LG Electronics

[Tutorial] Integrating Video Advertisement into Web Application

Version 1.1 – February 2012

LGDEV-055

Home Entertainment Company
LG Electronics, Inc.

Copyright

About This Document

Revision History

Document Version	Date	Comment
1.1	February 1, 2012	Section 3.3 Source code is updated.
1.0	October 17, 2011	Initial Version

Purpose

This document describes how to integrate video ads (advertisements) into Web applications.

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, ad	Advertisement
API	Application Programming Interface
GUI	Graphical User Interface

Contents

Tables

Figures

1 Introduction

This chapter provides an overview of the types of video ads and video ad API.

1.1 Types of Video Ads

LG Smart TV platform provides two types of video ads for your web application:

- Preroll: Video ad gets played before the main video of the application.
- Postroll: Video ad gets played after the main video of the application.

Developers can choose either one of the above or both.
This tutorial explains how to contain both Pre and Postroll Video ads.

1.2 Video Ad API

The VideoAd class contains four functions: startPreroll, startPostroll, pause, and removeAd.
These functions are used to:

- start preroll video ad
- start postroll video ad
- pause the video ad
- stop the video ad and release all the video ad resources

[Table 1] Description of the VideoAd class

Class	Function Name	Description
VideoAd	startPreroll()	Plays a video ad before the main video starts.
	startPostroll()	Plays a video ad after the main video.
	pause()	Pauses the video ad currently being played.
	removeAd()	Stops the video ad and releases all the resources allocated for video ads.

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

Note

In Emulator for LG Smart TV, Advertisement API will not operate normally.

2 Creating Web Applications Containing Video Ads

This chapter describes how to create web applications that contain video ads using LG SmartAd JavaScript API and HTML5 video tag.

Caution

- Video ad API is available only if you play the main video of the application using HTML5 video tag.
- Graphical User Interface (GUI) for the player should be provided by application developer.
  - ✓ i.e., play control buttons, progress bar, etc.
- It is developer's responsibility to block skipping, fast-forwardg, and changing play speed of video ads.

2.1 Importing Library JavaScript File

LG SmartAd library files for Web applications reside at http://smartservice.lgappstv.com. You need to import the videoAd.js file from the server using script tags such as the following:

```
  <script type="text/javascript" onload='scriptLoaded=true'
onerror='scriptLoaded=false'
src="http://smartservice.lgappstv.com/library/apps/ad/lib/videoAd.js"
></script>
```

Note that the value of the boolean variable scriptLoaded is set to true if the script is loaded successfully; otherwise, it is set to false. The application must run properly without the video ads even if it fails to access the library server. Success/failure of accessing the library server should not affect any other operations of the application. Hence, it is highly recommended you to check whether the script is loaded successfully.

2.2 Playing Video Ad

Follow the steps below to:

- play video ad before the main video of your Web application
- play video ad after the main video of your Web application

1. Declare an object variable for the VideoAd class.
2. Call the VideoAd constructor by passing the following five parameters:
   - div id that contains video tag
   - HTML5 video tag id that will play the main video of the application
   - Callback function to be called for the VideoAd Events
   - [Optional] title of the main video of the application, which is used for targeting
   - [Optional] duration of the main video of the application in second, which is used for targeting
3. Invoke startPreroll() function
4. Play the main video when the application captures the “ad_completed” event
5. Invoke startPostroll() function when the application after the main video is complete.

