LG Electronics

[Tutorial] LG Web_Controlling Voice Recognition

Version 1.0 – October 2012

LGDEV-083

Home Entertainment Company
LG Electronics, Inc.

Copyright

About This Document

Revision History

Document Version	Date	Comment
1.0	October 11, 2012	Initial Version

Purpose

This document describes how to control the voice recognition of the Magic Remote Control in web applications using LG Web Open API.

Reference Documents

Refer to the following documents:
- LG Web Application Development Guide
- LG Web Open API Reference Guide

Conventions

Codes

Source code and examples are indicated in the `grey Courier New` font.

Note, Caution

Note and caution are used to emphasize information.
The following samples describe when each is used.

Note

Contains information about something that is helpful to you.

Caution

Contains important information about something that you should know.

Abbreviation

The following table defines the abbreviations used in this document.

Abbreviation	Description
API	Application Programming Interface

Contents

Tables

Figures

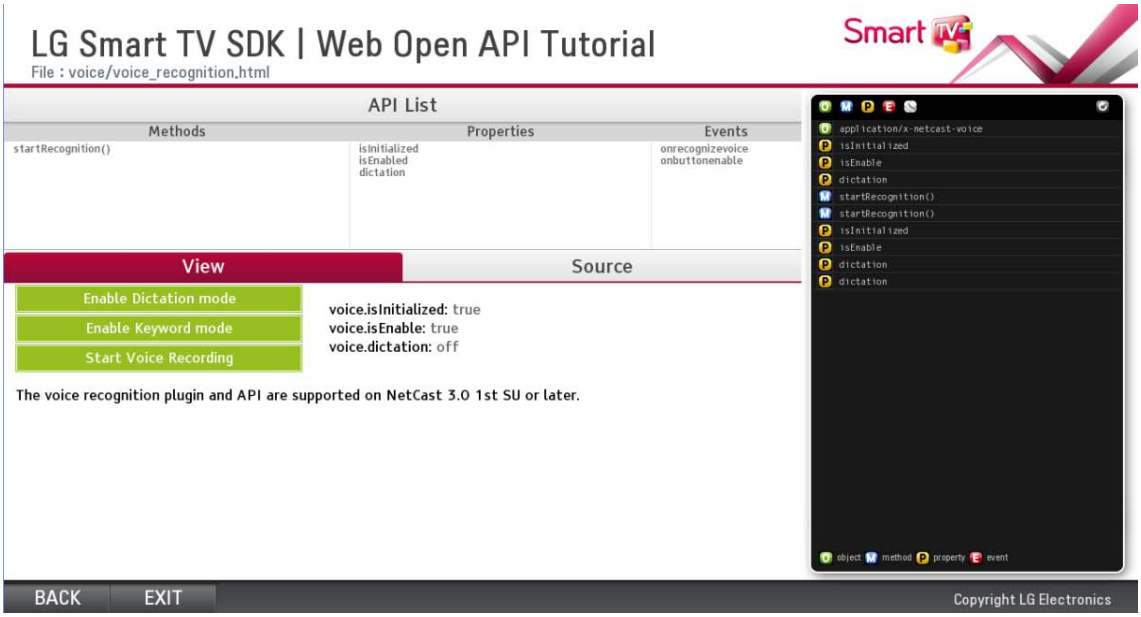
1 Introduction

This chapter provides an overview of this application and needed APIs.

1.1 Overview

This application is designed to show which method, properties, and events of LG Web Open API are used to use the voice recognition function of the Magic Remote Control on web applications in LG Smart TV.

There are two modes for the voice recognition function: dictation and keyword. (In keyword mode, the search keyword is selected from the voice recognition word list.) The language of the voice recognition function can be set under OPTION > Language > Voice Search Language. Now, it supports Dutch, English, French, German, Italian, Korean, Norwegian, Russian, Spanish, Swedish and UK English. Also, occurred and used events are shown, when status of voice recognition object is changed.



