Developing Web App

This document describes how to develop web-based application for LG Smart TV.

Quick Start Guide
This document is provided for beginners who want to start web app developing using LG IDE. With simple sample application, we explain how to setup the project and implement the codes.

App Development Guide
This document provides description on the NetCast platform architecture, supported media formats, protocols, minimum LG NetCast Platform capabilities and the HTML5/CSS specifications. Also, performance considerations are provided which web application developers should know.

Supported HTML5/CSS3 Standard for the NetCast Platform and Emulator
This document describes HTML5 and CSS3 supported by the NetCast platform and LG Smart TV Emulator.

Sample Tutorials: LG WebAPI Tutorials
This document includes sample tutorials for LG WebAPI Tutorial codes.
Contents

Quick Start Guide ......................................................................................... 8

App Development Guide .................................................................................. 9

LG Web Application Overview ........................................................................ 9

NetCast Platform Overview ............................................................................ 9

Specifications .................................................................................................. 11

Video Output .................................................................................................... 11

Web Engine ...................................................................................................... 11

  Cookie and Cache Support of LG Web Browser Engine .............................. 12

  userAgent String[1] ....................................................................................... 12

  The Accept-Language Header ....................................................................... 14

Protocols .......................................................................................................... 14

CODEC & Container ....................................................................................... 14

DRM .................................................................................................................. 14

  Using WMDRM/PlayReady ........................................................................ 15

  Using Verimatrix ......................................................................................... 15

OSD Resolution ............................................................................................... 16

Subtitle ............................................................................................................ 16

Images ............................................................................................................. 18

Cascading Style Sheets ................................................................................... 18

Streaming Method ............................................................................................ 18

  HTTP Streaming .......................................................................................... 18

  Http Live Streaming (HLS) - Adaptive Streaming ................................. 18

  Play List (ASX) ............................................................................................ 19

  Local File Play ............................................................................................. 19

Tips for Web Application Development .......................................................... 20

Web Browser Engine ....................................................................................... 20

  Document Scrolling ..................................................................................... 20

  Common Control Supporting .................................................................... 20

  Flushing Cookie and Cache Repositories ............................................... 20

Font and Font Style ........................................................................................ 20

  Font style Supporting ................................................................................ 21

  Representation of Control Characters[6] .................................................... 23

Platform Identification and Its Applications ................................................... 24

  Identifying and Differentiating with the userAgent String ...................... 24

  Identifying and Differentiating with Device Information APIs ............... 25

Input Device .................................................................................................... 25

  Traditional IR Remote ............................................................................... 25

  Magic Remote ............................................................................................. 25

  Traditional IR Remote Key Events .......................................................... 27

  Key Repeat Function .................................................................................. 27
Supported HTML5/CSS3 Standard for the NetCast Platform and Emulator .................................................................60

Introduction ..................................................................................61
Status Definition ...........................................................................61
HTML5 Standard ...........................................................................61
HTML5 Video[1],[2] ........................................................................61
Video Methods .............................................................................61
Video Attributes ..........................................................................61
Video Events ................................................................................63
HTML5 Audio[3] ...........................................................................64
Audio Methods .............................................................................64
Audio Attributes ..........................................................................64
Audio Events ................................................................................65
HTML5 Communication API[4][5] ................................................................. 66
  MessageEvent Attributes ........................................................................ 66
  Cross-Document Messaging Methods ..................................................... 66
  Channel Messaging Methods .................................................................. 66
  Channel Messaging Attributes ............................................................... 66
  WebSockets Attributes and Events[6] ...................................................... 66
  Server-sent Events Attributes and Events[7] ........................................... 67
HTML5 Canvas[8][9] ................................................................................... 67
  Colors, Styles, Transformations, and Pixel Manipulation Methods ........... 67
  Colors, Styles, Composing, Shadows, Line Styles, and Shapes Attributes . 67
  Interactivity, Animation, and Image[10] .................................................. 68
  Canvas Text Methods ............................................................................. 68
  Canvas Text Attributes .......................................................................... 69
HTML5 Inline SVG[11] ............................................................................... 69
  Gradients and Patterns .......................................................................... 69
  Filter Effects and Basic Shapes ............................................................. 69
  SVG Clipping Paths and Animation ....................................................... 70
  WebGL ..................................................................................................... 70
HTML5 Offline Web Applications API[12] .................................................. 70
  ApplicationCache Methods and Attributes ........................................... 70
  ApplicationCache Events ...................................................................... 71
HTML5 Web Storage API[13] .................................................................... 71
  Storage Attributes .................................................................................. 71
  Indexed Database Methods ..................................................................... 71
  WebSQL Database Methods .................................................................... 72
HTML5 User Interaction API[14][15] ............................................................ 72
  Drag&Drop Attributes ........................................................................... 72
  Drag&Drop Events ................................................................................ 72
  History Management Methods .............................................................. 72
  History Management Attributes ......................................................... 72
HTML5 Web Workers API[16] ................................................................... 72
  Workers .................................................................................................. 73
  Shared workers ..................................................................................... 73
CSS3 Standard .......................................................................................... 73
Dynamic CSS ............................................................................................ 73
  Transforms[17] ...................................................................................... 73
  Transitions[18] ..................................................................................... 74
  Animation[19] ....................................................................................... 74
Static CSS .................................................................................................. 75
  Background and Borders[20] ................................................................ 75
  Box[21] .................................................................................................. 76
  Color[22] ............................................................................................... 76
  Columns and Multi Column Layout[23] .................................................. 76
Sample Tutorials: LG WebAPI Tutorials

Activating or Deactivating Browser Page Loading Icon ........................................ 80
  Initializing the Page ................................................................................. 80
  Inputting Keys ....................................................................................... 81
  Handling Keys ......................................................................................... 81
  Handling Events ....................................................................................... 82
  Counting Time ......................................................................................... 82
  Source Code of pageloadingicon.html ...................................................... 83

Activating or Deactivating Magic Remote ..................................................... 87
  Initializing the Page ................................................................................. 87
  Inputting Keys ....................................................................................... 88
  Handling Events ..................................................................................... 88
  Counting Time ......................................................................................... 88
  Adding Mouse Event Handler ................................................................. 90
  Source Code of mouse.html ..................................................................... 90

Controlling Voice Recognition ...................................................................... 95
  Creating Web Applications ..................................................................... 96
    Initializing the Page .............................................................................. 96
    Check the NetCast Version ................................................................... 96
    Initializing Voice Recognition ............................................................... 97
    Handling Button Click Events ............................................................. 97
    Handling Voice Events .......................................................................... 98
    Displaying Text ................................................................................... 99
    Inputting Keys ..................................................................................... 100
    Setting Voice Recognition Object ....................................................... 100
    Source Code of voice_recognition.html ............................................. 100

Displaying Subtitle in Media Player .......................................................... 106
  Initializing the Page .............................................................................. 106
  Inputting Keys ..................................................................................... 107
  Displaying Text ..................................................................................... 107
  Controlling Play Mode and Subtitle ....................................................... 108
  Setting Media Object ............................................................................ 108
  Source Code of subtitle.html ................................................................. 108

Executing Windows Media Player 1 ............................................................ 116
  Initializing the Page .............................................................................. 116
  Handling Media Events ......................................................................... 117
  Controlling Play Mode ......................................................................... 119
  Displaying Text .................................................................................... 120
  Inputting Keys ..................................................................................... 121
  Setting Media Object ............................................................................ 121
  Source Code of mediaplayer.html ......................................................... 121

Annex A References ................................................................................... 78
Executing Windows Media Player 2 ................................................................. 128
  Initializing the Page .................................................................................. 128
  Inputting Keys ......................................................................................... 129
  Displaying Text ....................................................................................... 129
  Setting Media Object ................................................................................ 130
  Source Code of mediaplayer2.html ......................................................... 130
Executing Windows Media Player 3 .............................................................. 135
  Initializing the Page .................................................................................. 135
  Inputting Keys ......................................................................................... 136
  Displaying Text ....................................................................................... 136
  Setting Media Object ................................................................................ 138
  Source Code of mediaplayer3.html ......................................................... 138
Generating Outofmemory Event .................................................................. 143
  Initializing the Page .................................................................................. 143
  Inputting Keys ......................................................................................... 144
  Loading Images ....................................................................................... 144
  Displaying Information of Loaded Image and Event ............................... 146
  Handling Events ...................................................................................... 147
  Displaying usedMemorySize .................................................................... 148
  Source Code of outofmemory.html ......................................................... 148
Getting and Displaying Device Information ................................................ 154
  Initializing the Page .................................................................................. 154
  Inputting Keys ......................................................................................... 155
  Getting & Displaying Device Information ............................................... 155
  Setting Device Object ............................................................................ 156
  Source Code of deviceinfo01.html ......................................................... 157
Handling Key Inputs Using JavaScript Events .............................................. 162
  Initializing the Page .................................................................................. 162
  Handling Events ...................................................................................... 163
  Displaying Text ....................................................................................... 165
  Executing Test ......................................................................................... 166
  Source Code of keydownuppress.html .................................................... 168
Handling Key Repeat Using JavaScript Events .......................................... 176
  Initializing the Page .................................................................................. 176
  Handling Events ...................................................................................... 177
  Source Code of keyrepeat.html .............................................................. 179
Implementing Functionality of Back and Exit Keys ..................................... 185
  Initializing the Page .................................................................................. 185
  Inputting Keys ......................................................................................... 186
  Handling Keys ......................................................................................... 186
  Source Code of backexit.html ............................................................... 187
Implementing Functionality of Multi Audio ................................................ 191
  Initializing the Page .................................................................................. 191
  Inputting Keys ......................................................................................... 192
  Changing Language ................................................................................ 192
  Moving the Video Playback Position ...................................................... 193
  Setting Object ......................................................................................... 194
Source Code of multiaudio.html........................................................................................................194
Playing Media with ASX File........................................................................................................199
  Initializing the Page......................................................................................................................199
  Inputting Keys ........................................................................................................................200
  Displaying Text .........................................................................................................................200
  Executing Test ........................................................................................................................200
  Setting Object ........................................................................................................................202
Source Code of asx.html ................................................................................................................202
Playing Video in Full Screen Mode..............................................................................................207
  Initializing the Page......................................................................................................................207
  Inputting Keys ........................................................................................................................208
  Displaying and Playing Media Object in Full Screen Mode ......................................................208
  Setting Object ........................................................................................................................209
Source Code of fullscreenvideo.html ..........................................................................................209
Quick Start Guide

This document is provided for beginners who want to start web app developing using LG IDE. With simple sample application, we explain how to setup the project and implement the codes.

The following samples and tutorials are provided in [DISCOVER > Legacy Platform (NetCast) > Tools & Samples] menu in LG Developer website.

• Web_Quick Start Sample App (Tic Tac Toe)
This sample app and tutorial describe how developers develop and test web applications using the Sound API and background music in the application.

• Web_Quick Start Sample App (ImageViewer)
This sample app and tutorial describe how developers develop and test web applications using the Image Viewer Framework API.

• Web_Quick Start Sample App (MediaPluginVideoPlayer)
This sample app and tutorial describe how developers develop and test web applications using the Media Plugin Video Player Framework API.

• Web_Quick Start Sample App (HTML5VideoPlayer)
This sample app and tutorial describe how developers develop and test web applications using the HTML5 Video Player Framework API.
App Development Guide

This document has been prepared to be used by developers as the technical specification for the LG Smart TV.

In this document, the platform of LG Smart TV is referred to as the “NetCast Platform”. This document defines the platform architecture, supported media formats, protocols and minimum LG NetCast Platform capabilities.

This document specifies the NetCast Platform for applications running on an LG Smart TV which can receive broadcasting media (terrestrial, cable and satellite network based) and IP based media.

This document specifies the web based application programming environment for the NetCast Platform.

The NetCast Platform is built on the Linux operating system.

This chapter describes the following sections.

- LG Web Application Overview
- Specifications
- Tips for Web Application Development

LG Web Application Overview

Web application is an application that is accessed over a network such as the internet or an intranet. LG Smart TV can run web application through built-in web browser engine. Built-in LG web browser engine supports recent web technologies HTML5, Ajax, JSON, XML, and etc. Developers can make creative applications based on those web technologies.

Although developing LG web application is similar in many ways to developing general web application, it is not totally the same. As LG web application is executed by web browser engine, not web browser, there is no web browser’s UI components such as URL input field, scroll bar, system button, and popup. Developers should be aware of the following list:

- UI screen must adopt LG screen resolution size. It means that every element of web application should be displayed in 1280 x 720 pixel screen resolution size. Separate the elements into several pages if displaying elements in a screen is not available since scrolling is not supported.
- The smallest font must be readable and legible to users 3.5 meter (10 foot) away from the TV screen.
- Follow the “overscan area” rule because the overscan area may not be visible on some devices such as LG media devices.
- Application must be fully navigable using 4 direction and remote keys: Up/Down/Left/Right + OK + Back
- Implement the ‘Back’ button in application to move back to the previous NetCast menu.

For more detailed information, refer to the “UI Guidelines” in Developing > Designing section in this Library.

Web applications can be deployed in two types, hosted and packaged Web application, to LG Smart TV.

- Hosted (URL-based) web application: Developers can make a web application using just icon and URL. Hosted web application is suitable for the applications containing server-based web service.

- Packaged (downloadable) web application: Suitable for the applications containing small fixed resources such as icon, html, JavaScript, and media. (Supported from NetCast 2.0)

Note
Refer to Developing > Using SDK section in this Library for details on LG web application development, packaging, and deployment.

NetCast Platform Overview

The NetCast Platform uses the LG Browser, which is based on WebKit. This is not a restricted embedded web browser but provides full features of web technology.

The platform provides rich media playback functionalities including play, pause, stop and scaling. Various streaming protocols including HTTP and MMSH are supported by the media player plugin module. It also has a DRM manager to manage digital rights and a streaming manager to control the stream. The NetCast Platform supports WM-DRM 10
PD (Not supported in NetCast 3.0), PlayReady (Not supported in NetCast 2.0), Widevine, and Verimatrix (Supported in NetCast 3.0/4.0/4.5) as its DRM solutions. NetCast Platform also supports adaptive streaming solutions. NetCast Platform supports Widevine and HLS (Http Live Streaming) as its adaptive streaming solutions from NetCast 2.0.

The following figure illustrates the architecture of the NetCast Platform system. The NetCast Platform runs on the Linux operating system.

[Figure] System Block Diagram of the NetCast Platform
Specifications

This chapter describes features of LG Browser in addition to generic Web technologies as follows. With proper using of the feature, developer can create interactive LG Smart TV web applications.

- Video Output
- Web Engine
- Recommendations for DRM/Streaming Protocols
- Protocols
- CODEC & Container
- DRM
- OSD Resolution
- Subtitle
- Images
- Cascading Style Sheets
- Streaming Method
- Play List (ASX)
- Local File Play

Video Output

LG Electronics Television sets do not have video output for Web content except to the main panel display. Therefore, there is no need to provide any copy protection data with Web content as such content is only ever rendered to the screen. i.e., HDCP and Macrovision are not required for the NetCast Platform.

Web Engine

The NetCast Platform implements the LG browser engine, which is based on WebKit 537.1+ as an application programming environment for content providers and content aggregators. The supported features are listed in the following table.

<table>
<thead>
<tr>
<th>Specification</th>
<th>LG Web browser engine</th>
</tr>
</thead>
<tbody>
<tr>
<td>HTTP, HTTPS</td>
<td>Yes</td>
</tr>
<tr>
<td>HTML 4.01</td>
<td>Yes</td>
</tr>
<tr>
<td>HTML 5.0 [†]</td>
<td>Partly</td>
</tr>
<tr>
<td>XHTML 1.0/1.1</td>
<td>Yes</td>
</tr>
<tr>
<td>XML</td>
<td>Yes</td>
</tr>
<tr>
<td>XSLT</td>
<td>Yes</td>
</tr>
<tr>
<td>XPath</td>
<td>Yes</td>
</tr>
<tr>
<td>CSS 2.1 [†]</td>
<td>Yes</td>
</tr>
<tr>
<td>CSS 3 [†]</td>
<td>Partly</td>
</tr>
<tr>
<td>CSS 2D/3D Transforms, Transitions, Animation [†]</td>
<td>Partly</td>
</tr>
<tr>
<td>CSS TV Profile 1.0</td>
<td>Yes</td>
</tr>
<tr>
<td>Canvas [†]</td>
<td>Partly</td>
</tr>
<tr>
<td>SVG [†]</td>
<td>Partly</td>
</tr>
<tr>
<td>DOM 1 and 2</td>
<td>Yes</td>
</tr>
<tr>
<td>DOM 3</td>
<td>Partly</td>
</tr>
<tr>
<td>JavaScript (version &gt; 1.6)</td>
<td>Yes</td>
</tr>
<tr>
<td>XMLHttpRequest (AJAX)</td>
<td>Yes</td>
</tr>
</tbody>
</table>
**Specification**

<table>
<thead>
<tr>
<th>JSON</th>
<th>LG Web browser engine</th>
</tr>
</thead>
<tbody>
<tr>
<td>Yes</td>
<td></td>
</tr>
<tr>
<td>Java</td>
<td>No</td>
</tr>
</tbody>
</table>

* Partly: a level of satisfying some of specification (60~90%).

[*] Refer to [Supported HTML5/CSS3 Standard for the NetCast Platform and Emulator](#) for detailed information.

**Cookie and Cache Support of LG Web Browser Engine**

The LG web browser engine supports cookies and provides a cache for enhanced performance. The cache is used only when the web browser engine is running and no more used if the application is terminated.

**userAgent String**

The LG web browser engine supports a userAgent String as shown below.

```plaintext
Mozilla/5.0 (operating system information) AppleWebKit/WebKit version; LG Browser/LG Browser version(capabilities; LGE; modelName; softwareVersion; hardwareVersion); LG_NetCastVersion
```

A JavaScript application can get the string from the HTML DOM property, navigator.userAgent. The same string will be included in the HTTP request User-Agent header. The following shows an example of the userAgent string of LG browser engine.

**<NetCast 2.0>**

```plaintext
Mozilla/5.0 (DirectFB; U; Linux mips; en) AppleWebKit/531.2+ (KHTML, like Gecko, Safari/531.2+); LG Browser/4.0.9(+mouse+3D+SCREEN+TUNER; LGE; ModelName; 03.05.04; 0x00000001;); LG NetCast.TV-2011
```

**<NetCast 3.0>**

```plaintext
Mozilla/5.0 (DirectFB; Linux armv7l) AppleWebKit/534.26+ (KHTML, like Gecko) Version/5.0 Safari/534.26+ LG Browser/5.00.00(+mouse+3D+SCREEN+TUNER; LGE; ModelName; 09.00.00; 0x00000001;); LG NetCast.TV-2012
```

**<NetCast 4.0>**

```plaintext
Mozilla/5.0 (Unknown; Linux armv7l) AppleWebKit/537.1+ (KHTML, like Gecko) Safari/537.1+ LG Browser/6.00.00(+mouse+3D+SCREEN+TUNER; LGE; ModelName; 03.07.01; 0x00000001); LG NetCast.TV-2013/03.17.01 (LG, ModelName, wired)
```

The following table illustrates the fields in userAgent string and User-Agent HTTP header. The "capabilities" field identifies the capability of the device.

**[Table] Fields in userAgent string and User-Agent HTTP header**

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>LG Browser version</td>
<td>The version of LG web browser engine</td>
</tr>
<tr>
<td>capabilities</td>
<td>Zero or more concatenated optional feature strings (see also Table below)</td>
</tr>
<tr>
<td>modelName</td>
<td>String for the model name</td>
</tr>
<tr>
<td>softwareVersion</td>
<td>String for the version number of the firmware</td>
</tr>
<tr>
<td>hardwareVersion</td>
<td>String for the version number of the hardware</td>
</tr>
<tr>
<td>LG_NetCastVersion</td>
<td>String for the version of NetCast Platform</td>
</tr>
</tbody>
</table>

The following table illustrates the feature strings which can be used for "capabilities" field. See also section Device Info Plugin and API in Developing > API section in this Library for information on the supported features.

**[Table] Optional feature string for capabilities field**
### Feature string

<table>
<thead>
<tr>
<th>Feature string</th>
<th>Optional feature</th>
</tr>
</thead>
<tbody>
<tr>
<td>“+3D”</td>
<td>Support for 3D display feature</td>
</tr>
<tr>
<td>“+mouse”</td>
<td>Support for mouse feature (i.e. Magic Remote)</td>
</tr>
<tr>
<td>“+PORTAL_KEY”</td>
<td>Support for portal key (VK_PORTAL)</td>
</tr>
<tr>
<td>“+SCREEN”</td>
<td>Support for TV screen</td>
</tr>
<tr>
<td>“+TUNER”</td>
<td>Support for TV tuner (e.g. DVB-T/S/C, ATSC)</td>
</tr>
<tr>
<td>“+BDP”</td>
<td>Support for Blu-ray Player</td>
</tr>
</tbody>
</table>

### Other User-Agent strings

The above userAgent string is used in the LG web browser engine implementation. However, there are several other software components which use the User-Agent header. The components and their functionalities are listed in the following tables.

These include LG NetCast TV-2013 and 2013 is updated every year.

#### [Table] Software components which use the User-Agent string, NetCast 1.0 & 2.0

<table>
<thead>
<tr>
<th>Component</th>
<th>Functionality</th>
</tr>
</thead>
<tbody>
<tr>
<td>HTTP-client</td>
<td>Delivery of DRM related information</td>
</tr>
<tr>
<td>MSDL-client</td>
<td>Delivery of streaming media file information</td>
</tr>
<tr>
<td>MSDL-http</td>
<td>Delivery of HTTP streaming media</td>
</tr>
<tr>
<td>MSDL-mms</td>
<td>Delivery of MMS streaming media</td>
</tr>
</tbody>
</table>

#### [Table] Software components which use the User-Agent string, NetCast 3.0 & 4.0

<table>
<thead>
<tr>
<th>Component</th>
<th>Functionality</th>
</tr>
</thead>
<tbody>
<tr>
<td>HTTP-client</td>
<td>Delivery of HTTP streaming media and file information</td>
</tr>
<tr>
<td>HLS-client</td>
<td>Delivery of HLS streaming media</td>
</tr>
<tr>
<td>MMS-client</td>
<td>Delivery of MMS streaming media</td>
</tr>
<tr>
<td>DRM-client</td>
<td>Delivery of DRM related information</td>
</tr>
</tbody>
</table>

The software components listed in the table above have a different User-Agent string. The following tables describe the User-Agent strings which are used for the different software components in NetCast Platform.

#### [Table] User-Agent strings used for each software components, NetCast 1.0 & 2.0

<table>
<thead>
<tr>
<th>Component</th>
<th>Functionality</th>
</tr>
</thead>
<tbody>
<tr>
<td>HTTP-client</td>
<td>Mozilla/5.0 (compatible; LG-HttpClient-v1.0.3 DLNADOC/1.50 UPnP/1.1; MSIE 8.0; Windows NT 5.1; LG_UA; AD_LOGON=LGE.NET; .NET CLR 2.0.50727; .NET CLR 3.0.04506.30; .NET CLR 3.0.04506.648; LG_UA; AD_LOGON=LGE.NET; LG NetCast.TV-2011)</td>
</tr>
<tr>
<td>MSDL-client</td>
<td>Mozilla/5.0 (DirectFB; U: Linux mips; en) AppleWebKit/528.5+ (KHTML, like Gecko, Safari/528.5+) LG Browser (; LG NetCast.TV-2011)</td>
</tr>
<tr>
<td>MSDL-http</td>
<td>MSDL (compatible; LG NetCast.TV-2011)</td>
</tr>
<tr>
<td>MSDL-mms</td>
<td>NSPlayer/11.0.5721.5145 (compatible; LG NetCast.TV-2011)</td>
</tr>
</tbody>
</table>

#### [Table] User-Agent strings used for each software components, NetCast 3.0 & 4.0

<table>
<thead>
<tr>
<th>Component</th>
<th>Functionality</th>
</tr>
</thead>
<tbody>
<tr>
<td>Component</td>
<td>Functionality</td>
</tr>
<tr>
<td>---------------</td>
<td>-------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>HTTP-client</td>
<td>GStreamer soupsrc (compatible; LG NetCast.TV-2012) libsoup/2.34.2</td>
</tr>
<tr>
<td>HLS-client</td>
<td>HLS Client/2.0 (compatible; LG NetCast.TV-2012)</td>
</tr>
<tr>
<td>MMS-client</td>
<td>NSPlayer/4.1.0.3856 (compatible; LG NetCast.TV-2012)</td>
</tr>
<tr>
<td>DRM-client</td>
<td>libsoup/2.34.2 (compatible; LG NetCast.TV-2012)</td>
</tr>
</tbody>
</table>

The *Accept-Language* Header

The LG web browser engine supports the "Accept-Language" HTTP header[^2]. If the web server has multiple versions of resources, in different languages, it can provide content in user’s preferred language. The language code is based on the ISO 639-1 standard.

The Accept-Language header is set by the "Language" option in TV menu system. Therefore, the LG web browser engine will apply a language preference if the user changes the language option in the TV menu. The TV language management system of LG Electronics is based on ISO 639-2 using alpha-3 code space. Therefore, a conversion rule is required for linking ISO 639-1 alpha-2 code space. The NetCast Platform uses the official conversion rule between ISO 639-2 and ISO 639-2[^3]. Please see Annex C for full details.

Because ISO 639-2 and ISO 639-1 have different code space size, and ISO 639-2 has a bigger code space than ISO 639-1, the full ISO 639-2 code cannot be completely converted to the ISO 639-1 code. Some codes cannot be converted into ISO 639-1. In this case, the NetCast Platform assigns the default language. Currently, the default language is English, ISO 639-1 code ‘en’. In Annex C, the blank cells in ISO 639-1 column shall be default language.

The number of languages available is limited to those provided within the LGE TV environment. Therefore, not all languages in Annex C are supported.

Protocols

The NetCast Platform supports HTTP and HTTPS for its HTML file delivery and for its media (audio and video) delivery. The NetCast Platform prefers HTTP and HTTPS for HTML delivery; HTTP, HTTPS, HLS and Widevine for VoD media delivery; HLS and Widevine for linear broadcasting.

The transport protocols which are supported by the NetCast Platform can be summarized as shown below.

[Table] Transport protocols

<table>
<thead>
<tr>
<th>Use</th>
<th>Preferred protocols</th>
<th>Supported protocols</th>
</tr>
</thead>
<tbody>
<tr>
<td>HTML file transportation</td>
<td>HTTP, HTTPS</td>
<td>HTTP, HTTPS</td>
</tr>
<tr>
<td>VoD media file transportation</td>
<td>HTTP, HTTPS, HLS, Widevine</td>
<td>HTTP, HTTPS, HLS, Widevine</td>
</tr>
<tr>
<td>Linear or live broadcasting</td>
<td>HLS, Widevine</td>
<td>HLS, Widevine</td>
</tr>
</tbody>
</table>

CODEC & Container

The NetCast Platform supports WMV-9, MPEG-2 and H.264 codecs for its video decoding and WMA-9, MPEG-1 and HE-AAC for its audio decoding.

Annex B specifies all the supported codecs and containers by the NetCast Platform in more detail.

Note

To ensure the stability of streaming h.264, LG recommends you to use maximum of four reference frames.

DRM

The NetCast Platform supports PlayReady, Widevine, and Verimatrix DRM for its digital rights management system.

[Table] Supported DRM formats in NetCast 3.0, 4.0, and 4.5
PlayReady is backwards compatible with Windows Media DRM 10 content, meaning that content encrypted with WM DRM 10 will be played on a PlayReady terminal.

PlayReady supports both pre-delivery and post-delivery methods. PlayReady uses content formats such as *.wmv / *.wma (WMDRM), *.pyv / *.pya (PlayReady) (ASF Container).

Using WMDRM/PlayReady

Pre-Delivery method (WMDRM Only)

In pre-delivery method, drmClientInfo and setDrmLicenseInfo APIs are used. It is application server that delivers content information to license server. Methods like cookie can be used but the information is not stored in TV.

1. Application server requests client information to Smart TV.
2. Client information is received from Smart TV using drmClientInfo API and delivered to the application server.
3. Application server delivers the content information to license server with the client information received from Smart TV.
4. License server that received client and content information delivers the license of the content and client to application server.
5. Application server delivers the license that has been received from the license server to Smart TV.
6. Smart TV sets DRM license using setDrmLicenseInfo API.
7. Starts to play the content using content URL.

Post-Delivery method

In post-delivery method, no APIs are used.

1. Application server delivers content URL to Smart TV and the content is started to play.
2. Smart TV acquires content information and license server URL by extracting the header from content stream.
3. Based on the acquired content information, Smart TV delivers the license acquisition data to license server directly.
4. License server delivers the requested license to Smart TV directly.
5. Starts to play the content.

See also section “drm_type” in Developing > API section in this Library for detailed information about DRM solutions.

Using Verimatrix

For Verimatrix, you must call setVMconfigData method before playing the media. As the following example, you must set the service_type to 1 when calling setVMconfigData. See setVMconfigData method in API > Device Info Plugin and API section.

```
// Example of device object in HTML
<object type="application/x-netcast-info"
    id="device"
    width="0"
    height="0">
</object>

var device = document.getElementById("device");
var result = device.setVMconfigData(1, "CompanyName", 10.1.1.1); //service_type must be 1
if (result == 1) {
    console.log("Verimatrix Initialization Success!");
} else {
    console.log("Verimatrix Initialization Fail!");
}
```

Also, you must set drmType property in media object to “verimatrix”. See drmType property in API > Media Player.
Plugin and API section.

// Example of media object in HTML
<object type="application/x-netcast-av"
  width="1280"
  height="720"
  drmType="verimatrix"
  id="media">
</object>

OSD Resolution

The NetCast Platform supports 1280x720 for its OSD resolution and 32 bpp (bit per pixel) for its OSD color depth.

Subtitle

The subtitle can be applied only when a full size video is being played.
The NetCast Platform supports SAMI (Synchronized Accessible Media Interchange), CineCanvas file format and Timed Text format (W3C Timed text markup language (TTML) 1.0) for its subtitle decoding.

Note

Characters other than ASCII are recommended to be encoded with UTF-8. (ISO8859-* or UTF-16/32 may not work normally in some conditions.)

[Table] Supported subtitle file format of LG web browser engine

<table>
<thead>
<tr>
<th>Format</th>
<th>NetCast Platform 1.0</th>
<th>NetCast Platform 2.0 or higher version</th>
</tr>
</thead>
<tbody>
<tr>
<td>SAMI</td>
<td>O</td>
<td>O</td>
</tr>
<tr>
<td>CineCanvas</td>
<td>O</td>
<td>O</td>
</tr>
<tr>
<td>Timed Text</td>
<td>X</td>
<td>O</td>
</tr>
</tbody>
</table>

These are the Reference URLs for SAMI, CineCanvas and Timed Text.
- CineCanvas : [http://www.deluxedigital.co.uk/assets/pdf/DDL_DCI_Compliant_Subtitle_Files.pdf](http://www.deluxedigital.co.uk/assets/pdf/DDL_DCI_Compliant_Subtitle_Files.pdf)
- Timed Text : [http://www.w3.org/TR/ttaf1-dfxp/](http://www.w3.org/TR/ttaf1-dfxp/)

Following is the detailed specification.

<SAMI>

[Table] Detailed specification for SAMI

<table>
<thead>
<tr>
<th>Start</th>
<th>End [second]</th>
<th>Contents</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;!--</td>
<td>--&gt;</td>
<td>Comments symbol</td>
</tr>
<tr>
<td>&lt;head&gt;</td>
<td>&lt;/head&gt;</td>
<td>Container for processing information and metadata for an HTML document</td>
</tr>
<tr>
<td>&lt;body&gt;</td>
<td>&lt;/body&gt;</td>
<td>Container for the displayable content of an HTML document</td>
</tr>
<tr>
<td>&lt;sync start&gt;</td>
<td>&lt;/sync&gt;</td>
<td>Set the start time for the caption</td>
</tr>
<tr>
<td>&lt;p class&gt;</td>
<td>&lt;/p&gt;</td>
<td>Creates a paragraph</td>
</tr>
<tr>
<td>&lt;br&gt;</td>
<td></td>
<td>Forced line break</td>
</tr>
<tr>
<td> </td>
<td>&amp;nbsp</td>
<td>Non-breaking space</td>
</tr>
<tr>
<td>&lt;</td>
<td>&amp;lt</td>
<td>Inequality(less than)</td>
</tr>
<tr>
<td>&gt;</td>
<td>&amp;gt</td>
<td>Inequality(greater than)</td>
</tr>
<tr>
<td>&amp;</td>
<td>&amp;amp</td>
<td>Definition for ampersands in HTML</td>
</tr>
<tr>
<td>Start</td>
<td>End [second]</td>
<td>Contents</td>
</tr>
<tr>
<td>---------</td>
<td>-------------</td>
<td>-----------------------------------------------</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Replace for double quotation marks</td>
</tr>
</tbody>
</table>

### <CinecanvasType>

[Table] Detailed specification for Cinecanvas Type

<table>
<thead>
<tr>
<th>Start</th>
<th>End [second]</th>
<th>Contents</th>
</tr>
</thead>
<tbody>
<tr>
<td>&quot;</td>
<td>&quot;</td>
<td>Comments symbol</td>
</tr>
<tr>
<td>&lt;</td>
<td>&lt;</td>
<td>Specifies the version of subtitle XML</td>
</tr>
<tr>
<td>&gt;</td>
<td>&gt;</td>
<td>Beginning of a particular subtitle</td>
</tr>
<tr>
<td>&amp;</td>
<td>&amp;</td>
<td>Time of the subtitle first appears on screen</td>
</tr>
<tr>
<td>&quot;</td>
<td>&quot;</td>
<td>Time of a subtitle disappears on screen</td>
</tr>
<tr>
<td>&lt;</td>
<td>&lt;</td>
<td>Actual texts are rendered on screen</td>
</tr>
<tr>
<td>&lt;</td>
<td>&lt;</td>
<td>Font to use for the subtitle</td>
</tr>
<tr>
<td> </td>
<td> </td>
<td>Forced line break</td>
</tr>
<tr>
<td>&lt;</td>
<td>&lt;</td>
<td>Non-breaking space</td>
</tr>
<tr>
<td>&lt;</td>
<td>&lt;</td>
<td>Inequality(less than)</td>
</tr>
<tr>
<td>&gt;</td>
<td>&gt;</td>
<td>Inequality(greater than)</td>
</tr>
<tr>
<td>&amp;</td>
<td>&amp;</td>
<td>Definition for ampersands in HTML</td>
</tr>
<tr>
<td>&quot;</td>
<td>&quot;</td>
<td>Replace for double quotation marks</td>
</tr>
</tbody>
</table>

### <Timed Text>

[Table] Detailed specification for Timed Text

<table>
<thead>
<tr>
<th>Start</th>
<th>End [second]</th>
<th>Contents</th>
</tr>
</thead>
<tbody>
<tr>
<td>&quot;</td>
<td>&quot;</td>
<td>Comments symbol</td>
</tr>
<tr>
<td>&lt;</td>
<td>&lt;</td>
<td>Marking a group of consecutive elements</td>
</tr>
<tr>
<td>&lt;</td>
<td>&lt;</td>
<td>Represent a paragraph</td>
</tr>
<tr>
<td>&lt;</td>
<td>&lt;</td>
<td>Time of the subtitle first appears on screen</td>
</tr>
<tr>
<td>&lt;</td>
<td>&lt;</td>
<td>Time of a subtitle disappears on screen</td>
</tr>
<tr>
<td>&lt;</td>
<td>&lt;</td>
<td>Represents the main content of the document</td>
</tr>
<tr>
<td>&lt;</td>
<td>&lt;</td>
<td>Temporal beginning and ending of a document instance</td>
</tr>
<tr>
<td>&lt;</td>
<td>&lt;</td>
<td>Forced line break</td>
</tr>
<tr>
<td> </td>
<td> </td>
<td>Non-breaking space</td>
</tr>
<tr>
<td>&lt;</td>
<td>&lt;</td>
<td>Inequality(less than)</td>
</tr>
<tr>
<td>&lt;</td>
<td>&lt;</td>
<td>Inequality(greater than)</td>
</tr>
<tr>
<td>&amp;</td>
<td>&amp;</td>
<td>Definition for ampersands in HTML</td>
</tr>
<tr>
<td>&quot;</td>
<td>&quot;</td>
<td>Replace for double quotation marks</td>
</tr>
</tbody>
</table>

See also section “subtitleOn” and “subtitle” in **Developing > API** section in this Library for the detailed information about subtitling.
Images

There are several things to consider for performance when using images on web applications. Using too large image or too many images on a single page will delay the page loading and rendering time. It is usually recommended to set the size of image within the resolution of the TV screen. Currently, applications on the NetCast Platform support a resolution of 1280x720, so it is recommended not to use the image bigger than 1280x720.

Note

Limit image sizes to less than 1 megapixel (about 1280x720 pixel).

The number of images that can be used on a page of web application is determined by the memory available on the NetCast Platform. If there is no free space available, the Platform generates the “outofmemory” event to have the application recognize the current state. In order to prevent the web application from generating the “outofmemory” event, the web browser engine should be able to manage the memory by switching pages. And, if you know the size of the image to be loaded, rendering delay can be reduced by setting “width” and “height” in the `<img>` tag.

NetCast Platform supports image decoding. The following table lists the supported image formats.

<table>
<thead>
<tr>
<th>Supported image formats</th>
</tr>
</thead>
<tbody>
<tr>
<td>JPEG</td>
</tr>
<tr>
<td>PNG</td>
</tr>
<tr>
<td>GIF</td>
</tr>
</tbody>
</table>

Cascading Style Sheets

CSS is used often when creating web applications because it can reduce repetitive tasks and offers a variety of functions.

The CSS should be optimized to improve the performance of Web application. It is not recommended to indicate an element with the relative position and size by having the consecutive parent-child structure. If CSS is in this structure, the LG web browser engine has to recalculate an absolute position and size of each element based on the parent’s style whenever processing the layout of each page. Indicating the location and size of each element as an absolute location and size is more efficient because the LG web browser engine does not have to calculate them.

In addition, web application will run faster using a clear selector like “#id” or “a:link” than using relatively an obscure selector like “:link”.

Streaming Method

HTTP Streaming

The NetCast Platform supports HTTP streaming, which means ‘Progressive Download’. This method transfers a file over HTTP protocol and plays it when the decodable part of the file is received in buffer before the entire file is downloaded.

Http Live Streaming (HLS) - Adaptive Streaming

The NetCast Platform 2.0 or higher version supports HLS (Http Live Streaming). HLS is the adaptive streaming method that - Apple has documented as an Internet-Draft (Reference URL of HLS: http://tools.ietf.org/html/draft-pantos-http-live-streaming-04).

Application authors can use HLS to send audio and video over HTTP from an ordinary web server for playback. HLS supports both live-broadcasts and prerecorded content (video on demand). HLS supports multiple alternate streams at different bit rates, and the NetCast Platform can switch streams intelligently as network bandwidth changes. HLS also provides media encryption and user authentication over HTTPS, allowing publishers to protect contents.

It is recommended that application authors use SD or HD video resolutions on the NetCast Platform. Content below SD will not have adequate video quality.

Application authors are recommended not to change the resolution of a presentation, as flicker may occur.
Note
For HLS to function correctly, meet one of the following:

• The URL includes the media file extension (e.g. .m3u8, .mpd, or .mp4).
• The MIME type is explicitly defined in media object.

To find out the supported MIME types in NetCast platform, see Annex A.

Note
The NetCast Platform 2.0 or higher version does not support the following tags: EXT-X-PROGRAM-DATA-TIME, EXT-X-ALLOW-CACHE, EXT-X-DISCONTINUITY

LG HLS client was developed as HLS draft doc ver. 04 (http://tools.ietf.org/html/draft-pantos-http-live-streaming-04). So some added functions after ver. 04 does not work or behavior can differ from other HLS client which is developed as recent HLS doc.

If resolution of source is very low (under 320x240), video may be broken or does not play normally.
Media data for HLS must include both audio and video data. LGTV does not support HLS stream which has audio only stream. Therefore, LG strongly recommends not to use audio only stream. If HLS stream has audio only stream, “CODECS” attribute must be added in #EXT-X-STREAM-INF tag so LGTV can ignore the audio only stream. Otherwise playback error occurs while stream is playing.

For example,
#EXT-X-STREAM-INF:PROGRAM-ID=1, BANDWIDTH=59573, CODECS="mp4a.40.2"

Note
To ensure the stablility of streaming h.264, LG recommends you to use maximum of four reference frames.

Play List (ASX)
The NetCast Platform supports ASX[^1] as a play list file format. However, not all features are supported. Refer to the MSDN document for more information about ASX elements.[^3]

See also section “next” and “previous” in Developing > API > Media Player Plugin and API section in this Library for the detailed information about Play List.

The NetCast Platform supports the following elements.

<table>
<thead>
<tr>
<th>Table</th>
<th>Supported elements in ASX format</th>
</tr>
</thead>
<tbody>
<tr>
<td>Element name</td>
<td>Description</td>
</tr>
<tr>
<td>ASX</td>
<td>Defines a file as a Windows Media metafile.</td>
</tr>
<tr>
<td>COPYRIGHT</td>
<td>Contains the copyright information for an ASX or ENTRY element.</td>
</tr>
<tr>
<td>ENTRY</td>
<td>Specifies the path to a media clip.</td>
</tr>
<tr>
<td>PARAM</td>
<td>Specifies the value of a custom parameter associated with a playlist or an element of a playlist.</td>
</tr>
<tr>
<td>REF</td>
<td>Specifies a URL for a piece of digital media content.</td>
</tr>
<tr>
<td>TITLE</td>
<td>Contains the title of an ASX or ENTRY element.</td>
</tr>
<tr>
<td>REPEAT</td>
<td>Defines the number of times the media player repeats one or more ENTRY or ENTRYREF elements.</td>
</tr>
<tr>
<td>BASE</td>
<td>Defines a URL string appended to the front of URLs sent to the media player.</td>
</tr>
</tbody>
</table>

Local File Play
In packaged web application, local media files such as music and video files cannot be played.
Tips for Web Application Development

This chapter explains the useful tips for Web application development as follows. When you develop Web applications, these tips can be useful to set your environment or conditions.