Sample Code

```
   var videoAd = null;
   var adPlaying = false;
   var bPostrollCompleted = false;
   function initializeAd() {
       if (scriptLoaded) {
           videoAd = new VideoAd("html5_video_div", "html5_video",
adStatusListener,'prepost both', 20);
           videoAd.startPreroll();
       }
       else
           playContent(); // script for ad is not loaded, thus play content!
   }
   function playContent() {
       var videoElement = document.getElementById(videoId);
       videoElement.src = mediaURL;
       videoElement.load();
       videoElement.play();
       videoElement.addEventListener('ended', videoEnd, false);
   }
   function videoEnd() {
       if (!bPostrollCompleted) {
           bPostrollCompleted = true;
           videoAd.startPostroll();
       }
       else
           delete videoAd;
   }
   function adStatusListener (adStatus) {
       if (adStatus == "ad_playing") {
           adPlaying = true;
       }
       if (adStatus == "ad_completed") {
           if(!bPostrollCompleted)
               playContent();
           adPlaying = false;
       }
   }
   .
   .
   .

   <body onload="initializeAd()" onunload="videoAd.removeAd()"
onkeydown="processKeyDown(event)">
       <div id="html5_video_div" style="width: 640px; height: 360px; float:
left; position: absolute">
           <video id="html5_video" poster="" width=640 height=360>
```

2.3 Handling VideoAd Events

The VideoAd class contains four events as below:

[Table 2] Description of the VideoAd event

Class	Event Name	Description
VideoAd	ad_present (event)	This event is fired when there is a video ad to play.
	ad_absent (event)	This event is fired when there is no video ad to play
	ad_playing (event)	This event is fired when the video ad is present and currently playing.
	ad_completed (event)	This event is fired when the video ad is complete.

VideoAd class fires a few events about the video ad status. In order to listen to those events, developers should implement a callback function and pass the function name as the third parameter of the VideoAd constructor. When an ad is requested, sequence of events maybe fired. For example, when the ad is available, ad_present, ad_playing, and ad_completed are sequentially fired. In other case, when the ad is not available, ad_absent, then ad_completed are fired.
Note that ad_completed gets fired even there is no ad to be played back.

For sample codes on these events, refer to “LG Advertisement API Reference Guide.”

2.4 Releasing Video Ad Resources

You can stop and release all the VideoAd resources at anytime while your application is running. It is highly recommended you to invoke the removeAd() function before you exit your application.
A typical use case for invoking removeAd() function would be when the user leaves the Web page.

The following example handles such case:

Sample Code

```
<body onload="initializeAd()" onunload="videoAd.removeAd()">
```

3 Pre and Postroll Video Ad Sample Application

This chapter provides the directory structure and source code of sample application, called “Pre and Postroll Video Ad”.

3.1 Overview of Pre and Postroll Video Ad Sample Application

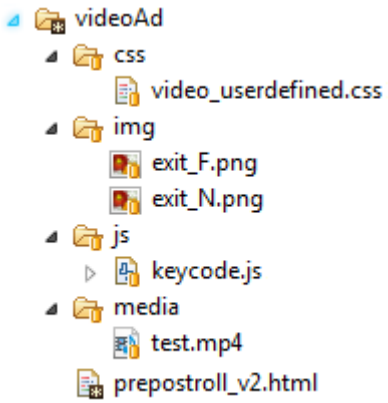
The sample application called Pre and Postroll Video Ad plays videos in following order:

1. Preroll video ad
2. Video contentf
3. Postroll video ad

Graphical User Interface (GUI) for the progress bar and the video control buttons is implemented in the sample code. However, developers should provide their own GUI for their HTML5 video.

3.2 Directory Structure of Pre and Postroll Video Ad Sample Application

The directory structure of Pre and Postroll Video Ad is shown in the figure below:



