

# **[Tutorial] LG Web\_Integrating VideoAd Media Plugin into Web Application**

---

Version 1.0 – October 2012

**LGDEV-080**

Home Entertainment Company  
LG Electronics, Inc.

## Copyright

**Copyright © 2012 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, 2012	Initial Version

## Purpose

This document describes how to integrate the video advertisements of media plugin into web applications using LG Web Open API.

## Reference Documents

Refer to the following documents:

- LG Web Application Development Guide
- LG Advertisement 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
AD	Advertisement
API	Application Programming Interface

## Contents

<b>1</b>	<b>Introduction.....</b>	<b>5</b>
1.1	Overview .....	6
1.2	Needed APIs .....	7
<b>2</b>	<b>Creating Application.....</b>	<b>8</b>
2.1	Initializing the Page .....	9
2.2	Initializing the Video Media Plugin .....	10
2.3	Handling Button Click Events .....	11
2.4	Handling VideoAdMediaPlugin Status Events .....	13
2.5	Handling Playback Buttons.....	16
2.6	Inputting Keys .....	18
2.7	Setting VideoMediaPlugin Object .....	19
2.8	Source Code of videoAdMediaPlugin.html .....	20

## Tables

[Table 1] Description of the Needed APIs.....	7
---	---

## Figures

[Figure 1] Web Application using VideoAdMediaPlugin API .....	6
[Figure 2] Displaying VideoAdMediaPlugin Status Event.....	15



