

## **[Tutorial] LG Web\_Executing Window Media Player #3**

---

Version 1.0 – October 2011

**LGDEV-052**

Home Entertainment Company  
LG Electronics, Inc.

## Copyright

**Copyright © 2011 LG Electronics, Inc. All Rights Reserved.**

Though every care has been taken to ensure the accuracy of this document, LG Electronics, Inc. cannot accept responsibility for any errors or omissions or for any loss occurred to any person, whether legal or natural, from acting, or refraining from action, as a result of the information contained herein. Information in this document is subject to change at any time without obligation to notify any person of such changes.

LG Electronics, Inc. may have patents or patent pending applications, trademarks copyrights or other intellectual property rights covering subject matter in this document. The provision of this document does not give the recipient or reader any license to these patents, trademarks copyrights or other intellectual property rights.

No part of this document may be communicated, distributed, reproduced or transmitted in any form or by any means, electronic or mechanical or otherwise, for any purpose, without the prior written permission of LG Electronics, Inc.

This document is subject to revision without further notice.

All brand names and product names mentioned in this document are trademarks or registered trademarks of their respective owners.

# About This Document

## Revision History

Document Version	Date	Comment
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 #1
- LG Web\_Executing Window Media Player #2
- 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

<b>1</b>	<b>Introduction.....</b>	<b>6</b>
1.1	Overview .....	7
1.2	Needed APIs .....	8
<b>2</b>	<b>Creating Application.....</b>	<b>9</b>
2.1	Initializing the Page .....	10
2.2	Inputting Key .....	11
2.3	Displaying Text.....	12
2.4	Setting Media Object .....	14
2.5	Source Code of mediaPlayer3.html .....	15

## Tables

[Table 1] Description of the Needed APIs.....	8
---	---

## Figures

[Figure 1] Window Media Player Execution Application #3 .....	7
---	---



# 1 Introduction

---

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

1.1 Overview

1.2 Needed APIs

## 1.1 Overview

Following the “LG Web\_Executing Window Media Player #2”, this application is designed to show 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 get property values.

LG Smart TV SDK | Web Open API Tutorial  
File : /mediaplayer/app/mediaplayer3.html

Smart TV

Web Open API List		
Methods	Properties	Events
stop() play()	data error	onError

View Source

Expected Error Code

Error Description

Result

Press Red-key to test error code.

object method property event

BACK EXIT TEST PREVIOUS PAGE Copyright LG Electronics

[Figure 1] Window Media Player Execution Application #3

## 1.2 Needed APIs

This application uses following Web Open API:

[Table 1] Description of the Needed APIs

API Class	Name	Description
Method	play()	Plays media.
	stop()	Stops media.
Property	data	Returns media URL with String type.
	error	Returns error code if error occurs during media play.
Event	onError	The event occurs when an error occurs during media play.

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 subtitle using Web open API.

2.1 Initializing the Page

2.2 Inputting Keys

2.3 Displaying Text

2.4 Setting Media Object

2.5 Source Code of mediaPlayer3.html

## 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 be executed when the corresponding button is pressed.
- 18: Call function that displays property values on screen.
- 21: Add onError event handler of 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",onClickListener);  
14 :     addEventHandler(document.getElementById("btn_red"),"click", onClickListener);  
15 :     addEventHandler(document.getElementById("btn_green"),"click",onClickListener);  
16 :     addEventHandler(document.getElementById("btn_exit"),"click",onClickListener);  
17 :  
18 :     var video = document.getElementById("video");  
19 :  
20 :     //add onError handler  
21 :     video.onError = processError;  
22 :  
23 :     jsLog.lgobject('application/x-netcast-av');  
24 : }
```

## 2.2 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.

- 05: When the Back key is pressed, this code is executed.
- 06: When the Red key is pressed, this code is executed.
- 07: When the Green key is pressed, this code is executed.

### Sample Code

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

## 2.3 Displaying Text

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

Declaring and initializing variables

- 01: Declare array variable errorsToTest.
- 03-12: Set description for each error code.
- 14: Declare and initialize currentStep.

### setNextStep

This function displays error code and message saved in each step.

- 15: Change currentStep value when this function is called.
- 16-20: When this function is called, the test is started and the button is set to next.
- 22: If the value of currentStep is bigger than the size of array the function ends.
- 24: Declare media.
- 25: Stop media.
- 27-29: Displays error code, description, and result error code of errorsToTest on screen.
- 30-31: Set text of test\_result class.
- 33-37: If the value of currentStep equals the size of errorsToTest array, hide the Next button and displays description text on screen.
- 39-46: Set data (media source) which is got with currentStep to media. If there is data, media is played.

### processError

This function is onError event handler; it displays corresponding error code on screen if event occurs.

- 53-54: Returns if video is not played.
- 56-57: Set error property of value to errorCode and displays returned number.

### Sample Code