[Figure 1] Web Application using Voice Recognition API

1.2 Needed APIs

This application uses following Web Open API:

[Table 1] Description of the Needed APIs

API Class	Name	Description
Method	startRecognition	Call native UI of the voice recognition function and receive the result as an event.
Property	isInitialized	Determine whether the voice recognition function is initialized and returns the value as Boolean.
	isEnable	Determine whether the MRCU is paired (including its type) and whether the voice recognition function is enabled, and returns the value as Boolean.
	dictation	Determine whether the voice recognition function is in dictation mode and returns the value as string (on/off). If the return value is “off”, the function is in keyword mode. This methods can also be used to select the mode.
Event	onrecognizevoice	Receive the voice recognition result from the TV.
	onbuttonenable	Enable or disable the voice recognition button

For more information on these functions, refer to “LG Web Open API Reference Guide”.

Note

Log is used for checking the sequence of Web Open API; this will not be covered in the sample code description.

2 Creating Application

This chapter describes how to use the voice recognition function using Web open API.

2.1 Initializing the Page

Use the **initPage** function to set the basic functions of the application.

04: Record the last visited page when running the application.
07: Initialize the page.
08: Get the source code of the page using the XMLHttpRequest object.
09: Initialize the Log function.
11: Check NetCast version.
14-15: Add event handlers which will be executed when the corresponding button is pressed.
16-17: Add event handlers related to voice.
20: Call function to initialize voice.
21: Call function to show property values.

Sample Code

```
01 : function initPage()
02 :  {
03 :         //save page as last visited page
04 :         setLastVisitPage();
05 :
06 :         //common initialize function
07 :         commonInitialize();
08 :         requestSourceCode();
09 :         jsLog.initLG();
10 :
11 :         checkNetCastVersion();
12 :
13 :         //add onclick event handler
14 :         addEventHandler(document.getElementById("btn_back"),     "click",
     onClickHandler);
15 :         addEventHandler(document.getElementById("btn_exit"),     "click",
     onClickHandler);
16 :         addEventHandler(document.getElementById("dictation_mode"),
     "click", onDictationModeHandler);
17 :         addEventHandler(document.getElementById("keyword_mode"),
     "click", onKeywordModeHandler);
18 :
19 :         jsLog.lgobject('application/x-netcast-voice');
20 :         initVoice();
21 :         showPropResult();
22 :  }
```

2.2 Check NetCast Version

Use **checkNetCastVersion** function to check NetCast version as the voice recognition plugin and API are supported on NetCast 3.0 1st SU or later.

03: Declare nBrowserVersion.
04-08: If return value of nBrowserVersion is 4, then hide the visibility of all three buttons :
a. dictation_mode
b. keyword_mode
c. start_Recording
11-17: If return value of nBrowserVersion is equal or greater than 5, then display the names for supported method, property and event in API list area.

Sample Code

```
01 :   function checkNetCastVersion()
02 :   {
03 :   var nBrowserVersion = getBrowserVersion();
04 :   if(nBrowserVersion == 4) // NetCast 2.0
05 :   {
06 :   document.getElementById('dictation_mode').style.visibility = "hidden";
07 :   document.getElementById('keyword_mode').style.visibility   =   "hidden";

08 :   document.getElementById('start_Recording').style.visibility = "hidden";

09 :   }
10 :   else if(nBrowserVersion >= 5) // NetCast 3.0
11 :   {

12 :   setInnerTextById("APIdescription", "The voice recognition plugin and
   API are supported on NetCast 3.0 1st SU or later.");
13 :   setInnerTextById("method","startRecognition()");
14 :   setInnerTextById("property","isInitialized<br>
   isEnabled<br>dictation");
15 :   setInnerTextById("events","onrecognizevoice<br>onbuttonenable");
16 :   }
01 :   }
```

2.3 Initializing the Voice Recognition

03: Declare device. Some LG Smart TVs support voice recognition of the Magic-RCU.
04: Declare voice.
06: device.supportVoiceRecog: Developers can check whether or not supports voice recognition in TV by using the “supportVoiceRecog" read-only property in the Device Information plugin object. It returns true if LG Smart TV supports voice recognition, otherwise, it returns false.
voice.isInitialized: Determine whether the voice recognition function is initialized and returns the value as Boolean.
voice.isEnable: Determine whether the MRCU is paired (including its type) and whether the voice recognition function is enabled, and returns the value as Boolean.
13: If any one of three conditions is true, then only handle event for ‘Start Recording’ button.
15-16: Handle voice events.

