API

This guide explains features, functions, and how-to-use information of the APIs required to create the Web application for LG Smart TV.

Main Features of LG Smart TV SDK 3.0 APIs
This section introduces the key API functions of LG Smart TV SDK V3.0 released in February 2013. This is also provided in Korean document. You can download it from [DISCOVER > Legacy Platform (NetCast) > Technical Notes].

Web API Overview
LG Web API provides APIs for developers to create web applications on LG Smart TV. The section shows the LG Web API service blocks.

NetCast API
This section provides descriptions on NetCast API methods and proprietary events.

Media Player Plugin and API
This section provides descriptions on Media Player Plugin and API methods and properties.

Device Info Plugin and API
This section provides descriptions on Device Info Plugin and API methods, properties, and events.

Voice Recognition Plugin and API
This section provides descriptions on Voice Recognition Plugin and API methods, properties, and events.

AppToApp Plugin and API
This section provides descriptions on AppToApp Plugin and API methods and events.

Sound Plugin and API
This section provides descriptions on Sound Plugin and API methods.

DRMAgent Plugin and API
This section provides descriptions on DRMAgent Plugin and API methods and events.

Media Device NetCast API
This section provides descriptions on Media Device NetCast API methods.

Image Viewer Framework API
This section provides descriptions on Image Viewer Framework API methods.

Media Plugin Video Player Framework API
This section provides descriptions on Media Plugin Video Player Framework API methods.

HTML5 Video Player Framework API
This section provides descriptions on HTML5 Video Player Framework API methods.

Web UI Component API
This section provides descriptions on Web UI Component API methods and parameters.

Annex A Differences in Media Devices
Media products partly support NetCast 3.0. Please see the detailed information below for Media products development.

Annex B Utilities for Using Web UI Components
This section describes utilities for using LG Web UI components.
Contents

Main Features of LG Smart TV SDK 3.0 APIs ................................................................. 14
  New Features of LG Smart TV SDK 3.0 APIs ............................................................... 14
  Main Features of LG Smart TV SDK ............................................................................... 14
  Media ............................................................................................................................... 15
  Voice Recognition ........................................................................................................... 16
  AppToApp ....................................................................................................................... 17
  Sound .............................................................................................................................. 18
  UI Component ............................................................................................................... 19

Web API Overview ............................................................................................................. 20

NetCast API ....................................................................................................................... 22
  Methods/Proprietary Events ......................................................................................... 22
    Methods ....................................................................................................................... 23
      window.NetCastExit ................................................................................................. 23
      window.NetCastBack ............................................................................................... 23
      window.NetCastSetPageLoadingIcon ....................................................................... 24
      window.NetCastSetDefaultAspectRatio .................................................................. 25
      window.NetCastLaunchQMENU .............................................................................. 26
      window.NetCastLaunchRATIO ............................................................................... 27
      window.NetCastMouseOff ....................................................................................... 28
      window.NetCastGetMouseOnOff ............................................................................ 29
      window.NetCastSetAutoMouseOff .......................................................................... 29
      window.NetCastGetUsedMemorySize ..................................................................... 30
      window.NetCastSystemKeyboardVisible ................................................................. 30
    Proprietary Events ..................................................................................................... 32
      mouseon ................................................................................................................... 32
      mouseoff .................................................................................................................. 32
      outofmemory .......................................................................................................... 33

Media Player Plugin and API............................................................................................ 35
  Media Object ................................................................................................................. 35
    Properties .................................................................................................................... 36
      mode3D ...................................................................................................................... 36
      preBufferingTime ..................................................................................................... 36
      oneshot_url .............................................................................................................. 36
      subtitleOn / subtitle .............................................................................................. 37
      drm_type .................................................................................................................. 37
      playCount ............................................................................................................... 38
  Audio Object ................................................................................................................ 38
  Media Type Resolving in Media Player Plugin .............................................................. 38
  Methods/Properties/Events .......................................................................................... 39
    Methods ....................................................................................................................... 40
      play .......................................................................................................................... 40
      stop ......................................................................................................................... 41
      next .......................................................................................................................... 41
      previous ................................................................................................................... 41
      seek .......................................................................................................................... 42
<table>
<thead>
<tr>
<th>Method/Property</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>mediaPlayInfo</td>
<td>42</td>
</tr>
<tr>
<td>setWidevineXXX</td>
<td>44</td>
</tr>
<tr>
<td>Properties</td>
<td>46</td>
</tr>
<tr>
<td>version</td>
<td>46</td>
</tr>
<tr>
<td>type</td>
<td>46</td>
</tr>
<tr>
<td>data</td>
<td>46</td>
</tr>
<tr>
<td>width / height</td>
<td>47</td>
</tr>
<tr>
<td>playTime</td>
<td>47</td>
</tr>
<tr>
<td>playPosition</td>
<td>48</td>
</tr>
<tr>
<td>playState</td>
<td>48</td>
</tr>
<tr>
<td>error</td>
<td>49</td>
</tr>
<tr>
<td>autoStart</td>
<td>50</td>
</tr>
<tr>
<td>isScannable</td>
<td>50</td>
</tr>
<tr>
<td>speed</td>
<td>51</td>
</tr>
<tr>
<td>bufferingProgress</td>
<td>52</td>
</tr>
<tr>
<td>subtitleOn</td>
<td>52</td>
</tr>
<tr>
<td>subtitle</td>
<td>53</td>
</tr>
<tr>
<td>mode3D</td>
<td>53</td>
</tr>
<tr>
<td>audioLanguage</td>
<td>54</td>
</tr>
<tr>
<td>Events</td>
<td>56</td>
</tr>
<tr>
<td>onPlayStateChange</td>
<td>56</td>
</tr>
<tr>
<td>onBuffering</td>
<td>56</td>
</tr>
<tr>
<td>onError</td>
<td>57</td>
</tr>
<tr>
<td>onDRMRightsError</td>
<td>58</td>
</tr>
</tbody>
</table>

**Device Info Plugin and API**

<table>
<thead>
<tr>
<th>Property</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>59</td>
</tr>
<tr>
<td>Device Object</td>
<td>59</td>
</tr>
<tr>
<td>Methods/Properties</td>
<td>59</td>
</tr>
<tr>
<td>Methods</td>
<td>60</td>
</tr>
<tr>
<td>getLocalTime</td>
<td>60</td>
</tr>
<tr>
<td>getSystemTime</td>
<td>60</td>
</tr>
<tr>
<td>pentouchInfo</td>
<td>61</td>
</tr>
<tr>
<td>setDrmLicenseInfo</td>
<td>62</td>
</tr>
<tr>
<td>setVMConfigData</td>
<td>63</td>
</tr>
<tr>
<td>Properties</td>
<td>64</td>
</tr>
<tr>
<td>version</td>
<td>64</td>
</tr>
<tr>
<td>manufacturer</td>
<td>64</td>
</tr>
<tr>
<td>modelName</td>
<td>64</td>
</tr>
<tr>
<td>serialNumber</td>
<td>65</td>
</tr>
<tr>
<td>swVersion</td>
<td>65</td>
</tr>
<tr>
<td>hwVersion</td>
<td>66</td>
</tr>
<tr>
<td>SDKVersion</td>
<td>66</td>
</tr>
<tr>
<td>osdResolution</td>
<td>66</td>
</tr>
<tr>
<td>networkType</td>
<td>67</td>
</tr>
<tr>
<td>net_macAddress</td>
<td>67</td>
</tr>
<tr>
<td>drmClientInfo</td>
<td>68</td>
</tr>
<tr>
<td>net_dhcp</td>
<td>68</td>
</tr>
<tr>
<td>net_isConnected</td>
<td>69</td>
</tr>
<tr>
<td>net_hasIP</td>
<td>69</td>
</tr>
<tr>
<td>net_ipAddress</td>
<td>69</td>
</tr>
<tr>
<td>net_netmask</td>
<td>70</td>
</tr>
<tr>
<td>net_gateway</td>
<td>70</td>
</tr>
<tr>
<td>net_dns1</td>
<td>70</td>
</tr>
<tr>
<td>net_dns2</td>
<td>71</td>
</tr>
<tr>
<td>supportMouse</td>
<td>71</td>
</tr>
<tr>
<td>supportVoiceRecog</td>
<td>72</td>
</tr>
<tr>
<td>supportPentouch</td>
<td>72</td>
</tr>
<tr>
<td>support3D</td>
<td>73</td>
</tr>
<tr>
<td>support3DMode</td>
<td>73</td>
</tr>
</tbody>
</table>
Voice Recognition Plugin and API ................................................................. 78