- **Web Browser Engine**
- **Font and Font Style**
- **Platform Identification and Its Applications**
- **Input Device**
- **Smart Text for NetCast 2.0**
- **Smart Text for NetCast 3.0 and 4.0**
- **Virtual Keyboard**
- **Performance Consideration**
- **JavaScript**

### Web Browser Engine

#### Document Scrolling

Although web page scrolling is not recommended by the NetCast Platform, due to the poor user experience, it is supported by pressing the "PR+" (Programme Up) and “PR-“ (Programme Down) buttons on the IR Remote. In the Win32 emulator, users can scroll web pages by pressing the “Page Up” and “Page Down” keys on the keyboard. Scrolling a web page in general decreases the user experience. In order to prevent users from inadvertently scrolling a web page, it is recommended that application authors adopt the code example as below.

```html
<body style='margin:0'>
<div style='width:1280px; height:720px; overflow:hidden;'>
<!-- place graphical objects here -->
</div>
</body>
```

#### Common Control Supporting

LG web application does not recommend common controls. LG Electronics recommends application authors not to use this kind of controls.

#### Flushing Cookie and Cache Repositories

Application authors may want to flush cookie and cache storage of the LG web browser engine. The NetCast Platform allows application authors to flush the cookie and cache repositories. Enter the TV main menu and select “Factory Reset” submenu in “Option” menu, then the repositories will be completely flushed.

Cookie and cache can be also removed by executing “Clear data” in LG internet browser setting.

### Font and Font Style

The NetCast 3.0 and 4.0 support two different weights of LG Display fonts and Tiresias v8.04 font. The LG Display fonts consist of ‘LG Display-Regular’ and ‘LG Display-Light’ including a Hong Kong variant. The ‘LG Display-Regular’ is the most preferred font for NetCast 3.0 and 4.0 platforms. For all of these, only a ‘plain’ style of the font is supported (i.e. not bold or italics, etc).

#### Note

It is highly recommended that application authors who start new service from NetCast 3.0 and 4.0 use ‘LG Display-Regular’ and ‘LG Display-Light’. However, those who have serviced from NetCast 2.0 or below might be able to use LG Display font for continuing services in NetCast 3.0 and 4.0.

The following example shows the using of a set of LG Display font and Tiresias font. Application authors can use this for all tags which permit the ‘style’ attribute.

```html
style='font-family:LG Display-Regular'

style='font-family:LG Display-Light'
```
The following table shows service countries which can use each font.

<table>
<thead>
<tr>
<th>Font</th>
<th>Applied countries</th>
</tr>
</thead>
<tbody>
<tr>
<td>LG Display-Regular</td>
<td>All countries except Hong Kong, Taiwan</td>
</tr>
<tr>
<td>LG Display-Light</td>
<td></td>
</tr>
<tr>
<td>LG Display</td>
<td></td>
</tr>
<tr>
<td>LG Display_HK-Regular</td>
<td>Hong Kong, Taiwan</td>
</tr>
<tr>
<td>LG Display_HK-Light</td>
<td></td>
</tr>
<tr>
<td>LG Display_HK</td>
<td></td>
</tr>
<tr>
<td>TiresiasScreenfont</td>
<td>Only Europe</td>
</tr>
</tbody>
</table>

If application authors want to apply global service, it is mandatory to use LG Display fonts (‘LG Display-Regular’ and ‘LG Display-Light’).

The NetCast Platform recommends four font sizes: ‘title large font’, ‘large font’, ‘medium font’ and ‘small font’. The following table lists the recommended size of fonts. For better display of text, application authors should follow the recommendation.

<table>
<thead>
<tr>
<th>Font size category</th>
<th>Font size (in pixel)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Title Large Font</td>
<td>45 px</td>
</tr>
<tr>
<td>Large Font</td>
<td>38 px</td>
</tr>
<tr>
<td>Medium Font</td>
<td>30 px</td>
</tr>
<tr>
<td>Small Font</td>
<td>26 px</td>
</tr>
</tbody>
</table>

Note

It is strongly recommended that LG Smart TV application authors specify the font family exactly. Different fonts could be applied for rendering a single paragraph, if LG Smart TV application author does not specify the exact font family. The rendering of text can be dirty when different fonts are applied in a single paragraph.

The LG Display fonts and Tiresias font are provided by LG Electronics to developers who have agreed to the following terms.

Font style Supporting

NetCast 3.0 and 4.0 support two different weight of fonts for suitable display. ‘LG Display-Light’ is a light weight version of ‘LG Display-Regular’ font. The difference is shown in the following figure. Moreover, the LG font-names of NetCast 1.0 and 2.0 are different. So, application authors who have been using the LG font from NetCast 1.0 must apply multi-font for NetCast 2.0 or 3.0/4.0. Otherwise, the rendering of text can be dirty when different fonts are applied in a single paragraph. Some examples of using multi-font will be given for better understanding.

Note
Application authors, who start service from NetCast 2.0 or do not want to use the LG font, do not need to use multi-font. However, application authors who service in both Netcast 2.0 and 3.0/4.0 should either use multi-font or ‘LG Display’ font only for backward compatibility.

[Figure] Comparison of thickness and width among LG Display fonts

[Table] LG font-name in NetCast 1.0

<table>
<thead>
<tr>
<th>Font</th>
<th>Applied countries</th>
</tr>
</thead>
<tbody>
<tr>
<td>LG Display_Eng</td>
<td>PAN-EU, Australia, Brazil</td>
</tr>
<tr>
<td>LG Display_Eng_Kor</td>
<td>Korea</td>
</tr>
<tr>
<td>FZHei-B01</td>
<td>China</td>
</tr>
<tr>
<td>LG_Big5HK_V</td>
<td>Hong Kong, Taiwan</td>
</tr>
</tbody>
</table>

[Table] LG font-name in NetCast 2.0

<table>
<thead>
<tr>
<th>Font</th>
<th>Applied countries</th>
</tr>
</thead>
<tbody>
<tr>
<td>LG Display</td>
<td>All countries except Hong Kong, Taiwan</td>
</tr>
<tr>
<td>LG Display_HK</td>
<td>Hong Kong, Taiwan</td>
</tr>
</tbody>
</table>

[Table] LG font-name in NetCast 3.0 and NetCast 4.0

<table>
<thead>
<tr>
<th>Font</th>
<th>Applied countries</th>
</tr>
</thead>
<tbody>
<tr>
<td>LG Display-Regular</td>
<td>All countries except Hong Kong, Taiwan</td>
</tr>
<tr>
<td>LG Display-Light</td>
<td></td>
</tr>
<tr>
<td>LG Display</td>
<td></td>
</tr>
<tr>
<td>LG Display_HK-Regular</td>
<td>Hong Kong, Taiwan</td>
</tr>
<tr>
<td>LG Display_HK-Light</td>
<td></td>
</tr>
<tr>
<td>LG Display_HK</td>
<td></td>
</tr>
</tbody>
</table>
The following shows the example of using multi-font.

\texttt{style='font-family: LG Display'}

\texttt{style='font-family: 'LG Display', 'LG Display-Regular'}

The following shows the example of using multi-font.

\texttt{style='font-family:LG Display_Eng'}

\texttt{style='font-family: 'LG Display_Eng', 'LG Display'}

The following shows the example of using multi-font.

\texttt{style='font-family: LG Display_Eng_Kor'}

\texttt{style='font-family: 'LG Display_Eng_Kor', 'LG Display'}

The following shows the example of using multi-font.

\texttt{style='font-family: FZHei-B01'}

\texttt{style='font-family: 'FZHei-B01', 'LG Display'}

The following shows the example of using multi-font.

\texttt{style='font-family: LG_Big5HK_V'}

\texttt{style='font-family: 'LG_Big5HK_V', 'LG Display_HK'}

**Representation of Control Characters**

The NetCast Platform does not support the correct rendering of control characters. In Basic Multilingual Plane of ISO/IEC 10646 (UCS or UNICODE), code positions U+0000 to U+001F, U+007F and U+0080 to U+009F are reserved for control characters. Those characters cannot have proper glyph, and the NetCast Platform represents these characters as a square box except U+0000, U+0009, U+000A, U+000B, U+000C and U+000D. Figure below shows the representation. Therefore, it is better for application authors to avoid rendering control characters.
There are several ways to uniquely identify the platform. These are:

- UserAgent string,
- User-Agent http header,
- Device information APIs (including Secure Device Identification)
- HTTPS client certificate. (Client Root CA related description is described in Annex E.)

**Identifying and Differentiating with the userAgent String**

The following example shows how application authors can identify the LG web browser engine version using the userAgent string. The application can identify the platform environment. See [userAgent String](#) for more information about the userAgent String.

```javascript
var userAgent = new String(navigator.userAgent);
if (userAgent.search(/LG Browser/) > -1)
{
  // platform dependant code
  // (in this example following code is dependent on LG web browser engine...
```

The following example shows how application authors can differentiate LGE devices using the userAgent string. The application can identify the platform environment. See [userAgent String](#) for more information about the userAgent
String.

```javascript
var userAgent = new String(navigator.userAgent);
if (userAgent.search(/LG NetCast.TV/) > -1)
{
    // platform dependant code
    // (in this example following code is dependent on LGTV device
} else if (userAgent.search(/LG NetCast.Media/) > -1)
{
    // platform dependant code
    // (in this example following code is dependent on LGMedia device
}
```

**Identifying and Differentiating with Device Information APIs**

The following example shows how application authors can differentiate an application using the device information API. The application can identify the platform environment. See section "Device Info Plugin and API" in Developing > API section in this Library for more information about the device information APIs.

```html
<object data="" id="deviceInfo" width=0 height=0 type="application/x-netcast-info">
</object>
<script>
var DeviceInfo = new String(deviceInfo.manufacturer);
if (DeviceInfo.search(/LGE/) > -1)
{
    // platform dependant code
    // (in this example following code is dependent on LGE device
... 
}
</script>
```

**Input Device**

**Traditional IR Remote**

All LG Smart TV models support the traditional IR Remote.

**Magic Remote**

Models that support the Magic Remote (a wand-like point and click IR Remote device) support all the features of the traditional IR Remote.

The Magic Remote is similar to a mouse in a computer system. The easiest way to understand the Magic Remote is to consider it as a mouse which has only left mouse button for click or double click and wheel scroll input. Figure below illustrates the Magic Remote.

Pressing the “OK” button generates “click” and “double click” events.

The other buttons are “power”, “Home”, “4-arrow key”, “volume up”, “volume down”, “channel up”, “channel down” and “mute”. In NetCast 3.0, “back”, “3D”, and “My Apps” buttons and wheel are added. In NetCast 4.0, “Voice Recognition” button is added.

“power”, “Home”, “Voice Recognition”, “volume up”, “volume down”, “3D” and “My Apps” buttons are not available for LG Smart TV application; they are used by the native TV application.

The “channel up” button generates “page up” (VK_PAGE_UP) and the “channel down” button generates “page down” (VK_PAGE_DOWN) in LG Smart TV applications.


The “back” button generates “back” (VK_BACK) in LG Smart TV applications.

Figure below illustrates the pointer which is rendered in the TV display screen when the Magic Remote is activated. Users can select one of several pointer shapes via the TV menu (Option menu).
Users activate the Magic Remote by pressing the “OK” button on the Magic Remote of NetCast 2.0 or by shaking the Magic Remote of NetCast 3.0 or higher and deactivate it by pressing any button on the traditional IR Remote. Whenever a device change occurs, a “mouseon” or “mouseoff” event is generated. From NetCast 3.0, the Magic Remote gets deactivated when the halt of the mouse movement continues for 3 seconds.

When users press any of the “4-arrow key” buttons on the Magic Remote, the pointer will disappear. (However it is still activated but without being displayed (sleep mode) in NetCast 2.0).

The “OK” button on the Magic Remote generates the “OK” keycode.

The pointers will re-appear when user shakes the Magic Remote.

Refer to Developing > API section in this Library to learn how to use the following events and methods.
- “mouseon” and “mouseoff”
- “window.NetCastGetMouseOnOff()” : How to get the mouse on or off status.
- “supportMouse”: How to check whether the TV supports Magic Remote or not.

[Figure] Magic Remote (Left: NetCast 2.0, Middle: NetCast 3.0, Right: NetCast 4.0)

[Figure] The Pointers of Magic Remote (Top: NetCast 2.0, Middle: NetCast 3.0, Bottom: NetCast 4.0)
Traditional IR Remote Key Events

Users can control an application using a traditional IR Remote. The following table lists the buttons and key events, which are available for application authors to build their applications.

**[Table] Supported image formats**

<table>
<thead>
<tr>
<th>Group</th>
<th>Key</th>
<th>Symbol</th>
<th>Mnemonic</th>
<th>Keyboard</th>
<th>Key code</th>
</tr>
</thead>
<tbody>
<tr>
<td>Numeric</td>
<td>0 - 9</td>
<td>0 - 9</td>
<td>VK_0 - VK_9</td>
<td>0 - 9</td>
<td>48-57</td>
</tr>
<tr>
<td>Arrow</td>
<td>UP</td>
<td>▲</td>
<td>VK_UP</td>
<td>↑</td>
<td>38</td>
</tr>
<tr>
<td></td>
<td>DOWN</td>
<td>▼</td>
<td>VK_DOWN</td>
<td>↓</td>
<td>40</td>
</tr>
<tr>
<td></td>
<td>LEFT</td>
<td>▼</td>
<td>VK_LEFT</td>
<td>←</td>
<td>37</td>
</tr>
<tr>
<td></td>
<td>RIGHT</td>
<td>▶</td>
<td>VK_RIGHT</td>
<td>→</td>
<td>39</td>
</tr>
<tr>
<td>Page Control</td>
<td>PAGE UP</td>
<td>↑ PAGE</td>
<td>VK_PAGE_UP</td>
<td>PgUp</td>
<td>33</td>
</tr>
<tr>
<td></td>
<td>PAGE DOWN</td>
<td>PAGE ↓</td>
<td>VK_PAGE_DOWN</td>
<td>PgDn</td>
<td>34</td>
</tr>
<tr>
<td>Selection</td>
<td>OK/SELECT</td>
<td>⊙</td>
<td>VK_ENTER</td>
<td>Enter</td>
<td>13</td>
</tr>
<tr>
<td></td>
<td>BACK/RETURN</td>
<td>BACK</td>
<td>VK_BACK</td>
<td>none</td>
<td>461</td>
</tr>
<tr>
<td>Colour</td>
<td>RED</td>
<td>●</td>
<td>VK_RED</td>
<td>none</td>
<td>403</td>
</tr>
<tr>
<td></td>
<td>GREEN</td>
<td>●</td>
<td>VK_GREEN</td>
<td>none</td>
<td>404</td>
</tr>
<tr>
<td></td>
<td>YELLOW</td>
<td>●</td>
<td>VK_YELLOW</td>
<td>none</td>
<td>405</td>
</tr>
<tr>
<td></td>
<td>BLUE</td>
<td>●</td>
<td>VK_BLUE</td>
<td>none</td>
<td>406</td>
</tr>
<tr>
<td>Playback</td>
<td>PLAY</td>
<td>▶</td>
<td>VK_PLAY</td>
<td>none</td>
<td>415</td>
</tr>
<tr>
<td></td>
<td>STOP</td>
<td>■</td>
<td>VK_STOP</td>
<td>none</td>
<td>413</td>
</tr>
<tr>
<td></td>
<td>PAUSE</td>
<td>∥</td>
<td>VK_PAUSE</td>
<td>none</td>
<td>19</td>
</tr>
<tr>
<td></td>
<td>REWIND</td>
<td>◀◀</td>
<td>VK_REWIND</td>
<td>none</td>
<td>412</td>
</tr>
<tr>
<td></td>
<td>FAST FORWARD</td>
<td>▶▶</td>
<td>VK_FAST_FWD</td>
<td>none</td>
<td>417</td>
</tr>
<tr>
<td>Service</td>
<td>INFO</td>
<td>INFO</td>
<td>VK_INFO</td>
<td>none</td>
<td>457</td>
</tr>
</tbody>
</table>

Basically, the key codes of the NetCast Platform follow those of CE-HTML, CEA-2014[7]. However the NetCast Platform does not support all key codes in CE-HTML. The full list of supported key codes are provided in the table above.

The ‘EXIT’ key is reserved for exiting the Web service function and is used by the native NetCast Platform. For this reason, the ‘EXIT’ key is not allowed to be used by application authors.

LG Electronics provide a virtual screen IR Remote for LG Smart TV Emulator. Application authors can use this emulator to develop their applications. Note that the PC emulator supports general keyboard key codes rather than IR Remote key codes.

**Key Repeat Function**

Some key events are expected to be generated repeatedly while a button is held down. The native TV software supports the key repeat function for the ‘channel up/down’, ‘page up/down’, ‘volume up/down’ and ‘arrow’, up/down/left/right, keys. However, only page up/down keys, arrow keys and rewind/fast forward keys are allowed in a web-based application. The following table shows the repeatable keys provided by the NetCast Platform.

**[Table] Repeatable keys**
<table>
<thead>
<tr>
<th>Group</th>
<th>Key</th>
<th>Symbol</th>
<th>Mnemonic</th>
<th>Keyboard</th>
<th>Key code</th>
</tr>
</thead>
<tbody>
<tr>
<td>Arrow</td>
<td>UP</td>
<td>▲</td>
<td>VK_UP</td>
<td>↑</td>
<td>38</td>
</tr>
<tr>
<td></td>
<td>DOWN</td>
<td>▼</td>
<td>VK_DOWN</td>
<td>↓</td>
<td>40</td>
</tr>
<tr>
<td></td>
<td>LEFT</td>
<td>◀</td>
<td>VK_LEFT</td>
<td>←</td>
<td>37</td>
</tr>
<tr>
<td></td>
<td>RIGHT</td>
<td>▶</td>
<td>VK_RIGHT</td>
<td>→</td>
<td>39</td>
</tr>
<tr>
<td>Playback</td>
<td>REWIND</td>
<td>◀◀</td>
<td>VK_REWIND</td>
<td>none</td>
<td>412</td>
</tr>
<tr>
<td></td>
<td>FAST FORWARD</td>
<td>▶▶</td>
<td>VK_FAST_FWD</td>
<td>none</td>
<td>417</td>
</tr>
<tr>
<td>Page Control</td>
<td>PAGE UP</td>
<td>↑ PAGE</td>
<td>VK_PAGE_UP</td>
<td>PgUp</td>
<td>33</td>
</tr>
<tr>
<td></td>
<td>PAGE DOWN</td>
<td>PAGE ↓</td>
<td>VK_PAGE_DOWN</td>
<td>PgDn</td>
<td>34</td>
</tr>
</tbody>
</table>

LG TV sets generate key events every 108 milliseconds. The theoretical maximum frequency is about 9.26 times per second. However, there are software overheads in processing the key events. The overall result of frequency is lower than the theoretical value of 9.26 times per second. The NetCast Platform does not guarantee the frequency of key repeat and it also depends on application’s processing load. An LG Smart TV application author can measure the frequency using a simple JavaScript application and use the measured value for further development of the application.

The key repeat function behavior of NetCast Platform

The NetCast Platform provides the same behavior of the key repeat function with the Webkit based PC web browser to aid LG Smart TV application authoring. The following figure illustrates the behavior of the key repeat function in NetCast Platform. A long key press produces multiple continuous key down events (onkeydown) and one key release event (onkeyup) at the end of pressing.

![Figure] The behavior of key repeat function

Key Event Handling

LG Smart TV application author can handle the key events by modifying the following example. See Traditional IR Remote Key Events for more information about available key codes.

```javascript
function processKeyDown(e) {
    var keycode;
    if(window.event) { // IE
        keycode = e.keyCode;
    } else if(e.which) { // Netscape/Firefox/Opera
        keycode = e.which;
    }
    switch(keycode) {
        case VK_UP: doup(); break;
        ...
    }
}
<body onkeydown="processKeyDown(event); return false">
    ...
</body>
```
Mouse Events
The following table lists and illustrates the available mouse events which are generated by the Magic Remote.

<table>
<thead>
<tr>
<th>Event</th>
<th>Generating condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>onclick</td>
<td>The Magic Remote clicks on an object</td>
</tr>
<tr>
<td>ondblclick</td>
<td>The Magic Remote double-clicks on an object</td>
</tr>
<tr>
<td>onmousedown</td>
<td>“OK” button on the Magic Remote is pressed</td>
</tr>
<tr>
<td>onmouseup</td>
<td>“OK” button on the Magic Remote is released</td>
</tr>
<tr>
<td>onmousemove</td>
<td>The Magic Remote is moved</td>
</tr>
<tr>
<td>onmouseout</td>
<td>The Magic Remote is moved off an element</td>
</tr>
<tr>
<td>onmouseover</td>
<td>The Magic Remote is moved over an element</td>
</tr>
<tr>
<td>onmousenewheel</td>
<td>The wheel of the Magic Remote is rotated (From NetCast 3.0)</td>
</tr>
</tbody>
</table>

Smart Text for NetCast 2.0
“LG Smart Text” function establishes the connection between a smartphone (Android phone and iPhone) and the TV using the text input feature of a smartphone application, “LG TV Remote 2011”.

LG web application of NetCast 2.0 supports the “LG TV Remote 2011” program (smartphone application) to control TV. With the virtual keyboard of the “LG TV Remote 2011” program, the “LG Smart Text” function allows users to enter characters into a text field of content provider’s application running on TV.

Using LG Smart Text
The following shows the “LG Smart Text” function running with the “LG TV Remote 2011” application.

[Figure] Using LG Smart Text (NetCast 2.0)

At the content provider’s application in Smart TV that needs input texts, users can input characters with their smartphone’s virtual keyboard.

System Requirements
[Table] System Requirements for Smart Text (NetCast 2.0)
<table>
<thead>
<tr>
<th>Type</th>
<th>Phone</th>
<th>TV</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hardware</td>
<td>- iPhone</td>
<td>LG Smart TV 2011</td>
</tr>
<tr>
<td></td>
<td>- Android phone (any version)</td>
<td>- NetCast 2.0</td>
</tr>
<tr>
<td>Software</td>
<td>LG TV Remote 2011 program</td>
<td>No additional software program is required.</td>
</tr>
<tr>
<td></td>
<td>(A free app that can be downloaded to a smartphone. Visit the Apple’s App Store or Android Market on your smartphone and search this app name to download.)</td>
<td></td>
</tr>
</tbody>
</table>

**Installing LG TV Remote 2011 Program into Android Phone**

This LG TV Remote 2011 program will work on both Android and iOS. The following process describes how to install the “LG TV Remote 2011” on your Android smartphone. It behaves the same for iPhone.

**Step1.** Visit the Android Market of your smartphone and select 🔍. Then, search and find the app you want.

![Step1](image1)

**Step2.** A virtual keyboard appears. To search “LG TV Remote 2011”, type the app name.

![Step2](image2)

**Step3.** After search, you can see the search result as shown in the figure.

![Step3](image3)
Step4. Select “LG TV Remote 2011” and click [Install]. Then, the app will be downloaded to your phone.

Step5. After download is complete, the following icon appears.
Pairing LG TV Remote 2011 Program with TV

The following describes how to pair "LG TV Remote 2011" program on your smartphone with LG TV.

Pairing means making a wireless connection between the "LG TV Remote 2011" program and LG Smart TV. Once you pair them, the TV searches and finds "LG TV Remote 2011" every time it is turned on.

Note
The following process is same for both iPhone and Android. The User Interface may be slightly different depending on the operating system.

Step1. Run "LG TV Remote 2011" on your smartphone and turn on your TV. At the bottom right of the TV screen, you can see the pairing key.

A pairing key consists of six random alphabet characters (A~Z.) When pairing is successfully performed or canceled, it disappears. After 1 minute, it disappears, too.

Step2. Your smartphone shows a list of connected TV sets. In the list, choose one and click "Connect" button.

The connected TV list means the list of TV sets that your phone is ready to connected. If you check the "Set as a default TV" radio button, the selected one becomes your default TV to connect. Therefore, you do not need to scroll and select the TV whenever you use "LG TV Remote 2011". The default TV is always marked with V or O.

Step3. After selecting the "Connect" button, this page appears on your smartphone. Select "Enter pairing key." If you want to go back to the previous TV list, select "Back to TV list."

In the figure below, "MODEL NAME" shows the actual TV model name that you are using (For example, 42L90QD)
“LG Smart Text” function establishes the connection between a smartphone (Android phone and iPhone) and the TV using the text input feature of a smartphone application, “LG TV Remote”.

LG web application of NetCast 3.0/4.0 supports the “LG TV Remote” program (smartphone application) to control TV. With the virtual keyboard of the “LG TV Remote” program, the “LG Smart Text” function allows users to enter characters into a text field of content provider’s application running on TV.

Using LG Smart Text

The following shows the “LG Smart Text” function running with the “LG TV Remote” application.

At the content provider’s application in Smart TV that needs input texts, users can input characters with their smartphone’s virtual keyboard.

System Requirements

[Table] System Requirements for Smart Text (NetCast 3.0 and 4.0)
<table>
<thead>
<tr>
<th>Type</th>
<th>Phone</th>
<th>TV</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hardware</td>
<td>- iPhone/iPad (4.1 and above)</td>
<td>LG Smart TV 2012 / LG Smart TV 2013</td>
</tr>
<tr>
<td></td>
<td>- Android phone/pad (any version)</td>
<td>- NetCast 3.0 / NetCast 4.0</td>
</tr>
<tr>
<td>Software</td>
<td>- LG TV Remote</td>
<td>No additional software program is required.</td>
</tr>
<tr>
<td></td>
<td>(A free app that can be downloaded to a smartphone. Visit the Apple's App Store or Android Market on your smartphone and search this app name to download.)</td>
<td></td>
</tr>
</tbody>
</table>

**Installing LG TV Remote Program into Android smartphone**

This LG TV Remote program will work on both Android and iOS. The following process describes how to install the "LG TV Remote" on your Android smartphone. It behaves the same for iPhone.

**Step1.** Visit the Android Market of your smartphone and select ![Search icon](image). Then, search and find the app you want.

![Search result](image)

**Step2.** A virtual keyboard appears. To search "LG TV Remote", type the app name.

![Keyboard](image)

**Step3.** After search, you can see the search result as shown in the figure.
Step 4. Select “LG TV Remote” and click [Install]. Then, the app will be downloaded to your phone.

Step 5. After download is complete, the following icon appears.
Pairing LG TV Remote Program with TV

The following describes how to pair “LG TV Remote” program on your smartphone with LG TV.

Pairing means making a wireless connection between the “LG TV Remote” program and LG Smart TV.

Note

The following process is same for both iPhone and Android. The User Interface may be slightly different depending on the operating system.

**Step1.** Run “LG TV Remote” on your smartphone and turn on your TV. At the bottom right of the TV screen, you can see the pairing key.

A pairing key consists of six random numeric characters (0~9). When pairing is successfully performed or canceled, it disappears. After 1 minute, it disappears, too.

**Step2.** Your smartphone shows a list of connected TV sets. In the list, choose one and click “Connect” button. The connected TV list means the list of TV sets that your phone is ready to connect.

**Step3.** After selecting a TV set, this page appears on your smartphone. Enter pairing key and select “Submit” button. If you want to go back to the previous TV list, select “Back” button.
Virtual Keyboard

Virtual Keyboard is designed to appear as below when a text input element is focused. Developers can easily use virtual keyboard by including virtual keyboard library.

USB keyboard input and Voice Recognition are supported since NetCast 3.0 (LG Smart TV 2012).

To download the Virtual Keyboard Library, go to [DISCOVER > Legacy Platform (NetCast) > Tools & Samples] menu in LG Developer (http://webostv.developer.lge.com) website.

Using Virtual Keyboard Library

To use the virtual keyboard, add the following script in the head of html file. In order to use virtual keyboard in iframes, it should be added in the html file connected to the iframe.
(Relative path of the jsLgVKeyboard folder should be modified according to the real path.)

In a normal or parent page
<link type="text/css" rel="Stylesheet" href="../jsLgVKeyboard/LgVKeyboard.css" />
<script id="mainVKScript" type="text/javascript" src="../jsLgVKeyboard/LgVKeyboard.js"></script>

In iframe page
<script type="text/javascript" src="../jsLgVKeyboard/LgVKeyboardIframe.js"></script>

As Virtual Keyboard uses keydown and keyup events, use the lgKb.onKeyDown and lgKb.onKeyUp methods as below to use keydown event in the application instead of setting handler of window.onkeydown event.

```javascript
function onKeyDown(event) {
    var keyCode = event.keyCode;
    switch(keyCode) {
    case VK_LEFT :
        // handle keydown event
    } // end case
} // end function
```
```javascript
function onKeyUp(event) {
    var keyCode = event.keyCode;
    switch(keyCode) {
        case VK_LEFT :
            ...
            break;
        case VK_RIGHT :
            ...
            break;
        ...
    }
}

function onKeyPress(event) {
    var keyCode = event.keyCode;
    switch(keyCode) {
        case VK_LEFT :
            ...
            break;
        case VK_RIGHT :
            ...
            break;
        ...
    }
}

lgKb.onKeyDown = onKeyPress;
lgKb.onKeyUp = onKeyPress;
</script>

Virtual Keyboard library calls lgKb.onKeyboardShow or lgKb.onKeyboardHide methods after Virtual Keyboard appear or disappear. Override the lgKb.onKeyboardShow and lgKb.onKeyboardHide methods as below for additional task.

```javascript
function onKeyboardShow(event) {
    ...
}

function onKeyboardHide(event) {
    ...
}

lgKb.onKeyboardShow = onKeyboardShow;
lgKb.onKeyboardHide = onKeyboardHide;
</script>

- Do not modify the folder structure and file name of jsLgVKeyboard.
- The script must include the following declaration: `id = "mainVKScript"`
- A web page should be set character set as UTF-8 to use virtual keyboard as following:

```html
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
```

- Virtual keyboard is enabled in the following tag/types:

<table>
<thead>
<tr>
<th>Tag</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>INPUT</td>
<td>TEXT</td>
</tr>
<tr>
<td>INPUT</td>
<td>PASSWORD</td>
</tr>
<tr>
<td>TEXTAREA</td>
<td>-</td>
</tr>
</tbody>
</table>
Beware of cross-domain problem when using iframe. If an application uses multiple html files those are loaded from different domains, cross-domain problem occurs and virtual keyboard cannot be used in iframe.

Supported Language
Virtual keyboard supports 42 languages, and 16 languages are supported for voice recognition of them. Refer to Developing > API > Voice Recognition Plugin and API section in this Library for detailed information.

Performance Consideration
This section describes the performance considerations for web application developers.

• Application screen size “1280 x 720” is recommended. If the width is bigger than the height, invisible area may occur.
• With the use of authoring tools, you can make Web applications that do not have complex structure and unnecessary codes. When using an Web authoring tool, you must check if it supports the standard markup and CSS layout. Also, check if grammers are correctly applied even after you finished the development. W3C validator can be used to find some mistakes in the content (http://validator.w3.org)
• Since TV has lower performance than PC, optimization must be done with JavaScript functions. A time-consuming task should be separated into several tasks so that it can be implemented with several functions.
• If animation and transition features are implemented without considering performance using UI Framework library which is based on JavaScript and CSS, performance issues can be caused.
• Although the Transparent function with the usage of CSS filter is useful for making fancy UI, it may cause performance issue due to the huge operations.
• If you minimize the usage of JavaScript functions in the initial page, the execution time will be reduced. Hence, we recommend to make the ‘onload’ function simple.
• For Web application’s optimization, we recommend to analyze and optimize the contents to the page speed. (https://developers.google.com/speed/pagespeed/)
• To close the application, call window.NetCastBack() function.
• The speed of Web application will be significantly different depending on the size of each page and how complex the Web application is. If a Web application consists of one page implemented with multiple content blocks including hidden blocks, this may cause performance issues. In this case, the application has to be divided into multiple images, and each page has to be consists of limited number of content blocks to improve the performance.
• It is recommended not to use a redirect page because it deteriorates the performance by increasing the loading time.

Note
Keep pagesize small and limit the number of elements in a page.

JavaScript

General
When implementing code, it is important to optimize the code as much as possible because the NetCast Platform has lower computing performance than general PC platforms. In general, it is better to avoid using complex expressions in order to optimize the speed of Web application. Nested loops should not be used as much as possible. Compressing the web application is not always necessary, but removing unnecessary tags and CSS codes and shortening the name of functions and variables will help to improve performance.

One of the most frequent mistakes made by inexperienced developers is trying to use the element within the <body> tag in the Javascript to initialize the web page before the all <body> tags loaded. This will cause an error because the element cannot be used in the Javascript. Therefore, when initializing a Web page using the element included in the <body> tag, you need to use the OnLoad event handler that is generated only after the web page is fully loaded.
The following is a code sample of how you should not call your Javascript functions:

```html
<script type="text/javascript">
    function testFunction()
    {
        document.getElementById('testDiv').innerHTML='Test text';
    }
    testFunction();
</script>
</head>
<body>
...<div id="testDiv">Initial text</div>
</body>
```

A correct code using the body-onload is provided as follows:

```html
<script type="text/javascript">
    function testFunction()
    {
        document.getElementById('testDiv').innerHTML='Test text';
    }
</script>
</head>
<body onload="testFunction();">
...<div id="testDiv">Initial text</div>
</body>
```

Note
You must use the OnLoad event handler of the <body> tag to initialize web pages.

Timers
Timer is often used to implement animation effects using Javascript. Using a delay of 100 msec or less on the NetCast Platform is insignificant.

Also, using multiple timers on a single page is not desirable. If you do, the NetCast Platform does not work properly.

Note
When using a timer on the NetCast Platform, the delay time should be set to longer than 100 msec.

XMLHttpRequest
The following is an excerpt for using the XMLHttpRequest object:

```javascript
xmlHttp = new XMLHttpRequest();
...
xmlHttp.onreadystatechange = handleStateChange;
xmlHttp.open("GET", "http://testserver/testdata.xml", true);
xmlHttp.send(null);

function handleStateChange() {
    ...
}
...```

To increase performance, one can use JavaScript Object Notation (JSON) instead. An example of a JSON data
structure is provided as follows:

