case-sensitiveness

- **Example**

```
// Use.
myCombo.SetMultiSelect(false);
myCombo.SetUseAutoComplete(true);
alert(myCombo.GetUseAutoComplete());
```

- **Supported version**

  7.0.0.0 (Case-sensitiveness 7.0.0.107)

UseEdit Method

- **Function**

  Set or check whether the edited characters will be kept or not.

  Enabled only when MultiSelect = false.

- **Syntax**

Syntax	Get	Obj.**GetUseEdit**()
	Set	Obj.**SetUseEdit**(Value)

- **Info**

Return	Boolean, Setting value (for Get Method)		
Parameter	Type	Required?	Description
Value	Boolean	Required	true: Use/ false: Don't use true: Keep the edited characters even though they do not exist in the combo list. If the edited characters exist in the combo list, they can be checked all using the GetSelectText, GetSelectIndex, and GetSelectCode function of the pertinent item. If the edited characters don't exist in the combo list, the GetSelectText function checks the edited characters without change, and the GetSelectIndex function returns -1, and the GetSelectCode function returns "". The function of searching by entering a code is also kept, and the OnChange event also occurs when the value is changed. False: The edited characters are kept only when it exists in the combo list or combo code. If the edited characters

			don’t exist, the entered characters are cleared and the blank character (“”) is displayed.

- **Example**

```
// Use.
myCombo.SetMultiSelect(false);
myCombo.SetUseEdit(true);
// Get the use status.
alert(myCombo.GetUseEdit());
```

- **Supported version**

7.0.0.0

UseImage Method

Function

Set or check whether the image column is used or not.

Syntax

Syntax	Get	Obj.**GetUseImage**()
	Set	Obj.**SetUseImage**(Value)

Info

Return	Boolean, Setting value (for Get Method)		
Parameter	Type	Required?	Description
Value	Boolean	Required	Image column use status

Example

```
// Use the image column.
myCombo.SetUseImage(true);
// Get the image column use status.
alert(myCombo.GetUseImage());
```

Supported version

7.0.0.0

UseSelectCount Method

- **Function**

  Set or check whether the text value will be displayed or "X item selected" will be displayed

  as the selected item when selecting multiple items.

- **Syntax**

Syntax	Get	Obj.**GetUseSelectCount**()
	Set	Obj.**SetUseSelectCount**(Value)

- **Info**

Return	Boolean, setting value (for Get Method)		
Parameter	Type	Required?	Description
Value	Boolean	Required	Whether the selected item will be displayed as "X item selected" or not.

- **Example**

```
// Display as "X item selected".
myCombo.SetMultiSelect(true);
myCombo.SetUseSelectCount(true);
// Check the setting value.
alert(myCombo.GetUseSelectCount());
```

- **Supported version**

  7.0.0.34

ValidChar Method

Function

Limit the characters that can be entered in the element, which displays the string of the selected item.

Syntax

Syntax	Obj.ValidChar(validChar1, validChar2);

Info

Return	None		
Parameter	Type	Required?	Description
validChar1	String	Required	Input limit setting validChar1 1: Input alphabets only. 2: Input uppercase alphabets only. 3: Input lowercase alphabets only. 4: Input Korean characters only. 5: Input numbers only.
validChar2	String	Optional	Set validChar2 when setting more after setting validChar1. 1: Numbers 2: Special character (!,@,#,$,^,&,%,*,(,),+,=,- ,[,],\,/,{,},|,:,<,>,?, , , .) 3: Number + special character

Example

```
myCombo.validChar( “1” );       // Input alphabets only.
myCombo.validChar( “1”, “1” );   // Input alphabets + numbers.
myCombo.validChar( “2”, “3” );   // Input uppercase alphabets + numbers + special characters.
myCombo.validChar( “4”, “2” );    // Input Korean characters + special characters.
```

- **Supported version**

  7.0.0.61

ViewText Method

- **Function**

  Set or check the text that will be discretionally displayed by the user, not the text of the selected item. Set “” to display the text of the selected item, as it is now.

- **Syntax**

Syntax	Get	Obj.**GetViewText**()
	Set	Obj.**SetViewText**(Value)

- **Info**

Return	String, Setting value (for Get Method)		
Parameter	Type	Required?	Description
Value	String	Required	Text that will be discretionally displayed by the user.

- **Example**

```
// Example of changing text to display each time an item is selected.
function myCombo_OnChange(oldIndex, oldText, oldCode, newIndex, newText,
newCode) {
        var sCode = myCombo.GetSelectCode();
        var sCodeLen = sCode.split("|").length;

        if (sCode != "" && sCodeLen > 1) {
                // Display “X item selected” if several items are selected.
                myCombo.SetViewText(sCodeLen + “item is selected");
        } else {
                // Display the selected item, if there is only one selected item.
                myCombo.SetViewText("");
        }
}
```

- **Supported version**

  7.0.0.36

Visble Method

- **Function**

  Set or check whether the control is visible or not.

- **Syntax**

Syntax	Get	Obj.**GetVisible**()
	Set	Obj.**SetVisible**(Value)

- **Info**

Return	Boolean, setting value (for Get Method)		
Parameter	Type	Required?	Description
Value	Boolean	Required	Control visibility

- **Example**

```
// Hide the control
myCombo.SetVisible(false);
// Get control visibility.
alert(myCombo.GetVisible());
```

- **Supported version**

  7.0.0.0