   Voice Object ......................................................................................... 80
   Properties/Methods/Events ...................................................................... 80
       Properties .......................................................................................... 81
           isInitialized .................................................................................. 81
           isEnable ...................................................................................... 81
           dictation ..................................................................................... 81
           language .................................................................................... 82
       Methods ............................................................................................ 83
           startRecognition ........................................................................... 83
       Events ............................................................................................... 84
           onrecognizevoice .......................................................................... 84
           onbuttonenable .............................................................................. 84

AppToApp Plugin and API ........................................................................... 85

   AppToApp Object .................................................................................. 85
   Methods/Events .................................................................................... 85
       Methods ............................................................................................ 86
           sendMessage ............................................................................... 86
           sendMessageTo ............................................................................ 86
       Events ............................................................................................... 88
           onreceivingmessage .................................................................... 88

Sound Plugin and API .................................................................................. 89

   Sound Object ........................................................................................ 89
   Methods ............................................................................................... 89
       Method ............................................................................................. 90
           play ............................................................................................. 90

DRMAgent Plugin and API .......................................................................... 91

   DRMAgent Object .................................................................................. 91
   Methods/Events .................................................................................... 91
       Methods ............................................................................................ 92
           sendDRMMessage ....................................................................... 92
       Events ............................................................................................... 94
           onDRMMessageResult .................................................................. 94
       DRMAgent Specific ........................................................................... 95

Media Device NetCast API ......................................................................... 98
Image Viewer Framework API

How to Use Image Viewer Framework ...................................................... 99
Markup Interface ..................................................................................... 100
JS Interface .............................................................................................. 100
Methods .................................................................................................... 100

Methods
createImgViewer .................................................................................... 101
addImage ................................................................................................. 101
removeImage .......................................................................................... 102
gotoImage ................................................................................................ 103
zoom ........................................................................................................ 103
rotate ....................................................................................................... 104
flip ............................................................................................................. 105
startSlideShow ....................................................................................... 105
stopSlideShow ......................................................................................... 106
setSlideShowSpeed .................................................................................. 107
getSlideShowSpeed ................................................................................. 107
setRandomSlideShow .............................................................................. 108
getRandomSlideShow .............................................................................. 108
next ......................................................................................................... 108
previous .................................................................................................. 109
first .......................................................................................................... 109
last .......................................................................................................... 110
setDisplayMode ....................................................................................... 110
getDisplayMode ...................................................................................... 111
setFrameworkFocus ................................................................................ 112
removeFrameworkFocus ......................................................................... 112
mouseWheel ............................................................................................ 113
focusNext .............................................................................................. 113
focusPrevious ........................................................................................ 114
focusUp ................................................................................................... 114
focusDown .............................................................................................. 115
doSelection ............................................................................................ 115

Media Plugin Video Player Framework API ............................................. 116

How to Use Media Plugin Video Player Framework .................................. 117
Markup Interface ..................................................................................... 117
JS Interface .............................................................................................. 117

VideoPlayer Framework Implementation ................................................. 118
Methods .................................................................................................... 118

Methods
createVideoPlayer ................................................................................... 119
play .......................................................................................................... 119
stop ......................................................................................................... 120
rewind ...................................................................................................... 120
fastForward ............................................................................................. 121
switchToFullScreenMode ......................................................................... 121
optionMedia ........................................................................................... 122

HTML5 Video Player Framework API ...................................................... 123

How to Use HTML5 Video Player Framework ......................................... 124
Web UI Component API ................................................................. 130

AnimatedImage ........................................................................... 131

Markup Interface .................................................................. 132
    animation-hover ......................................................... 132
    animation-click ...................................................... 132
JS Interface (Constructor Parameters) ................................. 133
    animate ................................................................. 133
    id .......................................................................... 133
    selector .................................................................. 133
    style ..................................................................... 134
    src ...................................................................... 134
JS Interface (Methods) .......................................................... 135
    getAnimation .......................................................... 135
    getSrc .................................................................... 135
    getStyle ................................................................. 136
    setAnimation .......................................................... 136
    setSrc ................................................................. 136
    setValue ............................................................... 137

Button ...................................................................................... 137

Markup Interface .................................................................. 138
    icon-position ......................................................... 138
    icon-type .............................................................. 138
    text-inline ............................................................ 138
    text-omit .............................................................. 139
JS Interface (Constructor Parameters) ................................. 140
    href .................................................................. 140
    id ..................................................................... 140
    iconPosition ......................................................... 141
    iconType .............................................................. 141
    selector ................................................................. 141
    style ................................................................. 141
    textOmit ............................................................... 142
    textInline ............................................................. 142
    type ................................................................... 142
    value ................................................................. 142
JS Interface (Methods) .......................................................... 144
    getHref ................................................................. 144
    getIconPosition .................................................... 144
    getIconType ........................................................ 145
    getText ................................................................. 145
    getTextInline ........................................................ 146
    getTextOmit .......................................................... 146
    setHref ................................................................. 146
setIconPosition ................................................................. 147
setIconType ................................................................. 147
setText ................................................................. 147
setTextInline ................................................................. 148
setTextOmit ................................................................. 148

CheckBox ........................................................................ 149

Markup Interface ................................................................. 150
text-inline ................................................................. 150

JS Interface (Constructor Parameters) .................................................. 151
checked ................................................................. 151
id ................................................................. 151
selector ................................................................. 151
style ................................................................. 152
text ................................................................. 152
textInline ................................................................. 152

JS Interface (Methods) ................................................................. 153
cGetCheck ................................................................. 153
cGetText ................................................................. 153
cGetTextInline ................................................................. 154
cSetCheck ................................................................. 154
cSetText ................................................................. 154
cSetTextInline ................................................................. 155

Focus ........................................................................ 155

JS Interface (Constructor Parameters) .................................................. 157
id ................................................................. 157
option ................................................................. 157

JS Interface (Methods) ................................................................. 159
setOption ................................................................. 159

GroupButton ........................................................................ 159

JS Interface (Constructor Parameters) .................................................. 161
boxOrient ................................................................. 161
items ................................................................. 161
selector ................................................................. 162

JS Interface (Methods) ................................................................. 163
addItem ................................................................. 163
cGetBoxOrient ................................................................. 163
cGetItem ................................................................. 164
cRemoveItem ................................................................. 164
cSetBoxOrient ................................................................. 164
cSetItem ................................................................. 165

GroupCheckBox ........................................................................ 165

JS Interface (Constructor Parameters) .................................................. 167
boxOrient ................................................................. 167
items ................................................................. 168
legend ................................................................. 169
selector ................................................................. 169

JS Interface (Methods) ................................................................. 170
addImageItem .......................................................... 170
getImageItem .......................................................... 170
getLegend .............................................................. 171
removeItem ............................................................ 171
setBoxOrient ........................................................... 172
setItem ................................................................. 172
setLegend .............................................................. 173

GroupRadio ................................................................ 173

Markup Interface ........................................................ 174
box-orient ............................................................... 174

JS Interface (Constructor Parameters) ................................. 175
boxOrient ............................................................... 175
legend ................................................................. 175
id ..................................................................... 176
items ................................................................. 176
selector ............................................................... 176

JS Interface (Methods) ................................................... 178
addImageItem .......................................................... 178
getBoxOrient ........................................................ 178
getImageItem ........................................................ 179
getLegend ............................................................ 179
removeItem ........................................................... 179
setBoxOrient ........................................................ 180
setItem ................................................................. 180
setLegend .............................................................. 181

ImageList .................................................................. 181

Markup Interface ........................................................ 182

JS Interface (Constructor Parameters) ................................. 183
id ..................................................................... 183
image ................................................................. 183
selector ............................................................... 184
style ................................................................. 184