Sample Code

```
01 :     function initVoice()
02 :       {
03 :              var device = document.getElementById("device");
04 :              var voice = document.getElementById('voice');
05 :
06 :              if(!(device.supportVoiceRecog || voice.isInitialized ||
         voice.isEnable))
07 :              {
08 :                     //alert("em");
09 :     document.getElementById("start_Recording").className="executeBigBut
         tonOff";
10 :              }
11 :              else
12 :              {
13 :     addEventHandler(document.getElementById("start_Recording"),
         "click", voiceHandler);
14 :              }
15 :              voice.onrecognizevoice = recognizeVoiceHandler;
16 :              voice.onbuttonenable = buttonStateHandler;
17 :       }
```

2.4 Handling Button Click Events

The following functions are event handlers which were added in **initPage** function.

dictationModeHandler
This function is called when 'dictation mode' button is clicked.

03: Declare voice.
04: Set 'dictation' ON to use dictation property of voice recognition function. By default, it is in keyword mode.
05: Call function to show property values.

keywordModeHandler
This function is called when 'keyword mode' button is clicked.

11: Declare voice.
12: Set 'dictation' OFF to use keyword property of voice function.
13: Call function to show property values.

Sample Code

```
01 :   function onDictationModeHandler()
02 :         {
03 :               var voice = document.getElementById('voice');
04 :               voice.dictation = "on";
05 :               showPropResult()
06 :               jsLog.lgproperty('dictation');
07 :         }
08 :
09 :         function onKeywordModeHandler()
10 :         {
11 :               var voice = document.getElementById('voice');
12 :               voice.dictation = "off";
13 :               showPropResult()
14 :               jsLog.lgproperty('dictation');
15 :         }
```

The following functions are event handlers which were added in **initVoice** function.

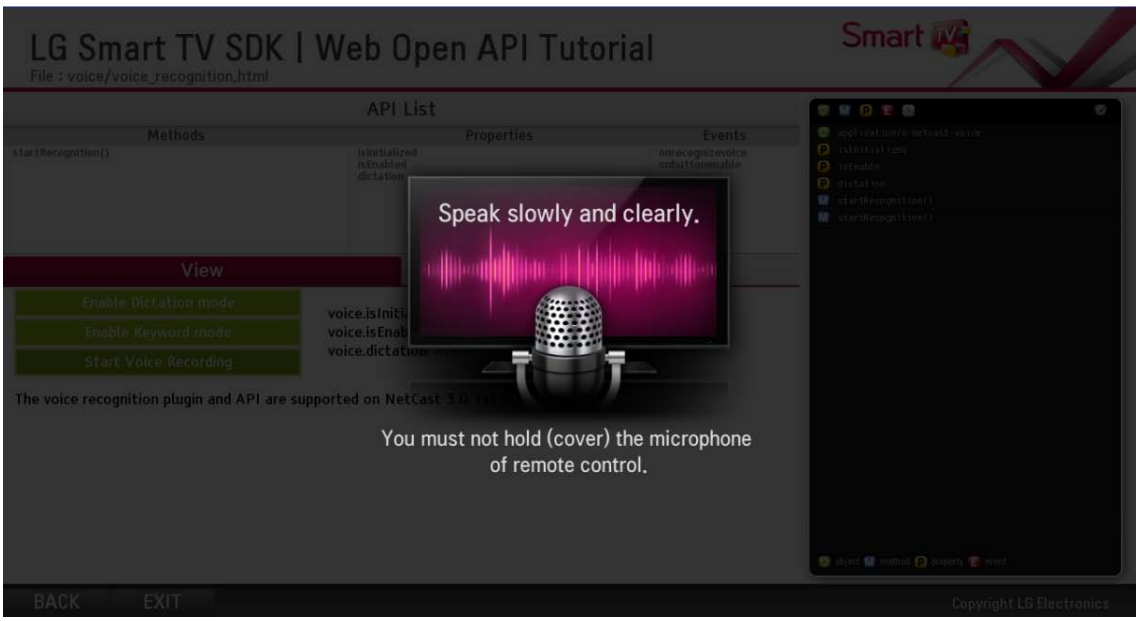
voiceHandler
This function is called when 'Start Recording' button is clicked.

03: Declare voice.
04-07: If voice recognition function is enabled, call startRecognition function and receive the result.