[Figure 1] Directory structure of Pre and Postroll Video Ad

3.3 Source Code of prepostroll_v2.html

Source code of prepostroll_v2.html is as follows:

```
  <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
  <html xmlns="http://www.w3.org/1999/xhtml">
  <head>
  <meta http-equiv="Content-Type" content="text/html; charset=utf-8"/>
  <title>Pre and postroll Video Ad</title>

  <link rel="stylesheet" href="./css/video_userdefined.css" type="text/css"/>
  <script type="text/javascript" onload='scriptLoaded=true'
onerror='scriptLoaded=false'
src="http://smartservice.lgappstv.com/library/apps/ad/lib/videoAd.js"></script>
  <!-- added by augustine for keycode -->
  <script text="text/javascript" src="./js/keycode.js"></script>
  <script type="text/javascript" language="javascript">
      var mediaURL = "./media/test.mp4";
      var videoAd = null;
      var eventAdderRemover = null;
      var bPostrollCompleted = false;
      var adPlaying = false;

      var divId = "html5_video_div";
      var videoId = "html5_video";
      var controlsId = "controls";
      var playControlId = "play";
      var progressControlId = "progress";
      var progressHolderId = "progress_box";
      var playProgressBarId = "play_progress";
      var currentTimeDisplayId = "current_time_display";
      var durationDisplayId = "duration_display";
      var volumeControlId = "volume";
      var volumeDisplayId = "volume_display";
      var tags = "";
      var title = "";
      var duration = "";
      var videoContolBar = null;

      var video;
      var controls;
      var fullScreenControl;
      var videoIsFullScreen;
      var progressControl;
      var progressHolder;
      var playProgressBar;
      var playProgressInterval;
      var currentTimeDisplay;
      var backButton;

      function adStatusListener (adStatus) {
          if (adStatus == "ad_playing") {
              adPlaying = true;
          }
          if (adStatus == "ad_completed") {
              if(!bPostrollCompleted)
                  playContent();

              adPlaying = false;
          }
      }


      function processKeyDown(e) {
          var keycode;
          if(window.event) {
              keycode = e.keyCode;
          } else if(e.which) {
              keycode = e.which;
          }
```

```
            switch(keycode) {
                case VK_BACK:  goDashBoard(); break;
            }
        }

        function goDashBoard() {
            window.NetCastBack();
        }
        function addProgressEvent() {

        }

        function initializeAd() {

            if (scriptLoaded) {
                videoAd = new VideoAd("html5_video_div", "html5_video",
adStatusListener,'prepost both', 20);
                videoAd.startPreroll();
            }
            else
                playContent(); // script for ad is not loaded, thus play content
directly!

            video = document.getElementById("html5_video");
            controls = document.getElementById("controls");
            playControl = document.getElementById("play");
            progressControl = document.getElementById("progress");
            progressHolder = document.getElementById("progress_box");
            playProgressBar = document.getElementById("play_progress");
            currentTimeDisplay =
document.getElementById("current_time_display");
            durationDisplay = document.getElementById("duration_display");
            volumeControl = document.getElementById("volume");
            volumeDisplay = document.getElementById("volume_display");
            fullScreenControl = document.getElementById("full_screen");
            backButton = document.getElementById("backButton");


            showController();
            positionController();

            playControl.addEventListener("click", function() {
                if (video.paused) {
                    playVideo();
                } else {
                    pauseVideo();
                }
            }, true);

            progressHolder.addEventListener("mousedown", function() {
                stopTrackingPlayProgress();

                if (video.paused) {
                    videoWasPlaying = false;
                } else {
                    videoWasPlaying = true;
                    videoAd.pause();
                    video.pause();
                }

                blockTextSelection();

                document.onmousemove = function(e) {
                    if (!adPlaying) {
                        setPlayProgress(e.pageX);
                    }
                }

                document.onmouseup = function() {
                unblockTextSelection();
```

```
                document.onmousemove = null;
                document.onmouseup = null;
                if (videoWasPlaying) {
                    video.play();
                    trackPlayProgress();
                }
            }

        }, true);

        progressHolder.addEventListener("mouseup", function(e) {