```json
{
  "tagName": "testObject",
  "tagProperties": {
    "data": "testData"
  },
  "tagContent": [
    "test content 1",
    "test content 2"
  ]
}
```

The NetCast platform has support for native JSON parsing using JSON.parse().

**Eval**

The eval function is very fast. However, it can compile and execute any JavaScript program, so there can be security issues. The use of eval is indicated when the source is trusted and competent. It is much safer to use a JSON parser. In web applications over XMLHttpRequest, communication is permitted only to the same origin that provide that page, so it is trusted. But it might not be competent. If the server is not rigorous in its JSON encoding, or if it does not scrupulously validate all of its inputs, then it could deliver invalid JSON text that could be carrying dangerous script. The eval function would execute the script, unleashing its malice.

To defend against this, a JSON parser should be used. A JSON parser will recognize only JSON text, rejecting all scripts. In NetCast platform, JSON parsers are also much faster than eval.

**Popups**

Popup is not supported because only one window is available for a Web application on the NetCast Platform. To implement the popup window effect, use the `<div>` tag, instead of window.open.

---

**Note**

The NetCast Platform does not support popups for a Web application.

---

**Annex A Complete List of Supported MIME Types**

The following table contains all MIME types which are currently supported by the NetCast platform.

<table>
<thead>
<tr>
<th>Container</th>
<th>MIME type</th>
</tr>
</thead>
<tbody>
<tr>
<td>general supported media</td>
<td>application/x-netcast-av</td>
</tr>
<tr>
<td>Device Information</td>
<td>application/x-netcast-info</td>
</tr>
<tr>
<td>WMV (ASF)</td>
<td>video/x-ms-asf, video/x-ms-wmv</td>
</tr>
<tr>
<td>WMA</td>
<td>audio/x-ms-wma</td>
</tr>
<tr>
<td>MP3</td>
<td>audio/mpeg, audio/mp3</td>
</tr>
<tr>
<td>MP4</td>
<td>video/mp4, video/mpeg</td>
</tr>
<tr>
<td>m3u8</td>
<td>application/vnd.apple.mpegurl</td>
</tr>
</tbody>
</table>
## Annex B Complete List of Supported Codecs and Containers

<table>
<thead>
<tr>
<th>Common File Extension</th>
<th>Media Container</th>
<th>A/V</th>
<th>Codec</th>
<th>Four CC</th>
<th>Profile Level</th>
<th>Max Bitrate</th>
<th>Max Resolution</th>
<th>Note</th>
</tr>
</thead>
<tbody>
<tr>
<td>*mp4</td>
<td>MP4</td>
<td>Video</td>
<td>H.264/AVC</td>
<td>H.264 AVC1</td>
<td>Baseline@L1.2 / L3.0</td>
<td>700 kbps (min)</td>
<td>4000 kbps</td>
<td>1920x1080</td>
</tr>
<tr>
<td>*.wv</td>
<td>ASF</td>
<td>Video</td>
<td>VC-1 Advanced Profile</td>
<td>WVC1</td>
<td>Advanced Profile@L1 / L3</td>
<td>6000 kbps</td>
<td>1280x720</td>
<td>30 fps</td>
</tr>
<tr>
<td>*.asf</td>
<td>ASX</td>
<td>Audio</td>
<td>WMA 9 Standard</td>
<td>WMA 2</td>
<td>32 ~ 192 kbps, 48kHz</td>
<td>Cannot support - over 48 KHz sampling rate - over 6 channel lossless - WMA10 Pro LBR v2, v3 (M0b profile)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>*.asf</td>
<td>ASX</td>
<td>Audio</td>
<td>WMA 9 PRO</td>
<td>WMA 3</td>
<td>48 ~ 384 kbps, 48kHz</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>*.wvm</td>
<td>Widevine proprietary</td>
<td>Video</td>
<td>H.264</td>
<td>H.264</td>
<td>Main Profile@L3.0 / L3.1 / L4.0</td>
<td>Adaptive streaming: 600kbps ~ 8300kbps</td>
<td>720x406 @ 25 fps</td>
<td>1280x720</td>
</tr>
<tr>
<td>*.ts</td>
<td>TS</td>
<td>Audio</td>
<td>H.264</td>
<td></td>
<td></td>
<td>3000 kbps</td>
<td>1920x1080</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Audio</td>
<td>AAC</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Note:** AAC-LC and AAC-HE
### Common File Extensions

<table>
<thead>
<tr>
<th>Extension</th>
<th>Media Container</th>
<th>A/V</th>
<th>Codec</th>
<th>Four CC</th>
<th>Profile Level</th>
<th>@ Max Bitrate</th>
<th>Max Resolution</th>
<th>Note</th>
</tr>
</thead>
<tbody>
<tr>
<td>*.mp3</td>
<td>MP3</td>
<td>Audio</td>
<td>MPEG1 Layer III</td>
<td></td>
<td></td>
<td>320 kbps</td>
<td></td>
<td></td>
</tr>
<tr>
<td>*.mp3</td>
<td>ASX</td>
<td>Audio</td>
<td>MP3</td>
<td></td>
<td></td>
<td>320 kbps</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
| *.wma     | ASF             | Audio | WMA 9 Standard | WMA 2  |               | 32 ~ 192 kbps, 48 kHz |      | Cannot support:  
  - over 48 KHz sampling rate  
  - over 6 channel lossless  
  - WMA10 Pro LBR v2, v3 (M0b profile) |

Video and audio data should be interleaved in time base when they are encoded in the file.

### Annex C ISO 639 Code Conversion Table

<table>
<thead>
<tr>
<th>ISO 639-2</th>
<th>ISO 639-1</th>
<th>Name of language</th>
</tr>
</thead>
<tbody>
<tr>
<td>aar</td>
<td>aa</td>
<td>Afar</td>
</tr>
<tr>
<td>abk</td>
<td>ab</td>
<td>Abkhazian</td>
</tr>
<tr>
<td>ace</td>
<td></td>
<td>Achinese</td>
</tr>
<tr>
<td>ach</td>
<td></td>
<td>Acoli</td>
</tr>
<tr>
<td>ad</td>
<td></td>
<td>Audio Description (France HD Forum)</td>
</tr>
<tr>
<td>ada</td>
<td></td>
<td>Adangme</td>
</tr>
<tr>
<td>ady</td>
<td></td>
<td>Adyghe; Adygei</td>
</tr>
<tr>
<td>afa</td>
<td></td>
<td>Afro-Asiatic (Other)</td>
</tr>
<tr>
<td>ath</td>
<td></td>
<td>Afrihili</td>
</tr>
<tr>
<td>afr</td>
<td>af</td>
<td>Afrikaans</td>
</tr>
<tr>
<td>ain</td>
<td></td>
<td>Ainu</td>
</tr>
<tr>
<td>aka</td>
<td>ak</td>
<td>Akan</td>
</tr>
<tr>
<td>akk</td>
<td></td>
<td>Akkadian</td>
</tr>
<tr>
<td>alb</td>
<td>sq</td>
<td>Albanian (ISO 639-2/B)</td>
</tr>
<tr>
<td>ale</td>
<td></td>
<td>Aleut</td>
</tr>
<tr>
<td>alg</td>
<td></td>
<td>Algonquian languages</td>
</tr>
<tr>
<td>alt</td>
<td></td>
<td>Southern Altai</td>
</tr>
<tr>
<td>amh</td>
<td>am</td>
<td>Amharic</td>
</tr>
<tr>
<td>ang</td>
<td></td>
<td>English, Old (ca. 450-1100)</td>
</tr>
<tr>
<td>anp</td>
<td></td>
<td>Angika</td>
</tr>
<tr>
<td>apa</td>
<td></td>
<td>Apache languages</td>
</tr>
<tr>
<td>ara</td>
<td>ar</td>
<td>Arabic</td>
</tr>
<tr>
<td>arc</td>
<td></td>
<td>Official Aramaic (700-300 BCE); Imperial Aramaic (700-300 BCE)</td>
</tr>
<tr>
<td>ISO 639-2</td>
<td>ISO 639-1</td>
<td>Name of language</td>
</tr>
<tr>
<td>----------</td>
<td>----------</td>
<td>--------------------------------------</td>
</tr>
<tr>
<td>arg</td>
<td>an</td>
<td>Aragonese</td>
</tr>
<tr>
<td>arm</td>
<td>hy</td>
<td>Armenian (ISO 639-2/B)</td>
</tr>
<tr>
<td>arn</td>
<td></td>
<td>Araucanian; Mapuche</td>
</tr>
<tr>
<td>arp</td>
<td></td>
<td>Arapahoe</td>
</tr>
<tr>
<td>art</td>
<td></td>
<td>Artificial (Other)</td>
</tr>
<tr>
<td>arw</td>
<td></td>
<td>Arawak</td>
</tr>
<tr>
<td>asm</td>
<td>as</td>
<td>Assamese</td>
</tr>
<tr>
<td>ast</td>
<td></td>
<td>Asturian; Bable; Leonese; Asturleonese</td>
</tr>
<tr>
<td>ath</td>
<td></td>
<td>Athapaskan languages</td>
</tr>
<tr>
<td>aus</td>
<td></td>
<td>Australian languages</td>
</tr>
<tr>
<td>ava</td>
<td>av</td>
<td>Avaric</td>
</tr>
<tr>
<td>ave</td>
<td>ae</td>
<td>Avestan</td>
</tr>
<tr>
<td>awa</td>
<td></td>
<td>Awadhi</td>
</tr>
<tr>
<td>ayam</td>
<td>ay</td>
<td>Aymara</td>
</tr>
<tr>
<td>aze</td>
<td>az</td>
<td>Azerbaijani</td>
</tr>
<tr>
<td>bad</td>
<td></td>
<td>Banda</td>
</tr>
<tr>
<td>bai</td>
<td></td>
<td>Bamileke languages</td>
</tr>
<tr>
<td>bak</td>
<td>ba</td>
<td>Bashkir</td>
</tr>
<tr>
<td>bal</td>
<td></td>
<td>Baluchi</td>
</tr>
<tr>
<td>bam</td>
<td>bm</td>
<td>Bambara</td>
</tr>
<tr>
<td>ban</td>
<td></td>
<td>Balinese</td>
</tr>
<tr>
<td>baq</td>
<td>eu</td>
<td>Basque (ISO 639-2/B)</td>
</tr>
<tr>
<td>bas</td>
<td></td>
<td>Basa</td>
</tr>
<tr>
<td>bat</td>
<td></td>
<td>Baltic (Other)</td>
</tr>
<tr>
<td>bej</td>
<td></td>
<td>Beja; Bedawiyet</td>
</tr>
<tr>
<td>bel</td>
<td>be</td>
<td>Belarusian</td>
</tr>
<tr>
<td>bem</td>
<td></td>
<td>Bemba</td>
</tr>
<tr>
<td>ben</td>
<td>bn</td>
<td>Bengali</td>
</tr>
<tr>
<td>ber</td>
<td></td>
<td>Berber (Other)</td>
</tr>
<tr>
<td>bho</td>
<td></td>
<td>Bhojpuri</td>
</tr>
<tr>
<td>bih</td>
<td>bh</td>
<td>Bihari</td>
</tr>
<tr>
<td>bik</td>
<td></td>
<td>Bikol</td>
</tr>
<tr>
<td>bin</td>
<td></td>
<td>Bini; Edo</td>
</tr>
<tr>
<td>bis</td>
<td>bi</td>
<td>Bislama</td>
</tr>
<tr>
<td>bla</td>
<td></td>
<td>Siksika</td>
</tr>
<tr>
<td>bnt</td>
<td></td>
<td>Bantu (Other)</td>
</tr>
<tr>
<td>bod</td>
<td>bo</td>
<td>Tibetan (ISO 639-2/T)</td>
</tr>
<tr>
<td>bos</td>
<td>bs</td>
<td>Bosnian (not from ISO 639-2 1998, but from Wekipedia)</td>
</tr>
<tr>
<td>bra</td>
<td></td>
<td>Braj</td>
</tr>
<tr>
<td>ISO 639-2</td>
<td>ISO 639-1</td>
<td>Name of language</td>
</tr>
<tr>
<td>-----------</td>
<td>-----------</td>
<td>------------------</td>
</tr>
<tr>
<td>bre</td>
<td>br</td>
<td>Breton</td>
</tr>
<tr>
<td>btk</td>
<td>btk</td>
<td>Batak (Indonesia)</td>
</tr>
<tr>
<td>bua</td>
<td>bu</td>
<td>Buriat</td>
</tr>
<tr>
<td>bug</td>
<td>bug</td>
<td>Buginese</td>
</tr>
<tr>
<td>bul</td>
<td>bg</td>
<td>Bulgarian</td>
</tr>
<tr>
<td>bur</td>
<td>my</td>
<td>Burmese (ISO 639-2/B)</td>
</tr>
<tr>
<td>byn</td>
<td>my</td>
<td>Blin; Blin</td>
</tr>
<tr>
<td>cad</td>
<td>cad</td>
<td>Caddo</td>
</tr>
<tr>
<td>cai</td>
<td>cai</td>
<td>Central American Indian (Other)</td>
</tr>
<tr>
<td>car</td>
<td>ca</td>
<td>Galibi; Carib</td>
</tr>
<tr>
<td>cat</td>
<td>ca</td>
<td>Catalan; Valencian</td>
</tr>
<tr>
<td>cau</td>
<td>ca</td>
<td>Caucasian (Other)</td>
</tr>
<tr>
<td>cdo</td>
<td>cdo</td>
<td>Min Dong Chinese (from ISO 639-3)</td>
</tr>
<tr>
<td>ceb</td>
<td>ceb</td>
<td>Cebuano</td>
</tr>
<tr>
<td>cel</td>
<td>cel</td>
<td>Celtic languages</td>
</tr>
<tr>
<td>ces</td>
<td>cs</td>
<td>Czech (ISO 639-2/T)</td>
</tr>
<tr>
<td>cha</td>
<td>ch</td>
<td>Chamorro</td>
</tr>
<tr>
<td>chb</td>
<td>ch</td>
<td>Chibcha</td>
</tr>
<tr>
<td>che</td>
<td>ce</td>
<td>Chechen</td>
</tr>
<tr>
<td>chg</td>
<td>ch</td>
<td>Chagatai</td>
</tr>
<tr>
<td>chi</td>
<td>zh</td>
<td>Chinese (ISO 639-2/B) : means Traditional Chinese in HK Subtitle (by Labelling Scheme Meeting 20080617.doc)</td>
</tr>
<tr>
<td>chk</td>
<td>ch</td>
<td>Chuukese</td>
</tr>
<tr>
<td>chm</td>
<td>ch</td>
<td>Mari</td>
</tr>
<tr>
<td>chn</td>
<td>ch</td>
<td>Chinook jargon</td>
</tr>
<tr>
<td>cho</td>
<td>ch</td>
<td>Choctaw</td>
</tr>
<tr>
<td>chp</td>
<td>ch</td>
<td>Chipewyan; Dene Suline</td>
</tr>
<tr>
<td>chr</td>
<td>ch</td>
<td>Cherokee</td>
</tr>
<tr>
<td>chs</td>
<td>ch</td>
<td>Simplified Chinese in HK Subtitle (by Labelling Scheme Meeting 20080617.doc) : Eventhough it conflicts with Chumash in ISO 639-3, Chumash is extinct.</td>
</tr>
<tr>
<td>chu</td>
<td>cu</td>
<td>Church Slavic; Old Slavonic; Church Slavonic; Old Bulgarian; Old Church Slavonic</td>
</tr>
<tr>
<td>chv</td>
<td>cv</td>
<td>Chuvash</td>
</tr>
<tr>
<td>chy</td>
<td>ch</td>
<td>Cheyenne</td>
</tr>
<tr>
<td>cmc</td>
<td>cm</td>
<td>Chamic languages</td>
</tr>
<tr>
<td>cmn</td>
<td>zh-cn</td>
<td>Mandarin Chinese (from ISO 639-3)</td>
</tr>
<tr>
<td>cop</td>
<td>cop</td>
<td>Coptic</td>
</tr>
<tr>
<td>cor</td>
<td>kw</td>
<td>Cornish</td>
</tr>
<tr>
<td>cos</td>
<td>co</td>
<td>Corsican</td>
</tr>
<tr>
<td>cpe</td>
<td>cpe</td>
<td>Creoles and pidgins, English based (Other)</td>
</tr>
<tr>
<td>cpf</td>
<td>cpf</td>
<td>Creoles and pidgins, French-based (Other)</td>
</tr>
<tr>
<td>ISO 639-2</td>
<td>ISO 639-1</td>
<td>Name of language</td>
</tr>
<tr>
<td>-----------</td>
<td>-----------</td>
<td>------------------</td>
</tr>
<tr>
<td>cpp</td>
<td></td>
<td>Creoles and pidgins, Portuguese-based (Other)</td>
</tr>
<tr>
<td>cre</td>
<td>cr</td>
<td>Cree</td>
</tr>
<tr>
<td>crh</td>
<td></td>
<td>Crimean Tatar; Crimean Turkish</td>
</tr>
<tr>
<td>crp</td>
<td></td>
<td>Creoles and pidgins (Other)</td>
</tr>
<tr>
<td>csb</td>
<td></td>
<td>Kashubian</td>
</tr>
<tr>
<td>cus</td>
<td></td>
<td>Cushitic (Other)</td>
</tr>
<tr>
<td>cym</td>
<td>cy</td>
<td>Welsh (ISO 639-2/T)</td>
</tr>
<tr>
<td>cze</td>
<td>cs</td>
<td>Czech (ISO 639-2/B)</td>
</tr>
<tr>
<td>dak</td>
<td></td>
<td>Dakota</td>
</tr>
<tr>
<td>dan</td>
<td>da</td>
<td>Danish</td>
</tr>
<tr>
<td>dar</td>
<td></td>
<td>Dargwa</td>
</tr>
<tr>
<td>day</td>
<td></td>
<td>Dayak</td>
</tr>
<tr>
<td>del</td>
<td></td>
<td>Delaware</td>
</tr>
<tr>
<td>den</td>
<td></td>
<td>Slave (Athapaskan)</td>
</tr>
<tr>
<td>deu</td>
<td>de</td>
<td>German (ISO 639-2/T)</td>
</tr>
<tr>
<td>dgr</td>
<td></td>
<td>Dogrib</td>
</tr>
<tr>
<td>din</td>
<td></td>
<td>Dinka</td>
</tr>
<tr>
<td>div</td>
<td>dv</td>
<td>Divehi; Dhivehi; Maldivian</td>
</tr>
<tr>
<td>doi</td>
<td></td>
<td>Dogri</td>
</tr>
<tr>
<td>dra</td>
<td></td>
<td>Dravidian (Other)</td>
</tr>
<tr>
<td>dsb</td>
<td></td>
<td>Lower Sorbian</td>
</tr>
<tr>
<td>dua</td>
<td></td>
<td>Duala</td>
</tr>
<tr>
<td>dum</td>
<td></td>
<td>Dutch, Middle (ca. 1050-1350)</td>
</tr>
<tr>
<td>dut</td>
<td>nl</td>
<td>Dutch; Flemish (ISO 639-2/B)</td>
</tr>
<tr>
<td>dyu</td>
<td></td>
<td>Dyula</td>
</tr>
<tr>
<td>dzo</td>
<td>dz</td>
<td>Dzongkha</td>
</tr>
<tr>
<td>efi</td>
<td></td>
<td>Efik</td>
</tr>
<tr>
<td>egy</td>
<td></td>
<td>Egyptian (Ancient)</td>
</tr>
<tr>
<td>eka</td>
<td></td>
<td>Ekajuk</td>
</tr>
<tr>
<td>ell</td>
<td>el</td>
<td>Greek, Modern (post 1453) (ISO 639-2/T)</td>
</tr>
<tr>
<td>elx</td>
<td></td>
<td>Elamite</td>
</tr>
<tr>
<td>eng</td>
<td>en-gb, en-au, en-us,</td>
<td>English, 'en-us' is for USA and Taiwan, 'en-au' is for Australia, 'en-gb' is for other nations</td>
</tr>
<tr>
<td>enm</td>
<td></td>
<td>English, Middle (1100-1500)</td>
</tr>
<tr>
<td>epo</td>
<td>eo</td>
<td>Esperanto</td>
</tr>
<tr>
<td>esp</td>
<td>es</td>
<td>Spanish; Castilian (ISO 639-2/T)</td>
</tr>
<tr>
<td>est</td>
<td>et</td>
<td>Estonian</td>
</tr>
<tr>
<td>eus</td>
<td>eu</td>
<td>Basque (ISO 639-2/T)</td>
</tr>
<tr>
<td>ewe</td>
<td>ee</td>
<td>Ewe</td>
</tr>
<tr>
<td>ISO 639-2</td>
<td>ISO 639-1</td>
<td>Name of language</td>
</tr>
<tr>
<td>-----------</td>
<td>-----------</td>
<td>------------------</td>
</tr>
<tr>
<td>ewo</td>
<td></td>
<td>Ewondo</td>
</tr>
<tr>
<td>fan</td>
<td></td>
<td>Fang</td>
</tr>
<tr>
<td>fao</td>
<td>fo</td>
<td>Faroese</td>
</tr>
<tr>
<td>fas</td>
<td>fa</td>
<td>Persian (ISO 639-2/T)</td>
</tr>
<tr>
<td>fat</td>
<td>fj</td>
<td>Fanti</td>
</tr>
<tr>
<td>fil</td>
<td>fi</td>
<td>Filipino; Pilipino</td>
</tr>
<tr>
<td>fin</td>
<td>fi</td>
<td>Finnish</td>
</tr>
<tr>
<td>fiu</td>
<td></td>
<td>Finno-Ugrian (Other)</td>
</tr>
<tr>
<td>fon</td>
<td></td>
<td>Fon</td>
</tr>
<tr>
<td>fra</td>
<td>fr, fr-ca</td>
<td>French (ISO 639-2/T), 'fr-ca' is for Canada</td>
</tr>
<tr>
<td>fre</td>
<td>fr, fr-ca</td>
<td>French (ISO 639-2/B), 'fr-ca' is for Canada</td>
</tr>
<tr>
<td>frm</td>
<td></td>
<td>French, Middle (ca. 1400-1600)</td>
</tr>
<tr>
<td>fro</td>
<td></td>
<td>French, Old (842-ca. 1400)</td>
</tr>
<tr>
<td>frr</td>
<td></td>
<td>Northern Frisian</td>
</tr>
<tr>
<td>frs</td>
<td></td>
<td>Eastern Frisian</td>
</tr>
<tr>
<td>fry</td>
<td>fy</td>
<td>Western Frisian</td>
</tr>
<tr>
<td>ful</td>
<td>ff</td>
<td>Fulah</td>
</tr>
<tr>
<td>fur</td>
<td></td>
<td>Friulian</td>
</tr>
<tr>
<td>gaa</td>
<td></td>
<td>Ga</td>
</tr>
<tr>
<td>gae</td>
<td></td>
<td>Gaelic (Scots) (not from ISO 639, versions below 5.0 of DBook specified this)</td>
</tr>
<tr>
<td>gay</td>
<td></td>
<td>Gayo</td>
</tr>
<tr>
<td>gba</td>
<td></td>
<td>Gbaya</td>
</tr>
<tr>
<td>gdh</td>
<td></td>
<td>Gaelic (Scots) (not from ISO 639-2 1998, maybe it was from old ver. of ISO 639)</td>
</tr>
<tr>
<td>gem</td>
<td></td>
<td>Germanic (Other)</td>
</tr>
<tr>
<td>geo</td>
<td>ka</td>
<td>Georgian (ISO 639-2/B)</td>
</tr>
<tr>
<td>ger</td>
<td>de</td>
<td>German (ISO 639-2/B)</td>
</tr>
<tr>
<td>gez</td>
<td></td>
<td>Geez</td>
</tr>
<tr>
<td>gil</td>
<td></td>
<td>Gilbertese</td>
</tr>
<tr>
<td>gla</td>
<td>gd</td>
<td>Gaelic; Scottish Gaelic</td>
</tr>
<tr>
<td>gle</td>
<td>ga</td>
<td>Irish</td>
</tr>
<tr>
<td>glg</td>
<td>gl</td>
<td>Galician</td>
</tr>
<tr>
<td>glv</td>
<td>gv</td>
<td>Manx</td>
</tr>
<tr>
<td>gmh</td>
<td></td>
<td>German, Middle High (ca. 1050-1500)</td>
</tr>
<tr>
<td>goh</td>
<td></td>
<td>German, Old High (ca. 750-1050)</td>
</tr>
<tr>
<td>gon</td>
<td></td>
<td>Gondi</td>
</tr>
<tr>
<td>gor</td>
<td></td>
<td>Gorontalo</td>
</tr>
<tr>
<td>got</td>
<td></td>
<td>Gothic</td>
</tr>
<tr>
<td>ISO 639-2</td>
<td>ISO 639-1</td>
<td>Name of language</td>
</tr>
<tr>
<td>-----------</td>
<td>-----------</td>
<td>------------------</td>
</tr>
<tr>
<td>grb</td>
<td></td>
<td>Grebo</td>
</tr>
<tr>
<td>grc</td>
<td></td>
<td>Greek, Ancient (to 1453)</td>
</tr>
<tr>
<td>gre</td>
<td>el</td>
<td>Greek, Modern (post 1453) (ISO 639-2/B)</td>
</tr>
<tr>
<td>grn</td>
<td>gn</td>
<td>Guaraní</td>
</tr>
<tr>
<td>gsw</td>
<td>gu</td>
<td>Swiss German; Alemannic; Alsatian</td>
</tr>
<tr>
<td>guj</td>
<td>gu</td>
<td>Gujarati</td>
</tr>
<tr>
<td>gwi</td>
<td></td>
<td>Gwich’in</td>
</tr>
<tr>
<td>hai</td>
<td></td>
<td>Haida</td>
</tr>
<tr>
<td>hat</td>
<td>ht</td>
<td>Haitian; Haitian Creole</td>
</tr>
<tr>
<td>hau</td>
<td>ha</td>
<td>Hausa</td>
</tr>
<tr>
<td>haw</td>
<td></td>
<td>Hawaiian</td>
</tr>
<tr>
<td>heb</td>
<td>he</td>
<td>Hebrew</td>
</tr>
<tr>
<td>her</td>
<td>hz</td>
<td>Herero</td>
</tr>
<tr>
<td>hil</td>
<td></td>
<td>Hiligaynon</td>
</tr>
<tr>
<td>him</td>
<td>hi</td>
<td>Himachali languages; Western Pahari languages</td>
</tr>
<tr>
<td>hin</td>
<td>hi</td>
<td>Hindi</td>
</tr>
<tr>
<td>hit</td>
<td></td>
<td>Hittite</td>
</tr>
<tr>
<td>hmn</td>
<td></td>
<td>Hmong</td>
</tr>
<tr>
<td>hmo</td>
<td>ho</td>
<td>Hiri Motu</td>
</tr>
<tr>
<td>hrv</td>
<td>hr</td>
<td>Croatian (ISO 639-2/T)</td>
</tr>
<tr>
<td>hsb</td>
<td></td>
<td>Upper Sorbian</td>
</tr>
<tr>
<td>hun</td>
<td>hu</td>
<td>Hungarian</td>
</tr>
<tr>
<td>hup</td>
<td></td>
<td>Hupa</td>
</tr>
<tr>
<td>hye</td>
<td>hy</td>
<td>Armenian (ISO 639-2/T)</td>
</tr>
<tr>
<td>iba</td>
<td></td>
<td>Iban</td>
</tr>
<tr>
<td>ibo</td>
<td>ig</td>
<td>Igbo</td>
</tr>
<tr>
<td>ice</td>
<td>is</td>
<td>Icelandic (ISO 639-2/B)</td>
</tr>
<tr>
<td>ido</td>
<td>io</td>
<td>Ido</td>
</tr>
<tr>
<td>iii</td>
<td>ii</td>
<td>Sichuan Yi; Nuosu</td>
</tr>
<tr>
<td>ijo</td>
<td></td>
<td>Ijo</td>
</tr>
<tr>
<td>iku</td>
<td>iu</td>
<td>Inuktitut</td>
</tr>
<tr>
<td>ile</td>
<td>ie</td>
<td>Interlingue; Occidental</td>
</tr>
<tr>
<td>ilo</td>
<td></td>
<td>Ilokó</td>
</tr>
<tr>
<td>ina</td>
<td>ia</td>
<td>Interlingua (International Auxiliary Language Association)</td>
</tr>
<tr>
<td>inc</td>
<td></td>
<td>Indic (Other)</td>
</tr>
<tr>
<td>ind</td>
<td>id</td>
<td>Indonesian</td>
</tr>
<tr>
<td>ine</td>
<td></td>
<td>Indo-European (Other)</td>
</tr>
<tr>
<td>inh</td>
<td></td>
<td>Ingush</td>
</tr>
<tr>
<td>ipk</td>
<td>ik</td>
<td>Inupiaq</td>
</tr>
<tr>
<td>ISO 639-2</td>
<td>ISO 639-1</td>
<td>Name of language</td>
</tr>
<tr>
<td>---------</td>
<td>---------</td>
<td>---------------------------------</td>
</tr>
<tr>
<td>ira</td>
<td></td>
<td>Iranian (Other)</td>
</tr>
<tr>
<td>iro</td>
<td></td>
<td>Iroquoian languages</td>
</tr>
<tr>
<td>isl</td>
<td>is</td>
<td>Icelandic (ISO 639-2/T)</td>
</tr>
<tr>
<td>ita</td>
<td>it</td>
<td>Italian</td>
</tr>
<tr>
<td>jav</td>
<td>jv</td>
<td>Javanese (ISO 639-2/B)</td>
</tr>
<tr>
<td>jbo</td>
<td></td>
<td>Lojban</td>
</tr>
<tr>
<td>jaw</td>
<td></td>
<td>Javanese (ISO 639-2/T)</td>
</tr>
<tr>
<td>jpn</td>
<td>ja</td>
<td>Japanese</td>
</tr>
<tr>
<td>jpr</td>
<td></td>
<td>Judeo-Persian</td>
</tr>
<tr>
<td>jrb</td>
<td></td>
<td>Judeo-Arabic</td>
</tr>
<tr>
<td>kaa</td>
<td></td>
<td>Kara-Kalpak</td>
</tr>
<tr>
<td>kab</td>
<td></td>
<td>Kabyle</td>
</tr>
<tr>
<td>kac</td>
<td></td>
<td>Kachin; Jingpho</td>
</tr>
<tr>
<td>kal</td>
<td>kl</td>
<td>Kalaallisut; Greenlandic</td>
</tr>
<tr>
<td>kam</td>
<td></td>
<td>Kamba</td>
</tr>
<tr>
<td>kan</td>
<td>kn</td>
<td>Kannada</td>
</tr>
<tr>
<td>kar</td>
<td></td>
<td>Karen</td>
</tr>
<tr>
<td>kas</td>
<td>ks</td>
<td>Kashmiri</td>
</tr>
<tr>
<td>kat</td>
<td>ka</td>
<td>Georgian (ISO 639-2/T)</td>
</tr>
<tr>
<td>kau</td>
<td>kr</td>
<td>Kanuri</td>
</tr>
<tr>
<td>kaw</td>
<td></td>
<td>Kawi</td>
</tr>
<tr>
<td>kaz</td>
<td>kk</td>
<td>Kazakh</td>
</tr>
<tr>
<td>kbd</td>
<td></td>
<td>Kabardian</td>
</tr>
<tr>
<td>kha</td>
<td></td>
<td>Khasi</td>
</tr>
<tr>
<td>khi</td>
<td></td>
<td>Khoisan (Other)</td>
</tr>
<tr>
<td>khm</td>
<td>km</td>
<td>Khmer</td>
</tr>
<tr>
<td>kho</td>
<td></td>
<td>Khotanese; Sakan</td>
</tr>
<tr>
<td>kik</td>
<td>ki</td>
<td>Kikuyu; Gikuyu</td>
</tr>
<tr>
<td>kin</td>
<td>rw</td>
<td>Kinyarwanda</td>
</tr>
<tr>
<td>kir</td>
<td>ky</td>
<td>Kirghiz; Kyrgyz</td>
</tr>
<tr>
<td>kmb</td>
<td></td>
<td>Kimbundu</td>
</tr>
<tr>
<td>kok</td>
<td></td>
<td>Konkani</td>
</tr>
<tr>
<td>kom</td>
<td>kv</td>
<td>Komi</td>
</tr>
<tr>
<td>kon</td>
<td>kg</td>
<td>Kongo</td>
</tr>
<tr>
<td>kor</td>
<td>ko</td>
<td>Korean</td>
</tr>
<tr>
<td>kos</td>
<td></td>
<td>Kosraean</td>
</tr>
<tr>
<td>kpe</td>
<td></td>
<td>Kpelle</td>
</tr>
<tr>
<td>krc</td>
<td></td>
<td>Karachay-Balkar</td>
</tr>
<tr>
<td>kri</td>
<td></td>
<td>Karelian</td>
</tr>
<tr>
<td>ISO 639-2</td>
<td>ISO 639-1</td>
<td>Name of language</td>
</tr>
<tr>
<td>-----------</td>
<td>-----------</td>
<td>-------------------</td>
</tr>
<tr>
<td>kro</td>
<td>Kru</td>
<td></td>
</tr>
<tr>
<td>kru</td>
<td>Kurukh</td>
<td></td>
</tr>
<tr>
<td>kua</td>
<td>kj</td>
<td>Kuanyama; Kwanyama</td>
</tr>
<tr>
<td>kum</td>
<td>Kumyk</td>
<td></td>
</tr>
<tr>
<td>kur</td>
<td>ku</td>
<td>Kurdish</td>
</tr>
<tr>
<td>kut</td>
<td>Kutenai</td>
<td></td>
</tr>
<tr>
<td>lad</td>
<td>Ladino</td>
<td></td>
</tr>
<tr>
<td>lah</td>
<td>Lahnda</td>
<td></td>
</tr>
<tr>
<td>lam</td>
<td>Lamba</td>
<td></td>
</tr>
<tr>
<td>lao</td>
<td>lo</td>
<td>Lao</td>
</tr>
<tr>
<td>lat</td>
<td>la</td>
<td>Latin</td>
</tr>
<tr>
<td>lav</td>
<td>lv</td>
<td>Latvian</td>
</tr>
<tr>
<td>lez</td>
<td>Lezghian</td>
<td></td>
</tr>
<tr>
<td>lim</td>
<td>li</td>
<td>Limburgan; Limburger; Limburgish</td>
</tr>
<tr>
<td>lin</td>
<td>ln</td>
<td>Lingala</td>
</tr>
<tr>
<td>lit</td>
<td>lt</td>
<td>Lithuanian</td>
</tr>
<tr>
<td>lol</td>
<td>Mongo</td>
<td></td>
</tr>
<tr>
<td>loz</td>
<td>Lozi</td>
<td></td>
</tr>
<tr>
<td>litz</td>
<td>lb</td>
<td>Luxembourgish; Letzeburgesch</td>
</tr>
<tr>
<td>lua</td>
<td></td>
<td>Luba-Lulu</td>
</tr>
<tr>
<td>lub</td>
<td>lu</td>
<td>Luba-Katanga</td>
</tr>
<tr>
<td>lug</td>
<td>lg</td>
<td>Ganda</td>
</tr>
<tr>
<td>lui</td>
<td>Luiseno</td>
<td></td>
</tr>
<tr>
<td>lun</td>
<td>Lunda</td>
<td></td>
</tr>
<tr>
<td>luo</td>
<td></td>
<td>Luo (Kenya and Tanzania)</td>
</tr>
<tr>
<td>lus</td>
<td>Lushai</td>
<td></td>
</tr>
<tr>
<td>mac</td>
<td>mk</td>
<td>Macedonian (ISO 639-2/B)</td>
</tr>
<tr>
<td>mad</td>
<td></td>
<td>Madurese</td>
</tr>
<tr>
<td>mag</td>
<td>Magahi</td>
<td></td>
</tr>
<tr>
<td>mah</td>
<td>mh</td>
<td>Marshall</td>
</tr>
<tr>
<td>mai</td>
<td>Maltihili</td>
<td></td>
</tr>
<tr>
<td>mak</td>
<td>Makasari</td>
<td></td>
</tr>
<tr>
<td>mal</td>
<td>ml</td>
<td>Malayalam</td>
</tr>
<tr>
<td>man</td>
<td></td>
<td>Mandingo</td>
</tr>
<tr>
<td>mao</td>
<td>mi</td>
<td>Maori</td>
</tr>
<tr>
<td>map</td>
<td></td>
<td>Austronesian (Other)</td>
</tr>
<tr>
<td>mar</td>
<td>mr</td>
<td>Marathi</td>
</tr>
<tr>
<td>mas</td>
<td></td>
<td>Masai</td>
</tr>
<tr>
<td>may</td>
<td>ms</td>
<td>Malay (ISO 639-2/B)</td>
</tr>
<tr>
<td>ISO 639-2</td>
<td>ISO 639-1</td>
<td>Name of language</td>
</tr>
<tr>
<td>-----------</td>
<td>-----------</td>
<td>------------------</td>
</tr>
<tr>
<td>mdf</td>
<td></td>
<td>Moksha</td>
</tr>
<tr>
<td>mdr</td>
<td></td>
<td>Mandar</td>
</tr>
<tr>
<td>men</td>
<td></td>
<td>Mende</td>
</tr>
<tr>
<td>mga</td>
<td></td>
<td>Irish, Middle (900-1200)</td>
</tr>
<tr>
<td>mic</td>
<td></td>
<td>Mi'kmaq; Micmac</td>
</tr>
<tr>
<td>min</td>
<td></td>
<td>Minangkabau</td>
</tr>
<tr>
<td>mis</td>
<td></td>
<td>Miscellaneous languages</td>
</tr>
<tr>
<td>mkd</td>
<td>mk</td>
<td>Macedonian (ISO 639-2/T)</td>
</tr>
<tr>
<td>mkh</td>
<td></td>
<td>Mon-Khmer (Other)</td>
</tr>
<tr>
<td>mlg</td>
<td>mg</td>
<td>Malagasy</td>
</tr>
<tr>
<td>mlt</td>
<td>mt</td>
<td>Maltese</td>
</tr>
<tr>
<td>mnc</td>
<td></td>
<td>Manchu</td>
</tr>
<tr>
<td>mni</td>
<td></td>
<td>Manipuri</td>
</tr>
<tr>
<td>mno</td>
<td></td>
<td>Manobo languages</td>
</tr>
<tr>
<td>moh</td>
<td></td>
<td>Mohawk</td>
</tr>
<tr>
<td>mol</td>
<td></td>
<td>Moldavian</td>
</tr>
<tr>
<td>mon</td>
<td>mn</td>
<td>Mongolian</td>
</tr>
<tr>
<td>mos</td>
<td></td>
<td>Mossi</td>
</tr>
<tr>
<td>mri</td>
<td>mi</td>
<td>Maori (ISO 639-2/T)</td>
</tr>
<tr>
<td>msa</td>
<td>ms</td>
<td>Malay (ISO 639-2/T)</td>
</tr>
<tr>
<td>mul</td>
<td></td>
<td>Multiple languages</td>
</tr>
<tr>
<td>mun</td>
<td></td>
<td>Munda languages</td>
</tr>
<tr>
<td>mus</td>
<td></td>
<td>Creek</td>
</tr>
<tr>
<td>mwl</td>
<td></td>
<td>Mirandese</td>
</tr>
<tr>
<td>mwr</td>
<td></td>
<td>Marwari</td>
</tr>
<tr>
<td>mya</td>
<td>my</td>
<td>Burmese (ISO 639-2/T)</td>
</tr>
<tr>
<td>myn</td>
<td></td>
<td>Mayan languages</td>
</tr>
<tr>
<td>myv</td>
<td></td>
<td>Erzya</td>
</tr>
<tr>
<td>nah</td>
<td></td>
<td>Nahuatl</td>
</tr>
<tr>
<td>nai</td>
<td></td>
<td>North American Indian</td>
</tr>
<tr>
<td>nap</td>
<td></td>
<td>Neapolitan</td>
</tr>
<tr>
<td>nan</td>
<td>zh-tw</td>
<td>Min Nan Chinese (from ISO 639-3)</td>
</tr>
<tr>
<td>nau</td>
<td>na</td>
<td>Nauru</td>
</tr>
<tr>
<td>nav</td>
<td>nv</td>
<td>Navajo; Navaho</td>
</tr>
<tr>
<td>nbl</td>
<td>nr</td>
<td>Ndebele, South; South Ndebele</td>
</tr>
<tr>
<td>nde</td>
<td>nd</td>
<td>Ndebele, North; North Ndebele</td>
</tr>
<tr>
<td>ndo</td>
<td>ng</td>
<td>Ndonga</td>
</tr>
<tr>
<td>nds</td>
<td></td>
<td>Low German; Low Saxon; German, Low; Saxon, Low</td>
</tr>
<tr>
<td>nep</td>
<td>ne</td>
<td>Nepali</td>
</tr>
<tr>
<td>ISO 639-2</td>
<td>ISO 639-1</td>
<td>Name of language</td>
</tr>
<tr>
<td>----------</td>
<td>----------</td>
<td>------------------</td>
</tr>
<tr>
<td>new</td>
<td></td>
<td>Nepal Bhasa; Newari</td>
</tr>
<tr>
<td>nia</td>
<td></td>
<td>Nias</td>
</tr>
<tr>
<td>nic</td>
<td></td>
<td>Niger-Kordofanian (Other)</td>
</tr>
<tr>
<td>niu</td>
<td></td>
<td>Niuean</td>
</tr>
<tr>
<td>nld</td>
<td>nl</td>
<td>Dutch; Flemish (ISO 639-2/T)</td>
</tr>
<tr>
<td>nno</td>
<td>nn</td>
<td>Norwegian Nynorsk; Nynorsk, Norwegian</td>
</tr>
<tr>
<td>nob</td>
<td>nb</td>
<td>Bokmal, Norwegian; Norwegian Bokmal</td>
</tr>
<tr>
<td>nog</td>
<td></td>
<td>Nogai</td>
</tr>
<tr>
<td>non</td>
<td></td>
<td>Norse, Old</td>
</tr>
<tr>
<td>nor</td>
<td>no</td>
<td>Norwegian</td>
</tr>
<tr>
<td>nqo</td>
<td></td>
<td>N'Ko</td>
</tr>
<tr>
<td>nso</td>
<td></td>
<td>Pedi; Sepedi; Northern Sotho</td>
</tr>
<tr>
<td>nub</td>
<td></td>
<td>Nubian languages</td>
</tr>
<tr>
<td>nwc</td>
<td></td>
<td>Classical Newari; Old Newari; Classical Nepal Bhasa</td>
</tr>
<tr>
<td>nya</td>
<td>ny</td>
<td>Chichewa; Chewa; Nyanja</td>
</tr>
<tr>
<td>nym</td>
<td></td>
<td>Nyamwezi</td>
</tr>
<tr>
<td>nyn</td>
<td></td>
<td>Nyankole</td>
</tr>
<tr>
<td>nyo</td>
<td></td>
<td>Nyoro</td>
</tr>
<tr>
<td>nzi</td>
<td></td>
<td>Nzima</td>
</tr>
<tr>
<td>oci</td>
<td>oc</td>
<td>Occitan (post 1500)</td>
</tr>
<tr>
<td>oji</td>
<td>oj</td>
<td>Ojibwa</td>
</tr>
<tr>
<td>ori</td>
<td>or</td>
<td>Oriya</td>
</tr>
<tr>
<td>orm</td>
<td>om</td>
<td>Oromo</td>
</tr>
<tr>
<td>osa</td>
<td></td>
<td>Osage</td>
</tr>
<tr>
<td>oss</td>
<td>os</td>
<td>Ossetian; Ossetic</td>
</tr>
<tr>
<td>ota</td>
<td></td>
<td>Turkish, Ottoman (1500-1928)</td>
</tr>
<tr>
<td>oto</td>
<td></td>
<td>Otomian languages</td>
</tr>
<tr>
<td>paa</td>
<td></td>
<td>Papuan (Other)</td>
</tr>
<tr>
<td>pag</td>
<td></td>
<td>Pangasinan</td>
</tr>
<tr>
<td>pal</td>
<td></td>
<td>Pahlavi</td>
</tr>
<tr>
<td>pam</td>
<td></td>
<td>Pampanga; Kapampangan</td>
</tr>
<tr>
<td>pan</td>
<td>pa</td>
<td>Panjabi; Punjabi</td>
</tr>
<tr>
<td>pap</td>
<td></td>
<td>Papiamento</td>
</tr>
<tr>
<td>pau</td>
<td></td>
<td>Palauan</td>
</tr>
<tr>
<td>peo</td>
<td></td>
<td>Persian, Old (ca. 600-400 B.C.)</td>
</tr>
<tr>
<td>per</td>
<td>fa</td>
<td>Persian (ISO 639-2/B)</td>
</tr>
<tr>
<td>phi</td>
<td></td>
<td>Philippine (Other)</td>
</tr>
<tr>
<td>phn</td>
<td></td>
<td>Phoenician</td>
</tr>
<tr>
<td>pli</td>
<td>pi</td>
<td>Pali</td>
</tr>
<tr>
<td>ISO 639-2</td>
<td>ISO 639-1</td>
<td>Name of language</td>
</tr>
<tr>
<td>-----------</td>
<td>-----------</td>
<td>------------------</td>
</tr>
<tr>
<td>pol</td>
<td>pl</td>
<td>Polish</td>
</tr>
<tr>
<td>pon</td>
<td></td>
<td>Pohnpeian</td>
</tr>
<tr>
<td>por</td>
<td>pt, pt-br</td>
<td>Portuguese, ‘pt-br’ is for Brazil</td>
</tr>
<tr>
<td>pra</td>
<td></td>
<td>Prakrit languages</td>
</tr>
<tr>
<td>pro</td>
<td></td>
<td>Provençal, Old (to 1500); Occitan, Old (to 1500)</td>
</tr>
<tr>
<td>pus</td>
<td>ps</td>
<td>Pushto; Pashto</td>
</tr>
<tr>
<td>qaa</td>
<td></td>
<td>original audio (not from ISO 639, from ‘EN 300 468 Annex F’)</td>
</tr>
<tr>
<td>qad</td>
<td></td>
<td>Audio Description (France HD Forum)</td>
</tr>
<tr>
<td>que</td>
<td>qu</td>
<td>Quechua</td>
</tr>
<tr>
<td>raj</td>
<td></td>
<td>Rajasthani</td>
</tr>
<tr>
<td>rap</td>
<td></td>
<td>Rapanui</td>
</tr>
<tr>
<td>rar</td>
<td></td>
<td>Rarotongan; Cook Islands Maori</td>
</tr>
<tr>
<td>roa</td>
<td></td>
<td>Romance (Other)</td>
</tr>
<tr>
<td>roh</td>
<td>rm</td>
<td>Raeto-Romance</td>
</tr>
<tr>
<td>rom</td>
<td></td>
<td>Romany</td>
</tr>
<tr>
<td>ron</td>
<td>ro</td>
<td>Romanian; Moldavian; Moldovan (ISO 639-2/T)</td>
</tr>
<tr>
<td>rum</td>
<td>ro</td>
<td>Romanian; Moldavian; Moldovan (ISO 639-2/B)</td>
</tr>
<tr>
<td>run</td>
<td>rm</td>
<td>Rundi</td>
</tr>
<tr>
<td>rup</td>
<td></td>
<td>Aromanian; Arumanian; Macedo-Romanian</td>
</tr>
<tr>
<td>rus</td>
<td>ru</td>
<td>Russian</td>
</tr>
<tr>
<td>sad</td>
<td></td>
<td>Sandawe</td>
</tr>
<tr>
<td>sag</td>
<td>sg</td>
<td>Sango</td>
</tr>
<tr>
<td>sah</td>
<td></td>
<td>Yakut</td>
</tr>
<tr>
<td>sai</td>
<td></td>
<td>South American Indian (Other)</td>
</tr>
<tr>
<td>sal</td>
<td></td>
<td>Salishan languages</td>
</tr>
<tr>
<td>sam</td>
<td></td>
<td>Samaritan Aramaic</td>
</tr>
<tr>
<td>san</td>
<td>sa</td>
<td>Sanskrit</td>
</tr>
<tr>
<td>sas</td>
<td></td>
<td>Sasak</td>
</tr>
<tr>
<td>sat</td>
<td></td>
<td>Santali</td>
</tr>
<tr>
<td>scn</td>
<td></td>
<td>Sicilian</td>
</tr>
<tr>
<td>scc</td>
<td></td>
<td>Serbian (ISO 639-2/B)</td>
</tr>
<tr>
<td>sco</td>
<td></td>
<td>Scots</td>
</tr>
<tr>
<td>scr</td>
<td></td>
<td>Croatian (ISO 639-2/B)</td>
</tr>
<tr>
<td>sel</td>
<td></td>
<td>Selkup</td>
</tr>
<tr>
<td>sem</td>
<td></td>
<td>Semitic (Other)</td>
</tr>
<tr>
<td>sga</td>
<td></td>
<td>Irish, Old (to 900)</td>
</tr>
<tr>
<td>shn</td>
<td></td>
<td>Shan</td>
</tr>
<tr>
<td>sid</td>
<td></td>
<td>Sidamo</td>
</tr>
<tr>
<td>sin</td>
<td>si</td>
<td>Sinhala; Sinhalese</td>
</tr>
<tr>
<td>ISO 639-2</td>
<td>ISO 639-1</td>
<td>Name of language</td>
</tr>
<tr>
<td>-----------</td>
<td>-----------</td>
<td>-------------------------------</td>
</tr>
<tr>
<td>sio</td>
<td></td>
<td>Siouan languages</td>
</tr>
<tr>
<td>sit</td>
<td></td>
<td>Sino-Tibetan (Other)</td>
</tr>
<tr>
<td>sla</td>
<td></td>
<td>Slavic (Other)</td>
</tr>
<tr>
<td>slk</td>
<td>sk</td>
<td>Slovak (ISO 639-2/T)</td>
</tr>
<tr>
<td>slo</td>
<td>sk</td>
<td>Slovak (ISO 639-2/B)</td>
</tr>
<tr>
<td>slv</td>
<td>sl</td>
<td>Slovenian</td>
</tr>
<tr>
<td>sma</td>
<td></td>
<td>Southern Sami</td>
</tr>
<tr>
<td>sme</td>
<td>se</td>
<td>Northern Sami</td>
</tr>
<tr>
<td>smi</td>
<td></td>
<td>Sami languages</td>
</tr>
<tr>
<td>smj</td>
<td></td>
<td>Lule Sami</td>
</tr>
<tr>
<td>smn</td>
<td></td>
<td>Inari Sami</td>
</tr>
<tr>
<td>smo</td>
<td>sm</td>
<td>Samoan</td>
</tr>
<tr>
<td>sms</td>
<td></td>
<td>Skolt Sami</td>
</tr>
<tr>
<td>sna</td>
<td>sn</td>
<td>Shona</td>
</tr>
<tr>
<td>snd</td>
<td>sd</td>
<td>Sindhi</td>
</tr>
<tr>
<td>snk</td>
<td></td>
<td>Soninke</td>
</tr>
<tr>
<td>sog</td>
<td></td>
<td>Sogdian</td>
</tr>
<tr>
<td>som</td>
<td>so</td>
<td>Somali</td>
</tr>
<tr>
<td>son</td>
<td></td>
<td>Songhai</td>
</tr>
<tr>
<td>sot</td>
<td>st</td>
<td>Sotho, Southern</td>
</tr>
<tr>
<td>spa</td>
<td>es</td>
<td>Spanish; Castilian (ISO 639-2/B)</td>
</tr>
<tr>
<td>sqi</td>
<td>sq</td>
<td>Albanian (ISO 639-2/T)</td>
</tr>
<tr>
<td>srd</td>
<td>sc</td>
<td>Sardinian</td>
</tr>
<tr>
<td>srm</td>
<td></td>
<td>Sranan Tongo</td>
</tr>
<tr>
<td>srp</td>
<td>sr</td>
<td>Serbian (ISO 639-2/T)</td>
</tr>
<tr>
<td>srr</td>
<td></td>
<td>Serer</td>
</tr>
<tr>
<td>ssa</td>
<td></td>
<td>Nilo-Saharan (Other)</td>
</tr>
<tr>
<td>ssu</td>
<td>ss</td>
<td>Swati</td>
</tr>
<tr>
<td>suk</td>
<td></td>
<td>Sukuma</td>
</tr>
<tr>
<td>sun</td>
<td>su</td>
<td>Sundanese</td>
</tr>
<tr>
<td>sus</td>
<td></td>
<td>Susu</td>
</tr>
<tr>
<td>sux</td>
<td></td>
<td>Sumerian</td>
</tr>
<tr>
<td>swa</td>
<td>sw</td>
<td>Swahili</td>
</tr>
<tr>
<td>swe</td>
<td>sv</td>
<td>Swedish</td>
</tr>
<tr>
<td>syc</td>
<td></td>
<td>Classical Syriac</td>
</tr>
<tr>
<td>syr</td>
<td></td>
<td>Syriac</td>
</tr>
<tr>
<td>tah</td>
<td>ty</td>
<td>Tahitian</td>
</tr>
<tr>
<td>tai</td>
<td></td>
<td>Tai (Other)</td>
</tr>
<tr>
<td>tam</td>
<td>ta</td>
<td>Tamil</td>
</tr>
<tr>
<td>ISO 639-2</td>
<td>ISO 639-1</td>
<td>Name of language</td>
</tr>
<tr>
<td>-----------</td>
<td>-----------</td>
<td>---------------------------</td>
</tr>
<tr>
<td>tat</td>
<td>tt</td>
<td>Tatar</td>
</tr>
<tr>
<td>tel</td>
<td>te</td>
<td>Telugu</td>
</tr>
<tr>
<td>tem</td>
<td></td>
<td>Timne</td>
</tr>
<tr>
<td>ter</td>
<td></td>
<td>Tereno</td>
</tr>
<tr>
<td>tet</td>
<td></td>
<td>Tetum</td>
</tr>
<tr>
<td>tgk</td>
<td>tg</td>
<td>Tajik</td>
</tr>
<tr>
<td>tgl</td>
<td>tl</td>
<td>Tagalog</td>
</tr>
<tr>
<td>tha</td>
<td>th</td>
<td>Thai</td>
</tr>
<tr>
<td>tib</td>
<td>bo</td>
<td>Tibetan (ISO 639-2/B)</td>
</tr>
<tr>
<td>tig</td>
<td></td>
<td>Tigre</td>
</tr>
<tr>
<td>tir</td>
<td>ti</td>
<td>Tigrinya</td>
</tr>
<tr>
<td>tiv</td>
<td></td>
<td>Tiv</td>
</tr>
<tr>
<td>tkl</td>
<td></td>
<td>Tokelau</td>
</tr>
<tr>
<td>thh</td>
<td></td>
<td>Klingon; thlIngan-Hol</td>
</tr>
<tr>
<td>tli</td>
<td></td>
<td>Tlingit</td>
</tr>
<tr>
<td>tmh</td>
<td></td>
<td>Tamashek</td>
</tr>
<tr>
<td>tog</td>
<td></td>
<td>Tonga (Nyasa)</td>
</tr>
<tr>
<td>ton</td>
<td>to</td>
<td>Tonga (Tonga Islands)</td>
</tr>
<tr>
<td>tpi</td>
<td></td>
<td>Tok Pisin</td>
</tr>
<tr>
<td>tsi</td>
<td></td>
<td>Tsimshian</td>
</tr>
<tr>
<td>tsn</td>
<td>tn</td>
<td>Tswana</td>
</tr>
<tr>
<td>tso</td>
<td>ts</td>
<td>Tsonga</td>
</tr>
<tr>
<td>tuk</td>
<td>tk</td>
<td>Turkmen</td>
</tr>
<tr>
<td>tum</td>
<td></td>
<td>Tumbuka</td>
</tr>
<tr>
<td>tup</td>
<td></td>
<td>Tupi languages</td>
</tr>
<tr>
<td>tur</td>
<td>tr</td>
<td>Turkish</td>
</tr>
<tr>
<td>tut</td>
<td></td>
<td>Altaic (Other)</td>
</tr>
<tr>
<td>tvl</td>
<td></td>
<td>Tuvalu</td>
</tr>
<tr>
<td>twi</td>
<td>tw</td>
<td>Twi</td>
</tr>
<tr>
<td>tyv</td>
<td></td>
<td>Tuvinian</td>
</tr>
<tr>
<td>uga</td>
<td></td>
<td>Ugaritic</td>
</tr>
<tr>
<td>uig</td>
<td>ug</td>
<td>Uighur; Uyghur</td>
</tr>
<tr>
<td>ukr</td>
<td>uk</td>
<td>Ukrainian</td>
</tr>
<tr>
<td>umb</td>
<td></td>
<td>Umbundu</td>
</tr>
<tr>
<td>und</td>
<td></td>
<td>Undetermined</td>
</tr>
<tr>
<td>urd</td>
<td>ur</td>
<td>Urdu</td>
</tr>
<tr>
<td>uzb</td>
<td>uz</td>
<td>Uzbek</td>
</tr>
<tr>
<td>vai</td>
<td></td>
<td>Vai</td>
</tr>
<tr>
<td>val</td>
<td></td>
<td>Valencian (used Spain)</td>
</tr>
<tr>
<td>ISO 639-2</td>
<td>ISO 639-1</td>
<td>Name of language</td>
</tr>
<tr>
<td>-----------</td>
<td>-----------</td>
<td>------------------</td>
</tr>
<tr>
<td>ven</td>
<td>ve</td>
<td>Venda</td>
</tr>
<tr>
<td>vie</td>
<td>vi</td>
<td>Vietnamese</td>
</tr>
<tr>
<td>vol</td>
<td>vo</td>
<td>Volapuk</td>
</tr>
<tr>
<td>vot</td>
<td></td>
<td>Votic</td>
</tr>
<tr>
<td>wak</td>
<td></td>
<td>Wakashan languages</td>
</tr>
<tr>
<td>wal</td>
<td></td>
<td>Wolaitta; Wolaytta</td>
</tr>
<tr>
<td>war</td>
<td></td>
<td>Waray</td>
</tr>
<tr>
<td>was</td>
<td></td>
<td>Washo</td>
</tr>
<tr>
<td>wel</td>
<td>cy</td>
<td>Welsh (ISO 639-2/B)</td>
</tr>
<tr>
<td>wen</td>
<td></td>
<td>Sorbian languages</td>
</tr>
<tr>
<td>wln</td>
<td>wa</td>
<td>Walloon</td>
</tr>
<tr>
<td>wol</td>
<td>wo</td>
<td>Wolof</td>
</tr>
<tr>
<td>xal</td>
<td></td>
<td>Kalmyk; Oirat</td>
</tr>
<tr>
<td>xho</td>
<td>xh</td>
<td>Xhosa</td>
</tr>
<tr>
<td>yao</td>
<td></td>
<td>Yao</td>
</tr>
<tr>
<td>yap</td>
<td></td>
<td>Yapes</td>
</tr>
<tr>
<td>yid</td>
<td>yi</td>
<td>Yiddish</td>
</tr>
<tr>
<td>yor</td>
<td>yo</td>
<td>Yoruba</td>
</tr>
<tr>
<td>ypk</td>
<td></td>
<td>Yupik languages</td>
</tr>
<tr>
<td>yue</td>
<td>zh-hk</td>
<td>Yue Chinese (Cantonese) (from ISO 639-3)</td>
</tr>
<tr>
<td>zap</td>
<td></td>
<td>Zapotec</td>
</tr>
<tr>
<td>zbl</td>
<td></td>
<td>Blissymbols; Blissymbolics; Bliss</td>
</tr>
<tr>
<td>zen</td>
<td></td>
<td>Zenaga</td>
</tr>
<tr>
<td>zha</td>
<td>za</td>
<td>Zhuang; Chuang</td>
</tr>
<tr>
<td>zho</td>
<td>zh</td>
<td>Chinese (ISO 639-2/T)</td>
</tr>
<tr>
<td>znd</td>
<td></td>
<td>Zande</td>
</tr>
<tr>
<td>zul</td>
<td>zu</td>
<td>Zulu</td>
</tr>
<tr>
<td>zun</td>
<td></td>
<td>Zuni</td>
</tr>
<tr>
<td>zxx</td>
<td></td>
<td>No linguistic content; Not applicable</td>
</tr>
<tr>
<td>zza</td>
<td></td>
<td>Zaza; Dimili; Dimli; Kirdiki; Kirmanji; Zazaki</td>
</tr>
</tbody>
</table>

**Annex D Differences in Media Devices**

<table>
<thead>
<tr>
<th>Chapter of this Document</th>
<th>Constraints and difference of Media Devices compared with TV</th>
</tr>
</thead>
</table>

|                         |                                                          |
Some of LG Media products have HDMI, Composite, Component and Scart as output connection types.

- **HDMI:**
  The video content is outputted on a HDCP-protected HDMI output interface only. If the TV does not support HDCP, the video contents can only be outputted on the analog video outputs.

- **Analog Outputs:**
  Analog video outputs are protected with CGMS-A. (CopyNever)

**Video Output**

**Web Engine**

Support only in AV tag from NetCast 3.0

**userAgent String**

A. All LG_NetCastVersion:
   Change LG NetCast.TV to LG NetCast.Media for Media products.

B. [Table] Optional feature string for capabilities field:
   - Add Feature String (+NO_NUM), Optional Feature (Support for non-numeric IR Remote).

C. Applications only need to use LG Smart TV for defining TV products and LG Media for defining Media products.

**Protocols**

<table>
<thead>
<tr>
<th>Use</th>
<th>Preferred protocols</th>
<th>Supported protocols</th>
</tr>
</thead>
<tbody>
<tr>
<td>HTML file transportation</td>
<td>HTTP, HTTPS</td>
<td>HTTPS, HTTPS</td>
</tr>
<tr>
<td>VoD media file transportation</td>
<td>HTTP, HTTPS, HLS, Widevine, MMSH/MMST</td>
<td>HTTP, HTTPS, HLS, Widevine, MMSH/MMST</td>
</tr>
<tr>
<td>Linear or live broadcasting</td>
<td>HLS, Widevine</td>
<td>HLS, Widevine</td>
</tr>
</tbody>
</table>

**DRM**

Media products do not support Adobe Access 3.0 DRM.

**Subtitle**

A. Media products support below subtitle formats.

<table>
<thead>
<tr>
<th>Format</th>
<th>2011 platform/SP520</th>
<th>2012 / 2013 platform</th>
</tr>
</thead>
<tbody>
<tr>
<td>SAMI</td>
<td>O</td>
<td>O</td>
</tr>
<tr>
<td>CineCanvas</td>
<td>O</td>
<td>X</td>
</tr>
<tr>
<td>Timed Text</td>
<td>O</td>
<td>X</td>
</tr>
<tr>
<td>SRT</td>
<td>O</td>
<td>O</td>
</tr>
<tr>
<td>SUB</td>
<td>O</td>
<td>O</td>
</tr>
<tr>
<td>SSA</td>
<td>X</td>
<td>O</td>
</tr>
</tbody>
</table>

B. Reference url add
- SRT : [http://matroska.org/technical/specs/subtitles/srt.html](http://matroska.org/technical/specs/subtitles/srt.html)
- SSA : [http://matroska.org/technical/specs/subtitles/ssa.html](http://matroska.org/technical/specs/subtitles/ssa.html)

C. Detailed specification for SRT