# 1 Introduction

---

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

1.1 Overview

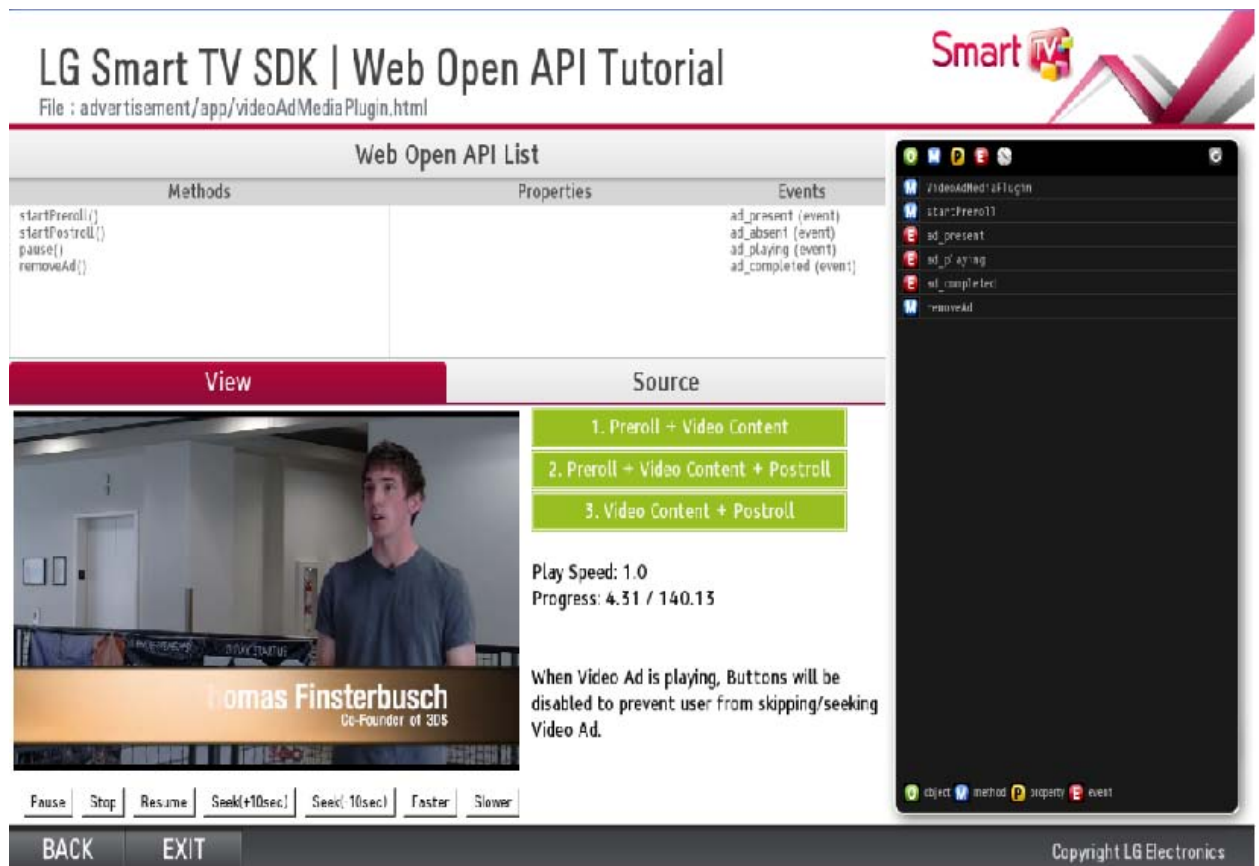
1.2 Needed APIs

## 1.1 Overview

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

The VideoAdMediaPlugin class is used to play a video advertisement using media plugin object. Currently, three types of video ad are available:

- Prefroll (before the main video)
- Postroll (after the main video)
- Both (Before and after main video)



[Figure 1] Web Application using VideoAdMediaPlugin API

## 1.2 Needed APIs

This application uses following Web Open API:

[Table 1] Description of the Needed APIs

API Class	Name	Description
Method	startPreroll	Plays a video ad before the main video starts..
	startPostroll	Plays a video ad after the main video is completed.
	pause	Pauses the video ad currently being played..
	removeAd	To remove all the resources allocated for video ads.
Event	ad_present	Fired when there is a video ad to play.
	ad_absent	Fired when there is no video ad to play.
	ad_completed	Fired when the video ad is completed or the event is fired immediately after ad_absent event.
	ad_playing	Fired when the video ad is present and currently playing

For more information on these functions, refer to “LG Advertisement 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 banner advertisement function using Web open API.

- 2.1 Initializing the Page
- 2.2 Initializing the Video Media Plugin
- 2.3 Handling Button Click Events
- 2.4 Handling VideoAdMediaPlugin Status Events
- 2.5 Handling Playback Buttons
- 2.6 Inputting Keys
- 2.7 Setting VideoAdMediaPlugin Object
- 2.8 Source Code of videoAdMediaPlugin.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.
- 06: Initialize the page.
- 07: Get the source code of the page using the XMLHttpRequest object.
- 08: Set the page id..
- 09: Initialize the Log function.
- 11: Call function to initialize voice.
- 14-20: Add event handlers which will be executed when the corresponding button is pressed.

### Sample Code

```
01 : function initPage()
02 : {
03 :     //save as last visited page
04 :     setLastVisitPage();
05 :     //common initialize function
06 :     commonInitialize();
07 :     requestSourceCode();
08 :     setPageID("VideoAdMediaPlugin");
09 :     jsLog.initLG();
10 :
11 :     initVideo();
12 :
13 :     //add onclick event handler
14 :     addEventHandler(document.getElementById("btn_back"), "click",
onClickHandler);
15 :     addEventHandler(document.getElementById("btn_exit"), "click",
onClickHandler);
16 :
17 :     addEventHandler(document.getElementById("preVideo"), "click",
handlePreroll);
18 :     addEventHandler(document.getElementById("preVideoPost"),
"click", handlePrepostroll);
19 :     addEventHandler(document.getElementById("videoPost"), "click",
handlePostroll);
20 : }
```

## 2.2 Initializing the Video Media Plugin

- 02: Declare video element of HTML5
- 03: Declare playInfo by calling mediaPlayInfo function of video.
- 04: Interval is set by using setInterval method of window class.
- 05: Setting call back function for onPlayStateChange event.
- 06: Setting call back function for onError event.
- 07: Initializing data value of video.