JS Interface (Methods) ................................................... 185
addImageItem .......................................................... 185
gGetCurrentIndex ..................................................... 185
moveImageItem ........................................................ 186
removeImageItem ..................................................... 186

ImageTile ................................................................ 186

Markup Interface ........................................................ 188

JS Interface (Constructor Parameters) ................................. 189
id ..................................................................... 189
option ................................................................. 190
selector ............................................................... 190
style ................................................................. 191
tables ................................................................. 191

JS Interface (Methods) ................................................... 193
gGetCurrentBackground ............................................. 193
gGetBorderSpacing .................................................... 193
gGetMargin .......................................................... 194
gGetNumToMove ...................................................... 194
gSetBackground ....................................................... 194
gSetBorderSpacing .................................................... 195
gSetMargin ........................................................... 195
gSetNumToMove ...................................................... 196

Label ..................................................................... 196

Markup Interface ........................................................ 197
<table>
<thead>
<tr>
<th>Function</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>JS Interface (Constructor Parameters)</td>
<td>198</td>
</tr>
<tr>
<td>id</td>
<td>198</td>
</tr>
<tr>
<td>selector</td>
<td>198</td>
</tr>
<tr>
<td>style</td>
<td>198</td>
</tr>
<tr>
<td>text</td>
<td>199</td>
</tr>
<tr>
<td>textInline</td>
<td>199</td>
</tr>
<tr>
<td>textScroll</td>
<td>199</td>
</tr>
<tr>
<td>textScrollDirection</td>
<td>199</td>
</tr>
<tr>
<td>JS Interface (Methods)</td>
<td>201</td>
</tr>
<tr>
<td>getText</td>
<td>201</td>
</tr>
<tr>
<td>getTextInline</td>
<td>201</td>
</tr>
<tr>
<td>getTextScroll</td>
<td>202</td>
</tr>
<tr>
<td>getTextScrollDirection</td>
<td>202</td>
</tr>
<tr>
<td>setTitle</td>
<td>202</td>
</tr>
<tr>
<td>setTitleRes</td>
<td>207</td>
</tr>
<tr>
<td>setMessage</td>
<td>207</td>
</tr>
<tr>
<td>messageRes</td>
<td>207</td>
</tr>
<tr>
<td>JS Interface (Constructor Parameters)</td>
<td>206</td>
</tr>
<tr>
<td>btnType</td>
<td>206</td>
</tr>
<tr>
<td>id</td>
<td>206</td>
</tr>
<tr>
<td>message</td>
<td>206</td>
</tr>
<tr>
<td>messageRes</td>
<td>207</td>
</tr>
<tr>
<td>title</td>
<td>207</td>
</tr>
<tr>
<td>titleRes</td>
<td>207</td>
</tr>
<tr>
<td>JS Interface (Methods)</td>
<td>208</td>
</tr>
<tr>
<td>addEventListener</td>
<td>208</td>
</tr>
<tr>
<td>close</td>
<td>208</td>
</tr>
<tr>
<td>doModal</td>
<td>209</td>
</tr>
<tr>
<td>getBtnType</td>
<td>209</td>
</tr>
<tr>
<td>getMessage</td>
<td>210</td>
</tr>
<tr>
<td>getTitle</td>
<td>210</td>
</tr>
<tr>
<td>setBtnType</td>
<td>210</td>
</tr>
<tr>
<td>setMessage</td>
<td>211</td>
</tr>
<tr>
<td>setMessageRes</td>
<td>211</td>
</tr>
<tr>
<td>setTitle</td>
<td>212</td>
</tr>
<tr>
<td>setTitleRes</td>
<td>212</td>
</tr>
<tr>
<td>Picker</td>
<td>212</td>
</tr>
<tr>
<td>Markup Interface</td>
<td>205</td>
</tr>
<tr>
<td>option</td>
<td>205</td>
</tr>
<tr>
<td>JS Interface (Constructor Parameters)</td>
<td>214</td>
</tr>
<tr>
<td>index</td>
<td>214</td>
</tr>
<tr>
<td>option</td>
<td>214</td>
</tr>
<tr>
<td>JS Interface (Constructor Parameters)</td>
<td>216</td>
</tr>
<tr>
<td>id</td>
<td>216</td>
</tr>
<tr>
<td>selector</td>
<td>216</td>
</tr>
<tr>
<td>text</td>
<td>217</td>
</tr>
<tr>
<td>textRes</td>
<td>217</td>
</tr>
<tr>
<td>type</td>
<td>217</td>
</tr>
<tr>
<td>width</td>
<td>218</td>
</tr>
<tr>
<td>JS Interface (Methods)</td>
<td>219</td>
</tr>
<tr>
<td>addItem</td>
<td>219</td>
</tr>
<tr>
<td>addItemWithRes</td>
<td>219</td>
</tr>
<tr>
<td>getOption</td>
<td>220</td>
</tr>
<tr>
<td>getSelected</td>
<td>220</td>
</tr>
<tr>
<td>removeItem</td>
<td>220</td>
</tr>
</tbody>
</table>
setOption ................................................................. 221
setPickerStyle .......................................................... 221
setSelect ................................................................. 222

Progress ........................................................................ 222

Markup Interface ............................................................ 223
JS Interface (Constructor Parameters) ................................ 224
    id .......................................................... 224
    max ................................................... 224
    min ................................................... 224
    style ................................................. 225
    selector ........................................... 225
    value ............................................. 225
JS Interface (Methods) .................................................. 227
    getMax ........................................ 227
    getMin ......................................... 227
    getStep ......................................... 228
    getValue ...................................... 228
    resetProgress ................................ 228
    setMax ...................................... 229
    setMin ...................................... 229
    setStep ...................................... 230
    setValue ................................... 230
    startProgress ................................ 231
    stopProgress ................................ 231

Radio ........................................................................... 231

Markup Interface ........................................................... 233
    text-inline ................................... 233
JS Interface (Constructor Parameters) ............................. 234
    checked ..................................... 234
    id .............................................. 234
    selector ..................................... 234
    style .......................................... 235
    text .......................................... 235
    textInline ................................... 235
JS Interface (Methods) .................................................... 236
    getCheck ..................................... 236
    getText ........................................ 236
    getTextInline .................................. 237
    setCheck ..................................... 237
    setText ....................................... 237
    setTextInline .................................. 238

Rating .......................................................................... 238

Markup Interface ............................................................ 239
JS Interface (Constructor Parameters) ................................ 240
    id .............................................. 240
    max ........................................... 240
    min ........................................... 240
    selector ..................................... 241
    style .......................................... 241
    value ........................................ 241
JS Interface (Methods) .................................................... 242
    getMax ..................................... 242
    getMin ..................................... 242
    getValue .................................... 243
    setMax ..................................... 243
    setMin ..................................... 243
    setValue .................................... 244
Scroller ....................................................................................................................... 244

Markup Interface ........................................................................................................ 246
  img-scroll-direction .............................................................................................. 246
JS Interface (Constructor Parameters) ........................................................................... 247
  id ............................................................................................................................. 247
  imgScrollDirection ................................................................................................. 247
  selector .................................................................................................................... 247
  style ......................................................................................................................... 248
JS Interface (Methods) ................................................................................................ 249
  addContents .............................................................................................................. 249
  getScrollDirection ................................................................................................. 249
  getScrollLock ......................................................................................................... 250
  setScrollDirection .................................................................................................. 250
  setScrollLock .......................................................................................................... 250

Slider ............................................................................................................................ 251

Markup Interface ........................................................................................................... 252
JS Interface (Constructor Parameters) ........................................................................... 253
  id ............................................................................................................................. 253
  max ......................................................................................................................... 253
  min .......................................................................................................................... 254
  style ......................................................................................................................... 254
  selector .................................................................................................................... 254
  value ....................................................................................................................... 254
JS Interface (Methods) ................................................................................................ 256
  getMax ..................................................................................................................... 256
  getMin ..................................................................................................................... 256
  getValue .................................................................................................................. 257
  setMax ..................................................................................................................... 257
  setMin ..................................................................................................................... 257
  setValue .................................................................................................................. 258