```
Start   End     Contents
--->     |          Expression time interval to display
&nbsp;     |          Non-breaking space
```

D. Detailed specification for SSA

```
Start   End     Contents
[Script Info] | Subtitle file header information
[V4 Styles]   | Style information
[Events]     | Represents the main content of the document
&nbsp;         | Non-breaking space
```
### Constraints and difference of Media Devices compared with TV

#### E. Detailed specification for SUB

<table>
<thead>
<tr>
<th>Start</th>
<th>End</th>
<th>Contents</th>
</tr>
</thead>
<tbody>
<tr>
<td>{}</td>
<td></td>
<td>Set the start and last time for the caption</td>
</tr>
<tr>
<td>&amp;nbsp</td>
<td></td>
<td>Non-breaking space</td>
</tr>
</tbody>
</table>

#### Images

- **A. JPEG/PNG**
  - i. Less than 4,000 x 3,000 x 24 bit/pixel
  - ii. Less than 3,000 x 3,000 x 32 bit/pixel

- **B. GIF**
  - Media products do not support GIF format.

#### HTTP Streaming

- Media products do not support Progressive download but HTTP/HTTPS streaming.

#### Document Scrolling

- "PR+" (Programme Up) and "PR-" (Programme Down) buttons are not available on Media products.

#### Font and Font Style

- Media products have only "LG Display", "LG Display_HK" and "TiresiasScreenfont" font.

#### Input Device

- Traditional IR Remote, Magic Remote and HID (USB Keyboard) are available.

#### Traditional IR Remote Key Events

- There are some unavailable keys on Media products which are VK_INFO, VK_PAGE_UP and VK_PAGE_DOWN. Moreover, some Media products do not support numeric key.

#### Key Repeat Function

- There are some unavailable keys on Media products which are VK_INFO, VK_PAGE_UP and VK_PAGE_DOWN

#### Smart Text

- Media products supports this function only in NetCast 3.0.

#### Complete List of Supported MIME Types

- Media products also support below MIME
  - AVI : video/x-mxvideo

#### Complete List of Supported Codecs and Containers

- *.mp4
  - Add codec (MPEG4 P2), FourCC (MP4V) and Max Resolution (1280x720@60FPS).
  - *.asf (- extension when Media container is ASX)
    - Change common File Extension from *.asf to *.asf, *.wmv, *.wma
  - *.mov
    - Media product does not support this.
  - *.ts
    - Change Video Max Bitrate to 800~4500Kbps
  - *.mp3
    - Change Media container from ID3V1, ID3V2 to mp3.
  - *.wav
    - Change Media Container from ASF to WAV
  - WMA PRO
    - Media product don't support WMA PRO

### Annex E HTTPS Server and Client Certification

The NetCast Platform supports HTTPS server certificate and client certificate.

1. Server certificate
   - A. The NetCast platform has trusted CAs to be able to verify Secure Socket Layer (SSL) server certificates. The list of
public CAs can be downloaded from [DISCOVER > Legacy Platform (NetCast) > Technical Notes > Web List of Public CAs and LG Root CA] in this site.

B. The NetCast platform supports the hostname matching with the Subject Alternative Name (SAN) extension and wildcard certificates.

2. Client certificate
A. If the server requests the client certificate during the Secure Socket Layer (SSL) connection, the LG Browser sends client certificate. However, the streaming module of media player does not support the client certificate. The client certificate is a standard X.509 v3 certificate and has been issued by LG custom CA. The LG root CA of TV and media can be downloaded with this document.

Annex F References

The following referenced documents are required for the application of this specification. For dated or versioned references, only the edition cited applies. For nonspecific references, the latest edition of the referenced document (including any amendments) applies.

<table>
<thead>
<tr>
<th>Index</th>
<th>Reference &amp; Version</th>
<th>Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>[1]</td>
<td>ETSI TS 102 796 V1.1.1 (2010-06)</td>
<td>Hybrid Broadcast Broadband TV</td>
</tr>
<tr>
<td>[6]</td>
<td>ISO/IEC 10646</td>
<td>Universal Multiple-Octet Coded Character Set (UCS)</td>
</tr>
<tr>
<td>[8]</td>
<td>ISO 639</td>
<td>Codes for the representation of names of languages</td>
</tr>
</tbody>
</table>
# Supported HTML5/CSS3 Standard for the NetCast Platform and Emulator

This document describes HTML5 and CSS3 supported by the NetCast platform and Emulator. The table below provides a brief description of each element or API.

<table>
<thead>
<tr>
<th>Topics</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>HTML5 Video and Audio</strong></td>
<td>This topic describes how to play a video or audio file on the web without plugins. It also lists the methods, attributes and events for Video and Audio. For the detail support information on the NetCast platform and Emulator, refer to the table on each Video and Audio section.</td>
</tr>
<tr>
<td><strong>HTML5 Communication</strong></td>
<td>This topic describes how to share data between multiple programs in an asynchronous manner. It also lists the methods, attributes and events for Cross-Document Messaging, Channel Messaging, MessageEvent, Web Sockets and Server-sent Events. For the detail support information on the NetCast platform and Emulator, refer to the table on each Cross-Document Messaging, Channel Messaging, MessageEvent, Web Sockets and Server-sent Events section.</td>
</tr>
<tr>
<td><strong>HTML5 Canvas</strong></td>
<td>This topic describes how to create graphics using javascript. It also lists the methods and attributes for Canvas. For the detail support information on the NetCast platform and Emulator, refer to the table on each Canvas section.</td>
</tr>
<tr>
<td><strong>HTML5 Inline SVG, WebGL</strong></td>
<td>This topic describes how to define vector-based graphics in XML format. It also lists the elements for Inline SVG and WebGL. For the detail support information on the NetCast platform and Emulator, refer to the table on each Inline SVG and WebGL section.</td>
</tr>
<tr>
<td><strong>HTML5 Offline Web Apps</strong></td>
<td>This topic describes how to operate web applications on offline state. It also lists the methods, attributes and events for Offline Web Applications. For the detail support information on the NetCast platform and Emulator, refer to the table on each Offline Web Applications section.</td>
</tr>
<tr>
<td><strong>HTML5 Web Storage</strong></td>
<td>This topic describes how to save the data on the disk of clients. It also lists the methods and attributes for Storage, Indexed Database and WebSQL Database. For the detail support information on the NetCast platform and Emulator, refer to the table on each Storage, Indexed Database and WebSQL Database section.</td>
</tr>
<tr>
<td><strong>HTML5 User Interactions</strong></td>
<td>This topic describes Drag and Drop and History Management. For the detail support information on the NetCast platform and Emulator, refer to the table on each Drag and Drop and History Management section.</td>
</tr>
<tr>
<td><strong>HTML5 Web Workers</strong></td>
<td>This topic describes how to use the multi-task in the web browser. It also lists the methods, attributes and events for Workers and Shared workers. For the detail support information on the NetCast platform and Emulator, refer to the table on each Workers and Shared workers section.</td>
</tr>
<tr>
<td><strong>CSS3</strong></td>
<td>This topic describes which properties of CSS3 are supported by the NetCast platform and LG Emulator. It also provides information on each Dynamic CSS and Static CSS. For the detail support information on the NetCast platform and Emulator, refer to the table on each Dynamic CSS and Static CSS section.</td>
</tr>
</tbody>
</table>
Introduction

This document describes the elements and properties of HTML5 and CSS3 which are currently supported by the NetCast platform and LG Smart TV Emulator.

The NetCast platform uses the LG Browser, which is based on Webkit as an application development environment. The NetCast platform is also compatible with Safari. See the table below for details.

<table>
<thead>
<tr>
<th>Status Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Status</td>
</tr>
<tr>
<td>O</td>
</tr>
<tr>
<td>X</td>
</tr>
</tbody>
</table>

**HTML5 Standard**

This section describes the elements of HTML5 which are currently supported by the NetCast platform and LG Smart TV Emulator.

- **HTML5 Video**
- **HTML5 Audio**
- **HTML5 Communication API**
- **HTML5 Canvas**
- **HTML5 Inline SVG**
- **HTML5 Offline Web Applications API**
- **HTML5 Web Storage API**
- **HTML5 User Interaction API**
- **HTML5 Web Workers API**

**HTML5 Video**[1],[2]

The `<video>` element is used to play videos and movies on the web without plugins. This element has methods, attributes, and events.

**Note**

- In NetCast 3.0, if the video clip is changed during playback in autoplay mode, it may not play.
- In NetCast 3.0, the `<video>` element can not play the video file which is in file system (local storage or external storage such as USB memory stick). Use media plugin object to play the video file from file system.

**Video Methods**

<table>
<thead>
<tr>
<th>Method</th>
<th>Remark</th>
<th>NetCast 3.0</th>
<th>NetCast 4.0</th>
<th>Emulator 2012</th>
<th>Emulator 2013</th>
</tr>
</thead>
<tbody>
<tr>
<td>canPlayType()</td>
<td>MIME type without a codecs parameter will not return ‘probably’.</td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>O</td>
</tr>
<tr>
<td>load()</td>
<td></td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>O</td>
</tr>
<tr>
<td>play()</td>
<td></td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>O</td>
</tr>
<tr>
<td>pause()</td>
<td></td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>O</td>
</tr>
</tbody>
</table>

**Video Attributes**
<table>
<thead>
<tr>
<th>Attribute</th>
<th>Value</th>
<th>Remark</th>
<th>NetCast 3.0</th>
<th>NetCast 4.0</th>
<th>Emulator 2012</th>
<th>Emulator 2013</th>
</tr>
</thead>
<tbody>
<tr>
<td>autoplay</td>
<td></td>
<td></td>
<td>O</td>
<td>O</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>controls</td>
<td></td>
<td>Control UI needs to be implemented in the application.</td>
<td>O</td>
<td>O</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>width/height</td>
<td>480x360</td>
<td></td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>O</td>
</tr>
<tr>
<td></td>
<td>640x360</td>
<td></td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>O</td>
</tr>
<tr>
<td></td>
<td>720x406</td>
<td></td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>O</td>
</tr>
<tr>
<td></td>
<td>720x486</td>
<td></td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>O</td>
</tr>
<tr>
<td></td>
<td>1280x720</td>
<td></td>
<td>O</td>
<td>O</td>
<td>X</td>
<td>O</td>
</tr>
<tr>
<td></td>
<td>1920x1080</td>
<td></td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>O</td>
</tr>
<tr>
<td>fullscreen</td>
<td></td>
<td></td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>O</td>
</tr>
<tr>
<td>loop</td>
<td></td>
<td>no repeat</td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>O</td>
</tr>
<tr>
<td></td>
<td></td>
<td>repeat once</td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>O</td>
</tr>
<tr>
<td></td>
<td></td>
<td>repeat ten times</td>
<td>O</td>
<td>O</td>
<td>X</td>
<td>O</td>
</tr>
<tr>
<td></td>
<td></td>
<td>repeat infinitely</td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>O</td>
</tr>
<tr>
<td>preload</td>
<td>auto</td>
<td></td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>O</td>
</tr>
<tr>
<td></td>
<td>metadata</td>
<td></td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>O</td>
</tr>
<tr>
<td></td>
<td>none</td>
<td></td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>O</td>
</tr>
<tr>
<td>src</td>
<td>http</td>
<td></td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>O</td>
</tr>
<tr>
<td></td>
<td>file</td>
<td></td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td></td>
<td>ftp</td>
<td></td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td></td>
<td>href</td>
<td></td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>O</td>
</tr>
<tr>
<td>media</td>
<td></td>
<td></td>
<td>X</td>
<td>O</td>
<td>O</td>
<td>O</td>
</tr>
<tr>
<td>type</td>
<td>H.264</td>
<td>e.g.) MIME Type: video/mp4</td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>O</td>
</tr>
<tr>
<td></td>
<td>MPEG-4</td>
<td>e.g.) MIME Type: video/mp4</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td></td>
<td>Ogg</td>
<td>e.g.) MIME Type: video/ogg</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>extend type</td>
<td>avi</td>
<td></td>
<td>X</td>
<td>O</td>
<td>X</td>
<td>O</td>
</tr>
<tr>
<td></td>
<td>asf</td>
<td></td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>O</td>
</tr>
<tr>
<td></td>
<td>wmv</td>
<td></td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td></td>
<td>mov</td>
<td></td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>O</td>
</tr>
<tr>
<td></td>
<td>mp4</td>
<td></td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>O</td>
</tr>
<tr>
<td></td>
<td>ts</td>
<td></td>
<td>X</td>
<td>O</td>
<td>O</td>
<td>O</td>
</tr>
<tr>
<td></td>
<td>swf</td>
<td></td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td></td>
<td>flv</td>
<td></td>
<td>X</td>
<td>X</td>
<td>O</td>
<td>O</td>
</tr>
<tr>
<td>poster</td>
<td></td>
<td></td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>O</td>
</tr>
<tr>
<td>error</td>
<td></td>
<td></td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>O</td>
</tr>
<tr>
<td>error.code</td>
<td></td>
<td></td>
<td>O</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>currentSrc</td>
<td></td>
<td></td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>O</td>
</tr>
<tr>
<td>networkState</td>
<td></td>
<td></td>
<td>O</td>
<td>O</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>preload</td>
<td></td>
<td></td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>O</td>
</tr>
<tr>
<td>buffered</td>
<td></td>
<td></td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>O</td>
</tr>
<tr>
<td>Attribute</td>
<td>Value</td>
<td>Remark</td>
<td>NetCast 3.0</td>
<td>NetCast 4.0</td>
<td>Emulator 2012</td>
<td>Emulator 2013</td>
</tr>
<tr>
<td>---------------</td>
<td>-------</td>
<td>--------</td>
<td>-------------</td>
<td>-------------</td>
<td>---------------</td>
<td>---------------</td>
</tr>
<tr>
<td>duration</td>
<td>O</td>
<td></td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>O</td>
</tr>
<tr>
<td>currentTime</td>
<td>O</td>
<td></td>
<td>O</td>
<td>X</td>
<td>O</td>
<td>O</td>
</tr>
<tr>
<td>initialTime</td>
<td>O</td>
<td></td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>O</td>
</tr>
<tr>
<td>readyState</td>
<td>O</td>
<td></td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>O</td>
</tr>
<tr>
<td>paused</td>
<td>O</td>
<td></td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>O</td>
</tr>
<tr>
<td>ended</td>
<td>O</td>
<td></td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>O</td>
</tr>
<tr>
<td>defaultPlaybackRate</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>playbackRate</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>played</td>
<td>O</td>
<td></td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>O</td>
</tr>
<tr>
<td>seeking</td>
<td>O</td>
<td></td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>O</td>
</tr>
<tr>
<td>seekable</td>
<td>O</td>
<td></td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>O</td>
</tr>
<tr>
<td>volume</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>muted</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Video Events**

<table>
<thead>
<tr>
<th>Event</th>
<th>Remark</th>
<th>NetCast 3.0</th>
<th>NetCast 4.0</th>
<th>Emulator 2012</th>
<th>Emulator 2013</th>
</tr>
</thead>
<tbody>
<tr>
<td>loadstart</td>
<td>X</td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>O</td>
</tr>
<tr>
<td>progress</td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>O</td>
</tr>
<tr>
<td>suspend</td>
<td>X</td>
<td>O</td>
<td>X</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>abort</td>
<td>X</td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>O</td>
</tr>
<tr>
<td>error</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>emptied</td>
<td>X</td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>O</td>
</tr>
<tr>
<td>stalled</td>
<td>X</td>
<td>O</td>
<td>X</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>play</td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>O</td>
</tr>
<tr>
<td>pause</td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>O</td>
</tr>
<tr>
<td>loadedmetadata</td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>O</td>
</tr>
<tr>
<td>loadeddata</td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>O</td>
</tr>
<tr>
<td>waiting</td>
<td>X</td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>O</td>
</tr>
<tr>
<td>playing</td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>O</td>
</tr>
<tr>
<td>canplay</td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>O</td>
</tr>
<tr>
<td>canplaythrough</td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>O</td>
</tr>
<tr>
<td>seeking</td>
<td>X</td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>O</td>
</tr>
<tr>
<td>sought</td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>O</td>
</tr>
<tr>
<td>timeupdate</td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>O</td>
</tr>
<tr>
<td>ended</td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>O</td>
</tr>
<tr>
<td>ratechange</td>
<td>O</td>
<td>X</td>
<td>O</td>
<td>O</td>
<td>X</td>
</tr>
<tr>
<td>durationchange</td>
<td>O</td>
<td>O</td>
<td>X</td>
<td>O</td>
<td>O</td>
</tr>
<tr>
<td>volumechange</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
</tbody>
</table>
HTML5 Audio

The `<audio>` element is used to play audio on the web without plugins. This element has methods, attributes, and events.

Note

When playing music on overseas websites with a slow network, there may be no reaction on web pages or any linked pages. For NetCast 4.0, music can be played but there is no reaction on linked pages.

---

**Audio Methods**

<table>
<thead>
<tr>
<th>Method</th>
<th>Remark</th>
<th>NetCast 3.0</th>
<th>NetCast 4.0</th>
<th>Emulator 2012</th>
<th>Emulator 2013</th>
</tr>
</thead>
<tbody>
<tr>
<td>canPlayType()</td>
<td>MIME type without a codecs parameter will not return 'probably'.</td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>O</td>
</tr>
<tr>
<td>load()</td>
<td></td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>O</td>
</tr>
<tr>
<td>play()</td>
<td></td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>O</td>
</tr>
<tr>
<td>pause()</td>
<td></td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>O</td>
</tr>
</tbody>
</table>

**Audio Attributes**

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Value</th>
<th>Remark</th>
<th>NetCast 3.0</th>
<th>NetCast 4.0</th>
<th>Emulator 2012</th>
<th>Emulator 2013</th>
</tr>
</thead>
<tbody>
<tr>
<td>autoplay</td>
<td></td>
<td></td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>O</td>
</tr>
<tr>
<td>controls</td>
<td></td>
<td>Control UI needs to be implemented in the application.</td>
<td>O</td>
<td>O</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>loop</td>
<td></td>
<td>no repeat</td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>O</td>
</tr>
<tr>
<td></td>
<td></td>
<td>repeat once</td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>O</td>
</tr>
<tr>
<td></td>
<td></td>
<td>repeat ten times</td>
<td>O</td>
<td>X</td>
<td>X</td>
<td>O</td>
</tr>
<tr>
<td></td>
<td></td>
<td>repeat infinitely</td>
<td>O</td>
<td>X</td>
<td>O</td>
<td>O</td>
</tr>
<tr>
<td>preload</td>
<td>auto</td>
<td></td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>O</td>
</tr>
<tr>
<td></td>
<td>metadata</td>
<td></td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>O</td>
</tr>
<tr>
<td></td>
<td>none</td>
<td></td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>O</td>
</tr>
<tr>
<td>src</td>
<td>http</td>
<td></td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>O</td>
</tr>
<tr>
<td></td>
<td>file</td>
<td></td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td></td>
<td>ftp</td>
<td></td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td></td>
<td>href</td>
<td></td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>O</td>
</tr>
<tr>
<td>media</td>
<td></td>
<td></td>
<td>X</td>
<td>O</td>
<td>X</td>
<td>O</td>
</tr>
<tr>
<td>type</td>
<td>PCM</td>
<td></td>
<td>O</td>
<td>O</td>
<td>X</td>
<td>O</td>
</tr>
<tr>
<td></td>
<td>MP3</td>
<td></td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>O</td>
</tr>
<tr>
<td></td>
<td>AAC</td>
<td></td>
<td>X</td>
<td>X</td>
<td>O</td>
<td>O</td>
</tr>
<tr>
<td></td>
<td>Ogg</td>
<td></td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td></td>
<td>WebM</td>
<td></td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>error</td>
<td></td>
<td></td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>O</td>
</tr>
<tr>
<td>error.code</td>
<td></td>
<td></td>
<td>O</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>currentSrc</td>
<td></td>
<td></td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>O</td>
</tr>
<tr>
<td>networkState</td>
<td></td>
<td></td>
<td>O</td>
<td>O</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Attribute</td>
<td>Value</td>
<td>Remark</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>-----------------</td>
<td>-------</td>
<td>--------</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>preload</td>
<td>O</td>
<td>O</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>buffered</td>
<td>O</td>
<td>O</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>duration</td>
<td>O</td>
<td>O</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>currentTime</td>
<td>O</td>
<td>O</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>initialTime</td>
<td>O</td>
<td>O</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>readyState</td>
<td>O</td>
<td>O</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>paused</td>
<td>O</td>
<td>O</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ended</td>
<td>O</td>
<td>O</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>defaultPlayback</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>playbackRate</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>played</td>
<td>O</td>
<td>O</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>seeking</td>
<td>X</td>
<td>O</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>seekable</td>
<td>O</td>
<td>O</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>volume</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>muted</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Audio Events

<table>
<thead>
<tr>
<th>Event</th>
<th>Value</th>
<th>Remark</th>
</tr>
</thead>
<tbody>
<tr>
<td>loadStart</td>
<td>X</td>
<td>O</td>
</tr>
<tr>
<td>progress</td>
<td>O</td>
<td>O</td>
</tr>
<tr>
<td>suspend</td>
<td>X</td>
<td>O</td>
</tr>
<tr>
<td>abort</td>
<td>X</td>
<td>O</td>
</tr>
<tr>
<td>error</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>emptied</td>
<td>X</td>
<td>O</td>
</tr>
<tr>
<td>stalled</td>
<td>X</td>
<td>O</td>
</tr>
<tr>
<td>play</td>
<td>O</td>
<td>O</td>
</tr>
<tr>
<td>pause</td>
<td>O</td>
<td>O</td>
</tr>
<tr>
<td>loadedmetadata</td>
<td>O</td>
<td>O</td>
</tr>
<tr>
<td>loadeddata</td>
<td>X</td>
<td>O</td>
</tr>
<tr>
<td>waiting</td>
<td>X</td>
<td>O</td>
</tr>
<tr>
<td>playing</td>
<td>O</td>
<td>O</td>
</tr>
<tr>
<td>canplay</td>
<td>O</td>
<td>O</td>
</tr>
<tr>
<td>canplaythrough</td>
<td>O</td>
<td>O</td>
</tr>
<tr>
<td>seeking</td>
<td>X</td>
<td>O</td>
</tr>
<tr>
<td>sought</td>
<td>O</td>
<td>O</td>
</tr>
<tr>
<td>timeupdate</td>
<td>O</td>
<td>O</td>
</tr>
<tr>
<td>ended</td>
<td>O</td>
<td>O</td>
</tr>
<tr>
<td>ratechange</td>
<td>O</td>
<td>X</td>
</tr>
</tbody>
</table>
HTML5 Communication API\[^{4},^{5}\]

Communication API is used to share data between multiple programs in an asynchronous manner. This manner is also used for Web Workers and Server-sent Events. Cross Document Messaging enables secure communication across the web pages. Channel Messaging uses two ports for many-to-many message communication. Two-way ports are used at each end for communication. So if one port sends messages, then it is delivered to another port, and vice versa. WebSockets is used to exchange data between web servers and browsers in real-time. Once a WebSocket connection is established, the client and server can send messages to each other at any time.

Message Event Attributes

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Value</th>
<th>Remark</th>
<th>NetCast 3.0</th>
<th>NetCast 4.0</th>
<th>Emulator 2012</th>
<th>Emulator 2013</th>
</tr>
</thead>
<tbody>
<tr>
<td>data</td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>O</td>
</tr>
<tr>
<td>origin</td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>O</td>
</tr>
<tr>
<td>lastEventId</td>
<td>X</td>
<td>O</td>
<td>X</td>
<td>O</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>source</td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>O</td>
</tr>
<tr>
<td>ports</td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>O</td>
</tr>
</tbody>
</table>

Cross-Document Messaging Methods

<table>
<thead>
<tr>
<th>Method</th>
<th>Remark</th>
<th>NetCast 3.0</th>
<th>NetCast 4.0</th>
<th>Emulator 2012</th>
<th>Emulator 2013</th>
</tr>
</thead>
<tbody>
<tr>
<td>postMessage( )</td>
<td>message, targetOrigin and ports can be used as arguments (ports argument is optional). Enter the URL of the domain as the targetOrigin argument which sent the messages.</td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>O</td>
</tr>
</tbody>
</table>

Channel Messaging Methods

<table>
<thead>
<tr>
<th>Method</th>
<th>Remark</th>
<th>NetCast 3.0</th>
<th>NetCast 4.0</th>
<th>Emulator 2012</th>
<th>Emulator 2013</th>
</tr>
</thead>
<tbody>
<tr>
<td>messageChannel</td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>O</td>
</tr>
<tr>
<td>postMessage()</td>
<td>message and ports can be used as arguments (ports arguments is optional).</td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>O</td>
</tr>
<tr>
<td>start()</td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>O</td>
</tr>
<tr>
<td>close()</td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>O</td>
</tr>
</tbody>
</table>

Channel Messaging Attributes

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Value</th>
<th>Remark</th>
<th>NetCast 3.0</th>
<th>NetCast 4.0</th>
<th>Emulator 2012</th>
<th>Emulator 2013</th>
</tr>
</thead>
<tbody>
<tr>
<td>port1</td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>O</td>
</tr>
<tr>
<td>port2</td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>O</td>
</tr>
</tbody>
</table>

WebSockets Attributes and Events\[^{6}\]

<table>
<thead>
<tr>
<th>Attribute/Event</th>
<th>Value</th>
<th>Remark</th>
<th>NetCast 3.0</th>
<th>NetCast 4.0</th>
<th>Emulator 2012</th>
<th>Emulator 2013</th>
</tr>
</thead>
<tbody>
<tr>
<td>readyState</td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>O</td>
</tr>
<tr>
<td>Attribute/Event</td>
<td>Value</td>
<td>Remark</td>
<td>NetCast 3.0</td>
<td>NetCast 4.0</td>
<td>Emulator 2012</td>
<td>Emulator 2013</td>
</tr>
<tr>
<td>----------------</td>
<td>-------</td>
<td>--------</td>
<td>-------------</td>
<td>-------------</td>
<td>---------------</td>
<td>---------------</td>
</tr>
<tr>
<td>bufferedAmount</td>
<td>O</td>
<td></td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>O</td>
</tr>
<tr>
<td>open</td>
<td>O</td>
<td></td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>O</td>
</tr>
<tr>
<td>message</td>
<td>O</td>
<td></td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>O</td>
</tr>
<tr>
<td>error</td>
<td>X</td>
<td></td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>close</td>
<td>O</td>
<td></td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>O</td>
</tr>
</tbody>
</table>

Server-sent Events Attributes and Events

<table>
<thead>
<tr>
<th>Attribute/Event</th>
<th>Value</th>
<th>Remark</th>
<th>NetCast 3.0</th>
<th>NetCast 4.0</th>
<th>Emulator 2012</th>
<th>Emulator 2013</th>
</tr>
</thead>
<tbody>
<tr>
<td>readyState</td>
<td>O</td>
<td></td>
<td>X</td>
<td>O</td>
<td>X</td>
<td>O</td>
</tr>
<tr>
<td>open</td>
<td>O</td>
<td></td>
<td>X</td>
<td>O</td>
<td>X</td>
<td>O</td>
</tr>
<tr>
<td>message</td>
<td>O</td>
<td></td>
<td>X</td>
<td>O</td>
<td>X</td>
<td>O</td>
</tr>
<tr>
<td>error</td>
<td>O</td>
<td></td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>O</td>
</tr>
</tbody>
</table>

HTML5 Canvas

The HTML5 Canvas element is used to create graphics using JavaScript. Text, videos or animations can be inserted in the canvas as well as two-dimensional shapes such as circles, squares, lines and images.

Colors, Styles, Transformations, and Pixel Manipulation Methods

<table>
<thead>
<tr>
<th>Method</th>
<th>Remark</th>
<th>NetCast 3.0</th>
<th>NetCast 4.0</th>
<th>Emulator 2012</th>
<th>Emulator 2013</th>
</tr>
</thead>
<tbody>
<tr>
<td>createPattern()</td>
<td>fillStyle</td>
<td>O</td>
<td>O</td>
<td>X</td>
<td>O</td>
</tr>
<tr>
<td>createLinearGradient()</td>
<td>fillStyle</td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>O</td>
</tr>
<tr>
<td>createRadialGradient()</td>
<td>fillStyle</td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>O</td>
</tr>
<tr>
<td>createPattern()</td>
<td>strokeStyle</td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>O</td>
</tr>
<tr>
<td>createLinearGradient()</td>
<td>strokeStyle</td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>O</td>
</tr>
<tr>
<td>createRadialGradient()</td>
<td>strokeStyle</td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>O</td>
</tr>
<tr>
<td>rotate()</td>
<td>strokeStyle</td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>O</td>
</tr>
<tr>
<td>scale()</td>
<td>strokeStyle</td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>O</td>
</tr>
<tr>
<td>translate()</td>
<td>strokeStyle</td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>O</td>
</tr>
<tr>
<td>setTransform()</td>
<td>strokeStyle</td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>O</td>
</tr>
<tr>
<td>getImageData()</td>
<td>strokeStyle</td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>O</td>
</tr>
<tr>
<td>putImageData()</td>
<td>strokeStyle</td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>O</td>
</tr>
<tr>
<td>arc()</td>
<td>Include all methods and attributes for arc</td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>O</td>
</tr>
<tr>
<td>bezierCurveTo()</td>
<td>Include all methods and attributes for bezier curve</td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>O</td>
</tr>
<tr>
<td>clip()</td>
<td>Include all methods and attributes for clip</td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>O</td>
</tr>
<tr>
<td>fillRect()</td>
<td>Include all methods and attributes for rectangle</td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>O</td>
</tr>
<tr>
<td>quadraticCurveTo()</td>
<td>Include all methods and attributes for quadratic curve</td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>O</td>
</tr>
</tbody>
</table>

Colors, Styles, Composing, Shadows, Line Styles, and Shapes Attributes
<table>
<thead>
<tr>
<th>Attribute</th>
<th>Value</th>
<th>Remark</th>
<th>NetCast 3.0</th>
<th>NetCast 4.0</th>
<th>Emulator 2012</th>
<th>Emulator 2013</th>
</tr>
</thead>
<tbody>
<tr>
<td>fillStyle</td>
<td>color</td>
<td></td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>O</td>
</tr>
<tr>
<td></td>
<td>rgba</td>
<td></td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>O</td>
</tr>
<tr>
<td>strokeLine</td>
<td>color</td>
<td></td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>O</td>
</tr>
<tr>
<td></td>
<td>rgba</td>
<td></td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>O</td>
</tr>
<tr>
<td>globalAlpha</td>
<td></td>
<td></td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>O</td>
</tr>
<tr>
<td>shadowOffsetX</td>
<td></td>
<td></td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>O</td>
</tr>
<tr>
<td>shadowOffsetY</td>
<td></td>
<td></td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>O</td>
</tr>
<tr>
<td>shadowBlur</td>
<td></td>
<td></td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>O</td>
</tr>
<tr>
<td>shadowColor</td>
<td></td>
<td></td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>O</td>
</tr>
<tr>
<td>lineWidth</td>
<td></td>
<td></td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>O</td>
</tr>
<tr>
<td>lineCap</td>
<td>round</td>
<td></td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>O</td>
</tr>
<tr>
<td></td>
<td>square</td>
<td></td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>O</td>
</tr>
<tr>
<td></td>
<td>butt</td>
<td></td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>O</td>
</tr>
<tr>
<td>lineJoin</td>
<td>bevel</td>
<td></td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>O</td>
</tr>
<tr>
<td></td>
<td>round</td>
<td></td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>O</td>
</tr>
<tr>
<td></td>
<td>miter</td>
<td></td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>O</td>
</tr>
<tr>
<td></td>
<td>miterLimit</td>
<td></td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>O</td>
</tr>
</tbody>
</table>

### Interactivity, Animation, and Image

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Value</th>
<th>Remark</th>
<th>NetCast 3.0</th>
<th>NetCast 4.0</th>
<th>Emulator 2012</th>
<th>Emulator 2013</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mouse</td>
<td>Mouse</td>
<td>Include all methods and attributes for Mouse</td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>O</td>
</tr>
<tr>
<td>Animation</td>
<td>Animation</td>
<td>Include all methods and attributes for Animation</td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>O</td>
</tr>
<tr>
<td>Images/video</td>
<td>drawImage()</td>
<td></td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>O</td>
</tr>
<tr>
<td>Video</td>
<td></td>
<td>Include all methods and attributes for Video</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
</tbody>
</table>

### Canvas Text Methods

<table>
<thead>
<tr>
<th>Method</th>
<th>Remark</th>
<th>NetCast 3.0</th>
<th>NetCast 4.0</th>
<th>Emulator 2012</th>
<th>Emulator 2013</th>
</tr>
</thead>
<tbody>
<tr>
<td>createPattern()</td>
<td>fillStyle</td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>O</td>
</tr>
<tr>
<td>createLinearGradient()</td>
<td>fillStyle</td>
<td>X</td>
<td>O</td>
<td>X</td>
<td>O</td>
</tr>
<tr>
<td>createRadialGradient()</td>
<td>fillStyle</td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>O</td>
</tr>
<tr>
<td>createPattern()</td>
<td>strokeStyle</td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>O</td>
</tr>
<tr>
<td>createLinearGradient()</td>
<td>strokeStyle</td>
<td>X</td>
<td>O</td>
<td>X</td>
<td>O</td>
</tr>
<tr>
<td>createRadialGradient()</td>
<td>strokeStyle</td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>O</td>
</tr>
<tr>
<td>fillText()</td>
<td></td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>O</td>
</tr>
<tr>
<td>strokeText()</td>
<td></td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>O</td>
</tr>
<tr>
<td>rotate()</td>
<td></td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>O</td>
</tr>
<tr>
<td>scale()</td>
<td></td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>O</td>
</tr>
<tr>
<td>Method</td>
<td>Remark</td>
<td>NetCast 3.0</td>
<td>NetCast 4.0</td>
<td>Emulator 2012</td>
<td>Emulator 2013</td>
</tr>
<tr>
<td>--------------</td>
<td>-------------------------</td>
<td>-------------</td>
<td>-------------</td>
<td>---------------</td>
<td>---------------</td>
</tr>
<tr>
<td>translate()</td>
<td></td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>O</td>
</tr>
<tr>
<td>setTransform()</td>
<td></td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>O</td>
</tr>
</tbody>
</table>

Canvas Text Attributes

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Value</th>
<th>Remark</th>
<th>NetCast 3.0</th>
<th>NetCast 4.0</th>
<th>Emulator 2012</th>
<th>Emulator 2013</th>
</tr>
</thead>
<tbody>
<tr>
<td>@font-family</td>
<td>font-family</td>
<td>Provides ten fonts: Graublau Sans Web, Fertigo Pro, Tallys, Diavlo, Fontin, Fontin Sans, Pykes Peak Zero, Kaffeesatz, Vollkorn, Tagesschrift</td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>O</td>
</tr>
<tr>
<td>font</td>
<td>fillText</td>
<td></td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>O</td>
</tr>
<tr>
<td>textAlign</td>
<td>fillStyle</td>
<td></td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>O</td>
</tr>
<tr>
<td>textBaseline</td>
<td></td>
<td></td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>O</td>
</tr>
<tr>
<td>fillStyle</td>
<td>color</td>
<td></td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>O</td>
</tr>
<tr>
<td></td>
<td>rgba</td>
<td></td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>O</td>
</tr>
<tr>
<td>strokeStyle</td>
<td>color</td>
<td></td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>O</td>
</tr>
<tr>
<td></td>
<td>rgba</td>
<td></td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>O</td>
</tr>
<tr>
<td>shadowOffsetX</td>
<td></td>
<td></td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>O</td>
</tr>
<tr>
<td>shadowOffsetY</td>
<td></td>
<td></td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>O</td>
</tr>
<tr>
<td>shadowBlur</td>
<td></td>
<td></td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>O</td>
</tr>
<tr>
<td>shadowColor</td>
<td></td>
<td></td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>O</td>
</tr>
<tr>
<td>animation</td>
<td>Include all methods and attributes for text animation</td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>O</td>
<td></td>
</tr>
</tbody>
</table>

HTML5 Inline SVG[11]

SVG is used to define vector-based graphics in XML format for the web. Two-dimensional graphics can be drawn using graphic tags like `<rect>` and `<line>`. It also helps to change graphics dynamically using JavaScript or to decorate graphics using CSS. SVG is better than Canvas for creating clear, vivid graphics.

WebGL is used to define three dimensional graphics. But NetCast and Emulator do not support this feature now.

Gradients and Patterns

<table>
<thead>
<tr>
<th>Element</th>
<th>Remark</th>
<th>NetCast 3.0</th>
<th>NetCast 4.0</th>
<th>Emulator 2012</th>
<th>Emulator 2013</th>
</tr>
</thead>
<tbody>
<tr>
<td>linearGradient</td>
<td>Include all methods and attributes for linearGradient element</td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>O</td>
</tr>
<tr>
<td>radialGradient</td>