### Sample Code

```
01 : function initVideo(){
02 :     video = document.getElementById("video");
03 :     playInfo = video.mediaPlayInfo();
04 :     setInterval(updatePlayProgress, updateInterval);
05 :     video.onPlayStateChange = handleContentEnd;
06 :     video.onError = handleError;
07 :     video.data = "";
08 : }
```

### Handling content end event

- 02: If video play state has been finished then check.
- 03: If the script loaded is true and the type is not preroll ,then declare VideoAdMediaPlugin where postrollAdStatusListener is a call back function that listens to advertisements events.
- 05: Plays a video ad after the main video is completed.
- 07-09: if adType is preroll then set videoPlaying false and hide progress status indicator.

### Sample Code

```
01 : function handleContentEnd(){
02 :     if (video.playState == 5){// video content playback is finished
03 :         if (scriptLoaded && adType != "preroll"){
04 :             videoAd = new VideoAdMediaPlugin(video,
05 :                 postrollAdStatusListener,'titanic', 900);
06 :             jsLog.lgmethod('VideoAdMediaPlugin');
07 :             videoAd.startPostroll();
08 :             jsLog.lgmethod('startPostroll');
09 :         }else if( adType == "preroll"){
10 :             videoPlaying = false;
11 :             statusIndicators.style.visibility= "hidden";
12 :         }
13 :     }
```

### Handling the error

- 02: Alerts message will display.

### Sample Code

```
01 : function handleError(){
02 :     alert ("Media Plugin error!");
03 : }
```

## 2.3 Handling Button Click Events

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

### handlePreroll

- 03: if adPlaying is true then return so that users should not able to skip or seek video ads.
- 06: set adType variable to "preroll".
- 07-08: if scriptLoaded is true declare videoAd and play the video advertisement. Where video is media object that will play back the ad and main video, prerollAdStatusListener is a call back function that listens to advertisement events, content title is the title of the main video for ad targeting. 30 is the duration of pre/post roll in seconds.
- 10: Plays a video ad before the main video starts.
- 11: lgmethod to print log in console.
- 13: If start playing main video content using playContent function.

### handlePrepostroll

- 11: If adPlaying is true then return so that users should not able to skip or seek video ads.
- 20: set adType variable to "prepostroll".
- 21-24: if scriptLoaded is true, declare videoAd and play the video advertisement.
- 25: lgmethod to print log in console.
- 27: else start playing main video content using playContent function.

### handle Postroll

- 30: if adPlaying is true then return so that users should not able to skip or seek video ads.
- 34: set adType variable to "postroll".
- 35: start playing main video content using playContent function.

### Sample Code

```

01 : function handlePreroll() {
02 :     if (adPlaying){
03 :         return;
04 :     }
05 :
06 :     adType = "preroll";
07 :     if (scriptLoaded) {
08 :         videoAd = new VideoAdMediaPlugin(video, prerollAdStatusListener,
'content title', 30);
09 :         jsLog.lgmethod('VideoAdMediaPlugin');
10 :         videoAd.startPreroll();
11 :         jsLog.lgmethod('startPreroll');
12 :     } else {
13 :         playContent();
14 :     }
15 : }
16 : function handlePrepostroll() {
17 :     if (adPlaying){
18 :         return;
19 :     }
20 :     adType = "prepostroll";
21 :     if (scriptLoaded) {
22 :         videoAd = new VideoAdMediaPlugin(video,
prepostrollAdStatusListner, 'hello kitty', 20);
23 :         jsLog.lgmethod('VideoAdMediaPlugin');
24 :         videoAd.startPreroll();
25 :         jsLog.lgmethod('startPreroll');
26 :     } else {
27 :         playContent();
28 :     }
29 : }
30 : function handlePostroll() {
31 :     if (adPlaying) {
32 :         return;
33 :     }
34 :     adType = "postroll";

```

```
35 :           playContent();  
36 :       }
```

## 2.4 Handling VideoAdMediaPlugin Status Events

### Status listener for preroll Ad

This is a call back function that listens to advertisements events. This refers to ad before the main video.

