LG Electronics

[Tutorial] LG Web_Executing Window Media Player #1

Version 1.1 – February 2012

LGDEV-050

Home Entertainment Company
LG Electronics, Inc.

Copyright

About This Document

Revision History

Document Version	Date	Comment
1.1	February 13, 2012	Needed APIs and Source codes are updated.
1.0	October 12, 2011	Initial Version

Purpose

This document describes how to execute Window Media Player by using Web Open API of LG Smart TV.

Reference Documents

Refer to the following documents:
- LG Web_Executing Window Media Player #2
- LG Web_Executing Window Media Player #3
- 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

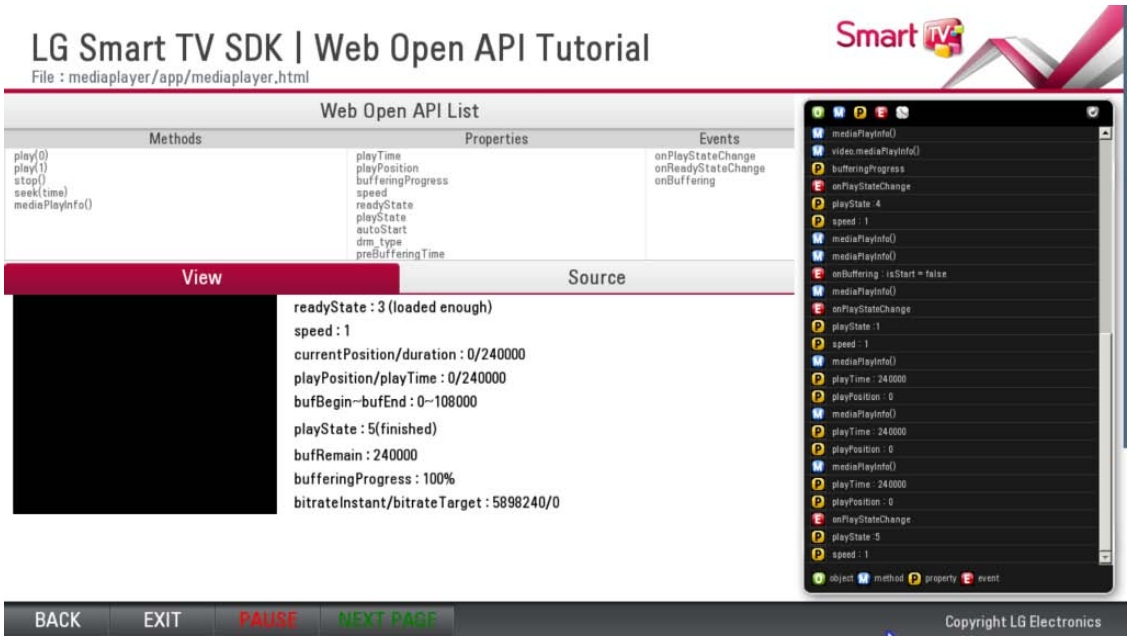
1 Introduction

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

1.1 Overview

This application is designed to shows which method, properties, and events of LG Web Open API are used to get basic information when executing media player in LG Smart TV.

This application shows how to control media and get property values. Also, occurred and used events when media status is changed are shown.



[Figure 1] Window Media Player Execution Application #1

1.2 Needed APIs

This application uses following Web Open API:

[Table 1] Description of the Needed APIs

API Class	Name	Description
Method	play(0)	Pauses media.
	play(1)	Plays media.
	stop()	Stops media.
	seek(time)	Seeks media to specific time.
	mediaPlayInfo	Media playbackinformation
Property	playTime	Total play time of media
	playPosition	Current play position
	bufferingProgress	Buffering status
	Speed	Play speed
	readyState	Returns current media ready status.
	playState	Returns media play status by number.
	autoStart	Set whether media file playout to be started automatically.
	drm_type	Set drm type.
	preBufferingTime	Adjust buffering time before playback.
Event	onPlayStateChange	The event occurs when play status of media item changes.