<td>Include all methods and attributes for radialGradient element</td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>O</td>
</tr>
<tr>
<td>pattern</td>
<td>Include all methods and attributes for pattern element</td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>O</td>
</tr>
</tbody>
</table>

Filter Effects and Basic Shapes

<table>
<thead>
<tr>
<th>Element</th>
<th>Remark</th>
<th>NetCast 3.0</th>
<th>NetCast 4.0</th>
<th>Emulator 2012</th>
<th>Emulator 2013</th>
</tr>
</thead>
<tbody>
<tr>
<td>feGaussianBlur</td>
<td>Include all methods and attributes for feGaussianBlur element</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>feBlend</td>
<td>Include all methods and attributes for feBlend element</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Element</td>
<td>Remark</td>
<td>NetCast 3.0</td>
<td>NetCast 4.0</td>
<td>Emulator 2012</td>
<td>Emulator 2013</td>
</tr>
<tr>
<td>------------------</td>
<td>-------------------------------------------------------------------------</td>
<td>-------------</td>
<td>-------------</td>
<td>---------------</td>
<td>---------------</td>
</tr>
<tr>
<td>feColorMatrix</td>
<td>Include all methods and attributes for feColorMatrix element</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>feComponentTransfer</td>
<td>Include all methods and attributes for feComponentTransfer element</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>feOffset</td>
<td>Include all methods and attributes for feOffset element</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>rect</td>
<td>Include all methods and attributes for rect (rectangle) element</td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>O</td>
</tr>
<tr>
<td>Rectangles with opacity</td>
<td>Include all attributes for rect element using fill-opacity and stroke-opacity.</td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>O</td>
</tr>
<tr>
<td>Rectangles with rounded corners</td>
<td>Include all attributes for rect element using rx and ry.</td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>O</td>
</tr>
<tr>
<td>circle</td>
<td>Include all methods and attributes for circle element</td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>O</td>
</tr>
<tr>
<td>ellipse</td>
<td>Include all methods and attributes for ellipse element</td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>O</td>
</tr>
<tr>
<td>line</td>
<td>Include all methods and attributes for line element</td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>O</td>
</tr>
<tr>
<td>polygon</td>
<td>Include all methods and attributes for polygon element</td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>O</td>
</tr>
<tr>
<td>polyline</td>
<td>Include all methods and attributes for polyline element</td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>O</td>
</tr>
<tr>
<td>path</td>
<td>Possible to draw spirals</td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>O</td>
</tr>
</tbody>
</table>

SVG Clipping Paths and Animation

<table>
<thead>
<tr>
<th>Element</th>
<th>Remark</th>
<th>NetCast 3.0</th>
<th>NetCast 4.0</th>
<th>Emulator 2012</th>
<th>Emulator 2013</th>
</tr>
</thead>
<tbody>
<tr>
<td>clipPath</td>
<td></td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>O</td>
</tr>
<tr>
<td>mask</td>
<td></td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>g</td>
<td>Include all methods and attributes for g element</td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>O</td>
</tr>
<tr>
<td>animateMotion</td>
<td>Include all methods and attributes for animateMotion element</td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>O</td>
</tr>
<tr>
<td>animateTransform</td>
<td>Include all methods and attributes for animateTransform element</td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>O</td>
</tr>
<tr>
<td>rect</td>
<td>Include all methods and attributes for rect element</td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>X</td>
</tr>
<tr>
<td>animate</td>
<td>Include all methods and attributes for animate element</td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>X</td>
</tr>
<tr>
<td>animateColor</td>
<td>Include all methods and attributes for animateColor element</td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>X</td>
</tr>
</tbody>
</table>

WebGL

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Value</th>
<th>Remark</th>
<th>NetCast 3.0</th>
<th>NetCast 4.0</th>
<th>Emulator 2012</th>
<th>Emulator 2013</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Activates 3D Canvas graphics</td>
<td></td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
</tbody>
</table>

HTML5 Offline Web Applications API[^12]

The Offline Web Application provides cache for working applications, even when they are offline. The application cache saves current web pages and links related to the current page to use the application without an internet connection. Therefore, the application can be used offline. Application cache also loads cached resources so the application runs quickly and the browser just receives changes from the server. This helps to reduce the burden.

ApplicationCache Methods and Attributes

<table>
<thead>
<tr>
<th>Method/Attribute</th>
<th>Value</th>
<th>Remark</th>
<th>NetCast 3.0</th>
<th>NetCast 4.0</th>
<th>Emulator 2012</th>
<th>Emulator 2013</th>
</tr>
</thead>
<tbody>
<tr>
<td>update()</td>
<td></td>
<td></td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>O</td>
</tr>
<tr>
<td>Method/Attribute</td>
<td>Value</td>
<td>Remark</td>
<td>NetCast 3.0</td>
<td>NetCast 4.0</td>
<td>Emulator 2012</td>
<td>Emulator 2013</td>
</tr>
<tr>
<td>------------------</td>
<td>-------</td>
<td>--------</td>
<td>-------------</td>
<td>-------------</td>
<td>---------------</td>
<td>--------------</td>
</tr>
<tr>
<td>swapCache()</td>
<td></td>
<td>The cache is changed in the background, but the screen does not refresh automatically so the reload() method should be used.</td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>X</td>
</tr>
<tr>
<td>status</td>
<td></td>
<td>There are six possible status: UNCACHED, IDLE, CHECKING, DOWNLOADING, UPDATEREADY, OBSOLETE</td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>O</td>
</tr>
</tbody>
</table>

**ApplicationCache Events**

<table>
<thead>
<tr>
<th>Event</th>
<th>Remark</th>
<th>NetCast 3.0</th>
<th>NetCast 4.0</th>
<th>Emulator 2012</th>
<th>Emulator 2013</th>
</tr>
</thead>
<tbody>
<tr>
<td>checking</td>
<td></td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>O</td>
</tr>
<tr>
<td>noupdate</td>
<td></td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>O</td>
</tr>
<tr>
<td>downloading</td>
<td></td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>O</td>
</tr>
<tr>
<td>progress</td>
<td></td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>O</td>
</tr>
<tr>
<td>cached</td>
<td></td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>O</td>
</tr>
<tr>
<td>updateready</td>
<td></td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>O</td>
</tr>
<tr>
<td>obsolete</td>
<td></td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>O</td>
</tr>
<tr>
<td>error</td>
<td></td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>O</td>
</tr>
</tbody>
</table>

**HTML5 Web Storage API**[13]

Web Storage is used to save a small amount of data into a client’s disk storage.

**Storage Attributes**

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Value</th>
<th>Remark</th>
<th>NetCast 3.0</th>
<th>NetCast 4.0</th>
<th>Emulator 2012</th>
<th>Emulator 2013</th>
</tr>
</thead>
<tbody>
<tr>
<td>sessionStorage</td>
<td></td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>O</td>
<td></td>
</tr>
<tr>
<td>localStorage</td>
<td></td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>O</td>
<td></td>
</tr>
</tbody>
</table>

**Indexed Database Methods**

<table>
<thead>
<tr>
<th>Method</th>
<th>Remark</th>
<th>NetCast 3.0</th>
<th>NetCast 4.0</th>
<th>Emulator 2012</th>
<th>Emulator 2013</th>
</tr>
</thead>
<tbody>
<tr>
<td>open()</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>createObjectStore()</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>transaction()</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>deleteObjectStore()</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>openCursor(range, direction)</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>put(value, key)</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>add(value, key)</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>get(key)</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>delete(key)</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>continue(key)</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
</tbody>
</table>
### WebSQL Database Methods

<table>
<thead>
<tr>
<th>Method</th>
<th>Remark</th>
<th>NetCast 3.0</th>
<th>NetCast 4.0</th>
<th>Emulator 2012</th>
<th>Emulator 2013</th>
</tr>
</thead>
<tbody>
<tr>
<td>openDatabase()</td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>O</td>
</tr>
<tr>
<td>transaction()</td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>O</td>
</tr>
<tr>
<td>executeSql()</td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>O</td>
</tr>
</tbody>
</table>

### HTML5 User Interaction API\textsuperscript{[14],[15]}

User Interaction provides two key features: Drag and Drop and History Management. Drag and Drop is used to move any object into another place, and Browser History is used for access to a browser’s history.

#### Drag&Drop Attributes

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Value</th>
<th>Remark</th>
<th>NetCast 3.0</th>
<th>NetCast 4.0</th>
<th>Emulator 2012</th>
<th>Emulator 2013</th>
</tr>
</thead>
<tbody>
<tr>
<td>draggable</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>dropzone</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
</tbody>
</table>

#### Drag&Drop Events

<table>
<thead>
<tr>
<th>Event</th>
<th>Remark</th>
<th>NetCast 3.0</th>
<th>NetCast 4.0</th>
<th>Emulator 2012</th>
<th>Emulator 2013</th>
</tr>
</thead>
<tbody>
<tr>
<td>drag</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>dragstart</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>dragenter</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>dragover</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>dragleave</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>drop</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>dragend</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
</tbody>
</table>

#### History Management Methods

<table>
<thead>
<tr>
<th>Method</th>
<th>Remark</th>
<th>NetCast 3.0</th>
<th>NetCast 4.0</th>
<th>Emulator 2012</th>
<th>Emulator 2013</th>
</tr>
</thead>
<tbody>
<tr>
<td>go()</td>
<td>O</td>
<td>O</td>
<td>X</td>
<td>O</td>
<td>O</td>
</tr>
<tr>
<td>back()</td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>O</td>
</tr>
<tr>
<td>forward()</td>
<td>O</td>
<td>O</td>
<td>X</td>
<td>O</td>
<td>O</td>
</tr>
<tr>
<td>pushState()</td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>O</td>
</tr>
<tr>
<td>replaceState()</td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>O</td>
</tr>
</tbody>
</table>

#### History Management Attributes

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Value</th>
<th>Remark</th>
<th>NetCast 3.0</th>
<th>NetCast 4.0</th>
<th>Emulator 2012</th>
<th>Emulator 2013</th>
</tr>
</thead>
<tbody>
<tr>
<td>length</td>
<td>O</td>
<td>O</td>
<td>X</td>
<td>O</td>
<td></td>
<td></td>
</tr>
<tr>
<td>state</td>
<td>O</td>
<td>O</td>
<td>X</td>
<td>O</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### HTML5 Web Workers API\textsuperscript{[16]}

Web Workers is used to support multi-tasking in the web browser. It runs JavaScript codes in parallel in the
background to optimize performance. There are dedicated workers and shared workers in Web Workers. For dedicated workers, there is a consistent one-to-one match between each worker object and each background process. For shared workers, the background processes are shared by a large number of worker objects.

### Workers

<table>
<thead>
<tr>
<th>Method</th>
<th>Remark</th>
<th>NetCast 3.0</th>
<th>NetCast 4.0</th>
<th>Emulator 2012</th>
<th>Emulator 2013</th>
</tr>
</thead>
<tbody>
<tr>
<td>Worker()</td>
<td></td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>O</td>
</tr>
<tr>
<td>terminate()</td>
<td></td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>O</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Event</th>
<th>Value</th>
<th>Remark</th>
<th>NetCast 3.0</th>
<th>NetCast 4.0</th>
<th>Emulator 2012</th>
<th>Emulator 2013</th>
</tr>
</thead>
<tbody>
<tr>
<td>onmessage</td>
<td></td>
<td></td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>O</td>
</tr>
</tbody>
</table>

### Shared workers

<table>
<thead>
<tr>
<th>Method</th>
<th>Remark</th>
<th>NetCast 3.0</th>
<th>NetCast 4.0</th>
<th>Emulator 2012</th>
<th>Emulator 2013</th>
</tr>
</thead>
<tbody>
<tr>
<td>SharedWorker()</td>
<td></td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>O</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Event</th>
<th>Value</th>
<th>Remark</th>
<th>NetCast 3.0</th>
<th>NetCast 4.0</th>
<th>Emulator 2012</th>
<th>Emulator 2013</th>
</tr>
</thead>
<tbody>
<tr>
<td>onmessage</td>
<td></td>
<td></td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>O</td>
</tr>
</tbody>
</table>

### CSS3 Standard

This section describes the properties of CSS3 which are currently supported by the NetCast platform and LG Smart TV Emulator.

- [Dynamic CSS](#)
- [Statics CSS](#)

#### Dynamic CSS

**Transforms**[^17]

<table>
<thead>
<tr>
<th>Property</th>
<th>Value</th>
<th>Remark</th>
<th>NetCast 3.0</th>
<th>NetCast 4.0</th>
<th>Emulator 2012</th>
<th>Emulator 2013</th>
</tr>
</thead>
<tbody>
<tr>
<td>matrix</td>
<td>matrix(n,n,n,n, n, n)</td>
<td></td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>O</td>
</tr>
<tr>
<td></td>
<td>matrix3d(n,n, n,n,n,n,n, n,n)</td>
<td></td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>translate</td>
<td>translate(x,y)</td>
<td></td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>O</td>
</tr>
<tr>
<td></td>
<td>translateZ(z)</td>
<td></td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td></td>
<td>translate3d(x, y,z)</td>
<td></td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>scale</td>
<td>scale(x,y)</td>
<td></td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>O</td>
</tr>
<tr>
<td></td>
<td>scale3d(x,y,z)</td>
<td></td>
<td>O</td>
<td>O</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>rotate</td>
<td>rotate(angle)</td>
<td></td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>O</td>
</tr>
<tr>
<td></td>
<td>rotateX(angle)</td>
<td></td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>O</td>
</tr>
<tr>
<td></td>
<td>rotateY(angle)</td>
<td></td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>O</td>
</tr>
<tr>
<td></td>
<td>rotateZ(angle)</td>
<td></td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>O</td>
</tr>
<tr>
<td></td>
<td>rotate3d(x,y,z)</td>
<td></td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>O</td>
</tr>
<tr>
<td>Property</td>
<td>Value</td>
<td>Remark</td>
<td>NetCast 3.0</td>
<td>NetCast 4.0</td>
<td>Emulator 2012</td>
<td>Emulator 2013</td>
</tr>
<tr>
<td>----------------</td>
<td>--------------------------------</td>
<td>--------</td>
<td>-------------</td>
<td>-------------</td>
<td>---------------</td>
<td>---------------</td>
</tr>
<tr>
<td>skew</td>
<td>skew(x-angle,y-angle)</td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>O</td>
</tr>
<tr>
<td>transform-style</td>
<td>preserve-3d</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>transform-origin</td>
<td>x-axis</td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>O</td>
</tr>
<tr>
<td></td>
<td>y-axis</td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>O</td>
</tr>
<tr>
<td></td>
<td>x-axis y-axis z-axis</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>perspective</td>
<td>number</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>perspective-origin</td>
<td>x-axis y-axis</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>backface-visibility</td>
<td>visible</td>
<td>hidden</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
</tbody>
</table>

**Transitions**\(^{[18]}\)

<table>
<thead>
<tr>
<th>Property</th>
<th>Value</th>
<th>Remark</th>
<th>NetCast 3.0</th>
<th>NetCast 4.0</th>
<th>Emulator 2012</th>
<th>Emulator 2013</th>
</tr>
</thead>
<tbody>
<tr>
<td>transition-property</td>
<td>all</td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>O</td>
</tr>
<tr>
<td>transition-duration</td>
<td>time (s)</td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>O</td>
</tr>
<tr>
<td>transition-timing-function</td>
<td>linear</td>
<td>cubic-bezier(n,n,n,n)</td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>O</td>
</tr>
<tr>
<td></td>
<td>ease</td>
<td>cubic-bezier(n,n,n,n)</td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>O</td>
</tr>
<tr>
<td></td>
<td>ease-in</td>
<td>cubic-bezier(n,n,n,n)</td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>O</td>
</tr>
<tr>
<td></td>
<td>ease-out</td>
<td>cubic-bezier(n,n,n,n)</td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>O</td>
</tr>
<tr>
<td></td>
<td>ease-in-out</td>
<td>cubic-bezier(n,n,n,n)</td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>O</td>
</tr>
<tr>
<td>transition-delay</td>
<td>time (s)</td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>O</td>
</tr>
</tbody>
</table>

**Animation**\(^{[19]}\)

<table>
<thead>
<tr>
<th>Property</th>
<th>Value</th>
<th>Remark</th>
<th>NetCast 3.0</th>
<th>NetCast 4.0</th>
<th>Emulator 2012</th>
<th>Emulator 2013</th>
</tr>
</thead>
<tbody>
<tr>
<td>@keyframes</td>
<td>animationname</td>
<td>keyframes-selector {css-styles;}</td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>O</td>
</tr>
<tr>
<td>animation-name</td>
<td>keyframename</td>
<td>none</td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>O</td>
</tr>
<tr>
<td>animation-duration</td>
<td>time (s)</td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>O</td>
</tr>
<tr>
<td>animation-timing-function</td>
<td>linear</td>
<td>cubic-bezier(n,n,n,n)</td>
<td>X</td>
<td>O</td>
<td>X</td>
<td>O</td>
</tr>
<tr>
<td></td>
<td>ease</td>
<td>cubic-bezier(n,n,n,n)</td>
<td>X</td>
<td>O</td>
<td>X</td>
<td>O</td>
</tr>
<tr>
<td>Property</td>
<td>Value</td>
<td>Remark</td>
<td>NetCast 3.0</td>
<td>NetCast 4.0</td>
<td>Emulator 2012</td>
<td>Emulator 2013</td>
</tr>
<tr>
<td>--------------------------</td>
<td>------------------------------</td>
<td>--------</td>
<td>-------------</td>
<td>-------------</td>
<td>----------------</td>
<td>----------------</td>
</tr>
<tr>
<td>bezier(n,n,n,n)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ease-in</td>
<td>cubic-bezier(n,n,n,n)</td>
<td>X</td>
<td>O</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ease-out</td>
<td>cubic-bezier(n,n,n,n)</td>
<td>X</td>
<td>O</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ease-in-out</td>
<td>cubic-bezier(n,n,n,n)</td>
<td>X</td>
<td>O</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>animation-delay</td>
<td>time (s)</td>
<td>O</td>
<td>O</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>animation-iteration-count</td>
<td>number</td>
<td>infinite</td>
<td>O</td>
<td>O</td>
<td></td>
<td></td>
</tr>
<tr>
<td>animation-direction</td>
<td>normal</td>
<td>O</td>
<td>O</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>alternate</td>
<td>O</td>
<td>O</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>animation-play-state</td>
<td>running</td>
<td>paused</td>
<td>O</td>
<td>O</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Static CSS**

**Background and Borders**

<table>
<thead>
<tr>
<th>Property</th>
<th>Value</th>
<th>Remark</th>
<th>NetCast 3.0</th>
<th>NetCast 4.0</th>
<th>Emulator 2012</th>
<th>Emulator 2013</th>
</tr>
</thead>
<tbody>
<tr>
<td>background-size</td>
<td>length (px)</td>
<td>O</td>
<td>O</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>percentage (%)</td>
<td>O</td>
<td>O</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>cover</td>
<td>O</td>
<td>O</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>contain</td>
<td>O</td>
<td>O</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>background-clip</td>
<td>padding-box</td>
<td>O</td>
<td>O</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>border-box</td>
<td>O</td>
<td>O</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>content-box</td>
<td>O</td>
<td>O</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>background-origin</td>
<td>padding-box</td>
<td>O</td>
<td>O</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>border-box</td>
<td>O</td>
<td>O</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>content-box</td>
<td>O</td>
<td>O</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>multiple-backgrounds</td>
<td></td>
<td>O</td>
<td>O</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>box-shadow</td>
<td>h-shadow</td>
<td>O</td>
<td>O</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>v-shadow</td>
<td>O</td>
<td>O</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>blur</td>
<td>O</td>
<td>O</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>spread</td>
<td>O</td>
<td>O</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>color</td>
<td>O</td>
<td>O</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>inset</td>
<td>O</td>
<td>O</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>border-bottom-left-radius</td>
<td></td>
<td>O</td>
<td>O</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Property</td>
<td>Value</td>
<td>Remark</td>
<td>NetCast 3.0</td>
<td>NetCast 4.0</td>
<td>Emulator 2012</td>
<td>Emulator 2013</td>
</tr>
<tr>
<td>------------------------</td>
<td>-------</td>
<td>--------</td>
<td>-------------</td>
<td>-------------</td>
<td>---------------</td>
<td>---------------</td>
</tr>
<tr>
<td>border-bottom-right-radius</td>
<td></td>
<td></td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>O</td>
</tr>
<tr>
<td>border-top-left-radius</td>
<td></td>
<td></td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>O</td>
</tr>
<tr>
<td>border-top-right-radius</td>
<td></td>
<td></td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>O</td>
</tr>
<tr>
<td>border-radius</td>
<td></td>
<td></td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>O</td>
</tr>
<tr>
<td>border-image</td>
<td></td>
<td></td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>O</td>
</tr>
</tbody>
</table>

**Box[21]**

<table>
<thead>
<tr>
<th>Property</th>
<th>Value</th>
<th>Remark</th>
<th>NetCast 3.0</th>
<th>NetCast 4.0</th>
<th>Emulator 2012</th>
<th>Emulator 2013</th>
</tr>
</thead>
<tbody>
<tr>
<td>overflow-x</td>
<td>visible</td>
<td></td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>O</td>
</tr>
<tr>
<td></td>
<td>hidden</td>
<td></td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>O</td>
</tr>
<tr>
<td></td>
<td>scroll</td>
<td></td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>O</td>
</tr>
<tr>
<td></td>
<td>auto</td>
<td></td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>O</td>
</tr>
<tr>
<td>overflow-y</td>
<td>visible</td>
<td></td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>O</td>
</tr>
<tr>
<td></td>
<td>hidden</td>
<td></td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>O</td>
</tr>
<tr>
<td></td>
<td>scroll</td>
<td></td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>O</td>
</tr>
<tr>
<td></td>
<td>auto</td>
<td></td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>O</td>
</tr>
</tbody>
</table>

**Color[22]**

<table>
<thead>
<tr>
<th>Property</th>
<th>Value</th>
<th>Remark</th>
<th>NetCast 3.0</th>
<th>NetCast 4.0</th>
<th>Emulator 2012</th>
<th>Emulator 2013</th>
</tr>
</thead>
<tbody>
<tr>
<td>HSL</td>
<td>RGB</td>
<td></td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>O</td>
</tr>
<tr>
<td>HSLA</td>
<td>RGB</td>
<td></td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>O</td>
</tr>
<tr>
<td>RGB &amp; opacity</td>
<td>RGB</td>
<td></td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>O</td>
</tr>
<tr>
<td>RGBA &amp; opacity</td>
<td>RGB</td>
<td></td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>O</td>
</tr>
<tr>
<td>RGBA</td>
<td>RGB</td>
<td></td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>O</td>
</tr>
</tbody>
</table>

**Columns and Multi Column Layout[23]**

<table>
<thead>
<tr>
<th>Property</th>
<th>Value</th>
<th>Remark</th>
<th>NetCast 3.0</th>
<th>NetCast 4.0</th>
<th>Emulator 2012</th>
<th>Emulator 2013</th>
</tr>
</thead>
<tbody>
<tr>
<td>column-width</td>
<td></td>
<td></td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>O</td>
</tr>
<tr>
<td>column-count</td>
<td></td>
<td></td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>O</td>
</tr>
<tr>
<td>column-gap</td>
<td></td>
<td></td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>O</td>
</tr>
<tr>
<td>column-rule</td>
<td></td>
<td></td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>O</td>
</tr>
<tr>
<td>column-rule-color</td>
<td>color</td>
<td></td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>O</td>
</tr>
<tr>
<td>column-rule-style</td>
<td>none</td>
<td></td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>O</td>
</tr>
<tr>
<td></td>
<td>hidden</td>
<td></td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>O</td>
</tr>
<tr>
<td>Property</td>
<td>Value</td>
<td>Remark</td>
<td>NetCast 3.0</td>
<td>NetCast 4.0</td>
<td>Emulator 2012</td>
<td>Emulator 2013</td>
</tr>
<tr>
<td>-------------------</td>
<td>-----------</td>
<td>--------</td>
<td>-------------</td>
<td>-------------</td>
<td>---------------</td>
<td>---------------</td>
</tr>
<tr>
<td>dotted</td>
<td>O</td>
<td></td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>O</td>
</tr>
<tr>
<td>dashed</td>
<td>O</td>
<td></td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>O</td>
</tr>
<tr>
<td>solid</td>
<td>O</td>
<td></td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>O</td>
</tr>
<tr>
<td>double</td>
<td>O</td>
<td></td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>O</td>
</tr>
<tr>
<td>groove</td>
<td>O</td>
<td></td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>O</td>
</tr>
<tr>
<td>ridge</td>
<td>O</td>
<td></td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>O</td>
</tr>
<tr>
<td>inset</td>
<td>O</td>
<td></td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>O</td>
</tr>
<tr>
<td>outset</td>
<td>O</td>
<td></td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>O</td>
</tr>
<tr>
<td>column-rule-width</td>
<td>thin</td>
<td></td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>O</td>
</tr>
<tr>
<td></td>
<td>medium</td>
<td></td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>O</td>
</tr>
<tr>
<td></td>
<td>thick</td>
<td></td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>O</td>
</tr>
<tr>
<td></td>
<td>length</td>
<td></td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>O</td>
</tr>
<tr>
<td>column-span</td>
<td>1</td>
<td></td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>O</td>
</tr>
<tr>
<td></td>
<td>all</td>
<td></td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>O</td>
</tr>
</tbody>
</table>

**Flexible Box**[^24] [^25]

<table>
<thead>
<tr>
<th>Property</th>
<th>Value</th>
<th>Remark</th>
<th>NetCast 3.0</th>
<th>NetCast 4.0</th>
<th>Emulator 2012</th>
<th>Emulator 2013</th>
</tr>
</thead>
<tbody>
<tr>
<td>box-align</td>
<td>start</td>
<td></td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>O</td>
</tr>
<tr>
<td></td>
<td>end</td>
<td></td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>O</td>
</tr>
<tr>
<td></td>
<td>center</td>
<td></td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>O</td>
</tr>
<tr>
<td></td>
<td>stretch</td>
<td></td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>O</td>
</tr>
<tr>
<td>box-direction</td>
<td>normal</td>
<td></td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>O</td>
</tr>
<tr>
<td></td>
<td>reverse</td>
<td></td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>O</td>
</tr>
<tr>
<td>box-flex</td>
<td>value</td>
<td></td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>O</td>
</tr>
<tr>
<td>box-ordinal-group</td>
<td>integer</td>
<td></td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>O</td>
</tr>
<tr>
<td>box-orient</td>
<td>horizontal</td>
<td></td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>O</td>
</tr>
<tr>
<td></td>
<td>vertical</td>
<td></td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>O</td>
</tr>
<tr>
<td></td>
<td>inline-axis</td>
<td></td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>O</td>
</tr>
<tr>
<td></td>
<td>block-axis</td>
<td></td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>O</td>
</tr>
<tr>
<td>box-pack</td>
<td>start</td>
<td></td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>O</td>
</tr>
<tr>
<td></td>
<td>end</td>
<td></td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>O</td>
</tr>
<tr>
<td></td>
<td>center</td>
<td></td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>O</td>
</tr>
<tr>
<td></td>
<td>justify</td>
<td></td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>O</td>
</tr>
</tbody>
</table>

**Text**[^26] [^27] [^28]

<table>
<thead>
<tr>
<th>Property</th>
<th>Value</th>
<th>Remark</th>
<th>NetCast 3.0</th>
<th>NetCast 4.0</th>
<th>Emulator 2012</th>
<th>Emulator 2013</th>
</tr>
</thead>
<tbody>
<tr>
<td>text-shadow</td>
<td>basic</td>
<td></td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>O</td>
</tr>
<tr>
<td></td>
<td>blur effect</td>
<td></td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>O</td>
</tr>
<tr>
<td>Property</td>
<td>Value</td>
<td>Remark</td>
<td>NetCast 3.0</td>
<td>NetCast 4.0</td>
<td>Emulator 2012</td>
<td>Emulator 2013</td>
</tr>
<tr>
<td>------------------</td>
<td>---------------</td>
<td>--------------</td>
<td>-------------</td>
<td>-------------</td>
<td>---------------</td>
<td>---------------</td>
</tr>
<tr>
<td>shadow on text</td>
<td></td>
<td></td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>O</td>
</tr>
<tr>
<td>neon glow</td>
<td></td>
<td></td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>O</td>
</tr>
<tr>
<td>multiple shadow</td>
<td></td>
<td></td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>O</td>
</tr>
<tr>
<td>quick and dirty</td>
<td>letterpress</td>
<td></td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>O</td>
</tr>
<tr>
<td>hard shadow</td>
<td></td>
<td></td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>O</td>
</tr>
<tr>
<td>double shadow</td>
<td></td>
<td></td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>O</td>
</tr>
<tr>
<td>down and distant</td>
<td></td>
<td></td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>O</td>
</tr>
<tr>
<td>close and heavy</td>
<td></td>
<td></td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>O</td>
</tr>
<tr>
<td>3D text</td>
<td></td>
<td></td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>O</td>
</tr>
<tr>
<td>true inset text</td>
<td></td>
<td></td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>O</td>
</tr>
<tr>
<td>Glowing</td>
<td></td>
<td></td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>O</td>
</tr>
<tr>
<td>Multiple Light Sources</td>
<td></td>
<td></td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>O</td>
</tr>
<tr>
<td>Soft Emboss</td>
<td></td>
<td></td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>O</td>
</tr>
<tr>
<td>text-overflow</td>
<td>clip</td>
<td></td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>O</td>
</tr>
<tr>
<td></td>
<td>ellipsis</td>
<td></td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>O</td>
</tr>
<tr>
<td>word-break</td>
<td>normal</td>
<td></td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>O</td>
</tr>
<tr>
<td></td>
<td>break-all</td>
<td></td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>O</td>
</tr>
<tr>
<td>word-wrap</td>
<td>normal</td>
<td></td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>O</td>
</tr>
<tr>
<td></td>
<td>break-word</td>
<td></td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>O</td>
</tr>
</tbody>
</table>

**Annex A References**

The following referenced websites are required for the application of this specification.

<table>
<thead>
<tr>
<th>Index</th>
<th>Reference</th>
<th>Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>Index</td>
<td>Reference</td>
<td>Title</td>
</tr>
<tr>
<td>-------</td>
<td>-----------</td>
<td>-------</td>
</tr>
<tr>
<td>[21]</td>
<td><a href="http://dev.w3.org/csswg/css3-box/">http://dev.w3.org/csswg/css3-box/</a></td>
<td>W3C Editor’s Draft 1 November 2012 – CSS Basic Box Model</td>
</tr>
<tr>
<td>[25]</td>
<td><a href="http://www.w3schools.com/cssref/#flexbox">http://www.w3schools.com/cssref/#flexbox</a></td>
<td>w3schools CSS Flexible Box Properties</td>
</tr>
</tbody>
</table>
Sample Tutorials: LG WebAPI Tutorials

LG WebAPI samples are provided to help developers create web application using LG Web APIs easily. Several sample codes related to various topics are provided as follows.

- Activating or Deactivating Browser Page Loading Icon
- Activating or Deactivating Magic Remote
- Controlling Voice Recognition
- Displaying Subtitle in Media Player
- Executing Windows Media Player 1
- Executing Windows Media Player 2
- Executing Windows Meida Player 3
- Generating Outofmemory Event
- Getting and Displaying Device Information
- Handling Key Inputs Using JavaScript Events
- Handling Key Repeat Using JavaScript Events
- Implementing Functionality of Back and Exit Keys
- Implementing Functionality of Multi Audio
- Playing Media with ASX File
- Playing Video in Full Screen Mode


Note
Refer to Developing > API section in this Library for detailed information on how to use Web APIs.

Activating or Deactivating Browser Page Loading Icon

This section describes how to activate and deactivate the browser page loading icon in web application.

- Initializing the Page
- Inputting Keys
- Handling Keys
- Handling Events
- Counting Time
- Source Code of pageloadingicon.html

It is recommended that application authors provide a “page loading icon” so that users are provided with an indication of the latency of data downloading from a server.

This application is designed to show how to activate or deactivate the “page loading icon” using the Web Open API of LG Smart TV.

This application shows which method is used to activate or deactivate the “page loading icon”.

1. Press Red-key to enable Page Loading Icon and test it.
2. Press Green-key to disable Page Loading Icon and test it.
### Needed APIs

This application uses following Web Open API:

<table>
<thead>
<tr>
<th>API Class</th>
<th>Function Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Method</td>
<td>window.NetCastSetPageLoadingIcon()</td>
<td>This API can be used by an LG Smart TV application author to activate or deactivate the browser's own page loading animation.</td>
</tr>
<tr>
<td>Property</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>Event</td>
<td>N/A</td>
<td>N/A</td>
</tr>
</tbody>
</table>

**Note**

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

### Initializing the Page

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

04: Records the last visited page when running the application.
07: Initializes the page.
08: Gets the source code of the page using the XMLHttpRequest object.
09: Sets the page ID.
10: Initializes the Log function.
13-16: Registers an event handler which will executed when the corresponding button is pressed.

```javascript
01 : function initPage()
02 : {
03 :     //save page as last visited page
04 :     setLastVisitPage();
05 : 
06 :     //common initialize function
07 :     commonInitialize();
08 :     requestSourceCode();
09 :     setPageID("Page Loading Icon");
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 : }
```

### Inputting Keys

The `onUserInput` function is called by the `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.

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

The following functions are executed when each key of onUserInput() is pressed.

**processRedKey**

This function is called when the Red (Enable) key is pressed.
05: Enables the page loading icon.
08: Shows on the screen that the page loading icon is enabled.
09: Maintains the text printed from code 10 through an alert until you press the [OK] button.
10: Reloads the page. When the page is reloaded, you can see the "Loading Icon" appearing on the screen.

**processGreenKey**

This function is called when the Green (Disable) key is pressed.
18: Calls the setCountDown() function.
19: Disables the page loading icon.
21: Shows on the screen that the page loading icon is disabled.

```
01 : function processRedKey()
02 : {
03 :   if(isThisLGEBrowser())
04 :     {
05 :     window.NetCastSetPageLoadingIcon('enabled');
06 :     jsLog.lgmethod('window.NetCastSetPageLoadingIcon()');
07 :     setInnerTextById("description", "Loading Icon is enabled!");
08 :     alert("Enable!");
09 :     window.location.reload();
10 :   }
11 : }
12 :
13 :
14 : function processGreenKey()
15 : {
16 :   if(isThisLGEBrowser())
17 :   {
18 :     setCountDown();
19 :     window.NetCastSetPageLoadingIcon('disabled');
20 :     jsLog.lgmethod('window.NetCastSetPageLoadingIcon()');
21 :     setInnerTextById("description", "Loading Icon is disabled!");
22 :   }
23 : }
```

Handling Events

The **setCountDown** function is called when the Green key is pressed.

07: Sets timerCount to 5 so it counts the remaining time of 5 seconds until the loading icon is disabled.
08: Calls the showLeftTime function every 1 second to display the remaining time until the loading icon is disabled.
Set the timer.
09: Shows the text on the screen.

```
01 : var timer;
02 : var timeCount = 5;
03 :
04 : //"GreenKey" click handler
05 : function setCountDown()
06 : {
07 :   timeCount = 5;
08 :   timer = setInterval(showLeftTime, 1000);
09 :   setInnerTextById("description", "Loading Icon will be off after 5 second");
10 : }
```
Counting Time
The following sample code counts and displays the remaining time until the "loading icon" is set to "disable".

The `showLeftTime` function counts the remaining time until the loading icon is disabled.

04: Decreases `timeCount` by 1.
05-06: If `timeCount` is greater than 0, the message "This page will be reloaded after + "timeCount" + " second" will be displayed on the id = "description" area.
09: If `timeCount` is 0, the clearInterval is called the setInterval function stops the specified timer.
10: Reloads the current page. When the page is reloaded, you can see the "Loading Icon" not appearing on the screen.