Tab .................................................................................................................................. 258

Markup Interface ........................................................................................................... 260
  index ....................................................................................................................... 260
JS Interface (Constructor Parameters) ........................................................................... 261
  id ............................................................................................................................. 261
  selector .................................................................................................................... 261
  text .......................................................................................................................... 261
JS Interface (Methods) ................................................................................................ 263
  addTab ..................................................................................................................... 263
  getSelected .............................................................................................................. 263
  getTabHeight .......................................................................................................... 264
  getTabWidth .......................................................................................................... 264
  setSelected ............................................................................................................. 265
  setTab ...................................................................................................................... 265
  setTabHeight .......................................................................................................... 265
  setTabWidth .......................................................................................................... 266

TextInput ....................................................................................................................... 266

Markup Interface ........................................................................................................... 268
  text-inline .............................................................................................................. 268
JS Interface (Constructor Parameters) ........................................................................... 269
  id ............................................................................................................................. 269
  selector .................................................................................................................... 269
  style ......................................................................................................................... 269
  text .......................................................................................................................... 270
  textinline ............................................................................................................... 270
  type .......................................................................................................................... 270
JS Interface (Methods) ................................................................................................ 271
getText..............................................................271
getTextInline .......................................................271
setText ............................................................272
setTextInline ......................................................272
ToggleSwitch .........................................................272
MarkUp Interface ....................................................274
text-inline ..........................................................274
JS Interface (Constructor Parameters) .........................275
id .................................................................275
on ...............................................................275
off ...............................................................275
selector .........................................................276
textinline .........................................................276
JS Interface (Methods) .............................................277
getSelected ......................................................277
getText ...........................................................277
g getTextInline .......................................................278
setTextRes .......................................................278
setSelected ....................................................278
setText ...........................................................279
setTextInline .......................................................279
TvHelp .............................................................280
MarkUp Interface ....................................................281
option ...........................................................281
JS Interface (Constructor Parameters) .........................282
id .................................................................282
items ...........................................................282
selector .........................................................283
style ............................................................283
JS Interface (Methods) .............................................284
addItem ..........................................................284
getLink ..........................................................284
getOption .......................................................285
getText ...........................................................285
removeItem .......................................................286
setItem ..........................................................286
PageEffects ........................................................286
JS Interface (Constructor Parameters) .........................288
currentPage ......................................................288
JS Interface (Methods) .............................................289
execute ..........................................................289
FadeEffect ........................................................290
FlipEffect ........................................................292
PopEffect ........................................................294
SlideEffect .........................................................296

Annex A Differences in Media Devices ..................................................298

Annex B Utilities for Using Web UI Components .................................299

Converting HTML Components into LG Components .........................299
JavaScript Handler for UI Components ....................................299
Defining the Next Focus .............................................300
Adding attribute in lge-attr ........................................301
Adding parameters in JavaScript constructor.................................................. 302
Calling direction key event............................................................................. 303
Main Features of LG Smart TV SDK 3.0 APIs

This document outlines the key API functions of LG Smart TV SDK V3.0 released in February 2013. It was created to help developers, content planners and department officials gain a better understanding of LG Smart TV applications (apps).

This chapter includes the following sections.
• New Features of LG Smart TV SDK 3.0 APIs
• Main Features of LG Smart TV SDK

New Features of LG Smart TV SDK 3.0 APIs

APIs provide the Web app programming environment for the LG Smart TV NetCast Platform. In particular, they provide an abundance of media playback functions, device information, DRM management functions and adaptive streaming solutions.

From the previous SDK version, this SDK 3.0 version added and modified a new API on Device Info Plugin and API, NetCast API.

NetCast API
- window.NetCastSetAutoMouseOff is added
- window.NetCastSystemKeyboardVisible is added

Device Info Plugin and API
- support3DMode is added

Main Features of LG Smart TV SDK

NetCast means that the platform is compatible with LG Smart TVs.

Developers can use the APIs provided by the LG Smart TV SDK to drive the NetCast Platform and develop creative and useful apps. This document outlines the following key functions:

- Media
- Voice Recognition
- AppToApp
- Sound
- UI Component
Media

The NetCast Platform supports the two following video playback methods in apps:

<table>
<thead>
<tr>
<th>Video Playback Method</th>
<th>Supported Streaming Protocol</th>
<th>DRM</th>
</tr>
</thead>
<tbody>
<tr>
<td>HTML5 Video Tag</td>
<td>HTTP (DRM not supported)</td>
<td></td>
</tr>
<tr>
<td>Media Object</td>
<td>HTTP, MMSH (for VOD media delivery)</td>
<td>PlayReady, Widevine, Verimatrix</td>
</tr>
<tr>
<td></td>
<td>HLS, Widevine (for Linear or live broadcasting)</td>
<td></td>
</tr>
</tbody>
</table>

Supported SDK and Emulator

- LG Smart TV SDK V1.5 or newer
- LG Smart TV Emulator 2011: HTML5 video tags are not supported, but media objects are supported.
- LG Smart TV Emulator 2012: Both HTML5 video tags and media objects are supported.

Related Document

See [Media Player Plugin and API](#) for detailed information.
Voice Recognition

The NetCast Platform makes it possible to use the Magic Remote’s voice recognition function in Smart TV Web apps. It provides the API for converting voice into text and allows users to use voice recognition to input characters while using their Smart TV, for example, when searching content and inputting their account details.

There are two voice recognition modes:
- Word mode: Keyword-level voice recognition (up to three similar voice recognition results will be displayed).
- Dictation mode: Sentence-level voice recognition (only one recognition result will be displayed).

The voice recognition plugin and API are supported since NetCast 3.0. The supported languages for the voice recognition are listed in the Voice Recognition Plugin and API.

Supported SDK and Emulator
- H12: Korea/North America/Europe/CIS/Australia/Brazil/China
- M12: Korea
- See the Voice Recognition Plugin and API for information on the supported languages in different regions.

Related Document
See Voice Recognition Plugin and API for detailed information.

Example of an Application

[Figure] Recognizing keywords using the voice recognition function in the search section of the Social Center
AppToApp

The NetCast Platform allows a Smart TV app and a mobile app (Android and iOS based), which can be called a second screen or companion app, to communicate with each other through the TV. An LG Smart TV app (host) uses the AppToApp API to communicate with a mobile app. The mobile app (controller) discovers, pairs with, launches, and communicates with the host based on the UDAP (Universal Discovery & Access Protocol) defined by LG.

Developers can offer a wide range of second screen-based experiences to Smart TV users by enabling interworking between Smart TV and mobile devices.

Currently, the AppToApp API specifications, UDAP, and UDAP service profile specifications are available separately. Sample code for mobile apps (controllers) will also be provided for easier implementation.

Supported SDK and Emulator
- LG Smart TV SDK V2.2 or newer
- LG Smart TV Emulator 2011: not supported
- LG Smart TV Emulator 2012: SDK 2.2 or newer

Related Document
See AppToApp Plugin and API for detailed information.

Example of an Application
The following app was developed using the AppToApp function. If users share the URL for the Webpage they wish to view on a big screen through the WatchBig! App, the webpage that appears on their mobile phones can be viewed on their Smart TV via its Web browser. The WatchBig! app will be available to download from LG Smart World (LG Apps TV) soon.

[Figure] LG Electronics Super App Contest Winner: WatchBig!
Sound
The NetCast Platform enables sound effects (a short WAVE file) to be inserted into the background music of a Smart TV video or program.
If the sound effect and background music are played at the same time, the Media Player plugin must be used to play the background music. It is also possible to play the sound effect on its own, without any background music. Several audio plugins can also be used simultaneously.

The WAVE files currently available for sound effects are as follows:
- Sampling rate: 44.1 K
- Audio bps: 16 bit
- Audio channel: mono

Supported SDK and Emulator
- LG Smart TV SDK V2.2 or newer
- LG Smart TV Emulator 2011: not supported
- LG Smart TV Emulator 2012: SDK 2.2 or newer

Related Document
See Sound Plugin and API for detailed information.
UI Component

The NetCast Platform provides Web UI components, whose look and feel are unique to LG Electronics. Developers can use the LG Web UI Components to develop apps that are compatible with the UI/UX of LG Smart TVs.