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 media player 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: Set the page ID.
10: Initialize the Log function.
13-16: Add an event handler which will executed when the corresponding button is pressed.
18: Set text of btn_red to ”PAUSE”.
21: Declare video.
24-28: Add event handlers related to media.

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 :   setPageID("Media Player");
10 :   jsLog.initLG();
11 :
12 :   //add onclick event handler
13 :   addEventHandler(document.getElementById("btn_back"),"click",onClickHandler);
14 :   addEventHandler(document.getElementById("btn_red"),"click", onClickHandler);
15 :   addEventHandler(document.getElementById("btn_green"),"click",onClickHandler);
16 :   addEventHandler(document.getElementById("btn_exit"),"click",onClickHandler);
17 :
18 :   setInnerTextById("btn_red", "PAUSE");
19 :
20 :   //add event handler for test
21 :   var video = document.getElementById("video");
22 :
23 :   //add onReadyStateChange event handler
24 :   video.onReadyStateChange = setReadyState;
25 :
26 :   //add playState event handler
27 :   video.onBuffering = processBufferingStateChangeFunction;
28 :   video.onPlayStateChange = processPlayStateChangeFunction;
29 :
30 :   jsLog.lgobject('application/x-netcast-av');
31 :   jsLog.lgproperty('autoStart');
32 :   jsLog.lgproperty('drm_type');
33 :   jsLog.lgproperty('preBufferingTime');
34 : }
```

2.2 Handling Media Events

The following functions are media event handlers which were added in initPage() function.

setReadyState

This function is handler of onReadStateChange event, displayed whenever readystate is changed.
03-05: Declare video, readyStateValue, and readyStateDesc.
07-11: Describe the value of readystate according to the return value of onReadStateChange event.
13: Display the value of readystate and description.

processBufferingStateChangeFunction

This function is handler of onBuffering event, displayed whenever bufferingstate is changed.
21: Call function that will display buffering information.
23-28: If buffering is started, update the buffering information on screen periodically by calling setBufferingInfo() function while buffering is proceeding.
29-32: If buffering ends, the calling of setBufferingInfo() function ends.

processPlayStateChangeFunction

This function is handler of onPlayStateChange event, displayed whenever playstate is changed.
39-41: Declare playStateExplain, video, and playState.
46: Display property speed.
48-54: Describe the value of playstate according to the return value ofonPlayStateChange event.
56: Display the value of playstate and description.
58-64: While the video is playing, update the playtime information on screen periodically by calling setPlayTimeInfo () function. If playing ends, the calling of setPlayTimeInfo() function ends.

Sample Code

```
01 : function setReadyState(readyState)
02 : {
03 :   var video = document.getElementById("video");
04 :   var readyStateValue = video.readyState;
05 :   var readyStateDesc = "";
06 :
07 :   if(readyStateValue == 0){readyStateDesc = "0 (not set)";}
08 :   else if(readyStateValue == 1){readyStateDesc = "1 (loading)";}
09 :   else if(readyStateValue == 3){readyStateDesc = "3 (loaded enough)";}
10 :   else if(readyStateValue == 4){readyStateDesc = "4 (loaded all)";}
11 :   else{readyStateDesc = readyStateValue + " (unknown state)";}
12 :
13 :   setInnerTextById("ready_state_value", "readyState : " + readyStateDesc);
14 :   jsLog.lgevent('onReadyStateChange : readyState = ' + readyState);
15 : }
16 :
17 : function processBufferingStateChangeFunction(isStarted)
18 : {
19 :   jsLog.lgevent('onBuffering : isStart = ' + isStarted);
20 :
21 :   setBufferingInfo();
22 :
23 :   if(isStarted)
24 :   {
25 :         buffertimer = setInterval(setBufferingInfo, 400);
26 :         jsLog.lgmethod('video.mediaPlayInfo()');
27 :         jsLog.lgproperty('bufferingProgress');
28 :   }
29 :   else
30 :   {
31 :         clearInterval(buffertimer);
32 :   }
33 : }
34 :
35 : function processPlayStateChangeFunction()
36 : {
37 :   jsLog.lgevent('onPlayStateChange');
38 :
```