- 03-06: If video ad is currently playing, disable videoPost and preVideoPost button so that users should not able to skip or seek video ads.
- 04: set adPlaying as true.
- 10: If video ad is completed to play, enable videoPost and preVideoPost buttons.
- 11: set adPlaying as false.
- 12: remove all the resources allocated for video ads.
- 13: lgmethod to print log in console.
- 14: Calling play content function to play main video content.

### Sample Code

```
01 : function prerollAdStatusListener(preEvent){
02 :   jsLog.lgevent(prerollStatus);
03 :   if (prerollStatus == "ad_playing") {
04 :     adPlaying = true;
05 :     document.getElementById("preVideoPost").className =
       "executeBigButtonOff";
06 :     document.getElementById("videoPost").className =
       "executeBigButtonOff";
07 :   } else if (prerollStatus == "ad_completed") {
08 :     document.getElementById("preVideoPost").className =
       "executeBigButton";
09 :     document.getElementById("videoPost").className =
       "executeBigButton";
10 :     adPlaying = false;
11 :     videoAd.removeAd();
12 :     jsLog.lgmethod('removeAd');
13 :     playContent();
14 :   }
15 : }
```

### Status listener for prepostroll Ad

This is a call back function that listens to advertisements events. This refers to ad before and after the main video.

- 03-06: if video ad is currently playing, disable preVideo and videoPost button so that users should not able to skip or seek video ads.
- 04: set adplaying as true.
- 07-10: else if video ad is completed enable preVideo and videoPost buttons.
- 11: set adPlaying as false.
- 12: remove all the resources allocated for video ads.
- 14: calling the playContent function to play main video content.

### Sample Code

```
01 : function prepostrollAdStatusListener(preEvent){
02 :   jsLog.lgevent(prepostrollStatus);
03 :   if (prepostrollStatus == "ad_playing") {
04 :     adPlaying = true;
05 :     document.getElementById("preVideo").className =
       "executeBigButtonOff";
06 :     document.getElementById("videoPost").className =
       "executeBigButtonOff";
07 :   } else if (prepostrollStatus == "ad_completed") {
08 :     document.getElementById("preVideo").className =
       "executeBigButton";
09 :     document.getElementById("videoPost").className =
       "executeBigButton";
10 :   }
11 : }
```

```

10 :
11 :   adPlaying = false;
12 :   videoAd.removeAd();
13 :   jsLog.lgmethod('removeAd');
14 :   playContent();
15 :   }
16 : }

```

### Status listener for postroll Ad

This is a call back function that listens to advertisements events. This refers to the ad after the main video.

04-11: if video ad is currently playing, in case of prepostroll disable preVideo and videoPost button and in case of post roll disable preVideo and prevideoPost button.

13: set adplaying as true.

14-22: else if video ad is completed playing, , in case of prepostroll enable preVideo and videoPost button and in case of post roll enable preVideo and prevideoPost.

24: set adPlaying as false.

25: remove all the resources allocated for video ads.

26: set adPlaying as false..

28: Hide the progress status indicator.