```javascript
01 : //show left time
02 : function showLeftTime()
03 : {
04 :     timeCount--;  
05 :     if(timeCount > 0)
06 :         innerTextById("description", "Loading Icon will be off after " +
07 :             timeCount + " second");
08 :     else
09 :         clearInterval(timer);
10 :         window.location.reload();
11 : }
12 : }
```

Source Code of pageloadingicon.html
Source code of pageloadingicon.html is as follows:

```html
<!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>
    //initialize page
    function initPage()
    {
        //save page as last visited page
        setLastVisitPage();

        //common initialize function
        commonInitialize();
        requestSourceCode();
        setPageID("Page Loading Icon");
        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);
    }
</script>
```

83
addEventHandler(document.getElementById("btn_exit"), "click", onClickHandler);
}

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

function processRedKey()
{
    if(isThisLGEBrowser())
    {
        window.NetCastSetPageLoadingIcon('enabled');
        jsLog.lgmethod('window.NetCastSetPageLoadingIcon()');
        setInnerTextById("description", "Loading Icon is enabled!");
        alert("Enable!");
        window.location.reload();
    }
}

function processGreenKey()
{
    if(isThisLGEBrowser())
    {
        setCountDown();
        window.NetCastSetPageLoadingIcon('disabled');
        jsLog.lgmethod('window.NetCastSetPageLoadingIcon()');
        setInnerTextById("description", "Loading Icon is disabled!");
    }
}

var timer;
var timeCount = 5;

//"GreenKey" click handler
function setCountDown()
{
    timeCount = 5;
    timer = setInterval(showLeftTime, 1000);
    setInnerTextById("description", "Loading Icon will be off after 5 second");
}

//show left time
function showLeftTime()
{
    timeCount--;
    if(timeCount > 0)
    {
        setInnerTextById("description", "This page will be reloaded after " +
        timeCount + " second");
    }
    else
    {
        clearInterval(timer);
    }
window.location.reload();
}
</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 : /netcast/app/pageloadingicon.html</div>
</div>

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

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

<div class='ViewTitleArea'>
  <div id='view' style='width:500px; margin-left: 100px;'>1. Press Red-key to enable Page Loading Icon and test it.</div>
</div>

<div class='ViewTitleArea'>
  <div id='view' style='width:500px; margin-left: 100px;'>2. Press Red-key again to disable Page Loading Icon and test it again.</div>
</div>

...
2. Press Green-key to disable Page Loading Icon and test it.

<table>
<thead>
<tr>
<th>width=500px align=left &gt;</th>
<th>width=200px align=left</th>
</tr>
</thead>
<tbody>
<tr>
<td>2. Press Green-key to disable Page Loading Icon and test it.</td>
<td></td>
</tr>
</tbody>
</table>
Activating or Deactivating Magic Remote

This section describes how to activate and deactivate the magic remote in web application.

- **Initializing the Page**
- **Inputting Keys**
- **Handling Events**
- **Counting Time**
- **Adding Mouse Event Handler**
- **Source Code of mouse.html**

This application is designed to show how to deactivate the Magic Remote using the Web Open API of LG Smart TV. This application shows which method is used to deactivate the Magic Remote and how Magic Remote handles activation/deactivation events.

### Needed APIs

This application uses following Web Open API:

<table>
<thead>
<tr>
<th>API Class</th>
<th>Function Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Method</td>
<td>Window.NetCastMouseOff(time)</td>
<td>This API can be used by an LG Smart TV application author to deactivate the Magic Remote and its pointer.</td>
</tr>
<tr>
<td></td>
<td>Window.NetCastGetMouseOnOff()</td>
<td>This API can be used by an LG Smart TV application author to get the on or off status of the Magic Remote. Its return value is “on” or “off”.</td>
</tr>
<tr>
<td>Property</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>Event</td>
<td>mouseon</td>
<td>This event is generated when the Magic Remote is activated.</td>
</tr>
<tr>
<td></td>
<td>mouseoff</td>
<td>This event is generated when the Magic Remote is deactivated.</td>
</tr>
</tbody>
</table>

**Note**

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

### Initializing the Page

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

- 04: Records the last visited page when running the application.
- 07: Initializes the page.
- 08: Gets the source code of the page using the XMLHttpRequest object.
- 09: Sets the page ID.
- 10: Initializes the Log function.
- 13-17: Adds an event handler which will executed when the corresponding button is pressed.
19: Calls the updateMouseStatus() function.

```javascript
01 : function initPage()
02 : {
03 :     // save page as last visited page
04 :     setLastVisitPage();
05 : 
06 :     // common initialize function
07 :     commonInitialize();
08 :     requestSourceCode();
09 :     setPageID("Mouse");
10 :     jsLog.initLG();
11 :
12 :     // add onclick event handler
13 :     addEventHandler(document.getElementById("btn_back"), "click", onClickHandler);
14 :     addEventHandler(document.getElementById("btn_exit"), "click", onClickHandler);
15 :
16 :     // add event handler for test
17 :     addEventHandler(document.getElementById("start"), "click", onClickMouseOff);
18 :
19 :     updateMouseStatus();
20 : }
```

### Inputting Keys

The `onUserInput` function is called by the `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.

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

### Handling Events

The following functions are for event handlers added by the `initPage` function.

#### mouseon_handler

The event handler called when the mouse is turned on.

04: `updateMouseStatus` is called. It defines functions which will be executed when the corresponding button is pressed.

#### mouseoff_handler

The event handler called when the mouse is turned off.

10: `updateMouseStatus` is called. It defines functions which will be executed when the corresponding button is pressed.

#### updateMouseStatus

Defines the action that will be taken depending on the mouse status.

16: Stores the mouse on or off status in the status variable.

17: Displays the value stored in the status variable.

19-23: If the mouse status is on and device is LG Browser4.0, Mouse off button is enabled and the message is printed.

24-27: If the mouse status is on and device is LG Browser5.0, the message is printed.

28-32: If the mouse status is off, Mouse off button is disabled and the message is printed.
checkNetCastVersion

Checks the LG Browser version and decides registration of mouseoff event.
35: Stores version information in the nBrowserVersion.
38-42: If the device is LG Browser 4.0, mouseoff event is registered and the message is printed.
43-47: If the device is LG Browser 5.0, Mouse off button is disabled and the message is printed.

onClickMouseOff

The event handler called when the Mouse off button is pressed.
53: Calls the window.NetCastMouseOff(time) API to turn off the mouse. To have the mouse turned off in 5 seconds, the parameter of 5 is entered.
54: Sets timeCount to 5 so it counts the remaining time of 5 seconds until the mouse is turned off.
55: Calls the showLeftTime function every 1 second to display the remaining time until the mouse is turned off. Set the timer.

```javascript
function mouseon_handler()
{
    jsLog.lgevent('mouseon');
    updateMouseStatus();
}

function mouseoff_handler()
{
    jsLog.lgevent('mouseoff');
    updateMouseStatus();
}

function updateMouseStatus()
{
    jsLog.lgmethod('window.NetCastGetMouseOnOff()');
    status = window.NetCastGetMouseOnOff();
    setInnerTextById("mouse_status", status);
    if(status == 'on' && nBrowserVersion == 4)
    {
        document.getElementById("mouse_off").style.visibility = "visible";
        setInnerTextById("description", "");
    }
    else if(status == 'on' && nBrowserVersion >= 5)
    {
        setInnerTextById("description", "");
    }
    else
    {
        document.getElementById("mouse_off").style.visibility = "hidden";
        setInnerTextById("description", "Press the OK button of mouse to use mouse");
    }
}

var nBrowserVersion = getBrowserVersion();
function checkNetCastVersion()
{
    if(nBrowserVersion == 4) // NetCast 2.0
    {
        addEventHandler(document.getElementById("mouse_off"), "click", onClickMouseOff);
        setInnerTextById("APIdescription", "In the LG DTV Emulator 2011, mouseoff event is not supported.");
    }
    else if(nBrowserVersion >= 5) // NetCast 3.0
    {
...
Counting Time

The `showLeftTime` function counts the remaining time until the mouse is turned off. The following sample code counts and displays the remaining time until the mouse is set to off by the `onClickMouseOff` function.

07: Decreases `timeCount` by 1.
08-09: If `timeCount` is greater than 0, the message "Mouse will be off after + "timeCount" + second" will be displayed on the id = "description" area.
10-11: If `timeCount` is 0, the clearInterval is called the setInterval function stops the specified timer.

Adding Mouse Event Handler

The following sample code adds the event handlers of mouseon and mouseoff.

Source Code of `mouse.html`

Source code of `mouse.html` is as follows:
//initialize page
function initPage() {
  //save page as last visited page
  setLastVisitPage();

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

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

  //add event handler for test
  checkNetCastVersion();
  //addEventHandler(document.getElementById("mouse_off"), "click", onClickMouseOff);

  updateMouseStatus();
}

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

function mouseon_handler() {
  jsLog.lgevent('mouseon');
  updateMouseStatus();
}

function mouseoff_handler() {
  jsLog.lgevent('mouseoff');
  updateMouseStatus();
}

function updateMouseStatus() {
  jsLog.lgmethod('window.NetCastGetMouseOnOff()');
  status = window.NetCastGetMouseOnOff();
  setInnerTextById("mouse_status", status);

  if(status == 'on' && nBrowserVersion == 4) {

document.getElementById("mouse_off").style.visibility = "visible";
setInnerTextById("description", "");

else if(status == 'on' && nBrowserVersion >= 5)
{
    setInnerTextById("description", "");
}
else
{
    document.getElementById("mouse_off").style.visibility = "hidden";
    setInnerTextById("description", "Press the OK button of mouse to use mouse");
}

var nBrowserVersion = getBrowserVersion();
function checkNetCastVersion()
{
    if(nBrowserVersion == 4) // NetCast 2.0
    {
        addEventHandler(document.getElementById("mouse_off"), "click", onClickMouseOff);
        setInnerTextById("APIDescription", "In the LG DTV Emulator 2011, mouseoff event is not supported.");
    }
    else if(nBrowserVersion >= 5) // NetCast 3.0
    {
        document.getElementById("mouse_off").style.visibility = "hidden";
        setInnerTextById("APIDescription", "In NetCast 3.0, NetCastMouseOff(time) API is not supported and the mouse gets deactivated when the halt of the mouse movement continues for 3 seconds only on TV set.");
    }
}

//"Mouse Off" click handler
function onClickMouseOff()
{
    jsLog.lgmethod('window.NetCastMouseOff(time)');
    window.NetCastMouseOff(5);
    timeCount = 5;
    timer = setInterval(showLeftTime, 1000);
    setInnerTextById("description", "Mouse will be off after 5 second");
}

//show left time
var timer;
var timeCount = 5;
function showLeftTime()
{
    timeCount--;
    if(timeCount > 0)
    {
        setInnerTextById("description", "Mouse will be off after " + timeCount + " second");
    }
    else
    {
        clearInterval(timer);
    }
}

window.onmouseoff = mouseoff_handler;
window.onmouseon = mouseon_handler;
<div class='MethodTitleArea'>Methods</div>
<div class='MethodListArea'>
window.NetCastMouseOff(time)<br>
window.NetCastGetMouseOnOff()
</div>
</div>

<div class='PropertyTitleArea'>Properties</div>
<div class='PropertyListArea'></div>
</div>
</div>

<div class='EventTitleArea'>Events</div>
<div class='EventListArea'>
mouseon<br>
mouseoff<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'>
<table border="0" cellpadding="0" cellspacing="0">
<tr height="50px">
<td width=200px align=left><div>Mouse Status :</div></td>
<td width=200px align="left"><div id="mouse_status"></div></td>
<td width=100px align="left"></td>
</tr>
</table>
</div>
</div>

<div id='tabCodeArea' class='UnselectedViewArea' style='float:right;' onclick='showCode();'>Source</div>
</div>
</div>
</body>
</html>
Controlling Voice Recognition

This section describes how to create the web application using LG Web Open APIs.

- Creating Web Applications

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

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

<table>
<thead>
<tr>
<th>API Class</th>
<th>Function Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Method</td>
<td>startRecognition</td>
<td>Call native UI of the voice recognition function and receive the result as an event.</td>
</tr>
<tr>
<td></td>
<td>isInitialized</td>
<td>Determine whether the voice recognition function is initialized and returns the value as Boolean.</td>
</tr>
<tr>
<td></td>
<td>isEnable</td>
<td>Determine whether the Magic Remote is paired (including its type) and whether the voice recognition function is enabled, and returns the value as Boolean.</td>
</tr>
<tr>
<td></td>
<td>dictation</td>
<td>Determine whether the voice recognition function is in dictation mode and returns the value as string (on/off). If the return value is &quot;off&quot;, the function is in keyword mode. This methods can also be used to select the mode.</td>
</tr>
<tr>
<td>Event</td>
<td>onrecognizevoice</td>
<td>Receive the voice recognition result from the TV.</td>
</tr>
<tr>
<td></td>
<td>onbuttonenable</td>
<td>Enable or disable the voice recognition button</td>
</tr>
</tbody>
</table>

For more information on these functions, refer to Developing > API > Voice Recognition API section in this Library.

Note

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

This section describes how to create web applications that controls the voice recognition of the Magic Remote using LG Web Open APIs.

- Initializing the Page
- Check the NetCast Version
- Initializing Voice Recognition
- Handling Button Click Events
- Handling Voice Events
- Displaying Text
- Inputting Keys
- Setting Voice Recognition Object
- Source Code of voice_recognition.html

Initializing the Page

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

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

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

Check the NetCast Version

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

03: Declare nBrowserVersion.
04-08: If return value of nBrowserVersion is 4, then hide the visibility of all three buttons : a. dictation_mode, b. keyword_mode, c. start_Recording
11-16: If return value of nBrowserVersion is equal or greater than 5, then display the names for supported method, property and event in API list area.
function checkNetCastVersion()
{
    var nBrowserVersion = getBrowserVersion();
    if(nBrowserVersion == 4) // NetCast 2.0
    {
        document.getElementById('dictation_mode').style.visibility = "hidden";
        document.getElementById('keyword_mode').style.visibility = "hidden";
        document.getElementById('start_Recording').style.visibility = "hidden";
    }
    else if(nBrowserVersion >= 5) // NetCast 3.0
    {
        setInnerTextById("APIdescription", "The voice recognition plugin and API are supported on NetCast 3.0 1st SU or later.");
        setInnerTextById("method", "startRecognition()");
        setInnerTextById("property", "isInitialized<br>.isEnabled<br>dictation");
        setInnerTextById("events", "onrecognizevoice<br>onbuttonenable");
    }
}

function initVoice()
{
    var device = document.getElementById("device");
    var voice = document.getElementById('voice');
    if(!(device.supportVoiceRecog || voice.isInitialized || voice.isEnable))
    {
        //alert("em");
        document.getElementById("start_Recording").className="executeBigButtonOff";
    }
    else
    {
        addEventHandler(document.getElementById("start_Recording"), "click", voiceHandler);
        voice.onrecognizevoice = recognizeVoiceHandler;
        voice.onbuttonenable = buttonStateHandler;
    }
}

Handling Button Click Events
The following functions are event handlers which were added in initPage function.

dictationModeHandler
This function is called when ‘dictation mode’ button is clicked.

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

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

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

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

**voiceHandler**
This function is called when ‘Start Recording’ button is clicked.

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

```
01 : function voiceHandler()
02 : {
03 :   var voice = document.getElementById('voice');
04 :   if(voice.isEnable == true)
05 :     {
06 :       voice.startRecognition();
07 :       jsLog/lgmethod('startRecognition()');
08 :     }
09 : }
```

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

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

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

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

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

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

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

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

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

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

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

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

Setting Voice Recognition Object
The following code shows how to set voice recognition object.

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

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

Source Code of voice_recognition.html
Source code of voice_recognition.html is as follows:

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

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

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

    checkNetCastVersion();

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

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

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

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

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

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

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

function voiceHandler()
{
  var voice = document.getElementById('voice');
  if(voice.isEnable == true)
  {
    voice.startRecognition();
    jsLog.lgmethod('startRecognition()');
  }
}

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

function buttonStateHandler(e)
{
  if(e)
  {
    if(e == true)
    {
      console.log(">O< buttonStateHandler is called");
      document.getElementById("start_recording").className="executeBigButton";
      addEventHandler(document.getElementById("start_recording"), "click", voiceHandler);
      jsLog.lgevent('buttonStateHandler : '+ true');
    }
    else
    {
      document.getElementById("start_recording").className="executeBigButtonOff";
      removeEventHandler(document.getElementById("start_recording"), "click", voiceHandler);
      jsLog.lgevent('buttonStateHandler : '+ false');
    }
  }
function onDictationModeHandler()
{
var voice = document.getElementById('voice');
voice.dictation = "on";
showPropResult()
jsLog/lgproperty('dictation');
}

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

function checkNetCastVersion()
{
var nBrowserVersion = getBrowserVersion();
if(nBrowserVersion == 4) // NetCast 2.0
{

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

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

</script>
</head>

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

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

<!-- navigation -->
<div class='SuiteNavigation'>File: voice/voice_recognition.html</div>

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

<div class='ApiListTitleArea'>API List</div>
<div class='ApiListArea'>

<div class='MethodTitleArea'>Methods</div>
<div class='MethodListArea'>

</div>

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

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

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

<!-- copyright -->
<div class='copyright'>Copyright LG Electronics</div>
</div>
</body>
</html>
Displaying Subtitle in Media Player

This section describes how to display the subtitle in media player.

- Initializing the Page
- Inputting Keys
- Displaying Text
- Controlling Play Mode and Subtitle
- Setting Media Object
- Source Code of subtitle.html

This application is designed to show how to display subtitle while media file is playing using the Web Open API of Smart TV.

This application shows which property is used to set and control on/off status of subtitle.

### Needed APIs

This application uses following Web Open API:

<table>
<thead>
<tr>
<th>API Class</th>
<th>Function Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Method</td>
<td>play(0)</td>
<td>Pauses media.</td>
</tr>
<tr>
<td></td>
<td>play(1)</td>
<td>Plays media.</td>
</tr>
<tr>
<td></td>
<td>stop()</td>
<td>Stops media.</td>
</tr>
<tr>
<td></td>
<td>seek(time)</td>
<td>Seeks media to specific time.</td>
</tr>
<tr>
<td></td>
<td>play(0)</td>
<td>Pauses media.</td>
</tr>
</tbody>
</table>

| Property  | subtitle       | Supports SAMI, CineCanvas, Timed Text subtitle. |
|           | subtitleOn     | Controls subtitle display by true/false. |
|           | autoStart      | Plays media automatically. |

### Note

Log is used for checking the sequence of Web Open API; this will not be covered in the sample code description.
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.
13-15: Add an event handler which will executed when the corresponding button is pressed.
17: Declare userAgent.
18~21: Check if the device uses LG Browser.

For information on userAgent string, refer to App Development Guide.

```
01 : function initPage()
02 : {
03 :   //save page as last visited page
04 :   setLastVisitPage();
05 :
06 :   //common initialize function
07 :   commonInitialize();
08 :   requestSourceCode();
09 :   setPageID("Subtitle");
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_exit"), "click", onClickHandler);
16 :   
17 :   var userAgent = new String(navigator.userAgent);
18 :   if (userAgent != null && userAgent.search(/LG Browser/) > -1)
19 :     isLGEBrowser = true;
20 :   else
21 :     isLGEBrowser = false;
22 : }
```

Inputting Keys

The onUserInput function is called by the 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.
07: When the Red key is pressed, this code is executed.

```
01 : function onUserInput(userInput)
02 : {
03 :   switch(userInput)
04 :   {
05 :     case VK_BACK : window.location.replace("../menu_mediaPlayer.html");
06 :       break;
07 :     case VK_RED : case 82 changeTestProgress(); break;
08 :       break; 09 : }
10 : }
```

Displaying Text

Use setTestProgress function to set the text of subtitle_test_description and btn_red.

```
01 : function setTestProgress(descString, buttonString)
02 : {
03 :   setInnerTextById("subtitle_test_description", descString);
```
Controlling Play Mode and Subtitle

The following code controls video play mode and subtitle settings.

The `changeTestProgress` function controls the play mode or on/off the subtitle according to the value of `testStep`.

01: Declare and initialize `testStep`.
05-06: Declare `video` and `sourcecode`.
07: Hide `sourcecode`.
09-96: Using switch-case, control the play mode or subtitle settings according to the value of `testStep`. Also, display the text for each mode. Refer to the comments of each case.
97: Reinitialize `testStep` if the value gets bigger than the setup value.
98: Whenever the function is called, the value of `testStep` is increased by 1.

01 : var testStep = 1;
02 :
03 : function changeTestProgress()
04 : {
05 :     var video = document.getElementById("video");
06 :     var sourcecode = document.getElementById("sourcecode");
07 :     sourcecode.style.visibility = "hidden";
08 :     switch(testStep)
09 :     {
10 :         case 1 :
11 :             //Play video and check whether subtitle is displayed properly
12 :                 if(isLGEBrowser)
13 :                     {
14 :                         video.play(1);
15 :                         video.subtitle="/ApiTutorial/mediafile/cinecanvas-timing.1.dcs"
16 :                         video.style.visibility="visible"
17 :                         bodycontent.style.visibility="hidden"
18 :                         subtitle_test_description.style.visibility="visible"
19 :                     }
20 :                 jsLog.lgobject('application/x-netcast-av');
21 :                 jsLog.lgmethod('video.play(1)');
22 :                 jsLog.lgproperty('subtitle');
23 :                 setTestProgress( "Check whether subtitle is displayed in proper sync.<br>Compare subtitle with time shown at the bottom.<br>Press RED-Key to pause.", "PAUSE" );
24 :                 break;
25 :         case 2 :
26 :             //Pause video which is being played.
27 :                 if(isLGEBrowser){video.play(0);}
28 :                 jsLog.lgmethod('video.play(0)');
29 :                 setTestProgress( "The video has been paused.<br>Play the video and then check subtitle.<br>Press RED-Key to play video.", "PLAY" );
30 :                 break;
31 :         case 3 :
32 :             //play video and check subtitle
33 :                 if(isLGEBrowser){video.play(1);}
34 :                 jsLog.lgmethod('video.play(1)');
35 :                 setTestProgress( "Check whether subtitle is displayed in proper sync.<br>Compare subtitle with time shown at the bottom.<br>Press RED-Key to move to 60 sec position.", "MOVE to 60 sec position" );
36 :                 break;
case 4:
   //seek video and check subtitle
   if(isLGEBrowser){video.seek(60000);}
   jsLog.lgmethod('video.seek(time)');
   setTestProgress( "The time position has been moved to 60 sec position.<br>Check whether subtitle is displayed in proper sync.<br>Compare subtitle with time shown at the bottom.<br>Press RED-Key to stop video.", "STOP" );
   break;

case 5:
   //Stop video which is being played.
   if(isLGEBrowser){video.stop();}
   jsLog.lgmethod('video.stop()');
   setTestProgress( "The video has been stopped.<br>Restart the video and then check subtitle.<br>Press RED-Key to restart video.", "RESTART" );
   break;

case 6:
   //Restart video and check subtitle
   if(isLGEBrowser){video.play(1);}
   jsLog.lgmethod('video.play(1)');
   setTestProgress( "Check whether subtitle is displayed in proper sync.<br>Compare subtitle with time shown at the bottom.<br>Press RED-Key to turn subtitle off.", "SUBTITLE OFF" );
   break;

case 7:
   //Turn subtitle off
   if(isLGEBrowser){video.subtitleOn=false;}
   jsLog.lgproperty('subtitleOn');
   setTestProgress( "The subtitle has been turned off.<br>Check that subtitle is no longer displayed.<br>Press RED-Key to turn subtitle on.", "SUBTITLE ON" );
   break;

case 8:
   //Turn subtitle on
   if(isLGEBrowser){video.subtitleOn=true;}
   jsLog.lgproperty('subtitleOn');
   setTestProgress( "The subtitle has been turned on.<br>Check whether subtitle is displayed in proper sync.<br>Compare subtitle with time shown at the bottom.<br>Press RED-Key to change subtitle.", "SUBTITLE CHANGE TO SMI" );
   break;

case 9:
   //change subtitle to different type of files :smi
   if(isLGEBrowser)
   {
   video.subtitle="/ApiTutorial/mediafile/smi-timing.1.smi";
   } 
   jsLog.lgproperty('subtitle');
   setTestProgress( "The subtitle has been changed.<br>Check that subtitle of 'SMI file' is displayed in proper sync.<br>Compare subtitle with time shown at the bottom.<br>Press RED-Key to change subtitle.", "SUBTITLE CHANGE TO XML" );
   break;

case 10:
   //change subtitle to different type of files :xml
   if(isLGEBrowser)
video.subtitle="/ApiTutorial/mediafile/danish.xml";
}
jsLog.lgproperty('subtitle');
setTestProgress("The subtitle has been changed.<br>Check that subtitle of 'XML file' is displayed in proper sync.<br>Compare subtitle with time shown at the bottom.<br>No more test is left on this page.", "");
document.getElementById("btn_red").style.display="none";
break;
}
if(testStep == 10) testStep = 1;
testStep ++;
}

Source Code of subtitle.html
Source code of subtitle.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>Subtitle Test Page</title>
<link rel="stylesheet" href="../../css/style.css">
<script language="javascript" src="../../js/common.js"></script>
<script language="javascript" src="../../js/keycode.js"></script>
<script language="javascript" src="../../js/media.js"></script>
<link type="text/css" rel="Stylesheet" href="../../blackbirdjs/blackbird.css" />
<style type="text/css">
.OverLay { position: absolute; z-index:1; opacity: 1; filter: alpha(opacity = 100); }
body { height: 100%; }
html { height: 100%; }
</style>
<script>
//initialize page
function initPage()
// save page as last visited page
setLastVisitPage();

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

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

var userAgent = new String(navigator.userAgent);
if (userAgent != null && userAgent.search(/LG Browser/) > -1)
isLGEBrowser = true;
else
isLGEBrowser = false;
}

function onUserInput(userInput)
{
switch(userInput)
{
case VK_BACK : window.location.replace("../menu_mediaPlayer.html"); break;
case VK_RED : case 82 : changeTestProgress(); break;
}
}

function setTestProgress(descString, buttonString)
{
setInnerTextById("subtitle_test_description", descString);
setInnerTextById("btn_red", buttonString);
}

var testStep = 1;
function changeTestProgress()
{
var video = document.getElementById("video");
var sourcecode = document.getElementById("sourcecode");
sourcecode.style.visibility = "hidden";

switch(testStep)
{
case 1 :
// Play video and check whether subtitle is displayed properly
if(isLGEBrowser)
{
video.play(1);
video.subtitle="/ApiTutorial/mediafile/cinecanvas-timing.1.dcs";
video.style.visibility="visible";
bodycontent.style.visibility="hidden";
subtitle_test_description.style.visibility="visible";
}
jsLog.lgobject('application/x-netcast-av');
jsLog.lgmethod('video.play(1)');
jsLog.lgproperty('subtitle');
setTestProgress( "Check whether subtitle is displayed in proper
sync.<br>Compare subtitle with time shown at the bottom.<br>Press RED-Key to pause.">, "PAUSE"));
break;
case 2 :
    //Pause video which is being played.
    if(isLGEBrowser){video.play(0);}
    jsLog.lgmethod('video.play(0)');
    setTestProgress( "The video has been paused.<br>Play the video and then check subtitle.<br>Press RED-Key to play video.:", "PLAY");
break;
case 3 :
    //Pause video which is being played.
    if(isLGEBrowser){video.play(1);}
    jsLog.lgmethod('video.play(1)');
    setTestProgress( "Check whether subtitle is displayed in proper sync.<br>Compare subtitle with time shown at the bottom.<br>Press RED-Key to move to 60 sec position."#, "MOVE to 60 sec position");
break;
case 4 :
    //Pause video which is being played.
    if(isLGEBrowser){video.seek(60000);}
    jsLog.lgmethod('video.seek(time)');
    setTestProgress( "The time position has been moved to 60 sec position.<br>Check whether subtitle is displayed in proper sync.<br>Compare subtitle with time shown at the bottom.<br>Press RED-Key to stop video.", "STOP");
break;
case 5 :
    //Pause video which is being played.
    if(isLGEBrowser){video.stop();}
    jsLog.lgmethod('video.stop()');
    setTestProgress( "The video has been stopped.<br>Restart the video and then check subtitle.<br>Press RED-Key to restart video.", "RESTART");
break;
case 6 :
    //Pause video which is being played.
    if(isLGEBrowser){video.play(1);}
    jsLog.lgmethod('video.play(1)');
    setTestProgress( "Check whether subtitle is displayed in proper sync.<br>Compare subtitle with time shown at the bottom.<br>Press RED-Key to turn subtitle off.", "SUBTITLE OFF");
break;
case 7 :
    //Pause video which is being played.
    if(isLGEBrowser){video.subtitleOn=false;}
    jsLog.lgproperty('subtitleOn');
    setTestProgress( "The subtitle has been turned off.<br>Check that subtitle is no longer displayed.<br>Press RED-Key to turn subtitle on.", "SUBTITLE ON");
break;
case 8 :
    //Pause video which is being played.
    if(isLGEBrowser){video.subtitleOn=true;}
    jsLog.lgproperty('subtitleOn');
    setTestProgress("The subtitle has been turned on.<br>Check whether subtitle is displayed in proper sync.<br>Compare subtitle with time shown at the bottom.<br>Press RED-Key to change subtitle.", "SUBTITLE CHANGE TO SMI");
break;
case 9 :
    //Pause video which is being played.
    if(isLGEBrowser){

video.subtitle="/ApiTutorial/mediafile/smi-timing.1.smi";

jsLog.lgproperty('subtitle');
setTestProgress("The subtitle has been changed.<br>Check that subtitle of 'SMI file' is displayed in proper sync.<br>Compare subtitle with time shown at the bottom.", "SUBTITLE CHANGE TO XML");
break;
case 10 :
//change subtitle to different type of files :xml
if(isLGEBrowser)
{
    video.subtitle="/ApiTutorial/mediafile/danish.xml";
}
jsLog.lgproperty('subtitle');
setTestProgress("The subtitle has been changed.<br>Check that subtitle of 'XML file' is displayed in proper sync.<br>Compare subtitle with time shown at the bottom.<br>No more test is left on this page.", "");
document.getElementById("btn_red").style.display="none";
break;
}
if(testStep == 10) testStep = 1;
testStep ++;
</script>
</head>

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

</div>
<object
    id="video"
    type="application/x-netcast-av"
    width=1280
    height=720
    data="/ApiTutorial/mediafile/timer.mp4"
    style="float: left; z-index: 1; visibility: hidden;"
    subtitle="/ApiTutorial/mediafile/cinecanvas-timing.1.dcs"
    subtitleOn=true
    autoStart=false
    PlayCount="0">
</object>

</div>

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

<!-- navigation -->
<div class='SuiteNavigation'>
    <div style="float:left;">File : /mediaplayer/subtitle/subtitle.html</div>
</div>

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

<!-- test contents -->
<div class='ContentArea'>
</div>

</div>

<div class='ApiListTitleArea'>API List</div>
Methods

```html
<play(0)<br>
<play(1)<br>
<stop()><br>
<seek(time)<br>
```

Properties

```html
subtitle<br>
subtitleOn<br>
autoStart<br>
```

Events

```html
```

View

```html
<Press RED-Key to start test.></div>
```

Source

```html
textarea class="SourceCodeArea" value="" id='sourcecode'></textarea>
```
Executing Windows Media Player 1

This section describes how to execute the windows media player in web application.

- Initializing the Page
- Handling Media Events
- Controlling Play Mode
- Displaying Text
- Inputting Keys
- Setting Media Object
- Source Code of mediaplayer.html

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 control media and get property values. Also, occurred and used events when media status is changed are shown.

**Needed APIs**
This application uses following Web Open API:

<table>
<thead>
<tr>
<th>API Class</th>
<th>Function Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Method</td>
<td>play(0)</td>
<td>Pauses media.</td>
</tr>
<tr>
<td></td>
<td>play(1)</td>
<td>Plays media.</td>
</tr>
<tr>
<td></td>
<td>stop()</td>
<td>Stops media.</td>
</tr>
<tr>
<td></td>
<td>seek(time)</td>
<td>Seeks media to specific time.</td>
</tr>
<tr>
<td></td>
<td>mediaPlayInfo</td>
<td>Media playbackinformation</td>
</tr>
<tr>
<td>Property</td>
<td>playTime</td>
<td>Total play time of media</td>
</tr>
<tr>
<td></td>
<td>currentPosition</td>
<td>Current play position</td>
</tr>
<tr>
<td></td>
<td>bufferingProgress</td>
<td>Buffering status</td>
</tr>
<tr>
<td></td>
<td>Speed</td>
<td>Play speed</td>
</tr>
<tr>
<td></td>
<td>readyState</td>
<td>Returns current media ready status.</td>
</tr>
<tr>
<td></td>
<td>playState</td>
<td>Returns media play status by number.</td>
</tr>
<tr>
<td></td>
<td>autoStart</td>
<td>Set whether media file playout to be started automatically.</td>
</tr>
<tr>
<td></td>
<td>drm_type</td>
<td>Set drm type.</td>
</tr>
</tbody>
</table>
### API Class
<table>
<thead>
<tr>
<th>Function Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>preBufferingTime</td>
<td>Adjust buffering time before playback.</td>
</tr>
<tr>
<td>Event</td>
<td>onPlayStateChange</td>
</tr>
</tbody>
</table>

**Note**

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

---

### 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.

```javascript
function initPage() {
    // save page as last visited page
    setLastVisitPage();
    // common initialize function
    commonInitialize();
    // get source code of the page
    requestSourceCode();
    // set page ID
    setPageID("Media Player");
    // initialize Log function
    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);
    // set text of btn_red to “PAUSE”
    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');
}
```

---

### 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 of onPlayStateChange 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.

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 :}
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.

01 : var testProcess = 2;
02 :
03 : function processRedKey()
04 : {
05 :   var video = document.getElementById("video");
06 :   switch(testProcess)
07 :     {
08 :       case 0 :
09 :         setInnerTextById("btn_red", "Press RED-key to play video.");
10 :       
11 :       case 1 : 
12 :         video.play(1);
13 :         setInnerTextById("btn_red", "PAUSE");
14 :         jsLog lgmethod('video.play(1)');
15 :         break;
16 :       case 2 : 
17 :         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;
}

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.

```
function setPlayTimeInfo()
{
    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");
```
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);
}

Inputting Keys

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

06: When the Back key is pressed, 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.

Setting Media Object

The following code shows how to set Media object.

03: Set data type. Refer to App Development Guide.
04: Set the media file playout to be started automatically.
05: Set drm type.
06: Set preBufferingTime.
07-08: Set width and height.

Source Code of mediaplayer.html

Source code of mediaplayer.html is as follows:
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)";};
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.");
        break;
    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();
```javascript
//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>

<!-- 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()<br>
      </div>
    </div>
    <div class='PropertyTitleArea'>
      Properties
      <div class='PropertyListArea'>
        playTime<br>
        playPosition<br>
        bufferingProgress<br>
        speed<br>
        readyState<br>
      </div>
    </div>
  </div>
</div>
```
playState<br>
autoStart<br>
drm_type<br>
preBufferingTime<br></div><div class='EventTitleArea'>Events<br>
<div class='EventListArea'>
onPlayStateChange<br>
onReadyStateChange<br>
onBuffering<br></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>
Executing Windows Media Player 2

This section describes how to execute the windows media player in web application.

- **Initializing the Page**
- **Inputting Keys**
- **Displaying Text**
- **Setting Media Object**
- **Source Code of mediaplayer2.html**

Following the "LG Web_Executing Window Media Play #1", this application shows which properties of LG Web Open API are used to get media information when executing media player in LG Smart TV.

This application shows how to get and set property values.

![LG Smart TV SDK | Web Open API Tutorial](Image)

**Needed APIs**
This application uses following Web Open API:

<table>
<thead>
<tr>
<th>API Class</th>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Property</td>
<td>playState</td>
<td>Returns current media play status by number.</td>
</tr>
<tr>
<td></td>
<td>version</td>
<td>Returns media version by String type.</td>
</tr>
<tr>
<td></td>
<td>type</td>
<td>Returns media type by String type.</td>
</tr>
<tr>
<td></td>
<td>data</td>
<td>Returns media URL by String type.</td>
</tr>
<tr>
<td></td>
<td>width / height</td>
<td>Returns media object size by String type.</td>
</tr>
<tr>
<td></td>
<td>isScannable</td>
<td>If returned value is true, fast forward or rewind is available.</td>
</tr>
</tbody>
</table>

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

**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.
Add an event handler which will execute when the corresponding button is pressed.

Call function that displays property values on screen.

```javascript
function initPage()
{
  // save page as last visited page
  setLastVisitPage();
  // common initialize function
  commonInitialize();
  requestSourceCode();
  setPageID("MediaPlayer");
  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);

  processTest();
}

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

var oldPlayState;

function processTest()
{
  var video = document.getElementById("video");
  jsLog.lgproperty('playState');
  // check if video is now being played
```
if((video.playState != 1) && (oldPlayState != video.playState))
{
oldPlayState = video.playState;
setInterval(processTest, 4000);
return;
}

//version
jsLog.lgproperty('version');
setInnerTextById("version_value", "1. version : " + video.version);

//type
jsLog.lgproperty('type');
setInnerTextById("type_value", "2. type : " + video.type);

//data
jsLog.lgproperty('data');
setInnerTextById("data_value", "3. data : " + video.data);

//width x height
jsLog.lgproperty('width');
jsLog.lgproperty('height');
setInnerTextById("width_height_value", "4. width x height : " + video.width + " x " + video.height);

//isScannable
jsLog.lgproperty('isScannable');
setInnerTextById("isScannable_value", "6. Check \"isScannable\" : " + video.isScannable);
}

Setting Media Object
The following code shows how to set Media object.

03: Set data type. Refer to App Development Guide.
04-05: Set width and height.
06: Set the path of media file which will be played.

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>

Source Code of mediaplayer2.html
Source code of mediaplayer2.html is as follows:

"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(2/3)</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>"
//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);

    processTest();

    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 : window.location.replace("./mediaplayer.html"); break;
        case VK_GREEN : case 71 : window.location.replace("./mediaplayer3.html"); break;
    }
}

var oldPlayState;
function processTest() {
    var video = document.getElementById("video");
    jsLog.lgproperty('playState');

    //check if video is now being played
    if((video.playState != 1) && (oldPlayState != video.playState)) {
        oldPlayState = video.playState;
        setInterval(processTest, 4000);
        return;
    }

    //version
    jsLog.lgproperty('version');
    setInnerTextById("version_value", "1. version : " + video.version);
//type
jsLog.lgproperty('type');
setInnerTextById("type_value", "2. type : " + video.type);

//data
jsLog.lgproperty('data');
setInnerTextById("data_value", "3. data : " + video.data);

//width x height
jsLog.lgproperty('width');
jsLog.lgproperty('height');
setInnerTextById("width_height_value", "4. width x height : " + video.width + " x " + video.height);

//isScannable
jsLog.lgproperty('isScannable');
setInnerTextById("isScannable_value", "6. Check \"isScannable\" : " + video.isScannable);
}
<div id='view'>
  <div class='ViewArea'>
  <![CDATA[
  <!-- video object -->
  <object
  id="video"
  type="application/x-netcast-av"
  width=300
  height=250
  data="/ApiTutorial/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="40px">
      <td width="100%" colspan="2" id="version_value">1. version :</td>
    </tr>
    <tr height="40px">
      <td width="100%" id="type_value">2. type :</td>
    </tr>
    <tr height="40px">
      <td width="100%" id="data_value">3. data :</td>
    </tr>
    <tr height="40px">
      <td width="100%" id="width_height_value">4. width x height :</td>
    </tr>
    <tr height="40px">
      <td width="100%" id="isScannable_value">5. Check "isScannable" :</td>
    </tr>
  </table>
  ]]>
</div>
</div>

<!-- description -->
<textarea class='SourceCodeArea' id='sourcecode'><?xml version="1.0" encoding="ISO-8859-1"?>
  <videoobject width="450" height="250" type="application/x-netcast-av"/>
  <table border="0" cellpadding="0" cellspacing="0" style="position: relative; left: 10px; width:450px; height:250px;">
    <tr height="40px">
      <td width="100%" id="version_value">1. version :</td>
    </tr>
    <tr height="40px">
      <td width="100%" id="type_value">2. type :</td>
    </tr>
    <tr height="40px">
      <td width="100%" id="data_value">3. data :</td>
    </tr>
    <tr height="40px">
      <td width="100%" id="width_height_value">4. width x height :</td>
    </tr>
    <tr height="40px">
      <td width="100%" id="isScannable_value">5. Check "isScannable" :</td>
    </tr>
  </table>
</textarea>
</div>

<!-- description -->
Check whether video is being played and retrieved values are correct.

<!-- button and copyright -->
<div class='SuiteButtonArea'>
  <!-- back key description -->
  <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'>PREVIOUS PAGE</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>
Executing Windows Media Player 3

This section describes how to activate and deactivate browser page loading icon.

- Initializing the Page
- Inputting Keys
- Displaying Text
- Setting Media Object
- Source Code of mediaplayer3.html

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.

Needed APIs
This application uses following Web Open API:

<table>
<thead>
<tr>
<th>API Class</th>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Method</td>
<td>play()</td>
<td>Plays media.</td>
</tr>
<tr>
<td></td>
<td>stop()</td>
<td>Stops media.</td>
</tr>
<tr>
<td>Property</td>
<td>data</td>
<td>Returns media URL with String type.</td>
</tr>
<tr>
<td></td>
<td>error</td>
<td>Returns error code if error occurs during media play.</td>
</tr>
<tr>
<td>Event</td>
<td>onError</td>
<td>The event occurs when an error occurs during media play.</td>
</tr>
<tr>
<td></td>
<td>play()</td>
<td>Plays media.</td>
</tr>
</tbody>
</table>

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

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: Call function that displays property values on screen.
21: Add onError event handler of media.

```javascript
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 :   var video = document.getElementById("video");
19 :
20 :   // add onError handler
21 :   video.onError = processError;
22 :
23 :   jsLog.lgobject('application/x-netcast-av');
24 : }
```

### Inputting Keys

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

06: When the Back key is pressed, 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.

```javascript
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 : window.location.replace("./mediaplayer.html"); break;
07 :     case VK_GREEN : case 71 : window.location.replace("./mediaplayer3.html"); break;
08 :   }
09 : }
```

### 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.
If the value of currentStep is bigger than the size of array the function ends.

Declare media.

Stop media.

Displays error code, description, and result error code of errorsToTest on screen.

Set text of test_result class.

If the value of currentStep equals the size of errorsToTest array, hide the Next button and displays description text on screen.

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.

Returns if video is not played.

Set error property of value to errorCode and displays returned number.

```
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 :     setInnerTextById("error_test_description", "Check whether expected error is occurred.<br>Press Red-key to test next error code.");
18 :     setInnerTextById("btn_red", "NEXT");
19 :   }  
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 :   setInnerTextById("error_test_description", "Check whether expected error is occurred.<br>No more test is left on this page.");
35 :   document.getElementById("btn_red").style.visibility = "hidden";
36 :  
37 :  
```
Setting Media Object

The following code shows how to set Media object.

03: Set data type. Refer to App Development Guide.
04-05: Set width and height.
06: Set the path of media file which will be played.