```
39 :   var playStateExplain = "";
40 :   var video = document.getElementById("video");
41 :   var playState = video.playState;
42 :
43 :   jsLog.lgproperty('playState :' + playState);
44 :   jsLog.lgproperty('speed : ' + video.speed);
45 :
46 :   setInnerTextById("speed_value", "speed : " + video.speed);
47 :
48 :   if(playState == 0){playStateExplain = "stopped";}
49 :   else if(playState == 1){playStateExplain = "playing";}
50 :   else if(playState == 2){playStateExplain = "paused";}
51 :   else if(playState == 3){playStateExplain = "connecting";}
52 :   else if(playState == 4){playStateExplain = "buffering";}
53 :   else if(playState == 5){playStateExplain = "finished";}
54 :   else if(playState == 6){playStateExplain = "error";}
55 :
56 :   setInnerTextById("play_state_value", "playState : " + playState + "(" +
     playStateExplain + ")");
57 :
58 :   if(playState == 1 )
59 :   {      playtimer = setInterval(setPlayTimeInfo,2000);
60 :   }
61 :   else
62 :   {
63 :          clearInterval(playtimer);
64 :   }
65 : }
```

2.3 Controlling Play Mode

The following code shows how to control media play mode.

processRedKey

This function changes play status according to the value of the testProcess.
01: Initialize testProcess.
05: Declare video.
07-41: Using switch-case, control the play mode according to the value of testProcess.
Set text of Red button for each status.
42: Whenever Red button is pressed, the value of testProcess is increased by 1.
43: Reinitialize testProcess if the value gets bigger than the setup value.

Sample Code

```
01 : var testProcess = 2;
02 :
03 : function processRedKey()
04 : {
05 :  var video = document.getElementById("video");
06 :
07 :  switch(testProcess)
08 :  {
09 :         case 0 :
10 :                setInnerTextById("btn_red",   "Press   RED-key   to   play
   video.");
11 :
12 :         case 1 :
13 :                video.play(1);
14 :                setInnerTextById("btn_red", "PAUSE");
15 :                jsLog.lgmethod('video.play(1)');
16 :                break;
17 :
18 :         case 2 :
19 :                video.play(0);
20 :                setInnerTextById("btn_red", "PLAY");
21 :                jsLog.lgmethod('video.play(0)');
22 :                break;
23 :
24 :         case 3 :
25 :                video.play(1);
26 :                setInnerTextById("btn_red", "SEEK");
27 :                jsLog.lgmethod('video.play(1)');
28 :                break;
29 :
30 :         case 4 :
31 :                video.seek(60 * 1000);
32 :                setInnerTextById("btn_red", "STOP");
33 :                jsLog.lgmethod('video.seek(time) ');
34 :                break;
35 :
36 :         case 5 :
37 :                video.stop();
38 :                setInnerTextById("btn_red",   "Press   RED-key   to   play
   video.");
39 :                jsLog.lgmethod('video.stop()');
40 :                break;
41 :  }
42 :  testProcess++;
43 :  if(testProcess==6) testProcess=1;
44 : }
```

2.4 Displaying Text

The following code displays the media object information by using properties and methods.

setPlayTimeInfo

This function displays playtime information on screen.
05: Declare playInfo and call mediaPlayInfo() function to save media play information into playInfo.
06-07: Display currentPosition, duration, and bufRemain saved in playInfo.
11: Display the values of playtime and playPosition properties.

setBufferingInfo

This function displays buffering information on screen.
16: Declare video.
19: Declare playInfo and call mediaPlayInfo() function to save media play information into playInfo.
21-24: Display the values of buffering properties saved in playInfo.

Sample Code