To use LG Web UI Components, you must install LG Smart TV SDK V2.2.0 or newer. You can generate UI components easily by means of drag and drop using the WYSIWYG Editor in the LG IDE. UI Components will be generated using the standard HTML markup or LG JavaScript interface.

Supported SDK and Emulator

- LG Smart TV SDK V2.2 or newer
- LG Smart TV Emulator 2011: not supported
- LG Smart TV Emulator 2012: SDK V2.2 or newer

Related Document

See [Web UI Component API](#) for detailed information.

Example of an Application

[Figure] Examples of apps using LG Web UI Components
Web API Overview

LG Web API provides APIs for developers to create web applications on LG Smart TV. The following figure illustrates the LG Web API service blocks.

Web Open API

[Figure] Service Block Diagram of LG Web API

**Window Extended (NetCast)**
NetCast supports the proprietary browser APIs for developers to use.

**Media Player**
Media Player Plugin and API contain CE-HTML and CEA-2014 compliant media player plugin.

**Device**
Device Info Plugin and API are for obtaining device information in the application.

**Voice Recognition**
Voice Recognition Plugin and API are for using voice recognition function of Magic Remote in the application.

**AppToApp**
AppToApp Plugin and API are used for interaction between LG Smart TV and external smart device such as Smart phone.

**Sound**
Sound Plugin and API are used for playing sound effect, a short wav file, while background music is being played using Media Player plugin. (It is possible to use Sound plugin with no background music being played.)

**DRMAgent**
DRMAgent Plugin and API are used for the generic playout of PlayReady content using the NetCast media object and using the drmAgent object.

**Media Device**
Media Device Plugin and API are used for PDP TV and Media device, not for LCD/LED TV device.

**Image Viewer Framework**
Image Viewer Framework API provides the functionalities of a photo gallery and image viewer.

**Media Plugin Video Player Framework**
Media Plugin Video Player Framework API provides the video player and its functionalities using the media object.

**HTML5 Video Player Framework**
HTML5 Video Player Framework API provides the video player and its functionalities using the HTML5 Video element.
**Web UI Component**

LG Smart TV SDK provides LG Web UI Components for developing LG Smart TV web applications. The LG Web UI components are designed to comply with the look and feel of LG Smart TV’s UI/UX. LG provides various components including Button, CheckBox, Focus and more. Also, 4 types of page effects are provided.

---

**Note**

According to our policy, the following Web Open APIs will not be compatible with the LG Smart TV 2014 platform.

- **AppToApp APIs** (Since LG will provide a new library that has the UDAP function in the next year, UDAP will not be supported in the LG Smart TV 2014 platform. If you want to deploy your app implemented using UDAP on the LG Smart TV 2014 platform, it must be partly reimplemented using new library. For more information, detailed reimplementation guide will be provided later.)

- **SNS APIs** (Removed from LG Web Open API on June, 2013.)
NetCast API

Following sections describe the proprietary browser APIs for LG Smart TV application authors to use.

- Methods/Proprietary Events

Methods/Proprietary Events

Methods and proprietary events are listed in the following sections.

- Methods
- Proprietary Events
Methods
Methods of NetCast API are as follows:

window.NetCastExit

Description
The NetCast Platform provides a proprietary API, ‘window.NetCastExit()’, to implement the exit function to AV. A JavaScript application can use this API for users to exit or quit the application to AV.

Note
Since LG Smart TV’s UX Guideline is changed, we do not recommend to use this API anymore. When an application is closed, you must use the window.NetCastBack() API to go back to the previous menu.

Syntax
window.NetCastExit();

Parameters
None

Return Value
None

Example
function processExit()
{
    window.NetCastExit();
}

Supported SDK / Emulator Version

<table>
<thead>
<tr>
<th>SDK Version</th>
<th>Emulator Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.5 or higher</td>
<td>LG Smart TV Emulator 2011 : SDK 1.5 or higher</td>
</tr>
<tr>
<td></td>
<td>LG Smart TV Emulator 2012 : SDK 2.0 or higher</td>
</tr>
<tr>
<td></td>
<td>LG Smart TV Emulator 2013 : SDK 3.0 or higher</td>
</tr>
</tbody>
</table>

window.NetCastBack

Description
The NetCast Platform provides a proprietary API, ‘window.NetCastBack()’, to implement the back function to the previous NetCast menu. A JavaScript application can use this API for users to move back to previous NetCast menu.

Syntax
window.NetCastBack();

Parameters
None

Return Value
None

Example
function processBack()
{
    if(window.NetCastBack) {
        window.NetCastBack();
    }
}

### Supported SDK / Emulator Version

<table>
<thead>
<tr>
<th>SDK Version</th>
<th>Emulator Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.5 or higher</td>
<td>LG Smart TV Emulator 2011 : SDK 1.5 or higher</td>
</tr>
<tr>
<td></td>
<td>LG Smart TV Emulator 2012 : SDK 2.0 or higher</td>
</tr>
<tr>
<td></td>
<td>LG Smart TV Emulator 2013 : SDK 3.0 or higher</td>
</tr>
</tbody>
</table>

**window.NetCastSetPageLoadingIcon**

#### Description
It is recommended that developers provide a “loading” icon so that users are provided with an indication of the latency of data downloading from a server. Developers can implement this feature using JavaScript, however, it may not be possible to do this, as there would not be any JavaScript running while a HTML page is loading. The NetCast Platform supports a proprietary API, `window.NetCastPageLoadingIcon()`, to provide the web application’s own page loading animation function. Developers can use this API during HTML page loading.

#### Note
This function will be applied while the next page is loaded.

The following examples show how application authors can enable and disable the browser’s page loading icon.

#### Syntax
`window.NetCastSetPageLoadingIcon(control);`

#### Parameters
- `control` [in] 'enabled' or 'disabled'

#### Return Value
None

#### Example
```javascript
function enablePageLoadingIcon() {
    window.NetCastSetPageLoadingIcon('enabled');
}
```

```javascript
function disablePageLoadingIcon() {
    window.NetCastSetPageLoadingIcon('disabled');
}
```
Developers may want to set the default aspect ratio for users to view full screen video with the correct aspect ratio. The NetCast Platform allows developers to set the default aspect ratio by using the `window.NetCastSetDefaultAspectRatio()` API. The setting only applies if the video runs in full screen mode, 1280 x 720. This API takes a string type of argument. The list of arguments and their behaviors are listed in the following table and figure.

**Supported SDK / Emulator Version**

<table>
<thead>
<tr>
<th>SDK Version</th>
<th>Emulator Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.5 or higher</td>
<td>LG Smart TV Emulator 2011 : SDK 1.5 or higher</td>
</tr>
<tr>
<td></td>
<td>LG Smart TV Emulator 2012 : SDK 2.0 or higher</td>
</tr>
<tr>
<td></td>
<td>LG Smart TV Emulator 2013 : SDK 3.0 or higher</td>
</tr>
</tbody>
</table>

**Description**

This API call applies only once for a whole application life cycle. Second and subsequent calls will be ignored by the LG Browser automatically. Therefore, it is recommended to locate the API call at the time of launching the application. If the application is launched again after exiting, the API will be enabled again.
Syntax

```javascript
window.NetCastSetDefaultAspectRatio(control);
```

Parameters

control 
[in] Aspect ratio control mode

[Table] Explanation of aspect ratio control mode

<table>
<thead>
<tr>
<th>Aspect ratio control mode</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>original</td>
<td>view original video image with original correct aspect ratio (TV screen may not be filled with video image)</td>
</tr>
<tr>
<td>zoom</td>
<td>fill the full TV screen with original aspect ratio video (there may be some cropping of the original video image)</td>
</tr>
<tr>
<td>full</td>
<td>fill the full TV screen with video (aspect ratio may be distorted, but with no loss of original video image)</td>
</tr>
</tbody>
</table>

Return Value

None

Example

```javascript
function setDefaultCloseOperation()
{
    window.NetCastSetDefaultAspectRatio('original');
}
```

Supported SDK / Emulator Version

<table>
<thead>
<tr>
<th>SDK Version</th>
<th>Emulator Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.5 or higher</td>
<td>LG Smart TV Emulator 2011 : SDK 1.5 or higher</td>
</tr>
<tr>
<td></td>
<td>LG Smart TV Emulator 2012 : SDK 2.0 or higher</td>
</tr>
<tr>
<td></td>
<td>LG Smart TV Emulator 2013 : SDK 3.0 or higher</td>
</tr>
</tbody>
</table>

`window.NetCastLaunchQMENU`
Description
The NetCast Platform provides a 'QMENU' (Quick Menu for Audio and Video Adjustment) for users to setup the aspect ratio for full screen video, picture quality adjustment and audio adjustment. The QMENU can be launched by users when playing video in full screen mode by pressing the 'QMENU' button on the remote control. This only works in full screen video mode.