### Sample Code

```

01 :   function postrollAdStatusListener(postEvent)
02 :     jsLog.lgevent(postrollStatus);
03 :     if (postrollStatus == "ad_playing") {
04 :       jsLog.lgmethod('postEvent = "ad_playing"');
05 :       if(adType=="prepostroll"){
06 :
07 :         document.getElementById("preVideo").className="executeBigButtonOff"
08 :         ;
09 :         document.getElementById("videoPost").className="executeBigButtonOff"
10 :         ";
11 :       }
12 :       if(adType=="postroll"){
13 :         document.getElementById("preVideo").className="executeBigButtonOff";
14 :         document.getElementById("preVideoPost").className="executeBigButtonOff";
15 :       }
16 :       adPlaying = true;
17 :     }
18 :     else if (postrollStatus == "ad_completed") {
19 :       if(adType=="prepostroll"){
20 :         document.getElementById("preVideo").className="executeBigButton";
21 :       }
22 :       document.getElementById("videoPost").className="executeBigButton";
23 :     }
24 :     if(adType=="postroll"){
25 :       document.getElementById("preVideo").className="executeBigButton";
26 :       document.getElementById("preVideoPost").className="executeBigButton";
27 :     }
28 :     adPlaying = false;
29 :     videoAd.removeAd();
30 :     jsLog.lgmethod('removeAd');
31 :     videoPlaying=false;
32 :     statusIndicators.style.visibility= "hidden";
33 :   }

```

```
30 : }
```

### To play the content

Content can be either audio or video.

02: mediaURL is set as the value for source.

03: video is played.

04: set videoPlaying to true.

### Sample Code

```
01 : function playContent(){
02 :   video.data = mediaURL;
03 :   video.play(1);
04 :   videoPlaying=true;
05 : }
```

## LG Smart TV SDK | Web Open API Tutorial

File : advertisement/app/videoAdMediaPlugin.html

### Web Open API List

Methods	Properties	Events
startPreroll()		ad_present (event)
startPostroll()		ad_absent (event)
pause()		ad_playing (event)
removeAd()		ad_completed (event)

View

Source

1. Preroll + Video Content

2. Preroll + Video Content + Postroll

3. Video Content + Postroll

Play Speed: 1.0

Progress: 11.29 / 14.97

When Video Ad is playing, Buttons will be disabled to prevent user from skipping/seeking Video Ad.

Pause Stop Resume Seek(+10sec) Seek(-10sec) Faster Slower

BACK EXIT

Copyright LG Electronics

[Figure 2] Displaying VideoAdMediaPlugin Status Event

## 2.5 Handling Playback Buttons

These are the buttons to resume, pause, stop, seek and to change the speed of the main video content. Only Pause and resume will work for video ad.

- 01:     handleResume function.
- 02:     video is played.
- 03:     set videoPlaying to true.
  
- 05:     handlePause function
- 06:     video is paused.
- 07:     set videoPlaying to false.
- 08:     if ad is playing then that videoAd is paused.
  
- 13:     handleStop function.
- 14:     The current playing video is stopped.
- 15:     set videoPlaying to false.
- 16:     hide the progress status bar.
  
- 18:     handleSeek function
- 19-20:   Calculate the new position after seek and seek it to new position.
  
- 21:     handleChangeSpeed function.
- 22:     if adPlaying is ture then return.
- 26:     increment playback rate by respective value.
  
- 29:     updatePlayProgress function.
- 30-31:   if adPlaying or videoPlaying is true then make progress status bar visible.
- 32-33:   on progress of the video. The play speed and the current time will update itself on progress of the video.
- 34-35:   Display new value to status bar.
- 36:     Hide the status bar if both are not playing.

### Sample Code

```

01 : function handleResume(){
02 :   video.play(1);
03 :   videoPlaying=true;
04 : }
05 : function handlePause() {
06 :   video.pause(0);
07 :   videoPlaying=false;
08 :   if(adPlaying){
09 :     videoAd.pause();
10 :     jsLog.lgmethod('pause');
11 :   }
12 : }
13 : function handleStop(){
14 :   video.stop();
15 :   videoPlaying=false;
16 :   statusIndicators.style.visibility= "hidden";
17 : }
18 : function handleSeek(secFromNow){
19 :   var toPosition = video.playPosition += secFromNow * 1000
20 :   video.seek(toPosition);
21 : }
22 : function handleChangeSpeed(rateIncBy){
23 :   if (adPlaying){
24 :     return;
25 :   }
26 :   if (video.speed + rateIncBy < 1) video.play(1);
27 :   else video.play(video.speed + rateIncBy);
28 : }
29 : function updatePlayProgress (){
30 :   if(adPlaying||videoPlaying){