```
01 : var errorsToTest = new Array();
02 :
03 : errorsToTest[errorsToTest.length] = [0, "A/V format not supported",
    getMediaFileUrl("samplevideo.flv")];
04 : errorsToTest[errorsToTest.length] = [1, "Cannot connect to server or
    connection lost",
    "http://not.exist.xxx/ncts/mediafiles/NetCastGeneratorClient_H.264_MP3.mov
    "];
05 : errorsToTest[errorsToTest.length] = [1000, "File is not found",
    getMediaFileUrl("not_exist.mp4")];
06 : errorsToTest[errorsToTest.length] = [1001, "Invalid protocol", "xxxx" +
    getMediaFileUrl("timer.mp4").substring(4)];
07 : errorsToTest[errorsToTest.length] = [1003, "Play list is empty",
    getMediaFileUrl("empty_list.asx")];
08 : errorsToTest[errorsToTest.length] = [1004, "Unrecognized play list",
    getMediaFileUrl("pls_test.pls")];
09 : errorsToTest[errorsToTest.length] = [1005, "Invalid ASX format",
    getMediaFileUrl("invalid_format.asx")];
10 :
11 : var currentStep = -1;
12 :
13 : function setNextStep()
14 : {
15 :     currentStep++;
16 :     if(currentStep == 0)
17 :     {
18 :         setInnerTextById("error_test_description", "Check whether expected
            error is occurred.<br>Press Red-key to test next error code.");
19 :         setInnerTextById("btn_red", "NEXT");
20 :     }
21 :
22 :     if(currentStep >= errorsToTest.length){return;}
23 :
24 :     var media = document.getElementById("video");
25 :     media.stop();
26 :     jsLog.lgmethod('stop()');
27 :     setInnerTextById("error_code", errorsToTest[currentStep][0]);
```

```
28 :   setInnerTextById("error_desc", errorsToTest[currentStep][1]);
29 :   setInnerTextById("result_error_code", "");
30 :   document.getElementById("test_result").className = "";
31 :   setInnerTextById("test_result", "");
32 :
33 :   if(currentStep == (errorsToTest.length - 1))
34 :   {
35 :       setInnerTextById("error_test_description", "Check whether expected
error is occurred.<br>No more test is left on this page.");
36 :       document.getElementById("btn_red").style.visibility = "hidden";
37 :   }
38 :
39 :   media.data = errorsToTest[currentStep][2];
40 :   jsLog.lgproperty('data');
41 :
42 :   if(errorsToTest[currentStep][2] != "")
43 :   {
44 :       media.play();
45 :       jsLog.lgmethod('play()');
46 :   }
47 : }
48 :
49 : function processError()
50 : {
51 :     jsLog.lgevent('onError');
52 :
53 :     if(currentStep < 0 || currentStep >= errorsToTest.length)
54 :         return;
55 :
56 :     var errorCode = document.getElementById("video").error;
57 :     setInnerTextById("result_error_code", errorCode);
58 : }
```

## 2.4 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.
- 04-05: Set width and height.
- 06: Set the path of media file which will be played.

### Sample Code

