

IME Module

Published 2014-10-28 | (Compatible with SDK 2.5,3.5,4.5,5.0,5.1 and 2011,2012,2013,2014 models)

Enables text input in applications.

Contents

[Methods](#)

[Global values](#)

class IME()

Methods

IMEShell()

Creates IME object.

Arguments:	inputObjId – id of the HTML input element callbackFunc (Function) – called when the IME object has been fully created context – this.context (optional)
Returns:	(Object) IME object

ime.getInputObj()

Gets the HTML input object attached to the IME object.

Returns:	(Object) input object from IME object
----------	--

ime.setAnyKeyFunc(func)

Sets a function to be called when any key is pressed.

Arguments:	func (Function) – callback function with following parameter: keyCode - code of the key pressed
------------	--

```
var callback = function (keyCode) {  
    alert("Key pressed, code " + keyCode);  
}  
imeobj.setAnyKeyFunc(callback);
```

ime.setKeyFunc(keyCode, func)

Sets a callback function to be called when a specific key is pressed.

Some keys are usually handled by IME itself. For example, numeric keys are used for text input, and the left and right keys are used for moving the cursor. It is possible to register callbacks for these special keys using setKeyFunc(). IME will call the callback function first, before handling the key itself. If the callback function returns true (or null), it will continue as normal to handle the key and take the usual action (like

entering text or moving the cursor). If the callback function returns false, IME will not take action for this key. This allows the client to block certain default behaviour of IME (like cursor movement). This feature should be used with care as it prevents normal functioning of IME.

Arguments:	keyCode – code for the specific key that should trigger the callback func (Function) – callback function with following parameters keyCode: code of the key pressed
------------	--

```
var tvKey = new Common.API.TVKeyValue(),
callback = function (keyCode) {
  if (keyCode == tvKey.KEY_LEFT) {
    alert("Blocking left key");
    return false;
  } else {
    alert("Key code received: " + keyCode + ", allow IME to process");
    /* Do some action here based on keyCode */
    return true;
  }
}
```

```
imeobj.setKeyFunc(tvKey.KEY_LEFT, callback);
imeobj.setKeyFunc(tvKey.KEY_INFO, callback);
imeobj.setKeyFunc(tvKey.KEY_MENU, callback);
```

ime.setKeypadPos(x, y, z)

Sets position of the IME keypad window

Arguments:	x (Number) – horizontal offset of the IME keypad from the left edge of the screen, in pixels y (Number) – vertical offset of the IME keypad from the top edge of the screen, in pixels z (Number) – [optional] z-index offset of the IME keypad. This value deside CSS z-index of the keypad. (default: 2100)
------------	--

```
imeobj.setKeypadPos(500, 40, 2100);
```

ime.setQWERTYPos(x, y, z)

Sets position of the display of the IME QWERTY keypad window

Arguments:	x (Number) – horizontal offset of the IME keypad display from the left edge of the screen, in pixels y (Number) – vertical offset of the IME keypad display from the top edge of the screen, in pixels z (Number) – [optional] z-index offset of the IME keypad. This value deside CSS z-index of the keypad. (default: 2100)
------------	--

```
imeobj.setQWERTYPos (500, 40, 2100);
```

ime.setEnterFunc(func)

Sets a callback function that will be called when the enter key is pressed on the input box

This is commonly used to indicate that entry of text on a form is completed

Arguments:	func (Function) – callback function with the following parameters: string - string currently entered in the input box, no return value
------------	---

```
var callback = function (string) {
  alert("User pressed enter, final text is " + string);
}
```

```
imeobj.setEnterFunc(callback);
```

ime.setMaxLengthFunc(func)

Sets a callback function that will be called when the maximum length of the HTML input element is reached.

The maximum length is specified as the HTML property “maxlength”. This callback will not be triggered if a “maxlength” property has not been set for the input element. Settting the maxlength property is required for 2010 platforms, but recommended to use on all platforms.

See also

[How to implement correctly the IME module?](#)

Troubleshooting common problems with IME module.

Arguments:	func (Function) – callback function with no parameters or return value
------------	---

```
var callback = function () {  
    alert("Maximum length reached");  
}  
imeobj.setMaxLengthFunc(callback);  
ime.setPrevEventOnLeftFunc(func)
```

Sets a callback function that will be called when the left key is pressed and the cursor position is at the start of the input box.

Arguments:	func (Function) – callback function with no parameters or return value
------------	---

```
var callback = function () {  
    alert("Left key pressed at left side of input box");  
}  
imeobj.setPrevEventOnLeftFunc(callback);  
ime.setOnCompleteFunc(func)
```

Method sets a callback function that will be called when the entry of one character is completed.

This function will be called when:

1. Right key pressed
2. Delete key pressed (delete key is indicated by IME display)
3. A space character is added
4. Entry of a character is completed

For example, in a text type <input>, the user may press a numeric key 3 times to choose the correct character. This callback will not be triggered until a short time is elapsed after this, or the user presses a different numeric key - this means that the user has chosen to keep this character in the string, and so the entry of one character is completed. This callback can be used, for example, to create a list of suggestions for completing entries in a search box.