```



```
31 : statusIndicators.style.visibility= "visible";
32 : var playspeedText = "Play Speed: " +
    video.playbackRate.toFixed(1);
33 : var ProgressText = "Progress: " + video.currentTime.toFixed(2) +
    " / " + video.duration.toFixed(2);
34 : setInnerTextById("playspeed", playspeedText);
35 : setInnerTextById("currentTime", ProgressText);
36 : }else statusIndicators.style.visibility= "hidden";}
```

## 2.6 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.

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

### Sample Code

```
01 : function onUserInput(userInput){
02 : switch(userInput){
03 : case VK_BACK :
    window.location.replace("../menu_advertisement.html");
04 : break;
05 : }
06 : }
```

## 2.7 Setting VideoMediaPlugin Object

The following code shows how to set VideoMediaPlugin object.

- 01: Set id value as 'video'.
- 02: Set data type as 'application/x-netcast-av'.
- 03: Set properties width and height.

### Sample Code

```
01 : <object id="video"  
02 : type="application/x-netcast-av"  
03 : width="520" height="300">  
04 : </object>
```

## 2.8 Source Code of videoAdMediaPlugin.html

Source code of videoAdMediaPlugin.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>App Template of API Unit Sample App</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 type="text/javascript" src="../../blackbirdjs/blackbird.js"></script>
<link type="text/css" rel="Stylesheet" href="../../blackbirdjs/blackbird.css"
/>
<script type="text/javascript" onload='scriptLoaded=true' onerror='scriptLoaded=false'
src="http://smartservice.lgappstv.com/library/apps/ad/lib/videoAd.js"></script>
<script type="text/javascript">
    var mediaURL = "http://smartservice.lgappstv.com/samples/videoAd/media/test.mp4";
    var adType = "";
    var video = null;
    var videoAd = null;
    var updateInterval = 100;
    var playInfo = null;
    var adPlaying = false; // a boolean variable to determine whether the video being played is an ad
    var videoPlaying = false; // a boolean variable to determine whether video is playing
    //initialize page
    function initPage() {
        //save as last visited page
        setLastVisitPage();

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

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

        addEventHandler(document.getElementById("preVideo"), "click", handlePreroll);
```

```

        addEventHandler(document.getElementById("preVideoPost"), "click", handlePrepostroll);
        addEventHandler(document.getElementById("videoPost"), "click", handlePostroll);
    }

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

    function initVideo() {
        video = document.getElementById("video");
        playInfo = video.mediaPlayInfo();
        setInterval(updatePlayProgress, updateInterval);
        video.onPlayStateChange = handleContentEnd;
        video.onError = handleError;
        video.data = "";
    }

    function handleError() {
        alert("Media Plugin error!");
    }

    function playContent() {
        video.data = mediaURL;
        video.play(1);
        videoPlaying=true;
    }

    function handleResume() {
        video.play(1);
        videoPlaying=true;
    }

    function handlePause() {
        video.play(0);
        videoPlaying=false;
        if (adPlaying) {
            // ad pause should be called in addition to content pause
            videoAd.pause();
            jsLog.lgmethod('pause');
        }
    }

```

```
    }

}

function handleStop() {
    video.stop();
    videoPlaying=false;
    statusIndicators.style.visibility= "hidden";
}

function handleSeek(secFromNow) {
    var toPosition = video.playPosition += secFromNow * 1000
    video.seek(toPosition);
}

function handleChangeSpeed(rateIncBy) {
    if (adPlaying) {
        return;
    }
    if (video.speed + rateIncBy < 1) video.play(1);
    else video.play(video.speed + rateIncBy);
}

function updatePlayProgress() {
    if(adPlaying||videoPlaying){
        statusIndicators.style.visibility= "visible";
        var playspeedText = "Play Speed: " + video.speed.toFixed(1);
        var ProgressText = "Progress: " + (video.playPosition / 1000).toFixed(2) + " / " + (video.playTime / 1000).toFixed(2);
        setInnerTextById("playspeed", playspeedText);
        setInnerTextById("currentTime", ProgressText);
    } else statusIndicators.style.visibility= "hidden";
}