```
01 : <object  
02 :   id="video"  
03 :   type="application/x-netcast-av"  
04 :   width=300  
05 :   height=250  
06 :   data="/ApiTutorial/mediafile/timer.mp4"  
07 :   style="float: left">  
08 : </object>
```

## 2.5 Source Code of mediaplayer3.html

Source code of mediaplayer3.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(3/3)</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
    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);

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

        //add onError handler
        video.onError = processError;

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

    //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 : setNextStep(); break;
            case VK_GREEN : case 71 :
window.location.replace("../mediaplayer2.html"); break;
        }
    }

    var errorsToTest = new Array();

    errorsToTest[errorsToTest.length] = [0, "A/V format not supported",
```

```

getMediaFileUrl("samplevideo.flv");
    errorsToTest[errorsToTest.length] = [1, "Cannot connect to server or
connection lost",
"http://not.exist.xxx/ncts/mediafiles/NetCastGeneratorClient_H.264_MP3.mov"];
    errorsToTest[errorsToTest.length] = [1000, "File is not found",
getMediaFileUrl("not_exist.mp4");
    errorsToTest[errorsToTest.length] = [1001, "Invalid protocol", "xxxx" +
getMediaFileUrl("timer.mp4").substring(4)];
    errorsToTest[errorsToTest.length] = [1003, "Play list is empty",
getMediaFileUrl("empty_list.asx");
    errorsToTest[errorsToTest.length] = [1004, "Unrecognized play list",
getMediaFileUrl("pls_test.pls");
    errorsToTest[errorsToTest.length] = [1005, "Invalid ASX format",
getMediaFileUrl("invalid_format.asx");

    var currentStep = -1;

    function setNextStep()
    {
        currentStep++;
        if(currentStep == 0)
        {
            setInnerTextById("error_test_description", "Check whether
expected error is occurred.<br>Press Red-key to test next error code.");
            setInnerTextById("btn_red", "NEXT");
        }

        if(currentStep >= errorsToTest.length){return;}

        var media = document.getElementById("video");
        media.stop();
        jsLog.lgmethod('stop()');
        setInnerTextById("error_code", errorsToTest[currentStep][0]);
        setInnerTextById("error_desc", errorsToTest[currentStep][1]);
        setInnerTextById("result_error_code", "");
        document.getElementById("test_result").className = "";
        setInnerTextById("test_result", "");

        if(currentStep == (errorsToTest.length - 1))
        {
            setInnerTextById("error_test_description", "Check whether
expected error is occurred.<br>No more test is left on this page.");
            document.getElementById("btn_red").style.visibility =
"hidden";
        }

        media.data = errorsToTest[currentStep][2];
        jsLog.lgproperty('data');

        if(errorsToTest[currentStep][2] != "")
        {
            media.play();
            jsLog.lgmethod('play()');
        }
    }

    function processError()
    {
        jsLog.lgevent('onError');

        if(currentStep < 0 || currentStep >= errorsToTest.length)
            return;

        var errorCode = document.getElementById("video").error;
        setInnerTextById("result_error_code", errorCode);
        jsLog.lgproperty('video.error');
    }
}
</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/mediaplayer3.html</div>
</div>

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

<!-- test contents -->
<div class='ContentArea ' id="content_body">
    <div class='ApiListTitleArea'>API List</div>
    <div class='ApiListArea'>
        <div class='MethodTitleArea'>
            Methods
            <div class='MethodListArea'>
                stop()<br>
                play()<br>
            </div>
        </div>
        <div class='PropertyTitleArea'>
            Properties
            <div class='PropertyListArea'>
                data<br>
                error
            </div>
        </div>
        <div class='EventTitleArea'>
            Events
            <div class='EventListArea'>
                onError<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"
                width=300
                height=250
                data="/ApiTutorial/mediafile/timer.mp4"
                style="float: left;">
            </object>

            <table id="temp_result" style="position: relative; left:
10px; width:570px; height:250px;" border="0" cellpadding="0" cellspacing="0">
                <tr height="80px">
                    <td width="40%"><div class="eachTestGuide
">Expected Error Code</div></td>
                    <td width="60%" colspan="2"><div
class="eachTestGuide " id="error_code"></div></td>
                </tr>
                <tr height="80px">
                    <td width="30%"><div class="eachTestGuide
">Error Description</div></td>

```

```
 <div class="eachTestGuide " id="error_desc"></div></td> </tr> | |  | | |
```