Note, it is possible for users to operate a LG Smart TV application using the Magic Remote, the pointing device of the NetCast Platform. There is no 'QMENU' button on the Magic Remote, Therefore, it is strongly recommended that the LG Smart TV developer implements a graphical user interface to launch the QMENU over a full screen video.

The NetCast Platform thus provides a proprietary API, `window.NetCastLaunchQMENU()` to enable this feature. If this API is called then the LG Smart TV will overlay the QMENU on the full screen video. Developers can launch the QMENU over a full screen video using this API.

Syntax
```
window.NetCastLaunchQMENU();
```

Parameters
None

Return Value
None

Example
```
function launchQMENU()
{
    if(window.NetCastLaunchQMENU) {
        window.NetCastLaunchQMENU();
    }
}
```

Supported SDK / Emulator Version

<table>
<thead>
<tr>
<th>SDK Version</th>
<th>Emulator Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.5 or higher</td>
<td>LG Smart TV Emulator 2011 : SDK 1.5 or higher</td>
</tr>
<tr>
<td></td>
<td>LG Smart TV Emulator 2012 : SDK 2.0 or higher</td>
</tr>
<tr>
<td></td>
<td>LG Smart TV Emulator 2013 : SDK 3.0 or higher</td>
</tr>
</tbody>
</table>

window.NetCastLaunchRATIO

Description
The NetCast Platform provides a 'RATIO' (Aspect Ratio Control) menu for users to setup the aspect ratio of a full screen video. The RATIO menu can be launched by a user while playing a video in a full screen mode by pressing the 'RATIO' button on the remote control. This only works in full screen video mode.

Note, it is possible for users to operate a LG Smart TV application using the Magic Remote, the pointing device of NetCast Platform. There is no 'RATIO' button on the Magic Remote, therefore, it is strongly recommended that the LG Smart TV developer implements a graphical user interface to launch the RATIO menu over a full screen video.

The NetCast Platform thus provides a proprietary API 'window.NetCastLaunchRATIO()' to enable this feature. If this API is called then the LG Smart TV will overlay the RATIO menu on the full screen video. Developers can launch the RATIO over a full screen video using this API.

Syntax
```
window.NetCastLaunchRATIO();
```

Parameters
None

Return Value
Example

```javascript
function launchRATIO()
{
    if(window.NetCastLaunchRATIO) {
        window.NetCastLaunchRATIO();
    }
}
```

### Supported SDK / Emulator Version

<table>
<thead>
<tr>
<th>SDK Version</th>
<th>Emulator Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.5 or higher</td>
<td>LG Smart TV Emulator 2011 : SDK 1.5 or higher</td>
</tr>
<tr>
<td></td>
<td>LG Smart TV Emulator 2012 : SDK 2.0 or higher</td>
</tr>
<tr>
<td></td>
<td>LG Smart TV Emulator 2013 : SDK 3.0 or higher</td>
</tr>
</tbody>
</table>

---

**window.NetCastMouseOff**

**Description**

This API can be used by an LG Smart TV developer to deactivate the Magic Remote and its pointer.

Refer to the section ‘Input Device’ and ‘userAgent String’ in Developing > Developing Web App > App Development Guide section in this Library, and `supportMouse`, `mouseon`, `mouseoff`, and `window.NetCastGetMouseOnOff()` for more information about the Magic Remote and its status related event and API.

Developers can deactivate the Magic Remote and its pointer using this API. In the following example, the “time” parameter is the time value after which the deactivation is applied. This parameter is processed to “second” level accuracy. For example, the Magic Remote pointer would disappear 5 seconds after calling “window.NetCastMouseOff(5);”.

**Note**

In NetCast 3.0 or higher, the mouse gets deactivated when the halt of the mouse movement continues for 3 seconds. Therefore, this function is supported only for backward compatibility and it does not do anything.

In other words, `window.NetCastMouseOff` function is ONLY activated in NetCast 2.0 (the model of 2011).

**Syntax**

```javascript
window.NetCastMouseOff(time);
```

**Parameters**

- **time**
  
  [in] Time value after which the deactivation is applied (in second)

**Return Value**

None

**Example**

```javascript
function mouseOff(time)
{
    if(window.NetCastMouseOff){
        window.NetCastMouseOff(time);
    }
}
```

### Supported SDK / Emulator Version

<table>
<thead>
<tr>
<th>SDK Version</th>
<th>Emulator Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.5 or higher</td>
<td>LG Smart TV Emulator 2011 : SDK 1.5 or higher</td>
</tr>
<tr>
<td></td>
<td>LG Smart TV Emulator 2012 : SDK 2.2 or higher</td>
</tr>
</tbody>
</table>
window.NetCastGetMouseOnOff

Description
This API can be used by an LG Smart TV developer to get the on or off status of the Magic Remote.

Refer to the section ‘Input Device’ and ‘userAgent String’ in Developing > Developing Web App > App Development Guide section in this Library, and supportMouse, mouseon, mouseoff, and window.NetCastGetMouseOff(time) for more information about the Magic Remote and its status related event and API.

Syntax
window.NetCastGetMouseOnOff();

Parameters
None

Return Value
Return value is “on” or “off”, the mouse status. (String type)

Example
var mouseOnOffStatus;

function getMouseOnOff()
{
    if(window.NetCastGetMouseOnOff) {
        mouseOnOffStatus = window.NetCastGetMouseOnOff();
    }
}

Supported SDK / Emulator Version

<table>
<thead>
<tr>
<th>SDK Version</th>
<th>Emulator Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.5 or higher</td>
<td>LG Smart TV Emulator 2011 : SDK 1.5 or higher</td>
</tr>
<tr>
<td></td>
<td>LG Smart TV Emulator 2012 : SDK 2.0 or higher</td>
</tr>
<tr>
<td></td>
<td>LG Smart TV Emulator 2013 : SDK 3.0 or higher</td>
</tr>
</tbody>
</table>

window.NetCastSetAutoMouseOff

Description
This API can be used by an LG Smart TV developer to set the auto mouse off property.

LG Smart TV turns off pointing feature automatically, if a user does not use the pointer of a mouse or Magic Remote for a specific time. However, some apps need to maintain the pointer at the specific place for a specific time. In this case, ‘NetCastSetAutoMouseOff’ API protects the mouse not to be turned off automatically.

Even when the mouse is set not to be turned off automatically, the pointing feature becomes off when a user presses the key of a common remote control.

Syntax
window.NetCastSetAutoMouseOff(control);
control [in] 'enable' or 'disable'

Return Value
None

Example
if(window.NetCastSetAutoMouseOff) {
    window.NetCastSetAutoMouseOff('disable');
}

Supported SDK / Emulator Version

<table>
<thead>
<tr>
<th>SDK Version</th>
<th>Emulator Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.0 or higher</td>
<td>LG Smart TV Emulator 2011: Not Supported</td>
</tr>
<tr>
<td></td>
<td>LG Smart TV Emulator 2012: Not Supported</td>
</tr>
<tr>
<td></td>
<td>LG Smart TV Emulator 2013: Not Supported</td>
</tr>
</tbody>
</table>

window.NetCastGetUsedMemorySize

Description
This API can be used by an LG Smart TV developer to get the total memory size used by the web application.