// deals with preroll + video content case
function handlePreroll() {
    if (adPlaying) {
        return;
    }

    adType = "preroll";
    if (scriptLoaded) {
        videoAd = new VideoAdMediaPlugin(video, prerollAdStatusListener, 'content title', 30);
        jsLog.lgmethod('VideoAdMediaPlugin');
        videoAd.startPreroll();
        jsLog.lgmethod('startPreroll');
    } else {
```

```

        playContent();
    }
}

// deals with preroll + video content + postroll case
function handlePrepostroll() {
    if (adPlaying) {
        return;
    }
    adType = "prepostroll";
    if (scriptLoaded) {
        videoAd = new VideoAdMediaPlugin(video, prepostrollAdStatusListner, 'hello kitty', 20);
        jsLog.lgmethod('VideoAdMediaPlugin');
        videoAd.startPreroll();
        jsLog.lgmethod('startPreroll');
    } else {
        playContent();
    }
}

// deals with video content + postroll case
function handlePostroll() {
    if (adPlaying) {
        return;
    }
    adType = "postroll";
    playContent();
}

function prerollAdStatusListener(prerollStatus) {
    jsLog.lgevent(prerollStatus);
    if (prerollStatus == "ad_playing") {
        adPlaying = true;
        document.getElementById("preVideoPost").className = "executeBigButtonOff";
        document.getElementById("videoPost").className = "executeBigButtonOff";
    } else if (prerollStatus == "ad_completed") {
        document.getElementById("preVideoPost").className = "executeBigButton";
        document.getElementById("videoPost").className = "executeBigButton";

        adPlaying = false;
        videoAd.removeAd();
        jsLog.lgmethod('removeAd');
        playContent();
    }
}

```

```
function prepostrollAdStatusListner(prepostrollStatus) {

    jsLog.lgevent(prepostrollStatus);
    if (prepostrollStatus == "ad_playing") {
        adPlaying = true;
        document.getElementById("preVideo").className = "executeBigButtonOff";
        document.getElementById("videoPost").className = "executeBigButtonOff";
    } else if (prepostrollStatus == "ad_completed") {
        document.getElementById("preVideo").className = "executeBigButton";
        document.getElementById("videoPost").className = "executeBigButton";

        adPlaying = false;
        videoAd.removeAd();
        jsLog.lgmethod('removeAd');
        playContent();
    }
}

function postrollAdStatusListener(postrollStatus) {
    jsLog.lgevent(postrollStatus);
    if (postrollStatus == "ad_playing") {
        jsLog.lgmethod('postEvent = "ad_playing"');
        if(adType=="prepostroll"){
            document.getElementById("preVideo").className="executeBigButtonOff";
            document.getElementById("videoPost").className="executeBigButtonOff";
        }

        if(adType=="postroll"){
            document.getElementById("preVideo").className="executeBigButtonOff";
            document.getElementById("preVideoPost").className="executeBigButtonOff";
        }

        adPlaying = true;
    }
    else if (postrollStatus == "ad_completed") {
        if(adType=="prepostroll"){
            document.getElementById("preVideo").className="executeBigButton";
            document.getElementById("videoPost").className="executeBigButton";
        }

        if(adType=="postroll"){
            document.getElementById("preVideo").className="executeBigButton";
            document.getElementById("preVideoPost").className="executeBigButton";
        }

        adPlaying = false;
        videoAd.removeAd();
        jsLog.lgmethod('removeAd');
```