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>

Source Code of mediaplayer3.html

Source code of mediaplayer3.html is as follows:

```html
<!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/media.js"></script>
<script type="text/javascript" src="../blackbirdjs/blackbird.js"></script>
<link type="text/css" rel="stylesheet" href="../blackbirdjs/blackbird.css" />

//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)
<div class='SuiteTitleLine'> </div>
</div><!-- test contents -->
<div class='ContentArea ' id='content_body'>
<div class='ApiListTitleArea'>API List</div>
<div class='ApiListArea'>
<div class='MethodTitleArea'>Methods</div>
<div class='MethodListArea'>
stop()<br>
play()<br>
</div>
</div>
<div class='PropertyTitleArea'>Properties</div>
<div class='PropertyListArea'>
data<br>
error<br>
</div>
</div>
<div class='EventTitleArea'>Events</div>
<div class='EventListArea'>
onError<br>
</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" 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>
<td width="60%" colspan="2"><div class="eachTestGuide ">Error Description</div></td>
</tr>
</table>
</div>
</div>
<table>
<thead>
<tr>
<th>Result</th>
<th>Error Code</th>
<th>Test Result</th>
</tr>
</thead>
</table>

Press Red key to test error code.

BACK
EXIT
TEST
PREVIOUS PAGE
Copyright LG Electronics
Generating Outofmemory Event

This section describes how to generate outofmemory event in web application.

- Initializing the Page
- Inputting Keys
- Loading Images
- Displaying Information of Loaded Image and Event
- Handling Events
- Displaying usedMemorySize
- Source Code of outofmemory.html

This application generates outofmemory event by loading image and increasing memory size. You can know how to generate outofmemory event and handle it through this application.

### Needed APIs

This application uses following Web Open API:

<table>
<thead>
<tr>
<th>API Class</th>
<th>Function Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Method</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>Property</td>
<td>NetCastGetUsedMemorySize</td>
<td>Total memory size used by the browser process</td>
</tr>
<tr>
<td>Event</td>
<td>outofmemory</td>
<td>This event gives the remaining memory size to application author.</td>
</tr>
</tbody>
</table>

**Note**

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

### Initializing the Page

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

04: Records the last visited page when running the application.
07: Initializes the page.
08: Gets the source code of the page using the XMLHttpRequest object.
09: Sets the page ID.
10: Initializes the Log function.
13-17: Registers an event handler which will executed when the corresponding button is pressed.
19: Calls displayStatus function.
22: Registers an event handler which will be executed when the outofmemory event occurs.
24: Displays the memory size used by the application on screen.
function initPage()
{
    // save page as last visited page
    setLastVisitPage();
    // common initialize function
    commonInitialize();
    requestSourceCode();
    setPageID("Out Of Memory");
    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_yellow"),"click", onClickHandler);
    addEventHandler(document.getElementById("btn_exit"),"click", onClickHandler);
    displayStatus();
    // add outofmemory event handler
    addEventHandler(window, "outofmemory", outOfMemoryHandler);
    getUsedMemorySize();
}

Inputting Keys
The onUserInput function is called by the 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.

06-07: When the Back key is pressed, this code is executed.
08: When the Red key is pressed, one image is loaded.
09: When the Green key is pressed, five images are loaded.
10: When the Yellow key is pressed, ten images are loaded.

function onUserInput(userInput)
{
    switch(userInput)
    {
        case VK_BACK:
            window.location.replace("../menu_netcast.html"); break;
        case VK_RED : case 82 : loadMoreImages(1); break;
        case VK_GREEN : case 71 : loadMoreImages(5); break;
        case VK_YELLOW : case 89 : loadMoreImages(10); break;
    }
}

Loading Images
The following functions load the specified number of images.

loadMoreImages(numberToLoad)
This function is called when multiple images are loaded.
04: Calls loadOneImage() function as the number of images.

loadOneImage:
Loads one image.
20-21: Returns if there is no image to load
24-25: Specifies size of the image to load.
Specifies the position that the image is loaded on the screen.
Specifies the image to load
Creates element of div tag.

Increases number of images by 1.
Increases number of images by 1.
Calls displayStatus() function.

```javascript
function loadMoreImages(numberToLoad)
{
    for(var i = 0 ; i < numberToLoad ; i++)
        loadOneImage();
}

var numberOfLoadedImages = 0;
var EACH_IMG_SIZE = 2880054;
var DEFAULT_LEFT = 0;
var DEFAULT_TOP = 0;
var DEFAULT_IMG_LEFT = 100;
var DEFAULT_IMG_TOP = 500;
var TOTAL_WIDTH = 900;
var TOTAL_HEIGHT = 50;
var MAX_COLUMN_INDEX = 50;
var MAX_ROW_INDEX = 2;

function loadOneImage()
{
    if(MAX_COLUMN_INDEX * MAX_ROW_INDEX == numberOfLoadedImages)
        return;
    
    //create div and set an image as background
    var widthToSet = Math.floor(TOTAL_WIDTH / MAX_COLUMN_INDEX);
    var heightToSet = Math.floor(TOTAL_HEIGHT / MAX_ROW_INDEX);
    var leftToSet = Math.floor((TOTAL_WIDTH / MAX_COLUMN_INDEX) *
        (numberOfLoadedImages % MAX_COLUMN_INDEX));
    var topToSet = Math.floor((TOTAL_HEIGHT / MAX_ROW_INDEX) *
        Math.floor(numberOfLoadedImages / MAX_COLUMN_INDEX));

    var bgUrlToSet = getMediaFileUrl("memoryTestImages") + "lake-nature_"
        + bgImgIndex + ".bmp";

    var divToAppend = document.createElement("div");
    divToAppend.style.position = "absolute";
    divToAppend.style.width = (widthToSet - 2) + "px";
    divToAppend.style.height = (heightToSet - 2) + "px";
    divToAppend.style.border = "1px solid #000000";
    divToAppend.style.left = leftToSet + "px";
    divToAppend.style.top = topToSet + "px";
    divToAppend.style.background = "url(" + bgUrlToSet + ");
    divToAppend.style.backgroundPosition = "+" + (DEFAULT_IMG_LEFT + leftToSet) +
        "px -" + (DEFAULT_IMG_TOP + topToSet) + "px";
    divToAppend.style.backgroundRepeat = "no-repeat";
    document.getElementById("ViewArea").appendChild(divToAppend);
    numberOfLoadedImages++;
    displayStatus();
```
Displaying Information of Loaded Image and Event

The following functions are for event handlers added by the initPage function.

**displayStatus:**

Displays information regarding the loaded image.
03: Displays the number of loaded images on the screen.
05: Displays size of loaded images on the screen.

**toCommaNotation(origString)**

Inserts commas in numbers.
15-22: If the number of digits is greater than three, comma is inserted in number for each three digits, which is saved in resultString.

```javascript
01 : function displayStatus()
02 : {
03 :     setInnerTextById("number_of_loaded_image_result", numberOfLoadedImages + "/" + (MAX_COLUMN_INDEX * MAX_ROW_INDEX));
04 :     var totalSize = EACH_IMG_SIZE * numberOfLoadedImages;
05 :     setInnerTextById("total_image_size_result", toCommaNotation(String(totalSize)) + " bytes");
06 : }
07 :
08 :
09 : function toCommaNotation(origString)
10 : {
11 :     var unitToSep = 3;
12 :     var resultString = ";
13 :     var processString = origString;
14 :     while(processString.length > unitToSep)
15 :         {
16 :             var cutIndex = processString.length - unitToSep;
17 :             var resultString = "," + processString.substring(cutIndex) + resultString;
18 :             processString = processString.substring(0, cutIndex);
19 :         }
20 :     resultString = processString + resultString;
21 :     return resultString;
22 : }
23 : }

51 : function loadMoreImages(numberToLoad)
52 : {
53 :     for(var i = 0 ; i < numberToLoad ; i++)
54 :         loadOneImage();
55 : }
56 :
57 : var numberOfLoadedImages = 0;
58 : var EACH_IMG_SIZE = 2880054;
59 : var DEFAULT_LEFT = 0;
60 : var DEFAULT_TOP = 0;
61 : var DEFAULT_IMG_LEFT = 100;
62 : var DEFAULT_IMG_TOP = 500;
63 : var TOTAL_WIDTH = 900;
64 : var TOTAL_HEIGHT = 50;
65 : var MAX_COLUMN_INDEX = 50;
66 : var MAX_ROW_INDEX = 2;
67 :
68 : function loadOneImage()
```
70: if (MAX_COLUMN_INDEX * MAX_ROW_INDEX == numberOfLoadedImages)
71:     return;
72:
73: //create div and set an image as background
74: var widthToSet = Math.floor(TOTAL_WIDTH / MAX_COLUMN_INDEX);
75: var heightToSet = Math.floor(TOTAL_HEIGHT / MAX_ROW_INDEX);
76: var leftToSet = Math.floor((TOTAL_WIDTH / MAX_COLUMN_INDEX) * (numberOfLoadedImages % MAX_COLUMN_INDEX));
77: var topToSet = Math.floor((TOTAL_HEIGHT / MAX_ROW_INDEX) * Math.floor(numberOfLoadedImages / MAX_COLUMN_INDEX));
78: var bgImgIndex = "";
79:
80: if (String(numberOfLoadedImages).length == 1){bgImgIndex = "00";}
81: else if (String(numberOfLoadedImages).length == 2){bgImgIndex = "0";}
82: else if (String(numberOfLoadedImages).length == 3){bgImgIndex = "";}
83:
84: bgImgIndex += String(numberOfLoadedImages);
85: var bgUrlToSet = getMediaFileUrl("memoryTestImages") + "lake-nature_" + bgImgIndex + ".bmp";
86: var divToAppend = document.createElement("div");
87:
88: divToAppend.style.position = "absolute";
89: divToAppend.style.width = (widthToSet - 2) + "px";
90: divToAppend.style.height = (heightToSet - 2) + "px";
91: divToAppend.style.border = "1px solid #000000";
92: divToAppend.style.left = leftToSet + "px";
93: divToAppend.style.top = topToSet + "px";
94: divToAppend.style.background = "url(" + bgUrlToSet + ")";
95: divToAppend.style.backgroundPosition = "-" + (DEFAULT_IMG_LEFT + leftToSet) + "px -" + (DEFAULT_IMG_TOP + topToSet) + "px";
96: divToAppend.style.backgroundRepeat = "no-repeat";
97: document.getElementById("ViewArea").appendChild(divToAppend);
98: numberOfLoadedImages++;
99: displayStatus();
100: }

Handling Events
The following function is for event handler, which handles outofmemory event.
outOfMemoryHandler is called when the outofmemory event occurs.

07: Increases eventCount by 1.
09-14: Saves counted value and event.available (size of usable memory) value to resultString and displays the values on the screen.

01: var resultString = " ";
02: var eventCount = 0;
03:
04: function outOfMemoryHandler()
05: {
06:     jsLog.lgevent('outofmemory');
07:     eventCount++;
08:
09:     if (eventCount == 1)
10:         resultString += eventCount + " - " + event.available + " MB";
11:     else
12:         resultString += "/ " + eventCount + " - " + event.available + " MB";
13:     setInnerTextById("out_of_memory_event_occurred", resultString);
14: }
Displaying usedMemorySize

The following function is for displaying the memory size

01: Set usedMemorySize.
04-09: If the return value of NetCastGetUsedMemorySize exists, display the value.

```
01 : var usedMemorySize;
02 : function getUsedMemorySize()
03 : {
04 :  if(window.NetCastGetUsedMemorySize)
05 :  {
06 :    usedMemorySize = window.NetCastGetUsedMemorySize();
07 :    setInnerTextById("used_memory_result", usedMemorySize);
08 :    jsLog.lgproperty('NetCastGetUsedMemorySize');
09 :  }
10 : }
```

Source Code of outofmemory.html

Source code of outofmemory.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>OutOfMemory Event Test Page</title>
<link rel="stylesheet" href="../../css/style.css">
<script language="javascript" src="../../js/keycode.js"></script>
<script language="javascript" src="../../js/media.js"></script>
<script language="javascript" src="../../js/common.js"></script>
<script language="javascript" src="../../js/menu.js"></script>
<link type="text/css" rel="Stylesheet" href="../blackbirdjs/blackbird.css" />
<script>
  //initialize page
  function initPage()
  {
    //save as last visited page
    setLastVisitPage();

    //common initialize function
    commonInitialize();
    requestSourceCode();
    setPageID("Out Of Memory");
    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_yellow"), "click", onClickHandler);
    addEventHandler(document.getElementById("btn_exit"), "click", onClickHandler);
    displayStatus();
  }
```

//add outofmemory event handler
addEventHandler(window, "outofmemory", outOfMemoryHandler);

getUsedMemorySize();
}

function onUserInput(userInput)
{
    switch(userInput)
    {
        case VK_BACK : window.location.replace("../menu_netcast.html"); break;
        case VK_RED : case 82 : loadMoreImages(1); break;
        case VK_GREEN : case 71 : loadMoreImages(5); break;
        case VK_YELLOW : case 89 : loadMoreImages(10); break;
    }
}

function loadMoreImages(numberToLoad)
{
    for(var i = 0 ; i < numberToLoad ; i++)
        loadOneImage();
}

var numberOfLoadedImages = 0;
var EACH_IMG_SIZE = 2880054;
var DEFAULT_LEFT = 0;
var DEFAULT_TOP = 0;
var DEFAULT_IMG_LEFT = 100;
var DEFAULT_IMG_TOP = 500;
var TOTAL_WIDTH = 900;
var TOTAL_HEIGHT = 50;
var MAX_COLUMN_INDEX = 50;
var MAX_ROW_INDEX = 2;

function loadOneImage()
{
    if(MAX_COLUMN_INDEX * MAX_ROW_INDEX == numberOfLoadedImages)
        return;

    //create div and set an image as background
    var widthToSet = Math.floor(TOTAL_WIDTH / MAX_COLUMN_INDEX);
    var heightToSet = Math.floor(TOTAL_HEIGHT / MAX_ROW_INDEX);
    var leftToSet = Math.floor((TOTAL_WIDTH / MAX_COLUMN_INDEX) *
        (numberOfLoadedImages % MAX_COLUMN_INDEX));
    var topToSet = Math.floor((TOTAL_HEIGHT / MAX_ROW_INDEX) *
        Math.floor(numberOfLoadedImages / MAX_COLUMN_INDEX));

    var bgImgIndex = ";";

    if(String(numberOfLoadedImages).length == 1){bgImgIndex = "00";}
    else if(String(numberOfLoadedImages).length == 2){bgImgIndex = "0";}
    else if(String(numberOfLoadedImages).length == 3){bgImgIndex = "";}

    bgImgIndex += String(numberOfLoadedImages);
    var bgUrlToSet = getMediaFileUrl("memoryTestImages") + "lake-nature_" +
        bgImgIndex + ".bmp";

    var divToAppend = document.createElement("div");
divToAppend.style.position = "absolute";
divToAppend.style.width = (widthToSet - 2) + "px";
divToAppend.style.height = (heightToSet - 2) + "px";
divToAppend.style.border = "1px solid #000000";
divToAppend.style.left = leftToSet + "px";
divToAppend.style.top = topToSet + "px";
divToAppend.style.background = "url(" + bgUrlToSet + ")";
divToAppend.style.backgroundPosition = "-" + (DEFAULT_IMG_LEFT + leftToSet) + "px" + (DEFAULT_IMG_TOP + topToSet) + "px";
divToAppend.style.backgroundRepeat = "no-repeat";
document.getElementById("ViewArea").appendChild(divToAppend);
numberOfLoadedImages++;
displayStatus();
}

function displayStatus()
{
    setInnerTextById("number_of_loaded_image_result", numberOfLoadedImages + "/" + (MAX_COLUMN_INDEX * MAX_ROW_INDEX));
    var totalSize = EACH_IMG_SIZE * numberOfLoadedImages;
    setInnerTextById("total_image_size_result", toCommaNotation(String(totalSize)) + " bytes");
}

function toCommaNotation(origString)
{
    var unitToSep = 3;
    var resultString = ";
    var processString = origString;

    while(processString.length > unitToSep)
    {
        var cutIndex = processString.length - unitToSep;
        var resultString = "," + processString.substring(cutIndex) + resultString;
        processString = processString.substring(0, cutIndex);
    }
    resultString = processString + resultString;
    return resultString;
}

var resultString = ";
var eventCount = 0;

function outOfMemoryHandler()
{
    jsLog.lg(event('outofmemory'));
    eventCount++;

    if(eventCount == 1)
        resultString += eventCount + "," + event.available + " MB"
    else
        resultString += "/" + eventCount + "," + event.available + " MB"

    setInnerTextById("out_of_memory_event_occurred", resultString);
}

var usedMemorySize;
function getUsedMemorySize()

```javascript
if (window.NetCastGetUsedMemorySize) {
    usedMemorySize = window.NetCastGetUsedMemorySize();
    setInnerTextById("used_memory_result", usedMemorySize);
    jsLog-lgproperty('NetCastGetUsedMemorySize');
}
```

<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 : netcast/app/outofmemory.html</div>
</div>

<!-- test contents -->
<div class="ContentArea">
    <div class="ApiListTitleArea">Web Open API List</div>
    <div class="ApiListArea">
        <div class="MethodTitleArea">
            Methods
        </div>
        <div class="PropertyTitleArea">
            Properties
            <div class="PropertyListArea">
                NetCastGetUsedMemorySize
            </div>
        </div>
        <div class="EventTitleArea">
            Events
            <div class="EventListArea">
                outofmemory
            </div>
        </div>
    </div>
    <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 class="ViewArea id='ViewArea' style="position: relative;">
    <table border="0" cellpadding="0" cellspacing="0">
        <tr height="50px">
```

```
<table>
<thead>
<tr>
<th>1. Each image size</th>
<th>2,880,054 bytes</th>
</tr>
</thead>
<tbody>
<tr>
<td>2. Number of loaded images</td>
<td>div id=&quot;number_of_loaded_image_result&quot;</td>
</tr>
<tr>
<td>3. Total image size</td>
<td>div id=&quot;total_image_size_result&quot;</td>
</tr>
<tr>
<td>4. OutOfMemory Event</td>
<td>div id=&quot;out_of_memory_event_occurred&quot;</td>
</tr>
<tr>
<td>5. Used Memory Size</td>
<td>div id=&quot;used_memory_result&quot;</td>
</tr>
</tbody>
</table>

Press Red / Green / Yellow key to load more image(s)

Check if "outofmemory" event occurs and available memory is acceptable.
<!-- yellow key description -->
<div id='btn_yellow' class='buttonDescription yellowColor'>LOAD 10</div>

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

</body>
</html>
Getting and Displaying Device Information

This section describes how to get and display the device information in web application.

- Initializing the Page
- Inputting Keys
- Getting & Displaying Device Information
- Setting Device Object
- Source Code of deviceinfo01.html

This application shows how to get LG Smart TV information using properties of LG Web Open API.

**Needed APIs**

This application uses following Web Open API:

<table>
<thead>
<tr>
<th>API Class</th>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Property</td>
<td>version</td>
<td>Version of Device Info Plugin.</td>
</tr>
<tr>
<td></td>
<td>manufacturer</td>
<td>Manufacturer ID (LGE)</td>
</tr>
<tr>
<td></td>
<td>modelName</td>
<td>Device model name</td>
</tr>
<tr>
<td></td>
<td>serialNumber</td>
<td>Device serial number</td>
</tr>
<tr>
<td></td>
<td>swVersion</td>
<td>Software version</td>
</tr>
<tr>
<td></td>
<td>hwVersion</td>
<td>Hardware version</td>
</tr>
<tr>
<td></td>
<td>osdResolution</td>
<td>OSD resolution (Returns in number)</td>
</tr>
<tr>
<td></td>
<td>preferredSubtitleLanguage</td>
<td>Subtitle language set in TV</td>
</tr>
<tr>
<td></td>
<td>preferredSubtitleStatus</td>
<td>On/off status of subtitle</td>
</tr>
</tbody>
</table>

**Note**

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

**Initializing the Page**

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

04: Record the last visited page when running the application.
07: Initialize the page.
08: Get the source code of the page using the XMLHttpRequest object.
09: Initialize the Log function.
12-14: Add an event handler which will executed when the corresponding button is pressed.
18: Display device information on screen by calling getDeviceInfo() method.

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

**Inputting Keys**

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

06: When the Back key is pressed, this code is executed.
08: When the Green key is pressed, this code is executed.

```
01: function onUserInput(userInput)  
02: {  
03:   switch(userInput)  
04:   {  
05:     case VK_BACK :  
06:       window.location.replace("./main_menu.html");  
07:       break;  
08:     case VK_GREEN :  
09:       window.location.replace("./deviceinfo02.html");  
10:       break;  
11:   }  
12: }  
```

**Getting & Displaying Device Information**

The following code shows how to get and display device information of LG Smart TV.

`getDeviceInfo`

This method displays the information got from the device when the application is started.

03: Declare device.
05: Call `getDeviceVersionInfo()` method to display version information.
07: Call `setInnerTextById` method to display manufacturer information.
10: Call `setInnerTextById` method to display modelName information.
13: Call `setInnerTextById` method to display serialNumber information.
16: Call `setInnerTextById` method to display swVersion information.
19: Call `setInnerTextById` method to display hwVersion information.
22: Call `setInnerTextById` method to display OSDResolution information.
25: Call `setInnerTextById` method to display preferredSubtitleLanguage information.
28: Call `setInnerTextById` method to display preferredAudioLanguage information.
31: Call `setInnerTextById` method to display preferredSubtitleStatus information.
**getDeviceVersionInfo**

This method is called in getDeviceInfo method and display the device version information.

38: Declare device.
40: Check the device version.
42: Call setInnerTextById method to display version information.

```javascript
01 : function getDeviceInfo()
02 : {
03 : var device = document.getElementById('device');
04 :
05 : getDeviceVersionInfo();
06 :
07 : setInnerTextById("manufacturer_value","manufacturer:" + device.manufacturer);
08 : jsLog.lgproperty('manufacturer');
09 :
10 : setInnerTextById("modelName_value","modelName : " + device.modelName);
11 : jsLog.lgproperty('modelName');
12 :
13 : setInnerTextById("serialNumber_value","serialNumber:" + device.serialNumber);
14 : jsLog.lgproperty('serialNumber');
15 :
16 : setInnerTextById("swVersion_value","swVersion : " + device.swVersion);
17 : jsLog.lgproperty('swVersion');
18 :
19 : setInnerTextById("hwVersion_value","hwVersion : " + device.hwVersion);
20 : jsLog.lgproperty('hwVersion');
21 :
22 : setInnerTextById("osdResolution_value","osdResolution:" + device.osdResolution);
23 : jsLog.lgproperty('osdResolution');
24 :
25 : setInnerTextById("preferredSubtitleLanguage_value","preferredSubtitleLanguage : " + device.preferredSubtitleLanguage);
26 : jsLog.lgproperty('preferredSubtitleLanguage');
27 :
28 : setInnerTextById("preferredAudioLanguage_value","preferredAudioLanguage : " + device.preferredAudioLanguage);
29 : jsLog.lgproperty('preferredAudioLanguage');
30 :
31 : setInnerTextById("preferredSubtitleStatus_value","preferredSubtitleStatus : " + device.preferredSubtitleStatus);
32 : jsLog.lgproperty('preferredSubtitleStatus');
33 :
34 : }
35 :
36 : function getDeviceVersionInfo()
37 : {
38 : var device = document.getElementById('device');
39 :
40 : if(device.version)
41 : {
42 : setInnerTextById("version_value","version : " + device.version);
43 : jsLog.lgproperty('version');
44 : }
45 : }
```

**Setting Device Object**

The following code shows how to set Media object.

03: Set deviceinfo type. Refer to App Development Guide.
04-05: Set width and height.
Source Code of deviceinfo01.html

Source code of deviceinfo01.html is as follows:

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

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

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

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

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

function onUserInput(userInput)
{
  switch(userInput)
  {
    case VK_BACK :
      window.location.replace("../main_menu.html");
      break;
    case VK_GREEN :
      window.location.replace("./deviceinfo02.html");
      break;
  }
```
function getDeviceInfo()
{
    var device = document.getElementById('device');

    getDeviceVersionInfo();

    setInnerTextById("manufacturer_value", "manufacturer : " +
    device.manufacturer);
    jsLog.lgproperty('manufacturer');

    setInnerTextById("modelName_value", "modelName : " + device.modelName);
    jsLog.lgproperty('modelName');

    setInnerTextById("serialNumber_value", "serialNumber : " +
    device.serialNumber);
    jsLog.lgproperty('serialNumber');

    setInnerTextById("swVersion_value", "swVersion : " + device.swVersion);
    jsLog.lgproperty('swVersion');

    setInnerTextById("hwVersion_value", "hwVersion : " + device.hwVersion);
    jsLog.lgproperty('hwVersion');

    setInnerTextById("osdResolution_value", "osdResolution : " +
    device.osdResolution);
    jsLog.lgproperty('osdResolution');

    setInnerTextById("preferredSubtitleLanguage_value",
    "preferredSubtitleLanguage : " + device.preferredSubtitleLanguage);
    jsLog.lgproperty('preferredSubtitleLanguage');

    setInnerTextById("preferredAudioLanguage_value", "preferredAudioLanguage : " +
    device.preferredAudioLanguage);
    jsLog.lgproperty('preferredAudioLanguage');

    setInnerTextById("preferredSubtitleStatus_value", "preferredSubtitleStatus : " +
    device.preferredSubtitleStatus);
    jsLog.lgproperty('preferredSubtitleStatus');
}

function getDeviceVersionInfo()
{
    var device = document.getElementById('device');

    if(device.version)
    {
        setInnerTextById("version_value", "version : " + device.version);
        jsLog.lgproperty('version');
    }
}
</script>
</head>
<body ondragstart='return false' onselectstart='return false'
onload="javascript:initPage();" >
API List

Methods

Properties
version
manufacturer
modelName
serialNumber
swVersion / hwVersion
osdResolution
preferredSubtitleLanguage
preferredAudioLanguage
preferredSubtitleStatus

Events

View

Source

object

type="application/x-netcast-info"

id="device"

width="0"

height="0"

style="float: left">
</object>

| 159 |
|-----------|----------------|-------------|---------------|-------------|-------------|----------------|---------------------------|---------------------------|-------------------------|
|           |                |             |               |             |             |               |                           |                           |                         |
|           |                |             |               |             |             |               |                           |                           |                         |
Handling Key Inputs Using JavaScript Events

This section describes how to handle the key inputs using JavaScript events.

- **Initializing the Page**
- **Handling Events**
- **Displaying Text**
- **Executing Test**
- **Source Code of keydownuppress.html**

This application shows how to use JavaScript events for general key inputs in applications. When receiving a key from IR Remote, you can check and control the received key using JavaScript event.

```
// Declare the variable.
01-15: Declare keysToTestArray and set key names and values.

initPage
This function initializes the page.
21: Record the last visited page when running the application.
24: Initialize the page.
25: Get the source code of the page using the XMLHttpRequest object.
26: Set the page ID.
27: Initialize the Log function.
30-31: Add an event handler which will executed when the corresponding button is pressed.
34: Add keydown event handler which will executed when menu keys (Back, Red, Green, and Yellow buttons) are pressed.
37: Call setKeysOnScreen() function.
40: Call setTimeout function, then, call initTest() function after 700 milliseconds.
42: Declare device.
43-46: Check device and determine whether to display portal key or not.

01: var keysToTestArray = ["div_01", "OK", 13, "d_p_u"]
02: , ["div_02", "PLAY", 415, "d_u"], ["div_03", "PAUSE", 19, "d_u"]
03: , ["div_04", "STOP", 413, "d_u"], ["div_05", "REWIND", 412, "d_u"]
04: , ["div_06", "FAST_FWD", 417, "d_u"], ["div_07", "PAGE_UP", 33, "d_u"]
05: , ["div_08", "PAGE_DOWN", 34, "d_u"], ["div_09", "LEFT", 37, "d_u"]
06: , ["div_10", "RIGHT", 39, "d_u"], ["div_11", "UP", 38, "d_u"]
07: , ["div_12", "DOWN", 40, "d_u"], ["div_13", "NUMBER 0", 48, "d_p_u"]
```

Press OK-Key to test Key-Events and Key-Code.
function initPage()
{
    //save page as last visited page
    setLastVisitPage();

    //common initialize function
    commonInitialize();

    requestSourceCode();

    setPageID("Key Down/Up/Press");

    jsLog.initLG();

    //add onclick event handler
    addEventHandler(document.getElementById("btn_back"), "click", backHandler);
    addEventHandler(document.getElementById("btn_reset"), "click", resetHandler);

    //add event handler for menu
    addEventHandler(document.body, "keydown", menuKeyHandler);

    //set keys to test
    setKeysOnScreen();

    //add event handler to test
    setTimeout(initTest, 700);

    var device = document.getElementById("device");
    if(! device.supportPortalKey)
    {
        document.getElementById("div_29").style.visibility = "hidden";
    }
}

Handling Events
The following code shows event handler functions which are added from initPage() function.

initTest
This function adds handlers of key events.
03: Add keydown event handler.
04: Add keyup event handler.
05: Add keypress event handler.
06: Call initTestProgress() function.

backHandler
When the Back button is pressed, the page moves to the assigned location.

resetHandler
When the Reset button is pressed, Key test is initialized by calling initTestProgress() function.

menuKeyHandler
This function is keydown event handler and executes corresponding function for each menu button.
19-21: Declare and initialize backToYellow, resetToGreen, and hasMenuExecuted. These variables are used to distinguish the test key and bottom menu.
25: Declare userInput and set key code of the event.
26-32: Using switch-case, execute the corresponding function for received userInput and set hasMenuExecuted to true.
28: When Back key is pressed and the Back button is not yellow, call backHandler() function.
29: When Red key is pressed and the Reset button is not green, call resetHandler() function.
30: When Green key is pressed and the Reset button is green, call resetHandler() function.
31: When Yellow key is pressed and the Back button is yellow, call backHandler() function.

**testKeyDown**

This function is a handler that is executed when keydown event occurs.
35: Declare and initialize keyEventGathering. keyEventGathering is used to check if the pressed key is for testing purpose.
39: Set "_down_" and inputted value of event key to keyEventGathering.

**testKeyPress**

This function is keypress event handler.
46: Check menuKeyHandler() function and if menu button is executed, just return.
47: If menu button is not executed and only key inputs are received, set "_press_" and inputted value of event key to keyEventGathering.

**testKeyUp**

This function is a handler that is executed when keyup event occurs.
52: Set "_up_" and inputted value of event key to keyEventGathering.
54-57: If hasMenuExecuted is true, initialize to false.
57-59: If hasMenuExecuted is false, call checkEachResult() function.

01 : function initTest()
02 : {
03 :     addEventHandler(document.body, "keydown", testKeyDown);
04 :     addEventHandler(document.body, "keyup", testKeyUp);
05 :     addEventHandler(document.body, "keypress", testKeyPress);
06 :     initTestProgress();
07 : }
08 :
09 : function backHandler()
10 : {
11 :     window.location.replace("../menu_commonjavascriptapi.html");
12 : }
13 :
14 : function resetHandler()
15 : {
16 :     initTestProgress();
17 : }
18 :
19 : var backToYellow = false;
20 : var resetToGreen = false;
21 : var hasMenuExecuted = false;
22 :
23 : function menuKeyHandler(event)
24 : {
25 :     var userInput = getCharCode(event);
26 :     switch(userInput) {
27 :         case VK_BACK : if(!backToYellow){backHandler(); hasMenuExecuted = true;} break;
28 :         case VK_RED : case 82 : if(!resetToGreen){resetHandler(); hasMenuExecuted = true;} break;
29 :         case VK_GREEN : case 71 : if(resetToGreen){resetHandler(); hasMenuExecuted = true;} break;
30 :     }
Displaying Text

The following code displays test keys on screen with text and change the color of text and background according to the proceeding status of the test.

**setSelectedKey**

Receives selectedKeyIndex and changes the background color of key which will be tested next.

03-10: The function is executed by number of the size of keysToTestArray.
05-07: Change the background color of key text of received selectedKeyIndex which is the key index of the next test.
07-09: Background colors of other key texts are set with default value.

**initTestProgress**

Set key text and class for testing.

13: Declare and initialize currentKeyIndex.
17: Initialize currentKeyIndex to 0.
18-21: Using For loop, set class to each key by number keysToTestArray size.
22: Call setSelectedKey() function and initialize to test the first key.
23: Display the description on screen.

**setKeysOnScreen**

Displays key text on screen which is assigned to keysToTestArray.

01 : function setSelectedKey(selectedKeyIndex)
02 : {
03 : for(var i = 0 ; i < keysToTestArray.length ; i++)
function checkEachResult()
{
    if (selectedKeyIndex == i) {
        document.getElementById(keysToTestArray[i][0]).style.backgroundColor = "#FF8000";
    } else {
        document.getElementById(keysToTestArray[i][0]).style.backgroundColor = "";
    }
}

var currentKeyIndex = 0;

function initTestProgress()
{
    currentKeyIndex = 0;
    for (var i = 0; i < keysToTestArray.length; i++)
    {
        document.getElementById(keysToTestArray[i][0]).className="eachTestGuide";
    }
    setSelectedKey(0);
    setInnerTextById("keydownuppress_test_description", "Press OK-Key to test Key-Events and Key-Code.");
}

function setKeysOnScreen()
{
    for (var i = 0; i < keysToTestArray.length; i++)
    {
        setInnerTextById(keysToTestArray[i][0], keysToTestArray[i][1]);
    }
}

Executing Test
Use checkEachResult function to test each key event and display in order.

03: Return if currentKeyIndex is bigger than the size of keysToTestArray
05: Declare and initialize keyGatheringToCompare.
07-16: Set the string to keyGatheringToCompare to compare if the correct key is pressed when key event is occurred according to the keys of keysToTestArray[currentKeyIndex].
18-20: If the value of keyEventGathering is same with the value of keyGatheringToCompare, the key text is displayed with green.
21-22: If it is not same, display with red.
24: Initialize keyEventGathering.
26-28: If currentKeyIndex is Red button, set resetToGreen to false, make button executable, and display button text with red.
31-34: If currentKeyIndex is Back button, set backToYellow to false, make button executable, and display button text with white.
36: Increase the value of currentKeyIndex by 1.
37: If the value of currentKeyIndex is smaller than the size of keysToTestArray execute the following codes.
38: Call setSelectedKey function with currentKeyIndex parameter.
40-42: If the value of keysToTestArray is "PORTAL" and the device does not support "PORTAL" key, display the corresponding text.
43-45: Otherwise, display the corresponding text.
47-50: If the value of keysToTestArray is Red and it is executed from test menu, set the key text in menu to change into green.
52-55: If the value of keysToTestArray is Back and it is executed in test key, set the key text in menu to change into yellow.
56-59: If the value of currentKeyIndex is bigger than the size of keysToTestArray, display the corresponding text.
if(currentKeyIndex >= keysToTestArray.length){return;}

var keyGatheringToCompare = "";

if(keysToTestArray[currentKeyIndex][3] == "d_p_u"){
    keyGatheringToCompare = ".down." + keysToTestArray[currentKeyIndex][2] + ".press." + keysToTestArray[currentKeyIndex][2] + ".up." + keysToTestArray[currentKeyIndex][2];
} else if(keysToTestArray[currentKeyIndex][3] == "d_u"){
    keyGatheringToCompare = ".down." + keysToTestArray[currentKeyIndex][2] + ".up." + keysToTestArray[currentKeyIndex][2];
} else {
    keyGatheringToCompare = "";
}

if(keyEventGathering == keyGatheringToCompare){
    document.getElementById(keysToTestArray[currentKeyIndex][0]).className += "greenColor";
} else {
    document.getElementById(keysToTestArray[currentKeyIndex][0]).className += "redColor";
}

keyEventGathering = "";

if(keysToTestArray[currentKeyIndex][1] == "RED"){
    resetToGreen = false;
    document.getElementById("btn_reset").className = "buttonDescription redColor";
} else {
    };

if(keysToTestArray[currentKeyIndex][1] == "BACK"){
    backToYellow = false;
    document.getElementById("btn_back").className = "buttonDescription whiteColor";
} else {
    currentKeyIndex++;
}

if(currentKeyIndex < keysToTestArray.length){
    setSelectedKey(currentKeyIndex);
}

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

if((keysToTestArray[currentKeyIndex][1]=="PORTAL") && !device.supportPortalKey){
    setInnerTextById("keydownuppress_test_description", "Press " + ".RESET" + ".-Key to test Key-Events and Key-Code.");
} else {
    setInnerTextById("keydownuppress_test_description", "Press " + keysToTestArray[currentKeyIndex][1] + ".-Key to test Key-Events and Key-Code.");
}

if(keysToTestArray[currentKeyIndex][1] == "RED"){
    resetToGreen = true;
    document.getElementById("btn_reset").className = "buttonDescription yellowColor";
} else {
    if(keysToTestArray[currentKeyIndex][1] == "BACK"){
        backToYellow = true;
        document.getElementById("btn_back").className = "buttonDescription whiteColor";
    } else {
        currentKeyIndex++;
    }
}

if(currentKeyIndex < keysToTestArray.length){
    setSelectedKey(currentKeyIndex);
}

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

if((keysToTestArray[currentKeyIndex][1]=="PORTAL") && !device.supportPortalKey){
    setInnerTextById("keydownuppress_test_description", "Press " + ".RESET" + ".-Key to test Key-Events and Key-Code.");
} else {
    setInnerTextById("keydownuppress_test_description", "Press " + keysToTestArray[currentKeyIndex][1] + ".-Key to test Key-Events and Key-Code.");
}

if(keysToTestArray[currentKeyIndex][1] == "RED"){
    resetToGreen = true;
    document.getElementById("btn_reset").className = "buttonDescription yellowColor";
} else {
    if(keysToTestArray[currentKeyIndex][1] == "BACK"){
        backToYellow = true;
        document.getElementById("btn_back").className = "buttonDescription whiteColor";
    } else {
        currentKeyIndex++;
    }
}

if(currentKeyIndex < keysToTestArray.length){
    setSelectedKey(currentKeyIndex);
}

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

if((keysToTestArray[currentKeyIndex][1]=="PORTAL") && !device.supportPortalKey){
    setInnerTextById("keydownuppress_test_description", "Press " + ".RESET" + ".-Key to test Key-Events and Key-Code.");
} else {
    setInnerTextById("keydownuppress_test_description", "Press " + keysToTestArray[currentKeyIndex][1] + ".-Key to test Key-Events and Key-Code.");
}

if(keysToTestArray[currentKeyIndex][1] == "RED"){
    resetToGreen = true;
    document.getElementById("btn_reset").className = "buttonDescription yellowColor";
} else {
    if(keysToTestArray[currentKeyIndex][1] == "BACK"){
        backToYellow = true;
        document.getElementById("btn_back").className = "buttonDescription whiteColor";
    } else {
        currentKeyIndex++;
    }
}
yellowColor};
55:    }
56:  }else{
57:    setSelectedKey(-1);
58:    setInnerTextById("keydownuppress_test_description", "No more test is left on this page.");
59:  }
60: }

Source Code of keydownuppress.html

Source code of keydownuppress.html is as follows:

```html
<!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>Key Down/Up/Press Test Page</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>
var keysToTestArray = [
  ["div_01", "OK", 13, "d_p_u"],
  ["div_02", "PLAY", 415, "d_u"],
  ["div_03", "PAUSE", 19, "d_u"],
  ["div_04", "STOP", 413, "d_u"],
  ["div_05", "REWIND", 412, "d_u"],
  ["div_06", "FAST_FWD", 417, "d_u"],
  ["div_07", "PAGE_UP", 33, "d_u"],
  ["div_08", "PAGE_DOWN", 34, "d_u"],
  ["div_09", "LEFT", 37, "d_u"],
  ["div_10", "RIGHT", 39, "d_u"],
  ["div_11", "UP", 38, "d_u"],
  ["div_12", "DOWN", 40, "d_u"],
  ["div_13", "NUMBER 0", 48, "d_p_u"],
  ["div_14", "NUMBER 1", 49, "d_p_u"],
  ["div_15", "NUMBER 2", 50, "d_p_u"],
  ["div_16", "NUMBER 3", 51, "d_p_u"],
  ["div_17", "NUMBER 4", 52, "d_p_u"],
  ["div_18", "NUMBER 5", 53, "d_p_u"],
  ["div_19", "NUMBER 6", 54, "d_p_u"],
  ["div_20", "NUMBER 7", 55, "d_p_u"],
  ["div_21", "NUMBER 8", 56, "d_p_u"],
  ["div_22", "NUMBER 9", 57, "d_p_u"],
  ["div_23", "RED", 403, "d_u"],
  ["div_24", "GREEN", 404, "d_u"],
  ["div_25", "YELLOW", 405, "d_u"],
  ["div_26", "BLUE", 406, "d_u"],
  ["div_27", "INFO", 457, "d_u"],
  ["div_28", "BACK", 461, "d_u"],
  ["div_29", "PORTAL", 1000, "d_u"]];
//initialize page
```
function initPage()
{
    // save page as last visited page
    setLastVisitPage();

    // common initialize function
    commonInitialize();
    requestSourceCode();
    setPageID("Key Down/Up/Press");
    jsLog.initLG();

    // add onclick event handler
    addEventHandler(document.getElementById("btn_back"), "click", backHandler);
    addEventHandler(document.getElementById("btn_reset"), "click", resetHandler);

    // add event handler for menu
    addEventHandler(document.body, "keydown", menuKeyHandler);

    // set keys to test
    setKeysOnScreen();

    // add event handler to test
    setTimeout(initTest, 700);

    var device = document.getElementById("device");
    if(! device.supportPortalKey)
    {
        document.getElementById("div_29").style.visibility = "hidden";
    }
}

function initTest()
{
    addEventHandler(document.body, "keydown", testKeyDown);
    addEventHandler(document.body, "keyup", testKeyUp);
    addEventHandler(document.body, "keypress", testKeyPress);
    initTestProgress();
}

function backHandler()
{
    window.location.replace("../menu_commonjavascriptapi.html");
}

function resetHandler()
{
    initTestProgress();
}

var backToYellow = false;
var resetToGreen = false;
var hasMenuExecuted = false;

function menuKeyHandler(event)
{
    var userInput = getKeyCode(event);
    switch(userInput)
    {
        case VK_BACK : if(!backToYellow){backHandler(); hasMenuExecuted = true;}
        break;
case VK_RED : case 82 : if(!resetToGreen){resetHandler(); hasMenuExecuted = true;} break;
case VK_GREEN : case 71 : if(resetToGreen){resetHandler(); hasMenuExecuted = true;} break;
case VK_YELLOW : case 89 : if(backToYellow){backHandler(); hasMenuExecuted = true;} break;
}
}

var keyEventGathering = "";

function testKeyDown(event)
{
    keyEventGathering += "_down_" + get.keyCode(event);
    jsLog.lgevent('javascript: Key Event Handling : KeyDown, Key : ' +
    getKeyName(event));
}

function testKeyPress(event)
{
    jsLog.lgevent('javascript: Key Event Handling : KeyPress, Key : ' +
    getKeyName(event));
    if(menuKeyHandler(event, true) == "menuExecuted") {return;}
    keyEventGathering += "_press_" + get.keyCode(event);
}

function testKeyUp(event)
{
    keyEventGathering += "_up_" + get.keyCode(event);
    jsLog.lgevent('javascript: Key Event Handling : KeyUp, Key : ' +
    getKeyName(event));
    if(hasMenuExecuted){
        hasMenuExecuted = false;
        keyEventGathering = "";
    }else{
        checkEachResult();
    }
}

function setSelectedKey(selectedKeyIndex)
{
for(var i = 0 ; i < keysToTestArray.length ; i++)
{
    if(selectedKeyIndex == i){
        document.getElementById(keysToTestArray[i][0]).style.backgroundColor =
        "#FF8000";
    }else{
        document.getElementById(keysToTestArray[i][0]).style.backgroundColor =
        "";
    }
}
}

var currentKeyIndex = 0;

function initTestProgress()
{
    currentKeyIndex = 0;
    for(var i = 0 ; i < keysToTestArray.length ; i++)
    {

document.getElementById(keysToTestArray[i][0]).className = "eachTestGuide";
}
setSelectedKey(0);
setInnerTextById("keydownuppress_test_description", "Press OK-Key to test Key-Events and Key-Code.");
}

//set keys to test
function setKeysOnScreen()
{
    for(var i = 0 ; i < keysToTestArray.length ; i++)
    {
        setInnerTextById(keysToTestArray[i][0], keysToTestArray[i][1]);
    }
}

function checkEachResult()
{
    if(currentKeyIndex >= keysToTestArray.length){return;}
    var keyGatheringToCompare = ";

    if(keysToTestArray[currentKeyIndex][3] == "d_p_u"){
        keyGatheringToCompare = "_down_" + keysToTestArray[currentKeyIndex][2]
        + "_press_" + keysToTestArray[currentKeyIndex][2]
        + "_up_" + keysToTestArray[currentKeyIndex][2];
    }else if(keysToTestArray[currentKeyIndex][3] == "d_u"){
        keyGatheringToCompare = "_down_" + keysToTestArray[currentKeyIndex][2]
        + "_up_" + keysToTestArray[currentKeyIndex][2];
    }else{
        keyGatheringToCompare = "";
    }

    if(keyEventGathering == keyGatheringToCompare){
        document.getElementById(keysToTestArray[currentKeyIndex][0]).className += "
        greenColor";
    }else{
        document.getElementById(keysToTestArray[currentKeyIndex][0]).className += "
        redColor";
    }

    keyEventGathering = "";
    //in case just tested red key
    if(keysToTestArray[currentKeyIndex][1] == "RED"){
        resetToGreen = false;
        document.getElementById("btn_reset").className = "buttonDescription redColor";
    }
    //in case just tested back key
    if(keysToTestArray[currentKeyIndex][1] == "BACK"){
        backToYellow = false;
        document.getElementById("btn_back").className = "buttonDescription whiteColor";
    }

    currentKeyIndex++;
    if(currentKeyIndex < keysToTestArray.length){
        setSelectedKey(currentKeyIndex);
        var device = document.getElementById("device");
        if((keysToTestArray[currentKeyIndex][1]=="PORTAL") &
        & !device.supportPortalKey){
    }
if(! device.supportPortalKey){
    setInnerTextById("keydownuppress_test_description", "Press " + "RESET" + "-Key to test Key-Events and Key-Code.");
}
else{
    setInnerTextById("keydownuppress_test_description", "Press " + keysToTestArray[currentKeyIndex][1] + "-Key to test Key-Events and Key-Code.");
}

//in case next test is red key
if(keysToTestArray[currentKeyIndex][1] == "RED"){
    resetToGreen = true;
    document.getElementById("btn_reset").className = "buttonDescription yellowColor";
}
//in case next test is back key
if(keysToTestArray[currentKeyIndex][1] == "BACK"){
    backToYellow = true;
    document.getElementById("btn_back").className = "buttonDescription yellowColor";
}
}else{
    setSelectedKey(-1);
    setInnerTextById("keydownuppress_test_description", "No more test is left on this page.");
}
}

function getKeyName(event)
{
    var key = getKeyCode(event);
    for(var i = 0 ; i < keysToTestArray.length ; i++)
    {
        if(keysToTestArray[i][2] == key)
        {
            return keysToTestArray[i][1];
        }
    }
    return "";
}

</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>
<object type="application/x-netcast-info"
    id="device">
</object>
<!-- navigation -->

<div class='SuiteNavigation'>
    <div style="float:left;" File : /commonjavascriptapi/app/keyrepeat.html</div>
</div>
<div class='SuiteTitleLine' ></div>
<!-- test contents -->
<p>| | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
<div id='btn_exit' class='buttonDescription'>EXIT</div>

<div id='btn_reset' class='buttonDescription redColor'>RESET</div>

<div class='copyright'>Copyright LG Electronics</div>
Handling Key Repeat Using JavaScript Events

This section describes how to handle the key repeat using JavaScript events.

- Initializing the Page
- Handling Events
- Source Code of keyrepeat.html

This application shows how JavaScript events are occurred when a key is pressed from IR Remote and to use these events.

---

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.
19-20: Add event handlers for key inputs (keydown, keyup) from IR Remote.
23: Set the background color of key to test.
24: Display description for test on screen.

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

  // common initialize function
  commonInitialize();

  // request source code
  requestSourceCode();

  // set page ID
  setPageID("Key Repeat");

  // initialize Log function
  jsLog.initLG();

  // add onclick event handler
```

---

Note

Log is used for checking the sequence of Web Open API; this will not be covered in the sample code description.
Handling Events
The following code shows event handler functions which are added from initPage() function.