Arguments:	func (Function) – callback function with no parameters or return value
------------	---

```
var callback = function () {  
    alert("Entry of a character is complete");  
}  
imeobj.setOnCompleteFunc(callback);  
ime.setBlockSpace([block])
```

Blocks entry of the space character in an input box. For example, this can be useful when entering a user id or password.

Arguments:	block (Boolean) – [optional] Sets whether space should be blocked or not true: IME will not allow the space character to be entered false: IME will allow the space character to be entered
------------	--

```
imeobj.setBlockSpace (true);  
ime.setString(string)
```

Sets the value attribute of the <input> associated with the IME object.

This function should be used instead of directly setting the value property of the HTML input box (element.value="..."). Directly setting the value property will cause unexpected behaviour of IME.

Arguments:	string (String) – string to display in the input box
------------	---

```
imeobj.setString("Hello world");  
ime.setKeypadChangeFunc(string, func)
```

Set a callback function that will be called when exchange to 12key or qwerty with the IME object.

Change position of the HTML Inputbox div when the IME keypad change event times.

Arguments:	string (String) – keypad layer type, '12key' or 'qwerty' func (Function) – callback function with no parameters or return value
------------	--

```
imeobj.setKeypadChangeFunc('12key', move12KeyPosition);
```

```
imeobj.setKeypadChangeFunc('qwerty', moveqwertyPosition);
```

```
ime.setInputHighlightFunc(func)
```

Set a callback function that will be called when input box highlight.

12key mode - always highlight focus

qwerty mode - move highlight focus with remocon direction keypress

Arguments:	func (Function) – callback function with no parameters or return value
------------	---

```
imeobj.setInputHighlightFunc(inputHighlight);
```

```
ime.setMode(string)
```

Set the input mode for start 12key keypad with the IME object.

Only '12key' mode - set input mode

_latin_small - small letters: "abcd", "boy", "girl"

_latin_cap - first letter capital: "Abcd", "Boy", "Girl"

_latin_big - capital letters: "ABCD", "BOY", "GIRL"

_num - numeric letters: 1, 2, 3, 4

_special - special letters: !, @, #, \$

string: Only 12key keypad input mode type.

Global: _latin_small, _latin_cap, _latin_big, _num, _special

Korean: _kor, _latin_small, _latin_big, _num, _special

Password : _latin_small, _latin_big, _num, _special

SDK : _latin_small, _latin_big, _num, _special

Returns:	(Boolean) true or false, true is successfull, false is not
----------	---

```
if (ret = imeobj.setMode("_num")) {  
    alert("ret : true " + ret);  
}
```

```
ime._focus()
```

Set show and start of IME keypad when mouse use.

IME show and start function is defferent when Mouse use on Apps. If Apps not use mouse, show IME keypad by focus() event. If Apps use mouse, show IME keypad by _focus() function.

Note

This function is supported on 2012 TV only

```

if (_g_ime.pluginMouse_use_YN) {
    imeobj._focus();
    document.getElementById('searchText1').focus();
} else {
    document.getElementById('searchText1').focus();
}
ime._blur()

```

Set hide and close of IME keypad when mouse use.

IME show and start function is different when Mouse use on Apps. If Apps not use mouse, hide IME keypad by blur() event. If Apps use mouse, hide IME keypad by _blur() function.

Note

This function is supported on 2012 TV only

```

if (_g_ime.pluginMouse_use_YN) {
    imeobj._blur();
    imeobj2._focus();
}
document.getElementById('searchText2').focus();

```

Global values

`_g_ime.dim_use_YN`

(Boolean)

default: true

Sets whether or not dimmed background div should be used with IME when mouse is used.

If mouse is used in app, focus moves to input box when background dim is clicked.

It's Global value of IME object.

Add position: before new IMEShell function.

Note

This value is supported on 2012 TV only

```

_g_ime.dimm_use_YN = true;
ime = new IMEShell("searchText1", ime_init_text, this);
if (!ime) {
    alert("object for IMEShell create failed", 3);
}
_g_ime.Recog_use_YN

```

default: true

Shows or hides voice recognition guide message of IME keypad bottom depending on the value of `_g_ime.Recog_use_YN`. Sets whether voice recognition is used or not.

If TV support voice recognition, Apps can set show or hide voice recognition guide message of IME keypad bottom.

It's Global value of IME object.

Voice recognition is not supported on SDK.

Add position: before new IMEShell function.

Note

This value is support by 2012th TV only

```
_g_ime.recog_use_YN = false;  
ime = new IMEShell("searchText1", ime_init_text,this);  
if (!ime) {  
    alert("object for IMEShell create failed", 3);  
}
```

_g_ime.pluginMouse_use_YN

Get mouse support with the IME object.

(Boolean) default: true

Mouse use or not use on Apps.

If Apps support mouse, Apps can get status or mouse use.

It's Global value of IME object.

Add position: after new IMEShell function

Note

This value is only support by 2012th TV

```
if (_g_ime.pluginMouse_use_YN) {  
    imeobj._focus();  
    document.getElementById('searchText1').focus();  
} else {  
    document.getElementById('searchText1').focus();  
}
```