Sample Code

```
01 :   function voiceHandler()
02 :         {
03 :               var voice = document.getElementById('voice');

04 :               if(voice.isEnable == true)
05 :               {
06 :                     voice.startRecognition();
07 :                     jsLog.lgmethod('startRecognition()');
08 :               }
09 :         }
```



[Figure 2] Start Recording Voice

2.5 Handling Voice Events

The following functions are voice event handlers which were added in **initVoice** function.

recognizeVoiceHandler
This function is handler of onrecognizevoice event , in order to receive the voice recognition result from the TV.

04-11: If event occurs, display its value in log window. Otherwise display “No Event”.

buttonStateHandler
This function is handler of onbuttonenable event , in order to enable or disable the voice recognition button. Upon pairing, the event receives the availability of the voice recognition according to the MRCU type from the TV. If the MRCU with the voice recognition disabled is paired with the RV, false is returned; otherwise, true. The voice recognition button is enabled or disabled based on the value returned by this event.

19-21: If event value is true, then enable “Start Recording” button.
22: Add event handler for “Start Recording” button.
23: If event value is false, then disable “Start Recording” button.
29: Remove event handler for “Start Recording” button.

Sample Code

```
01 :   function recognizeVoiceHandler(e)
02 :       {
03 :         console.log(">O< recognizeVoiceHandler is called.");
04 :         if(e)
05 :         {
06 :                result = e;
07 :                jsLog.lgevent('recognizeVoiceHandler : '+'result = '+
           result);
08 :         }
09 :         else
10 :         {
11 :                jsLog.lgevent('recognizeVoiceHandler : '+' No Event');
12 :         }
13 :       }
14 :
15 :       function buttonStateHandler(e)
16 :       {
17 :         if(e)
18 :         {
19 :                if(e == true)
20 :                {
21 :                       console.log(">O< buttonStateHandler is called");
22 :
             document.getElementById("start_Recording").className="executeBig
           Button";
23 :
             addEventHandler(document.getElementById("start_Recording"),
           "click", voiceHandler);
24 :                       jsLog.lgevent('buttonStateHandler : '+' true');
25 :                }
26 :                else
27 :                {
28 :
             document.getElementById("start_Recording").className="executeBig
           ButtonOff";
29 :
             removeEventHandler(document.getElementById("start_Recording"),
           "click", voiceHandler);
30 :                       jsLog.lgevent('buttonStateHandler : '+' false');
31 :                }
32 :         }
33 :       }
```

2.6 Displaying Text

The following code displays the voice recognition object information on screen by using properties.

showPropResult

04: Display the value of ‘voice.isInitialized’ according to the return value of ‘voice.isInitialized’ property.
06: Display the value of ‘voice.isEnable’ according to the return value of ‘voice.isEnable’ property.
08: Display the value of ‘voice.dictation’ according to the return value of ‘voice.dictation’ property.

Sample Code

```
01 :   function showPropResult(){
02 :                var msgArea = document.getElementById('displayArea');
03 :                var htmlStr="";
04 :                htmlStr+=(voice.isInitialized!=undefined)     ?     "<font
   style='color:#000000;         font-weight:bold'>voice.isInitialized:</font>
   "+voice.isInitialized : "";
05 :                jsLog.lgproperty('isInitialized');
06 :                htmlStr+=(voice.isEnable!=undefined)?           "<br><font
   style='color:#000000;             font-weight:bold'>voice.isEnable:</font>
   "+voice.isEnable : "";
07 :                jsLog.lgproperty('isEnable');
08 :                htmlStr+=(voice.dictation!=undefined)?          "<br><font
   style='color:#000000;            font-weight:bold'>voice.dictation:</font>
   "+voice.dictation : "";
09 :                jsLog.lgproperty('dictation');
10 :                msgArea.innerHTML =htmlStr;
11 :         }
```

2.7 Inputting Keys

Use **onUserInput** function is called by onClickHandler function; it receives a key value as the userInput parameter from onClickHandler and creates the corresponding function for each key value to operate the key.

06: When the back key is pressed, it will open previous page.

Sample Code

```
01 : //onUserInput function should be implemented
02 : function onUserInput(userInput)
03 : {
04 :  switch(userInput)
05 :  {
06 :   case VK_BACK :
07 :         window.location.replace("../main_menu.html");
08 :         break;
09 :
10 :  }
11 : }
```

2.8 Setting Voice Recognition Object

The following code shows how to set voice recognition object.