Syntax
window.NetCastGetUsedMemorySize();

Parameters
None

Return Value
Returns the memory size used in the application. (Int type)

Example
var usedMemorySize;

function getUsedMemorySize()
{
    if(window.NetCastGetUsedMemorySize) {
        usedMemorySize = window.NetCastGetUsedMemorySize();
    }
}

Supported SDK / Emulator Version

<table>
<thead>
<tr>
<th>SDK Version</th>
<th>Emulator Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.5 or higher</td>
<td>LG Smart TV Emulator 2011: Not Supported</td>
</tr>
<tr>
<td></td>
<td>LG Smart TV Emulator 2012: SDK 2.2 or higher</td>
</tr>
<tr>
<td></td>
<td>LG Smart TV Emulator 2013: SDK 3.0 or higher</td>
</tr>
</tbody>
</table>

window.NetCastSystemKeyboardVisible

Description
The NetCast Platform provides a proprietary API, ‘window.NetCastSystemKeyboardVisible’ to use the system keyboard of NetCast Platform.
To use a system keyboard at an application, set the input parameter as TRUE. To use the JavaScript keyboard provided by an application, set the input parameter as FALSE. If an application does not call this API, the JavaScript keyboard provided by an application is assumed.
The NetCast Platform supports the system keyboard since the NetCast 4.0 (2013)

Syntax

window.NetCastSystemKeyboardVisible(boolean);

Parameters

boolean [in] true or false

Return Value

None

Example

window.NetCastSystemKeyboardVisible(true);

Supported SDK / Emulator Version

<table>
<thead>
<tr>
<th>SDK Version</th>
<th>Emulator Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.0 or higher</td>
<td>LG Smart TV Emulator 2011 : Not Supported</td>
</tr>
<tr>
<td></td>
<td>LG Smart TV Emulator 2012 : Not Supported</td>
</tr>
<tr>
<td></td>
<td>LG Smart TV Emulator 2013 : Not Supported</td>
</tr>
</tbody>
</table>
Proprietary Events

Proprietary events of NetCast API are as follows:

**mouseover**

**Description**
This event is generated when the Magic Remote is activated. Refer to the section ‘Input Device’ in *Developing > Developing Web App > App Development Guide* section in this Library, and *Device Info API (Properties)* and *supportMouse* for more detailed information about the Magic Remote and input devices.

The following examples illustrate how developers can register the “mouseover” event handler in three ways.

**Example**

```html
// Registering "mouseover" event (in body tag)
<body
   onkeydown='processKeyDown(event)'
   style='margin:0; font-family:TiresiasScreenfont'
   ondragstart='return false'
   onselectstart='return false'
   onmouseon='mouseon_handler()'>
   . . .
</body>
```

```javascript
// Registering "mouseover" event (window property)
<script language='javascript'>
   if(window.onmouseon) {
      window.onmouseon = mouseon_handler;
   }
</script>
```

```javascript
// Registering "mouseover" event (DOM Level 2 event)
<script language='javascript'>
   if(window.onmouseover) {
      window.addEventListener('mouseover', mouseon_handler, true);
   }
</script>
```

See Also
- *window.NetCastMouseOff*
- *window.NetCastGetMouseOnOff*

**Supported SDK / Emulator Version**

<table>
<thead>
<tr>
<th>SDK Version</th>
<th>Emulator Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.5 or higher</td>
<td>LG Smart TV Emulator 2011: Not Supported</td>
</tr>
<tr>
<td></td>
<td>LG Smart TV Emulator 2012: SDK 2.2 or higher</td>
</tr>
<tr>
<td></td>
<td>LG Smart TV Emulator 2013: SDK 3.0 or higher</td>
</tr>
</tbody>
</table>

**mouseout**

**Description**
This event is generated when the Magic Remote is deactivated. Refer to the section ‘Input Device’ and ‘userAgent String’ in *Developing > Developing Web App > App Development Guide* section in this Library, and *supportMouse* for more detailed information about the Magic Remote and input devices.

The following examples illustrate how developers can register the “mouseout” event handler in three ways.

**Example**
// Registering “mouseoff” event (in body tag)
<body
    onkeydown='processKeyDown(event)'
    style='margin:0; font-family:TiresiasScreenfont'
    ondragstart='return false'
    onselectstart='return false'
    onmouseoff='mouseoff_handler()'>
</body>

// Registering “mouseoff” event (window property)
<script language='javascript'>
    if(window.onmouseoff) {
        window.onmouseoff = mouseoff_handler;
    }
</script>

// Registering “mouseoff” event (DOM Level 2 event)
<script language='javascript'>
    if(window.onmouseoff) {
        window.addEventListener('mouseoff', mouseoff_handler, true);
    }
</script>

See Also
window.NetCastMouseOff
window.NetCastGetMouseOnOff

Supported SDK / Emulator Version

<table>
<thead>
<tr>
<th>SDK Version</th>
<th>Emulator Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.5 or higher</td>
<td>LG Smart TV Emulator 2011 : Not Supported</td>
</tr>
<tr>
<td></td>
<td>LG Smart TV Emulator 2012 : SDK 2.2 or higher</td>
</tr>
<tr>
<td></td>
<td>LG Smart TV Emulator 2013 : SDK 3.0 or higher</td>
</tr>
</tbody>
</table>

outofmemory

Description
Web application gets force shut down when the TV system memory goes under 20 MB. The ‘outofmemory’ event gives the remaining memory size to developer before the web application gets force shut down. This event occurs three times (at about 50 MB (20 MB + 30 MB), 40 MB (20 MB + 20 MB), 30 MB (20 MB + 10 MB)). If you use ‘event.available’ property, you can get the current available memory size.

The following examples illustrate how developer can register the ‘outofmemory’ event handler in three ways.

Example
// Registering “outofmemory” event (in body tag)
<body
    onkeydown='processKeyDown(event)'
    style='margin:0; font-family:TiresiasScreenfont'
    ondragstart='return false'
    onselectstart='return false'
    onoutofmemory = 'outofmemory_handler()'>
    ...
</body>

// Registering “outofmemory” event (window property)
<script language='javascript'>
    if(window.onoutofmemory) {
        window.onoutofmemory = outofmemory_handler;
    }
</script>
// Registering "outofmemory" event (DOM Level 2 event)

```javascript
if(window.onoutofmemory) {
    window.addEventListener(outofmemory, outofmemory_handler, true);
}
</script>

## Supported SDK / Emulator Version

<table>
<thead>
<tr>
<th>SDK Version</th>
<th>Emulator Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.5 or higher</td>
<td>LG Smart TV Emulator 2011 : Not Supported</td>
</tr>
<tr>
<td></td>
<td>LG Smart TV Emulator 2012 : Not Supported</td>
</tr>
<tr>
<td></td>
<td>LG Smart TV Emulator 2013 : Not Supported</td>
</tr>
</tbody>
</table>
Media Player Plugin and API

The NetCast Platform supports CE-HTML and CEA-2014 compliant media player plugin. The following sub-sections describe the API of CE-HTML compliant media player plugin. The supported API list is a subset of CE-HTML, and it is not the whole set of media player APIs as provided by CE-HTML.

- Media Object
- Audio Object
- Media Type Resolving in Media Player Plugin
- Methods/Properties/Events

Media Object

An example MIME type for the media object is video/x-ms-wmv for NetCast Platform media player plugin. The media object supports the data URL, width, height, id, preBufferingTime, oneshot_url, subtitle, subtitleOn and drm_type, and playCount properties.

The NetCast Platform supports only one instance of the media object at any one time, so developers must not attempt to use more than one media object simultaneously.

Refer to Annex A Complete List of Supported MIME Types in Developing > Developing Web App > App Development Guide section in this Library. See also Media Type Resolving in Media Player Plugin for the media type resolving rule.

The following sample code is an example of using the wmv media object in HTML.