```
01 : function setPlayTimeInfo()
02 : {
03 :  jsLog.lgmethod('mediaPlayInfo()');
04 :
05 :  var playInfo = document.getElementById("video").mediaPlayInfo();
06 :  setInnerTextById("currentPosition_duration_value",
     "currentPosition/duration  :  "  +  playInfo.currentPosition  +  "/"  +
     playInfo.duration);
07 :  setInnerTextById("buf_remain_value",     "bufRemain     :     "     +
     playInfo.bufRemain);
08 :
09 :  jsLog.lgproperty('playTime : ' + video.playTime);
10 :  jsLog.lgproperty('playPosition : '+ video.playPosition);
11 :  setInnerTextById("playPosition_playTime_value",  "playPosition/playTime  :
     " + video.playPosition + "/" + video.playTime);
12 : }
13 :
14 : function setBufferingInfo()
15 : {
16 :  var video = document.getElementById("video");
17 :
18 :  jsLog.lgmethod('mediaPlayInfo()');
19 :  var playInfo = video.mediaPlayInfo();
20 :
21 :  setInnerTextById("buf_Begin_end_value",    "bufBegin~bufEnd    :    "    +
     playInfo.bufBegin + "~" + playInfo.bufEnd);
22 :  setInnerTextById("buf_remain_value",     "bufRemain     :     "     +
     playInfo.bufRemain);
23 :  setInnerTextById("buf_progress_value",   "bufferingProgress   :   "   +
     video.bufferingProgress + "%");
24 :  setInnerTextById("bitrate_value",  "bitrateInstant/bitrateTarget  :  "  +
     playInfo.bitrateInstant + "/" + playInfo.bitrateTarget);
25 : }
```

2.5 Inputting Key

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, this code is executed.
07: When the Red key is pressed, this code is executed.
08: When the Green key is pressed, this code is executed.

Sample Code

```
01 : <//onUserInput function should be implemented
02 : function onUserInput(userInput)
03 : {
04 :  switch(userInput)
05 :  {
06 :   case VK_BACK : window.location.replace("../menu_mediaPlayer.html"); break;
07 :   case VK_RED : case 82 : processRedKey(); break;
08 :   case VK_GREEN : case 71 : window.location.replace("./mediaplayer2.html");
                    break;
09 :  }
10 : }
```

2.6 Setting Media Object

The following code shows how to set Media object.

03: Set data type. Refer to “LG Web Application Development Guide” for related information.
04: Set the media file playout to be started automatically.
05: Set drm type.
06: Set preBufferingTime.
07-08: Set width and height.

Sample Code