02: Set data type. Refer to “LG Web Application Development Guide” for related information.
03: Set id property as ‘voice’.
04: The dictation property can be used when the voice recognition function is in dictation mode. The default is the keyword mode.

Sample Code

```
01 : <object
02 : type="application/x-netcast-voice"
03 : id="voice"
04 : dictation=”on”>
05 : </object>
```

2.9 Source Code of voice_recognition.html

Source code of voice_recognition.html is as follows:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=EUC-KR">
<title>Voice recognition API Test</title>
<link rel="stylesheet" href="../css/style.css">
<script language="javascript" src="../js/common.js"></script>
<script language="javascript" src="../js/keycode.js"></script>
<script language="javascript" src="../js/menu.js"></script>

<script type="text/javascript" src="../blackbirdjs/blackbird.js"></script>
<link type="text/css" rel="Stylesheet" href="../blackbirdjs/blackbird.css" />

<script>

	//initialize page
	function initPage()
	{
		//save page as last visited page
		setLastVisitPage();

		//common initialize function
		commonInitialize();
		requestSourceCode();
		jsLog.initLG();

		checkNetCastVersion();

		//add onclick event handler
		addEventHandler(document.getElementById("btn_back"), "click", onClickHandler);
		addEventHandler(document.getElementById("btn_exit"), "click", onClickHandler);
		addEventHandler(document.getElementById("dictation_mode"), "click", onDictationModeHandler);
		addEventHandler(document.getElementById("keyword_mode"), "click", onKeywordModeHandler);

		jsLog.lgobject('application/x-netcast-voice');
		initVoice();
		showPropResult();
	}

	function showPropResult(){
		var msgArea = document.getElementById('displayArea');
```

```
            var htmlStr="";
            htmlStr+=(voice.isInitialized!=undefined) ? "<font style='color:#000000; font-weight:bold'>voice.isInitialized:</font>
"+voice.isInitialized : "";
            jsLog.lgproperty('isInitialized');
            htmlStr+=(voice.isEnable!=undefined)? "<br><font style='color:#000000; font-weight:bold'>voice.isEnable:</font>
"+voice.isEnable : "";
            jsLog.lgproperty('isEnable');
            htmlStr+=(voice.dictation!=undefined)? "<br><font style='color:#000000; font-weight:bold'>voice.dictation:</font>
"+voice.dictation : "";
            jsLog.lgproperty('dictation');
            msgArea.innerHTML =htmlStr;
      }

      //onUserInput function should be implemented
      function onUserInput(userInput)
      {
            switch(userInput)
            {
                  case VK_BACK :
                        window.location.replace("../main_menu.html");
                        break;
            }
      }

      function initVoice()
      {
            var device = document.getElementById("device");
            var voice = document.getElementById('voice');

            if(!(device.supportVoiceRecog || voice.isInitialized || voice.isEnable))
            {
                  //alert("em");
                  document.getElementById("start_Recording").className="executeBigButtonOff";
            }
            else
            {
                  addEventHandler(document.getElementById("start_Recording"), "click", voiceHandler);
            }

            voice.onrecognizevoice = recognizeVoiceHandler;
            voice.onbuttonenable = buttonStateHandler;
      }

      function voiceHandler()
      {
```

```
		var voice = document.getElementById('voice');

		if(voice.isEnable == true)
		{
			voice.startRecognition();
			jsLog.lgmethod('startRecognition()');
		}
	}

	function recognizeVoiceHandler(e)
	{
		console.log(">O< recognizeVoiceHandler is called.");
		if(e)
		{
			result = e;
			jsLog.lgevent('recognizeVoiceHandler : '+'result = '+ result);
		}
		else
		{
			jsLog.lgevent('recognizeVoiceHandler : '+' No Event');
		}
	}

	function buttonStateHandler(e)
	{
		if(e)
		{
			if(e == true)
			{
				console.log(">O< buttonStateHandler is called");
				document.getElementById("start_Recording").className="executeBigButton";
				addEventHandler(document.getElementById("start_Recording"), "click", voiceHandler);
				jsLog.lgevent('buttonStateHandler : '+' true');
			}
			else
			{
				document.getElementById("start_Recording").className="executeBigButtonOff";
				removeEventHandler(document.getElementById("start_Recording"), "click", voiceHandler);
				jsLog.lgevent('buttonStateHandler : '+' false');
			}
		}
	}