            if (!adPlaying) {
                setPlayProgress(e.pageX);
            }
        }, true);

        volumeControl.addEventListener("mousedown", function() {
            blockTextSelection();
            document.onmousemove = function(e) {
                setVolume(e.pageX);
            }
            document.onmouseup = function() {
                unblockTextSelection();
                document.onmousemove = null;
                document.onmouseup = null;
            }
        }, true);

        volumeControl.addEventListener("mouseup", function(e) {
            setVolume(e.pageX);
        }, true);

        updateVolumeDisplay();

        fullScreenControl.addEventListener("click", function() {
            if (!videoIsFullScreen) {
                fullScreenOn();
            } else {
                fullScreenOff();
            }
        }, true);

        // added playVide() function owe to progress bar and time bar was
not increased from the first
        playVideo();
    }

    function playVideo() {
        video.play();
        playControl.className = "pause control";
        trackPlayProgress();
    }

    function pauseVideo() {
        video.pause();
        if (adPlaying)
            videoAd.pause();
        playControl.className = "play control";
        stopTrackingPlayProgress();
    }

    function positionController() {
        // controls.style.top = (video.offsetHeight - controls.offsetHeight)
+ "px";
        // added +35 for gap between video screen and controls
        controls.style.top = (video.offsetHeight - controls.offsetHeight) +
35 + "px";
        controls.style.left = "0px";
        controls.style.width = video.offsetWidth + "px";
        sizeProgressBar();
```

```
        }

        function positionControllerFullScreenOn() {
            controls.style.top = (video.offsetHeight - controls.offsetHeight) -
10 + "px";
            // added +35 for gap between video screen and controls
            // controls.style.top = (video.offsetHeight - controls.offsetHeight)
+ 35 + "px";
            controls.style.left = "0px";
            controls.style.width = video.offsetWidth + "px";
            sizeProgressBar();
        }

        function showController() {
            controls.style.display = "block";
        }

        function hideController() {
            controls.style.display = "none";
        }

        function sizeProgressBar() {
            progressControl.style.width = (controls.offsetWidth - 125) + "px";
            progressHolder.style.width = (progressControl.offsetWidth - 80) +
"px";
            updatePlayProgress();
        }

        function trackPlayProgress() {
            playProgressInterval = setInterval(updatePlayProgress, 33);
        }

        function stopTrackingPlayProgress() {
            clearInterval(playProgressInterval);
        }

        function updatePlayProgress() {
            playProgressBar.style.width = ((video.currentTime / video.duration)
* (progressHolder.offsetWidth - 2))
                    + "px";
            updateTimeDisplay();
        }

        function setPlayProgress(clickX) {

            var newPercent = Math.max(0, Math.min(1,
                    (clickX - findPosX(progressHolder))
                            / progressHolder.offsetWidth));
            video.currentTime = newPercent * video.duration;
            playProgressBar.style.width = newPercent
                    * (progressHolder.offsetWidth - 2) + "px";
            updateTimeDisplay();
        }

        function updateTimeDisplay() {
            currentTimeDisplay.innerHTML = formatTime(video.currentTime);
            if (video.duration)
                durationDisplay.innerHTML = formatTime(video.duration);
        }

        function setVolume(clickX) {
            var newVol = (clickX - findPosX(volumeControl))
                    / volumeControl.offsetWidth;
            if (newVol > 1) {
                newVol = 1;
            } else if (newVol < 0) {
                newVol = 0;
            }
            video.volume = newVol;
            updateVolumeDisplay();
        }
```

```
// Unique to these controls.
function updateVolumeDisplay() {
    var volNum = Math.floor(video.volume * 6);
    for ( var i = 0; i < 6; i++) {
        if (i < volNum) {
            volumeDisplay.children[i].style.borderColor = "#fff";
        } else {
            volumeDisplay.children[i].style.borderColor = "#555";
        }
    }
}

function fullScreenOn() {
    videoIsFullScreen = true;
    videoOrigWidth = video.offsetWidth;
    videoOrigHeight = video.offsetHeight;

    video.style.width = window.innerWidth + "px";
    video.style.height = window.innerHeight + "px";
    video.style.position = "fixed";
    video.style.left = 0;
    video.style.top = 0;
    controls.style.position = "fixed";
    // positionController();
    positionControllerFullScreenOn();

    fullScreenControl.className = "fs-active control";
    backButton.style.display = "none";