```
01 : <object
02 :   id="video"
03 :   type="application/x-netcast-av"
04 :   autoStart="true"
05 :   drm_type="wm-drm"
06 :   preBufferingTime = 5
07 :   width=300
08 :   height=250
09 :   data="../../mediafile/timer.mp4"
10 :   style="float: left">
11 : </object>
```

2.7 Source Code of mediaplayer.html

Source code of mediaplayer.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=UTF-8">
<title>Windows Media Player API Test Page(1/3)</title>
<link rel="stylesheet" href="../../css/style.css">
<script language="javascript" src="../../js/keycode.js"></script>
<script language="javascript" src="../../js/common.js"></script>
<script language="javascript" src="../../js/menu.js"></script>
<script language="javascript" src="../../js/media.js"></script>

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

<script>

	//initialize page
	var buffertimer;
	var playtimer;

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

		//common initialize function
		commonInitialize();
		requestSourceCode();
		setPageID("Media Player");
		jsLog.initLG();

		//add onclick event handler
		addEventHandler(document.getElementById("btn_back"), "click",
onClickHandler);
		addEventHandler(document.getElementById("btn_red"), "click",
onClickHandler);
		addEventHandler(document.getElementById("btn_green"), "click",
onClickHandler);
		addEventHandler(document.getElementById("btn_exit"), "click",
onClickHandler);

		setInnerTextById("btn_red", "PAUSE");

		//add event handler for test
		var video = document.getElementById("video");

		//add onReadyStateChange event handler
		video.onReadyStateChange = setReadyState;

		//add playState event handler
		video.onBuffering = processBufferingStateChangeFunction;
		video.onPlayStateChange = processPlayStateChangeFunction;

		jsLog.lgobject('application/x-netcast-av');
```

```
            jsLog.lgproperty('autoStart');
            jsLog.lgproperty('drm_type');
            jsLog.lgproperty('preBufferingTime');
      }

      function setReadyState(readyState)
      {
            var video = document.getElementById("video");
            var readyStateValue = video.readyState;
            var readyStateDesc = "";

            if(readyStateValue == 0){readyStateDesc = "0 (not set)";}
            else if(readyStateValue == 1){readyStateDesc = "1 (loading)";}
            else if(readyStateValue == 3){readyStateDesc = "3 (loaded
enough)";}
            else if(readyStateValue == 4){readyStateDesc = "4 (loaded
all)";}
            else{readyStateDesc = readyStateValue + " (unknown state)";}

            setInnerTextById("ready_state_value", "readyState : " +
readyStateDesc);
            jsLog.lgevent('onReadyStateChange : readyState = ' +
readyState);
      }

      function processBufferingStateChangeFunction(isStarted)
      {
            jsLog.lgevent('onBuffering : isStart = ' + isStarted);
            setBufferingInfo();

             if(isStarted)
             {
                  buffertimer = setInterval(setBufferingInfo, 400);
                  jsLog.lgmethod('video.mediaPlayInfo()');
                  jsLog.lgproperty('bufferingProgress');
             }
             else
             {
                  clearInterval(buffertimer);
             }
      }

      function processPlayStateChangeFunction()
      {
            jsLog.lgevent('onPlayStateChange');

            var playStateExplain = "";
            var video = document.getElementById("video");
            var playState = video.playState;

            jsLog.lgproperty('playState :' + playState);
            jsLog.lgproperty('speed : ' + video.speed);
            setInnerTextById("speed_value", "speed : " + video.speed);

            if(playState == 0){playStateExplain = "stopped";}
            else if(playState == 1){playStateExplain = "playing";}
            else if(playState == 2){playStateExplain = "paused";}
            else if(playState == 3){playStateExplain = "connecting";}
            else if(playState == 4){playStateExplain = "buffering";}
            else if(playState == 5){playStateExplain = "finished";}
            else if(playState == 6){playStateExplain = "error";}
```

```
		setInnerTextById("play_state_value", "playState : " + playState
+ "(" + playStateExplain + ")");

		if(playState == 1 ){
			playtimer = setInterval(setPlayTimeInfo,2000);
		}
		else {
			clearInterval(playtimer);
		}
	}

	var testProcess = 2;
	function processRedKey()
	{
		var video = document.getElementById("video");

		switch(testProcess)
		{
			case 0 :
				setInnerTextById("btn_red", "Press RED-key to play
video.");

			case 1 :
				video.play(1);
				setInnerTextById("btn_red", "PAUSE");
				jsLog.lgmethod('video.play(1)');
				break;

			case 2 :
				video.play(0);
				setInnerTextById("btn_red", "PLAY");
				jsLog.lgmethod('video.play(0)');
				break;

			case 3 :
				video.play(1);
				setInnerTextById("btn_red", "SEEK");
				jsLog.lgmethod('video.play(1)');
				break;

			case 4 :
				video.seek(60 * 1000);
				setInnerTextById("btn_red", "STOP");
				jsLog.lgmethod('video.seek(time) ');
				break;

			case 5 :
				video.stop();
				setInnerTextById("btn_red", "Press RED-key to play
video.");
				jsLog.lgmethod('video.stop()');
				break;
		}
		testProcess++;
		if(testProcess==6) testProcess=1;
	}

	function setPlayTimeInfo()
	{
		jsLog.lgmethod('mediaPlayInfo()');
		var playInfo =
document.getElementById("video").mediaPlayInfo();
```

```
            setInnerTextById("currentPosition_duration_value",
"currentPosition/duration : " + playInfo.currentPosition + "/" +
playInfo.duration);
            setInnerTextById("buf_remain_value", "bufRemain : " +
playInfo.bufRemain);

            jsLog.lgproperty('playTime : ' + video.playTime);
            jsLog.lgproperty('playPosition : '+ video.playPosition);
            setInnerTextById("playPosition_playTime_value",
"playPosition/playTime : " + video.playPosition + "/" + video.playTime);
      }

      function setBufferingInfo()
      {
            var video = document.getElementById("video");

            jsLog.lgmethod('mediaPlayInfo()');
            var playInfo = video.mediaPlayInfo();

            setInnerTextById("buf_Begin_end_value", "bufBegin~bufEnd : " +
playInfo.bufBegin + "~" + playInfo.bufEnd);
            setInnerTextById("buf_remain_value", "bufRemain : " +
playInfo.bufRemain);
            setInnerTextById("buf_progress_value", "bufferingProgress : " +
video.bufferingProgress + "%");
            setInnerTextById("bitrate_value",
"bitrateInstant/bitrateTarget : " + playInfo.bitrateInstant + "/" +
playInfo.bitrateTarget);
      }

      //onUserInput function should be implemented
      function onUserInput(userInput)
      {
            switch(userInput)
            {
                  case VK_BACK :
window.location.replace("../menu_mediaPlayer.html"); break;
                  case VK_RED : case 82 : processRedKey(); break;
                  case VK_GREEN : case 71 :
window.location.replace("./mediaplayer2.html"); break;
            }
      }

</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 : mediaplayer/app/mediaplayer.html</div>
</div>

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

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