```

        videoPlaying=false;
        statusIndicators.style.visibility= "hidden";
    }
}

function handleContentEnd() {
    if (video.playState == 5) { // video content playback is finished
        if (scriptLoaded && adType != "preroll") {
            videoAd = new VideoAdMediaPlugin(video, postrollAdStatusListener, 'titanic', 900);
            jsLog.lgmethod('VideoAdMediaPlugin');
            videoAd.startPostroll();
            jsLog.lgmethod('startPostroll');
        } else if( adType == "preroll"){
            videoPlaying = false;
            statusIndicators.style.visibility= "hidden";
        }
    }
}
}
</script>
</head>

<body ondragstart='return false' onselectstart='return false' onload="javascript:initPage();">
    <div id='mainBody' class='mainBody'>
        <!-- title -->
        <div class='SuiteTitle'>LG Smart TV SDK | Web Open API Tutorial</div>
        <!-- navigation -->
        <div class='SuiteNavigation'>
            <div style="float:left;">File : advertisement/app/videoAdMediaPlugin.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'>startPreroll()
                        <br>startPostroll()
                        <br>pause()
                        <br>removeAd()
                    </div>
                </div>
                <div class='PropertyTitleArea'>Properties
                    <div class='PropertyListArea'></div>
                </div>
                <div class='EventTitleArea'>Events

```

```
                <div class='EventListArea'>ad_present (event)
                    <br>ad_absent (event)
                    <br>ad_playing (event)
                    <br>ad_completed (event)
                </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' class='ViewArea'>
            <table border="0">
                <tr>
                    <td rowspan="4" width="520" height="300">
                        <div id="html5_video_div" style="top: 5px; left: 5px; width: 520px; height: 300px;
float: left; position: absolute; background-color:black;">
                            <object id="video" type="application/x-netcast-av" width="520"
height="300"></object>
                        </div>
                    </td>
                    <td>
                        <div id="preVideo" class="executeBigButton">1. Preroll + Video Content</div>
                    </td>
                </tr>
                <tr>
                    <td>
                        <div id="preVideoPost" class="executeBigButton">2. Preroll + Video Content +
Postroll</div>
                    </td>
                </tr>
                <tr>
                    <td>
                        <div id="videoPost" class="executeBigButton">3. Video Content + Postroll</div>
                    </td>
                </tr>
                <tr>
                    <td height="60">
                        <div id="statusIndicators" style="visibility:hidden">
                            <div id="playspeed"></div>
                            <div id="currentTime"></div>
                        </div>
                        <div>
                            <br>
                        </div>
                    </td>
                </tr>
            </table>
        </div>
```

```

        <br>When Video Ad is playing, Buttons will be disabled to prevent user from
        skipping/seeking Video Ad.
    </div>
</td>
</tr>
<tr>
    <td width="520" height="50">
        <div id="controlButtonArea">
            <button id="pause" onclick="handlePause()">Pause</button>
            <button id="stop" onclick="handleStop()">Stop</button>
            <button id="resume" onclick="handleResume()">Resume</button>
            <button id="seekPlus5" onclick="handleSeek(10)">Seek(+10sec)</button>
            <button id="seekMinus5" onclick="handleSeek(-10)">Seek(-10sec)</button>
            <button id="incPlaySpeed" onclick="handleChangeSpeed(0.5)">Faster</button>
            <button id="decPlaySpeed" onclick="handleChangeSpeed(-0.5)">Slower</button>
        </div>
    </td>
</tr>
</table>
</div>
<div style="visibility: hidden" id='codeview'>
    <textarea class="SourceCodeArea" value="" id='sourcecode'></textarea>
</div>
</div>
<!-- description -->
<div class='SuiteDescription' id='back_exit_test_description'></div>
<!-- button and copyright -->
<div id='SuiteButtonArea' class='SuiteButtonArea'>
    <!-- back key description -->
    <div id='btn_back' class='buttonDescription'>BACK</div>
    <!-- exit button -->
    <div id='btn_exit' class='buttonDescription'>EXIT</div>
    <!-- copyright -->
    <div class='copyright'>Copyright LG Electronics</div>
</div>
</div>
<!-- end of mainBody DIV -->
</body>
</html>

```