	function onDictationModeHandler()
	{
```

```
		var voice = document.getElementById('voice');
		voice.dictation = "on";
		showPropResult()
		jsLog.lgproperty('dictation');
	}

	function onKeywordModeHandler()
	{
		var voice = document.getElementById('voice');
		voice.dictation = "off";
		showPropResult()
		jsLog.lgproperty('dictation');
	}

	function checkNetCastVersion()
	{
		var nBrowserVersion = getBrowserVersion();

		if(nBrowserVersion == 4) // NetCast 2.0
		{
			document.getElementById('dictation_mode').style.visibility = "hidden";
			document.getElementById('keyword_mode').style.visibility = "hidden";
			document.getElementById('start_Recording').style.visibility = "hidden";
		}
		else if(nBrowserVersion >= 5) // NetCast 3.0
		{

			setInnerTextById("APIdescription", "The voice recognition plugin and API are supported on NetCast 3.0 1st SU or
later.");
			setInnerTextById("method","startRecognition()");
		setInnerTextById("property","isInitialized<br> isEnabled<br>dictation");
		setInnerTextById("events","onrecognizevoice<br>onbuttonenable");
		 }
	}

</script>
</head>

<body ondragstart='return false' onselectstart='return false' onload="javascript:initPage();">

<!-- title -->
<div class='SuiteTitle' >LG Smart TV SDK | Web Open API Tutorial</div>

<!-- navigation -->
<div class='SuiteNavigation'>
```

```
      <div style="float:left;">File : voice/voice_recognition.html</div>
</div>

<div class='SuiteTitleLine'> </div>

<!-- test contents -->
<div id='content_body' class='ContentArea'>

      <div class='ApiListTitleArea'>API List</div>
      <div class='ApiListArea'>
            <div class='MethodTitleArea'>
                  Methods
                  <div class='MethodListArea' id="method"></div>
            </div>
            <div class='PropertyTitleArea'> Properties
            <div class='PropertyListArea' id="property"> </div>
         </div>
         <div class='EventTitleArea'> Events
            <div class='EventListArea'  id="events"> </div>
         </div>
      </div>

      <div class='ViewTitleArea'>
            <div id='tabViewArea' class='SelectedViewArea' style='float:left;' onclick="showView();">View</div>
            <div id='tabCodeArea' class='UnselectedViewArea' style='float:right;' onclick='showCode();'>Source</div>
      </div>

      <div id='view'>
        <div class='ViewArea'>
            <object
            id="voice"
            type="application/x-netcast-voice"
               width="0"
                        height="0"
                        style="float: left"
         >
       </object>
       <object
            id="device"
            type="application/x-netcast-info"
               width="0"
                        height="0"
                        style="float: left"
         >
       </object>
```

```
              <table >
                     <tr ><td width="350"><div id="dictation_mode" class="executeBigButton" >Enable Dictation mode</div></td>
              <td rowspan="3"><div id="displayArea" class="displayAreaStyle">Hi this is the area for the item</div></td>
              </tr>
                     <tr ><td><div id="keyword_mode" class="executeBigButton">Enable Keyword mode</div></td>
                   </tr>
                     <tr ><td><div id="start_Recording" class="executeBigButton">Start Voice Recording</div></td>
                   </tr>

                     <tr height="50px">
                           <td colspan="2" align=left><div id="APIdescription" ></div></td>
                     </tr>
              </table>

              </div>
       </div>

       <div style="visibility: hidden" id='codeview'>
              <textarea class="SourceCodeArea" value=""  id='sourcecode'></textarea>
       </div>

</div>


<!-- button and copyright -->
<div class='SuiteButtonArea'>

       <!-- button -->
       <div id='btn_back' class='buttonDescription'>BACK</div>

       <!-- exit key description -->
       <div id='btn_exit' class='buttonDescription'>EXIT</div>

       <!-- copyright -->
       <div class='copyright'>Copyright LG Electronics</div>
</div>

</body>
</html>
```