```
	<div class='ApiListTitleArea'>Web Open API List</div>
	<div class='ApiListArea'>
		<div class='MethodTitleArea'>
			Methods
			<div class='MethodListArea'>
				play(0)<br>
				play(1)<br>
				stop()<br>
				seek(time)<br>
				mediaPlayInfo()
			</div>
		</div>
		<div class='PropertyTitleArea'>
			Properties
			<div class='PropertyListArea'>
				playTime<br>
				playPosition<br>
				bufferingProgress<br>
				speed<br>readyState<br>
				playState<br>
				autoStart<br>
				drm_type<br>
				preBufferingTime
			</div>
		</div>
		<div class='EventTitleArea'>
			Events
			<div class='EventListArea'>
				onPlayStateChange<br>
				onReadyStateChange<br>
				onBuffering<br>

			</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="video"
			type="application/x-netcast-av"
			autoStart="true"
			drm_type="wm-drm"
			preBufferingTime = 5
			width=300
			height=250
			data="../../mediafile/timer.mp4"
			style="float: left">
		</object>

		<table border="0" cellpadding="0" cellspacing="0"
style="position: relative; left: 10px; width:450px; height:250px;">
			<tr height="11%" >
				<td ><div class="eachTestGuide "
id="ready_state_value" >readyState :</div></td>
```

```
                    </tr>
                    <tr height=11%>
                          <td ><div class="eachTestGuide "
id="speed_value">speed :</div></td>
                    </tr>
                    <tr height="11%">
                          <td ><div class="eachTestGuide "
id="currentPosition_duration_value">currentPosition/duration :</div></td>
                    </tr>
                    <tr height="11%">
                          <td><div class="eachTestGuide "
id="playPosition_playTime_value">playPosition/playTime :</div></td>
                    </tr>
                    <tr height="11%">
                          <td ><div class="eachTestGuide "
id="buf_Begin_end_value">bufBegin - bufEnd :</div></td>
                    </tr>
                    <tr>
                          <td><div class="eachTestGuide "
id="play_state_value">playState :</div></td>
                    </tr>
                    <tr height="11%">
                          <td><div class="eachTestGuide "
id="buf_remain_value">bufRemain :</div></td>
                    </tr>
                    <tr height="11%">
                          <td><div class="eachTestGuide "
id="buf_progress_value">bufferingProgress :</div></td>
                    </tr>
                    <tr height="11%">
                          <td><div class="eachTestGuide "
id="bitrate_value">bitrateInstant/bitrateTarget :</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>

      <!-- red key description -->
      <div id='btn_red' class='buttonDescription redColor'>PLAY</div>

      <!-- green key description -->
      <div id='btn_green' class='buttonDescription greenColor'>NEXT
PAGE</div>

      <!-- copyright -->
```

```
        <div class='copyright'>Copyright LG Electronics</div>
</div>

</body>
</html>
```