}

function fullScreenOff() {
    videoIsFullScreen = false;
    video.style.width = videoOrigWidth + "px";
    video.style.height = videoOrigHeight + "px";
    video.style.position = "static";
    controls.style.position = "absolute";
    positionController();
    fullScreenControl.className = "control";
    backButton.style.display = "block";
}

function blockTextSelection() {
    document.body.focus();
    document.onselectstart = function() {
        return false;
    };
}

function unblockTextSelection() {
    document.onselectstart = function() {
        return true;
    };
}

// Return seconds as MM:SS
function formatTime(seconds) {
    seconds = Math.round(seconds);
    minutes = Math.floor(seconds / 60);
    minutes = (minutes >= 10) ? minutes : "0" + minutes;
    seconds = Math.floor(seconds % 60);
    seconds = (seconds >= 10) ? seconds : "0" + seconds;
    return minutes + ":" + seconds;
}

// Get objects position on the page
function findPosX(obj) {
    var curleft = obj.offsetLeft;
    while (obj = obj.offsetParent) {
        curleft += obj.offsetLeft;
    }
```

```
            return curleft;
        }

        // modified by Augustine, sean modified on 7/13
        function playContent() {
            var videoElement = document.getElementById(videoId);
            videoElement.src = mediaURL;
            videoElement.load();
            videoElement.play();
            videoElement.addEventListener('ended', videoEnd, false);
        }

        function videoEnd() {
            if (!bPostrollCompleted) {
                bPostrollCompleted = true;
                videoAd.startPostroll();
            }
            else
                delete videoAd;
        }
    </script>
    </head>
    <body onload="initializeAd()" onunload="videoAd.removeAd()"
onkeydown="processKeyDown(event)">
        <h1> Testing Pre and Postroll Video Ad </h1>
        <!-- <p> 1. Preroll advertisement </p>
        <p> 2. LG Infinia Video </p>
        <p> 3. Postroll advertisement </p>
        -->

        <div style="height: 390px; width: 480px;">
            <div id="html5_video_div"
                style="width: 640px; height: 360px; float: left; position:
absolute">
                <video id="html5_video" poster="" width=640 height=360>
                <i> Browser doesn't support video tag</i> </video>
                <div id="controls">
                    <div id="play" class="play control">
                        <span></span>
                    </div>
                    <div id="progress" class="control">
                        <div id="progress_box">
                            <span id="load_progress"><span
id="play_progress"></span>
                            </span>
                        </div>
                        <div id="play_time">
                            <span id="current_time_display">00:00</span> / <span
                                id="duration_display">00:00</span>
                        </div>
                    </div>

                    <div id="volume" class="control">
                        <span>
                            <ul id="volume_display">
                                <li id="0"
onclick="player_volumeChangeHandler(this.id);"></li>
                                <li id="1"
onclick="player_volumeChangeHandler(this.id);"></li>
                                <li id="2"
onclick="player_volumeChangeHandler(this.id);"></li>
                                <li id="3"
onclick="player_volumeChangeHandler(this.id);"></li>
                                <li id="4"
onclick="player_volumeChangeHandler(this.id);"></li>
                                <li id="5"
onclick="player_volumeChangeHandler(this.id);"></li>
                            </ul> </span>
                    </div>

                    <div id="full_screen" class="control">
```

```
                        <span>
                            <table border="0" cellpadding="0" cellspacing="0">
                                <tr>
                                    <td><div id="fs_top_left" class="fs-
corner"></div></td>
                                    <td><div id="fs_top_right" class="fs-
corner"></div></td>
                                </tr>
                                <tr>
                                    <td><div id="fs_bottom_left" class="fs-
corner"></div></td>
                                    <td><div id="fs_bottom_right" class="fs-
corner"></div></td>
                                </tr>
                            </table> </span>
                    </div>
                </div>
            </div>
        </div>
        <div style="top: 622px; left: 1068px; position: absolute">
            <a href="Javascript:goDashBoard();"> <img id="backButton"
src="./img/exit_N.png"
                onMouseover="this.src='./img/exit_F.png'"
                onMouseOut="this.src='./img/exit_N.png'">
            </a>
        </div>
    </body>
    </html>
```