02-16: Declare and initialize keyToTest, currentTestPosition, maxRepeatCount, maxGaugeWidth, repeatCount, startTime, and progressStatus.

onKeyDownHandler
This event handler called when key down is occurred.
20: Declare userInput and save key code of the event.
23: If userInput is same with setup key and the repeat number is not bigger than the maximum value, execute the following codes.
25-30: If repeatCount is 0, declare currentDate and initialize startTime to current time.
31: Increase repeatCount by 1.
32: Increase gauge bar by number of repeatCount.
34-40: If number of key repeat is bigger than maxRepeatCount, declare elapsedTime, count the key repeat time, and display it on screen.

onKeyUpHandler
This event handler is called when key up is occurred.
46: Declare userInput and save key code of the event.
49-52: If inputted key value is same with setup key value and the repeat number does not exceed the maximum number, change the color of corresponding gauge bar.

resetCurrentTest
Execute reset function.
57-59: Initialize repeatCount, startTime, and progressStatus.
60: Delete the child nodes of currentTestPosition.
61-63: Initialize the bar length of currentTestPosition and display the corresponding text on screen.

changeTestPosition
Change key to test.
68-70: Initialize repeatCount, startTime, and progressStatus.
71: Set class to unselect the selected key.
73-78: To go to previous key, decrease the value of currentTestPosition by 1 and move the position of the key. If the value of currentTestPosition gets smaller than 0, move the position to the last key position according to the size of keysToTest.
80-85: To go to the next key, increase the value of currentTestPosition by 1 and move the key position. If the value of currentTestPosition gets bigger than keysToTest, set to 0 and move to the first key position.
87-88: Changing the background color of class to orange to show that the testing key is changed. Also, display description text on screen.
var currentTestPosition = 0;
var maxRepeatCount = 40;
var maxGaugeWidth = 273;
var repeatCount = 0;
var startTime = 0;
var progressStatus = "";

function onKeyDownHandler(event) {
    var userInput = getKeyCode(event);
    jsLog.lgevent('javascript: Key Event Handling : KeyDown, Key : ' + getKeyName(event));
    if(userInput==keysToTest[currentTestPosition][2]&&repeatCount<maxRepeatCount) {
        if(repeatCount == 0) {
            var currentDate = new Date();
            startTime = currentDate.getTime();
            setInnerTextById(keysToTest[currentTestPosition][1], "");
        }
        repeatCount++;
        document.getElementById(keysToTest[currentTestPosition][1]).style.width = Math.floor(maxGaugeWidth * (repeatCount / maxRepeatCount)) + "px";
        if(repeatCount >= maxRepeatCount) {
            var currentDate = new Date();
            var elapsedTime = currentDate.getTime() - startTime;
            setInnerTextById(keysToTest[currentTestPosition][1], (elapsedTime / repeatCount) + "ms");
            setInnerTextById("keyrepeat_test_description", "Press YELLOW-Key to test next key.");
        }
    }
}

function onKeyUpHandler(event) {
    var userInput = getKeyCode(event);
    jsLog.lgevent('javascript: Key Event Handling : KeyUp, Key : ' + getKeyName(event));
    if(userInput==keysToTest[currentTestPosition][2]&&repeatCount<maxRepeatCount) {
        document.getElementById(keysToTest[currentTestPosition][1]).style.backgroundColor="#FF0000";
    }
}

function resetCurrentTest() {
    repeatCount = 0;
}
startTime = 0;
progressStatus = "";
clearChildNodes(document.getElementById(keysToTest[currentTestPosition][1]));
document.getElementById(keysToTest[currentTestPosition][1]).style.width="0px";
document.getElementById(keysToTest[currentTestPosition][1]).style.backgroundColor = "#008000";
setInnerTextById("keyrepeat_test_description", "Press and hold " + keysToTest[currentTestPosition][3] + "-key to test key repeat.");
}

function changeTestPosition(direction)
{
repeatCount = 0;
startTime = 0;
progressStatus = "";
document.getElementById(keysToTest[currentTestPosition][0]).className="eachTestGuide";
if(direction == "PREVIOUS"){
currentTestPosition--;
if(currentTestPosition < 0){
currentTestPosition = keysToTest.length - 1;
}
}
if(direction == "NEXT"){
currentTestPosition++;
if(currentTestPosition >= keysToTest.length){
currentTestPosition = 0;
}
}
document.getElementById(keysToTest[currentTestPosition][0]).className = "eachTestGuide orangeBgColor";
setInnerTextById("keyrepeat_test_description", "Press and hold " + keysToTest[currentTestPosition][3] + "-key to test key repeat.");
}

Source Code of keyrepeat.html
Source code of keyrepeat.html is as follows:

```html
<!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>Key Repeat Test Page</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>
<link type="text/css" rel="Stylesheet" href="../../blackbirdjs/blackbird.css" />
<script>
    //initialize page
    function initPage()
```
function onUserInput(userInput) {
  switch (userInput) {
    case VK_BACK: window.location.replace("../menu_commonjavascriptapi.html"); break;
    case VK_RED: case 82: resetCurrentTest(); break;
    case VK_GREEN: case 71: changeTestPosition("PREVIOUS"); break;
    case VK_YELLOW: case 89: changeTestPosition("NEXT"); break;
  }
}

// array of keys to test
// [0]: title id, [1]: gauge id, [2]: key code, [3]: key name
var keysToTest = [
  ["up_title", "up_progress", 38, "UP"],
  ["down_title", "down_progress", 40, "DOWN"],
  ["left_title", "left_progress", 37, "LEFT"],
  ["right_title", "right_progress", 39, "RIGHT"],
  ["rewind_title", "rewind_progress", 412, "REWIND"],
  ["fast_forward_title", "fast_forward_progress", 417, "FAST_FORWARD"],
  ["page_up_title", "page_up_progress", 33, "PAGE_UP"],
  ["page_down_title", "page_down_progress", 34, "PAGE_DOWN"]
];

var currentTestPosition = 0;
var maxRepeatCount = 40;
var maxGaugeWidth = 273;
var repeatCount = 0;
var startTime = 0;
var progressStatus = "";

function onKeyDownHandler(event) {

}
```
var userInput = getKeyCode(event);
jsLog.lgevent('javascript: Key Event Handling : KeyDown, Key : ' +
getKeyName(event));

if(userInput == keysToTest[currentTestPosition][2] && repeatCount <
maxRepeatCount){
    if(repeatCount == 0){
        var currentDate = new Date();
        startTime = currentDate.getTime();
        setInnerTextById(keysToTest[currentTestPosition][1], "");
    }
    repeatCount++;
    document.getElementById(keysToTest[currentTestPosition][1]).style.width =
Math.floor((maxGaugeWidth * (repeatCount / maxRepeatCount))) + "px";
    if(repeatCount >= maxRepeatCount){
        var currentDate = new Date();
        var elapsedTime = currentDate.getTime() - startTime;
        setInnerTextById(keysToTest[currentTestPosition][1], (elapsedTime /
repeatCount) + "ms");
        setInnerTextById("keyrepeat_test_description", "Press YELLOW-Key to test
next key.");
    }
}

function onKeyUpHandler(event)
{
    var userInput = getKeyCode(event);
    jsLog.lgevent('javascript: Key Event Handling : KeyUp, Key : ' +
getKeyName(event));

    if(userInput == keysToTest[currentTestPosition][2] && repeatCount <
maxRepeatCount){
        document.getElementById(keysToTest[currentTestPosition][1]).style.backgroundColor =
"#FF0000";
    }
}

// reset current key test status
function resetCurrentTest()
{
    repeatCount = 0;
    startTime = 0;
    progressStatus = "";
    clearChildNodes(document.getElementById(keysToTest[currentTestPosition][1]));
document.getElementById(keysToTest[currentTestPosition][1]).style.width =
"0px";

document.getElementById(keysToTest[currentTestPosition][1]).style.backgroundColor =
"#008000";
    setInnerTextById("keyrepeat_test_description", "Press and hold " +
keysToTest[currentTestPosition][3] + "-key to test key repeat.");
}

// change test position
function changeTestPosition(direction)
{
    repeatCount = 0;
    startTime = 0;
    progressStatus = "";
    document.getElementById(keysToTest[currentTestPosition][0]).className =
```
if(direction == "PREVIOUS") {
    currentTestPosition--;
    if(currentTestPosition < 0) {
        currentTestPosition = keysToTest.length - 1;
    }
}
if(direction == "NEXT") {
    currentTestPosition++;
    if(currentTestPosition >= keysToTest.length) {
        currentTestPosition = 0;
    }
}
document.getElementById(keysToTest[currentTestPosition][0]).className = "eachTestGuide orangeBgColor";
setInnerTextById("keyrepeat_test_description", "Press and hold " + keysToTest[currentTestPosition][3] + "-key to test key repeat.");
}

function getKeyName(event) {
    var key = getKeyCode(event);
    for(var i = 0; i < keysToTest.length; i++) {
        if(keysToTest[i][2] == key) {
            return keysToTest[i][3];
        }
    }
    return "-";
}

*/
</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 : /commonjavascriptapi/app/keyrepeat.html</div>
</div>
<div class='SuiteTitleLine'></div>

<!-- test contents -->
<div id='ContentArea' class='ContentArea' >
    <div class='ApiListTitleArea'>Web Open API List</div>
    <div class='ApiListArea'>
        <div class='MethodTitleArea'>Methods</div>
        <div class='MethodListArea'>
        </div>
        <div class='PropertyTitleArea'>Properties</div>
    </div>
</div>

/*
<table>
<thead>
<tr>
<th>Event</th>
<th>Interval Average</th>
<th>Interval Average</th>
</tr>
</thead>
<tbody>
<tr>
<td>UP</td>
<td><img src="#" alt="Gauge UP" /></td>
<td><img src="#" alt="Gauge Rewind" /></td>
</tr>
<tr>
<td>DOWN</td>
<td><img src="#" alt="Gauge Down" /></td>
<td><img src="#" alt="Gauge Fast Forward" /></td>
</tr>
<tr>
<td>LEFT</td>
<td><img src="#" alt="Gauge Left" /></td>
<td><img src="#" alt="Gauge Page Up" /></td>
</tr>
<tr>
<td>RIGHT</td>
<td><img src="#" alt="Gauge Right" /></td>
<td><img src="#" alt="Gauge Page Down" /></td>
</tr>
</tbody>
</table>
Implementing Functionality of Back and Exit Keys

This section describes how to implement functionality of Back and Exit keys in web application.

- Initializing the Page
- Inputting Keys
- Handling Keys
- Source Code of backexit.html

This application is designed to show how to implement the functionality of Back, and Exit using Web open API. This application shows which method is used to exit the application and move back to the MyApps menu or TV video in each page.

### Needed APIs

This application uses following Web Open API:

<table>
<thead>
<tr>
<th>API Class</th>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Method</td>
<td>window.NetCastExit()</td>
<td>This API is used to exit or quit the application.</td>
</tr>
<tr>
<td></td>
<td>window.NetCastBack()</td>
<td>This API is used to move back to the MyApps menu.</td>
</tr>
<tr>
<td>Property</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>Event</td>
<td>N/A</td>
<td>N/A</td>
</tr>
</tbody>
</table>

**Note**

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

### Initializing the Page

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

04: Records the last visited page when running the application.
07: Initializes the page.
08: Gets the source code of the page using the XMLHttpRequest object.
09: Sets the page ID.
10: Initializes the Log function.
13-16: Registers an event handler which will executed when the corresponding button is pressed.
18-21: Deletes the cookie related the application if it has been saved.

```javascript
01 : function initPage()
02 : {
03 :   //save page as last visited page
```
04:     setLastVisitPage();
05: 
06:     //common initialize function
07:     commonInitialize();
08:     requestSourceCode();
09:     setPageID("Back, Exit");
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:     if(readCookie("back_exit_test_status") != "")
19:     {
20:       //clear back_exit_test_status cookie if it have been set
21:       deleteCookie("back_exit_test_status", "/");
22:     }
23: }

Inputting Keys

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

06: When the Back key is pressed, 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.

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

Handling Keys

The following functions are executed when each key of onUserInput() is pressed.

processRedKey

This function is called when the Red key is pressed.
05-07: Gets the type and version of the browser through the navigator object.
11: Saves "back_exit_test_status" cookie.
13: Moves back to the MyApps menu.

processGreenKey

This function is called when the Green key is pressed.
27: Exits the application.
30: Shows the text on the screen if the browser of app is not a LG Browser.

Note

If you open the main menu when "back_exit_test_status" cookie has been saved, you can move to the current application through a new popup window.
function processRedKey()
{
    if(isThisLGEBrowser())
    {
        var userAgentString = new String(navigator.userAgent);
        var BrowserVer = userAgentString.substring(userAgentString.search(/LG Browser/) + 11);
        BrowserVer = BrowserVer.split("\", 1);
        if(BrowserVer >= "3.0.23")
        {
            //set cookie which will be checked at main_menu.html
            writeCookie("return_back_exit_test_status", "under_testing", "/", 1 * 24 * 60 * 60 * 1000);
            jsLog.lgmethod('window.NetCastBack()');
            window.NetCastBack();
        }
        else
        {
            setInnerTextById("result_3", "Need Version over 3.0.23");
        }
    }
    else
    {
        setInnerTextById("result_3", "This is not a LG Browser");
    }
}

function processGreenKey()
{
    if(isThisLGEBrowser())
    {
        jsLog.lgmethod('window.NetCastExit()');
        window.NetCastExit();
    }
    else
    {
        setInnerTextById("result_4", "This is not a LG Browser");
    }
}

Source Code of backexit.html
Source code of backexit.html is as follows:

```html
<!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" />

//initialize page
function initPage()
{
    //save as last visited page
    setLastVisitPage();
```
commonInitialize();
requestSourceCode();
setPageID("Back, Exit");
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);

if(readCookie("back_exit_test_status") != "") {
    // clear back_exit_test_status cookie if it have been set
    deleteCookie("back_exit_test_status", "/");
}

function onUserInput(userInput) {
    switch(userInput) {
        case VK_BACK : window.location.replace("../menu_netcast.html"); break;
        case VK_RED : case 82 : processRedKey(); break;
        case VK_GREEN : case 71 : processGreenKey(); break;
    }
}

function processRedKey() {
    if(isThisLGEBrowser()) {
        var userAgentString = new String(navigator.userAgent);
        var BrowserVer = userAgentString.substring(userAgentString.search(/LG Browser/) + 11);
        BrowserVer = BrowserVer.split("", 1);
        if(BrowserVer >= "3.0.23") {
            // set cookie which will be checked at main_menu.html
            writeCookie("back_exit_test_status", "under_testing", "/", 1 * 24 * 60 * 60 * 1000);
            jsLog.lgmethod('window.NetCastBack()');
            window.NetCastBack();
        } else {
            setInnerTextById("result_3", "Need Version over 3.0.23");
        }
    } else {
        setInnerTextById("result_3", "This is not a LG Browser");
    }
}

function processGreenKey() {
    if(isThisLGEBrowser()) {
        jsLog.lgmethod('window.NetCastExit()');
        window.NetCastExit();
    }
else
    setInnerTextById("result_4", "This is not a LG Browser");
}
</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 : /netcast/app/backexit.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'>
                window.NetCastBack()<br>
                window.NetCastExit()
            </div>
        </div>
        <div class='PropertyTitleArea'>Properties</div>
        <div class='EventTitleArea'>Events</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" cellpadding="0" cellspacing="0">
            <tr height="50px">
                <td width=500px align=left><div>1. Press Red-key to test NetCastBack()</div></td>
                <td width=200px align="left"><div id="result_1"></div></td>
                <td width=100px align="left"></td>
            </tr>
        </table>
    </div>
</div>
2. Press Green-key to test NetCastExit()
Implementing Functionality of Multi Audio

This section describes how to implement functionality of multi audio in web application.

- Initializing the Page
- Inputting Keys
- Changing Language
- Moving the Video Playback Position
- Setting Object
- Source Code of multiaudio.html

This application shows how to use the Multi Audio function. The Multi Audio function plays a video supporting multiple languages and allows the user to listen to voice data in desired language.

Needed APIs
This application uses following Web Open API:

<table>
<thead>
<tr>
<th>API Class</th>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Method</td>
<td>Seek(time)</td>
<td>Sets the time position of playback.</td>
</tr>
<tr>
<td>Property</td>
<td>playTime</td>
<td>Returns the duration of the currently playing media item</td>
</tr>
<tr>
<td></td>
<td>playPosition</td>
<td>Returns the play position of the currently playing media item</td>
</tr>
<tr>
<td></td>
<td>playState</td>
<td>Returns the play state of the currently playing media item as an enumerated number.</td>
</tr>
<tr>
<td>Event</td>
<td>N/A</td>
<td>N/A</td>
</tr>
</tbody>
</table>

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

Initializing the Page

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

04: Records the last visited page when running the application.
07: Initializes the page.
08: Gets the source code of the page using the XMLHttpRequest object.
09: Sets the page ID.
10: Initializes the Log function.
13-17: Registers an event handler which will executed when the corresponding button is pressed.
19: Calls checkInitStatus() function.
```javascript
function initPage() {
  // save page as last visited page
  setLastVisitPage();
  // common initialize function
  commonInitialize();
  requestSourceCode();
  setPageID("Multi Audio");
  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_yellow"), "click", onClickHandler);
  addEventHandler(document.getElementById("btn_exit"), "click", onClickHandler);

  checkInitStatus();
}
```

**Inputting Keys**

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

06: When the Back key is pressed, this code is executed.
07: When the Red key is pressed, `toggleAudioLanguage()` is called.
08: When the Green key is pressed, `movePosition(-60)` is called.
09: When the Yellow key is pressed, `movePosition(60)` is called.

```
function onUserInput(userInput) {
  switch(userInput) {
    case VK_BACK: window.location.replace("../menu_commonjavascriptapi.html"); break;
    case VK_RED : toggleAudioLanguage(); break;
    case VK_GREEN : movePosition(-60); break;
    case VK_YELLOW : movePosition(60); break;
    default: break;
  }
}
```

**Changing Language**

The following code shows that the language changes when the video plays back.

`checkInitStatus`

Checks the playback status.
06-11: If playState is 1 (playing), `toggleAudioLanguage()` is called.
12: If playState is not 1, the `setTimeout()` method is called to call `checkInitStatus()` every 1 ms.

`toggleAudioLanguage`

Changes the language.
16-17: Declares the `curLangIdx` variable and stores the supported languages in `supportLanguage`.
21-24: Since two languages are supported, they are stored in `curLangIdx` and `nextLangIdx` alternately.
25: Changes the language using the `audioLanguage` property.
28-31: Displays the currently selected language on the screen.
Moving the Video Playback Position

The following code moves the current playback position of the video to the desired position using the seek API.

movePosition

04: Stores the video object in the videoObj variable.
05: Gets the total playback time of the video using the playTime property.
08: Adds the setToMove parameter value passed to this function when it was called to the current playback position (playPosition) and stores the result in the positionToSet variable.
11: If positionToSet is less than 0, positionToSet is set to 0 to move the playback position of the video to the start position.
12: If positionToSet is greater than the total playback time of the video, positionToSet is set to the totalPlayTime value to move the playback position of the video to the end position.
14: Moves the current playback position of the video to the position stored in positionToSet.
12 : if(positionToSet > totalPlayTime){positionToSet = totalPlayTime;}
13 :
14 : video.seek(positionToSet);
15 : jsLog.lgmethod('seek(time)');
16 : }

Setting Object
The following code shows how to set object.

03: Set data type. Refer to App Development Guide.
04-05: Set width and height.

01 : <object
02 : id="video"
03 : type="application/x-netcast-av"
04 : width=300
05 : height=250
06 : data="../mediafile/ multi_audio2.wmv"
07 : style="float: left">
08 : </object>

Source Code of multiaudio.html
Source code of multiaudio.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>Multi Audio Test Page</title>
<link rel="stylesheet" href="../../css/style.css">
<script language="javascript" src="../../js/common.js"></script>
<script language="javascript" src="../../js/keycode.js"></script>
<script language="javascript" src="../../js/media.js"></script>
<script language="javascript" src="../../js/menu.js"></script>
<script language="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("Multi Audio");
    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_yellow"), "click", onClickHandler);
    addEventHandler(document.getElementById("btn_exit"), "click", onClickHandler);
checkInitStatus();

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

function onUserInput(userInput)
{
    switch(userInput)
    {
        case VK_BACK : window.location.replace("../menu_commonjavascriptapi.html"); break;
        case VK_RED : toggleAudioLanguage(); break;
        case VK_GREEN : movePosition(-60); break;
        case VK_YELLOW : movePosition(60); break;
    }
}

// check until playstate become 1(playing) to set initial "audioLanguage"
function checkInitStatus()
{
    var videoObj = document.getElementById("video");
    if(videoObj.playState == 1)
    {
        jsLog lgproperty('playState : ' + videoObj.playState);
        toggleAudioLanguage();
    }
    else{setTimeout(checkInitStatus, 1);}  
}

// toggle "audioLanguage" among supportLanguage
var curLangIdx = -1;
var supportLanguage = ["en", "fr"];

function toggleAudioLanguage()
{
    curLangIdx++;
    if(curLangIdx >= supportLanguage.length){curLangIdx = 0;}
    var nextLangIdx = curLangIdx + 1;
    if(nextLangIdx >= supportLanguage.length){nextLangIdx = 0;}
    document.getElementById("video").audioLanguage = supportLanguage[curLangIdx];
    var guideStrToSet = ""
    guideStrToSet += 'Current "audioLanguage" is "' + supportLanguage[curLangIdx] + '". <br>
    setInnerText(document.getElementById("multiAudioTestGuide"), guideStrToSet);
}

// move position
function movePosition(secToMove)
{
    var videoObj = document.getElementById("video");
    var totalPlayTime = videoObj.playTime;
    jsLog lgproperty('playTime : ' + totalPlayTime);
var positionToSet = videoObj.playPosition + (secToMove * 1000);
jsLog.lgproperty('playPosition : ' + videoObj.playPosition);

if(positionToSet < 0 ) {positionToSet = 0;}
if(positionToSet > totalPlayTime) {positionToSet = totalPlayTime;}

video.seek(positionToSet);
jsLog.lgmethod('seek(time)');

</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 : /commonjavascriptapi/app/multiaudio.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>
    <div class='MethodListArea'>
      seek(time)
    </div>
  </div>
  <div class='PropertyTitleArea'>Properties</div>
  <div class='PropertyListArea'>
    playTime<br>
    playPosition<br>
    playState<br>
    audioLanguage<br>
  </div>
  <div class='EventTitleArea'>Events</div>
  <div class='EventListArea'>
  </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>
<table border="0" cellpadding="0" cellspacing="0" style="position: relative; left: 10px; width: 600px; height: 150px;">
<tr height="10%">
<td><div class="centerTestGuide">Press Green-key to move 60 sec backward.</div></td>
</tr>
<tr height="10%">
<td><div class="centerTestGuide">Press Yellow-key to move 60 sec forward.</div></td>
</tr>
<tr height="10%">
<td><div class="centerTestGuide">Press Red-key to change "audioLanguage".</div></td>
</tr>
<tr height="10%">
<td align="center" valign="middle"><div class="centerTestGuide"></div></td>
</tr>
</table>
</div>
Playing Media with ASX File

This section describes how to play media with asx file.

- Initializing the Page
- Inputting Keys
- Displaying Text
- Executing Test
- Setting Object
- Source Code of asx.html

This application shows how to play media with ASX file, which is play list file format, using the Web Open API of Smart TV. Refer to App Development Guide > Specifications > Play list (ASX) for supported elements.

**Needed APIs**

This application uses following Web Open API:

<table>
<thead>
<tr>
<th>API Class</th>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Method</td>
<td>play(1)</td>
<td>Plays media.</td>
</tr>
<tr>
<td></td>
<td>next()</td>
<td>Plays the next media.</td>
</tr>
<tr>
<td></td>
<td>previous()</td>
<td>Plays the previous media.</td>
</tr>
</tbody>
</table>

**Note**

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

**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.
13-15: Add an event handler which will be executed when the corresponding button is pressed.
18: Declare userAgent and set the value of navigator.userAgent.
20~23: Check if the device uses LG Browser and set true or false to isLGEBrowser.

Refer to App Development Guide > Specifications > userAgent string.
01 : function initPage()
02 : {
03 :     // save page as last visited page
04 :         setLastVisitPage();
05 : 
06 :     // common initialize function
07 :         commonInitialize();
08 :     requestSourceCode();
09 :     setPageID("Asx");
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_exit"),"click", onClickHandler);
16 : 
17 :     // check if this is LG Browser
18 :     var userAgent = new String(navigator.userAgent);
19 :
20 :         if (userAgent != null && userAgent.search(/LG Browser/) > -1)
21 :             isLGEBrowser = true;
22 :         else
23 :             isLGEBrowser = false;
24 :
25 :     jsLoglgobject('application/x-netcast-av');
26 : }

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

```
01 : function onUserInput(userInput)
02 : {
03 :     switch(userInput)
04 :     {
05 :         case VK_BACK : window.location.replace ("../menu_mediaPlayer.html");
06 :             break;
07 :         case VK_RED : case 82 changeTestProgress(); break;
08 :             break;
09 :     }
10 : }
```

**Displaying Text**

Use `setVideoDescription` function receives text as the `descriptionToSet` parameter and display the text on corresponding ID.

```
01 : function setVideoDescription(descriptionToSet)
02 : {
03 :     document.getElementById("video_description").style.zIndex = "2";
04 :     document.getElementById("test_result").style.zIndex = "1";
05 :     setInnerTextById("video_description", descriptionToSet);
06 : }
```

**Executing Test**

The `changeTestProgress` function is called when user pressed Red button.
Using switch-case, execute testing according to the value of testStep.

01: Declare initialize testStep.
05-08: Declare video, asxPropertiesDiv, descriptionDiv, and redButtonDiv and save each element.
10-53: Using switch-case, execute the functions according to the value of testStep.
27-34: (case1) If the browser is LG Browser, declare video and play media.
36-43: (case3) If the browser is LG Browser, move to the next media.
45-52: (case4) If the browser is LG Browser, move to the previous media.
54: Increase testStep by 1.

01 : var testStep = 1;
02 :
03 : function changeTestProgress()
04 : {
05 :     var video = document.getElementById("video");
06 :     var asxPropertiesDiv = document.getElementById("asxProperties");
07 :     var descriptionDiv = document.getElementById("buttonDescription");
08 :     var redButtonDiv = document.getElementById("btn_red");
09 :
10 :     switch(testStep)
11 :     {
12 :         case 1:
13 :             if(isLGEBrowser)
14 :             {
15 :                 var video = document.getElementById("video");
16 :                 video.play(1);
17 :                 jsLog.lgmethod('video.play()');
18 :             }
19 :             else
20 :                 setVideoDescription("This is not a LG Browser");
21 :
22 :                 setVideoDescription("Check whether snow boarding video is displayed.");
23 :                 setInnerTextById("buttonDescription", "Press RED-Key to play next video.");
24 :                 setInnerTextById("btn_red", "PLAY NEXT");
25 :                 break;
26 :
27 :         case 2:
28 :             if(isLGEBrowser){video.next();}
29 :             jsLog.lgmethod('video.next()');
30 :
31 :             setVideoDescription("Check whether NetCast advertising video is displayed.");
32 :             setInnerTextById("buttonDescription", "Press RED-Key to play next video.");
33 :             setInnerTextById("btn_red", "PLAY NEXT");
34 :             break;
35 :
36 :         case 3:
37 :             if(isLGEBrowser){ video.next(); }
38 :             jsLog.lgmethod('video.next()');
39 :
40 :             setVideoDescription("Check whether video with timer at the bottom is displayed.");
41 :             setInnerTextById("buttonDescription", "Press RED-Key to play previous video.");
42 :             setInnerTextById("btn_red", "PLAY PREVIOUS");
43 :             break;
44 :
45 :         case 4:
Setting Object
The following code shows how to set object.

03: Set data type. Refer to App Development Guide.
04-05: Set width and height.

```
<object id="video"
type="application/x-netcast-av"
width=300
height=250
data="../../mediafile/asx_test.asx"
stye="float: left">
autoStart=false>
</object>
```

Source Code of asx.html
Source code of asx.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>ASX Test Page</title>
<link rel="stylesheet" href="../../css/style.css">
<script language="javascript" src="../../js/common.js"></script>
<script language="javascript" src="../js/keycode.js"></script>
<script language="javascript" src="../js/menu.js"></script>
<script 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("Asx");
}
```
jsLog.initLG();

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

//check if this is LG Browser
var userAgent = new String(navigator.userAgent);
if (userAgent != null && userAgent.search(/LG Browser/) > -1)
isLGEBrowser = true;
else
    isLGEBrowser = false;
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 : changeTestProgress(); break;
    }
}

//set video description
function setVideoDescription(descriptionToSet)
{
    document.getElementById("video_description").style.zIndex = "2";
    document.getElementById("test_result").style.zIndex = "1";
    setInnerTextById("video_description", descriptionToSet);
}

//change test progress
var testStep = 1;
function changeTestProgress()
{
    var video = document.getElementById("video");
    var asxPropertiesDiv = document.getElementById("asxProperties");
    var descriptionDiv = document.getElementById("buttonDescription");
    var redButtonDiv = document.getElementById("btn_red");

    switch(testStep)
    {
        case 1:
            if(isLGEBrowser)
            {
                var video = document.getElementById("video");
                video.play(1);
                jsLog.lgmethod('video.play()');
            }
            else
            {
                setVideoDescription("This is not a LG Browser");

                setVideoDescription("Check whether snow boarding video is displayed.");
                setInnerTextById("buttonDescription", "Press RED-Key to play next test step in LG Browser.");
            }
            break;
    }
}
case 2:
if(isLGEBrowser){video.next();}
jsLog.lgmethod('video.next()');

setVideoDescription("Check whether NetCast advertising video is
displayed.");
setInnerTextById("buttonDescription", "Press RED-Key to play next
video.");
setInnerTextById("btn_red", "PLAY NEXT");
break;

case 3:
if(isLGEBrowser){video.next();}
jsLog.lgmethod('video.next()');

setVideoDescription("Check whether video with timer at the bottom is
displayed.");
setInnerTextById("buttonDescription", "Press RED-Key to play previous
video.");
setInnerTextById("btn_red", "PLAY PREVIOUS");
break;

case 4:
if(isLGEBrowser){video.previous();}
jsLog.lgmethod('video.previous()');

setVideoDescription("Check whether NetCast advertising video is
displayed.");
setInnerTextById("buttonDescription", "No more test is left on this
page.");
document.getElementById("btn_red").style.visibility = "hidden";
break;

testStep++;
}

</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/asx/asx.html</div>
</div>

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

<!-- test contents -->
<div class='ContentArea'>
  <div class='ApiListTitleArea'>API List</div>
  <div class='ApiListArea'>

```javascript

```
<div class='MethodTitleArea'>
Methods
</div>
<div class='MethodListArea'>
play()<br>
next()<br>
previous()<br>
</div>
</div>
<div class='PropertyTitleArea'>
Properties
</div>
</div>
</div>
</div>
</div>
</div>
</div>
</div>
</div>
</div>

<div class='ViewTitleArea'>
</div>
</div>
</div>

<iframe width=300 height=250 src="/ApiTutorial/mediafile/asx_test.asx" style="float: left" autoStart=false>

</iframe>
<!-- video description -->

</div>
<br><br>

<div id='buttonDescription' style="font-size: 30px; text-align: center; ">
Press RED-Key to play ASX.
</div>
</div>
</div>

<codeview>
<!-- button and copyright -->
<div class='SuiteButtonArea'>
  <!-- back key description -->
  <div id='btn_back' class='buttonDescription '><a>BACK</a></div>

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

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

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

</body>
</html>
Playing Video in Full Screen Mode

This section describes how to play video in full screen mode.

- Initializing the Page
- Inputting Keys
- Displaying and Playing Media Object in Full Screen Mode
- Setting Object
- Source Code of fullscreenvideo.html

This application shows how to play a video in full screen mode. The LG Smart TV SDK provides no separate Web Open API that plays media content in full screen mode.

Basically, when the start position of the media object is set to (0, 0) and its size is set to 1280 X 720, the content is played in full screen mode. In this application, the width and height are set within the <body> and </body> tags and the elements used to compose the screen are hidden using the code 'style.visibility = "hidden"' so that only the video is played in full screen mode.

**Needed APIs**

This application uses following Web Open API:

<table>
<thead>
<tr>
<th>API Class</th>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Method</td>
<td>Play()</td>
<td>Plays media.</td>
</tr>
<tr>
<td></td>
<td>NetCastLaunchQMENU</td>
<td>Launches QMENU over a full screen video.</td>
</tr>
<tr>
<td></td>
<td>NetCastLaunchRATIO</td>
<td>Sets aspect ratio of full screen video.</td>
</tr>
<tr>
<td>Property</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>Event</td>
<td>N/A</td>
<td>N/A</td>
</tr>
</tbody>
</table>

**Note**

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

**Initializing the Page**

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

04: Records the last visited page when running the application.
07: Initializes the page.
08: Gets the source code of the page using the XMLHttpRequest object.
09: Sets the page ID.
10: Initializes the Log function.
13-15: Registers an event handler which will executed when the corresponding button is pressed.

```javascript
01 : function initPage()
02 : {
03 : //save page as last visited page
04 : setLastVisitPage();
05 :
06 : //common initialize function
07 : commonInitialize();
08 : requestSourceCode();
09 : setPageID("Full Screen");
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_exit"),"click",onClickHandler);
16 :
17 : jsLog.lgobject('application/x-netcast-av');
18 : }
```

**Inputting Keys**

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

06: When the Back key is pressed, this code is executed.
07: When the Red key is pressed, `changeTestProgress()` is called.
08: When the Green key is pressed, `NetCastLaunchQMENU()` is called.
10: When the Yellow key is pressed, `NetCastLaunchRATIO()` is called.

```javascript
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 : changeTestProgress(); break;
08 : case VK_GREEN : window.NetCastLaunchQMENU();
09 : jsLog.lgmethod('NetCastLaunchQMENU()'); break;
10 : case VK_YELLOW : window.NetCastLaunchRATIO();
11 : jsLog.lgmethod('NetCastLaunchRATIO()'); break;
12 : }
13 : }
```

**Displaying and Playing Media Object in Full Screen Mode**

The following function displays and plays a media object in full screen mode.

**changeTestProgress**

Changes the test progress.
03-05: Declares the `video`, `bodycontent`, and `sourcecode` variables.
07: Plays the video.
09: Sets the visibility of the video to "visible" to display it on the screen.
11-12: Sets the visibility of both of the `bodycontent` and `sourcecode` to "hidden" to hide it from the screen.
13: Sets the visibility of the Red key to "hidden" to hide it from the menu bar.

```javascript
01 : function changeTestProgress()
02 : {
03 : var video = document.getElementById("video");
04 : var bodycontent = document.getElementById("bodycontent");
```
Setting Object
The following code shows how to set object.

03: Set data type. Refer to App Development Guide.
04-05: Set width and height.

```
01 : <object
02 : id="video"
03 : type="application/x-netcast-av"
04 : width=1280
05 : height=720
06 : style="float: left; z-index: 1; visibility: hidden;"
07 : autoStart=false
08 : data="../../mediafile/NetCastGeneratorClient.avi">
09 : </object>
```

Source Code of fullscreenvideo.html
Source code of fullscreenvideo.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>Full Screen Video Test Page</title>
<link rel="stylesheet" href="../../css/style.css">
<script language="javascript" src="../../js/common.js"></script>
<script language="javascript" src="../../js/keycode.js"></script>
<script language="javascript" src="../../js/menu.js"></script>
<script 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" />
<style type="text/css">
 .OverLay { position: relative; z-index:10; opacity: 1; filter: alpha(opacity=100); } 
body { height: 100%; } 
html { height: 100%; } 
</style>
<script>
 //initialize page
 function initPage() 
 {  
    //save page as last visited page
    setLastVisitPage();
    //common initialize function
```
commonInitialize();
requestSourceCode();
setPageID("Full Screen");
jsLog.initLG();

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

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

function onUserInput(userInput)
{
    switch(userInput)
    {
    case VK_BACK : window.location.replace("../menu_mediaPlayer.html"); break;
    case VK_RED : case 82 : changeTestProgress(); break;
    case VK_GREEN : window.NetCastLaunchQMENU();
        jsLog.logmethod('NetCastLaunchQMENU()'); break;
    case VK_YELLOW : window.NetCastLaunchRATIO();
        jsLog.logmethod('NetCastLaunchRATIO()'); break;
    }
}

function changeTestProgress()
{
    var video = document.getElementById("video");
    var bodycontent = document.getElementById("bodycontent");
    var sourcecode = document.getElementById("sourcecode");

    video.play(1);
    jsLog.logmethod('video.play(1)');
    video.style.visibility="visible";

    bodycontent.style.visibility="hidden";
    sourcecode.style.visibility = "hidden";
    document.getElementById("btn_red").style.visibility="hidden";
}
</script>
</head>

<body ondragstart='return false' onselectstart='return false'
onload="javascript:initPage();">
<div id='bodycontent'>
<div>
<object id="video"
type="application/x-netcast-av"
width="1280"
height="720"
style="float: left; z-index: 1; visibility: hidden;"
autoStart=false
data="../../mediafile/NetCastGeneratorClient.avi">
</object>
</div>
<div class='SuiteTitle' >LG Smart TV SDK | Web Open API Tutorial</div>

File: mediaplayer/app/fullscreenvideo.html

API List

Methods
- play()
- NetCastLaunchQMENU()
- NetCastLaunchRATIO()

Properties

Events

View

Source

Check whether NetCast advertising video is being displayed in full screen. Press RED Key to start test.