```
// Example of media object in HTML
<object type="video/x-ms-wmv"
   data="http://192.168.1.50/example.wmv"
   width="1280"
   height="720"
   VideoMaxWidth="1280"
   VideoMaxHeight="720"
   id="media">
</object>
```

Following section describes the properties of Media Object.

- Properties
Properties
Properties of Media Object are as follows:

mode3D

Description
The NetCast Platform supports 3D formats such as 2D-to-3D, Side-by-Side, Top-and-Bottom, and Checker Board. It provides mode3D write-only property for developers to set a specific 3D format in 3D mode. The NetCast Platform will automatically display 3D video with defined 3D format if this property is set properly. The default value of this property is off and NetCast Platform displays original 2D video if this property is not defined. Note that NetCast 3.0 supports 3D rendering for any size of video screen. However, NetCast 2.0 supports it only for full screen mode. You can check if the TV supports 3D by using support3D in Device API.

<table>
<thead>
<tr>
<th>Variable</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>off</td>
<td>Original 2D Format</td>
</tr>
<tr>
<td>from_2d_to_3d</td>
<td>2D-to-3D Conversion Format</td>
</tr>
<tr>
<td>side_by_side</td>
<td>Side-by-Side Frame Compatible 3D (Left / Right)</td>
</tr>
<tr>
<td>side_by_side_rl</td>
<td>Side-by-Side Frame Compatible 3D (Right / Left)</td>
</tr>
<tr>
<td>top_bottom</td>
<td>Top-and-Bottom (or Over-Under) Frame Compatible 3D</td>
</tr>
<tr>
<td>checker_bd</td>
<td>Checker board Frame Compatible 3D only available for HD format (1080p @ 30Hz)</td>
</tr>
</tbody>
</table>

Example
<object type="video/x-ms-wmv"
    data="http://192.168.1.50/3dexample.wmv"
    mode3D="side_by_side"
    width="1280" height="720"
    id="media">
</object>

preBufferingTime

Description
Since NetCast 2.0, the preBufferingTime property is supported. Developers can adjust buffering time through preBufferingTime before playback. The unit of this property is an integer number of seconds. In the following example, developer is requesting that the media content starts playing after performing 5 seconds buffering time through calling "preBufferingTime = 5".

Example
<object type="video/x-ms-wmv"
    data="http://192.168.1.50/example.wmv"
    preBufferingTime = 5
    width="1280"
    height="720"
    id="media">
</object>

oneshot_url
Description
Developers may want to use a one shot URL to help prevent unwarranted content downloading. A “one shot URL” means a URL which is not available again after having been accessed once. The NetCast Platform supports a property, oneshot_url, for developers to play content linked to the one shot URL.

Note
To avoid multiple accesses to one shot URL, the NetCast Platform does not perform content type checking by reading the head of content file. Therefore, developers have to describe the exact MIME type. For example, “application/x-netcast-av” should be avoided because it does not describe the exact MIME type. ASX file should be avoided for one shot URL.

Example
<object type="video/x-ms-wmv"
   data="http://192.168.1.50/example.wmv"
   oneshot_url=true
   width="1280"
   height="720"
   id="media">
</object>

subtitleOn / subtitle

Description
The NetCast Platform supports subtitle decoding. Since NetCast 2.0, SAMI (Synchronized Accessible Media Interchange), CineCanvas and Timed Text subtitle formats are supported. The NetCast Platform supports the subtitle and subtitleOn properties.

The subtitle can be applied only when a full size video is being played.

Note
The NetCast Platform does not support multiplexing multiple subtitle tracks in one file.

Example
<object type="video/x-ms-wmv"
   data="http://192.168.1.50/example.wmv"
   subtitleOn=true
   subtitle="http://192.168.1.50/example.smi"
   width="1280"
   height="720"
   id="media">
</object>

drm_type

Description
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. The NetCast Platform supports a property, drm_type, for developers to set the DRM type. The default value of this property is “wm-drm” and NetCast Platform will use the WM-DRM solution if this property is not defined.

<table>
<thead>
<tr>
<th>Value</th>
<th>Meaning</th>
</tr>
</thead>
</table>

[Table] Available values for “drm_type” property
<table>
<thead>
<tr>
<th>Value</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>wm-drm</td>
<td>WM-DRM 10 PD or PlayReady (default value)</td>
</tr>
<tr>
<td>widevine</td>
<td>Widevine DRM and its adaptive/live streaming</td>
</tr>
<tr>
<td>verimatrix</td>
<td>Verimatri DRM</td>
</tr>
</tbody>
</table>

**Example**

```html
<object type="video/x-ms-wmv"
    data="http://192.168.1.50/example.wmv"
    drm_type="widevine"
    width="1280"
    height="720"
    id="media">
</object>
```

**playCount**

**Description**

To set play count for media, use "playCount" property. If you set the value to "0", it means the media will be played in an infinite loop. If you set the value to a number, 1 or over, it means the media will be played "n" times.

**Example**

```html
<object type="video/x-ms-wmv"
    data="http://192.168.1.50/example.wmv"
    playCount="2"
    width="1280"
    height="720"
    id="media">
</object>
```

**Audio Object**

As opposed to a video object, an audio object cannot have width and height. However, the NetCast Platform supports only one media player plugin and media object for both video and audio objects, so developers must specify the width and height for audio object as 0. This kind of example can be used for "radio like" services.

```html
// Example of audio object in HTML
<object type="audio/x-ms-wma"
    data="http://192.168.1.50/example.wma"
    width="0"
    height="0"
    id="media">
</object>
```

**Media Type Resolving in Media Player Plugin**

The following figure describes the decision tree for resolving media types on the NetCast Platform media player plugin. Refer to Annex A Complete List of Supported MIME Types in Developing > Developing Web App > App Development Guide section in this Library.
Methods/Properties/Events

Methods, properties, and events are listed in the following sections.

- Methods
- Properties
- Events
Methods

Methods of Media Player Plugin and API are as follows:

In this version of specification, the NetCast Platform does not support mute/unmute APIs. Users can mute and unmute audio using the mute function in the TV native system.

play

Description
Developers can play media at normal speed using media.play(speed) API.

Developers can also implement the trick mode play using media.play(speed) API only in mms streaming.

Note
In mms streaming, the speed parameter is transparently transmitted to the server without any conversion. Therefore, developers are responsible for matching the speed parameter between the JavaScript application and the server.

Syntax

```javascript
media.play(speed);
```

Parameters

<table>
<thead>
<tr>
<th>speed</th>
<th>[in] The range of allowed values of speed is from -30.0 to 30.0.</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Normal play speed. Default value is 1</td>
</tr>
<tr>
<td>0</td>
<td>Pause</td>
</tr>
</tbody>
</table>

Return Value
None

Example

```javascript
// Example of 'play'
var media = document.getElementById("media");
media.play(1);

// Example of 'trick mode play'
var media = document.getElementById("media");
media.play(-1.0);

// Example of 'pause'
var media = document.getElementById("media");
media.play(0);
```

See Also
isScannable
speed

Supported SDK / Emulator Version

<table>
<thead>
<tr>
<th>SDK Version</th>
<th>Emulator Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.5 or higher</td>
<td>LG Smart TV Emulator 2011 : SDK 1.5 or higher</td>
</tr>
<tr>
<td></td>
<td>LG Smart TV Emulator 2012 : SDK 2.0 or higher</td>
</tr>
<tr>
<td></td>
<td>- play(1), play(0) is supported</td>
</tr>
<tr>
<td></td>
<td>- play(speed) is not supported</td>
</tr>
<tr>
<td></td>
<td>LG Smart TV Emulator 2013 : SDK 3.0 or higher</td>
</tr>
</tbody>
</table>
### stop

**Description**
Developers can stop media using `media.stop()` API.

**Syntax**
```
media.stop();
```

**Parameters**
None

**Return Value**
None

**Example**
```
var media = document.getElementById("media");
media.stop();
```

<table>
<thead>
<tr>
<th>Supported SDK / Emulator Version</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>SDK Version</strong></td>
</tr>
<tr>
<td>1.5 or higher</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
</tbody>
</table>

### next

**Description**
The NetCast Platform supports playlist (ASX), and developers can play the next media by calling `media.next()` API.

Refer to section ‘userAgent String’ in Developing > Developing Web App > App Development Guide section in this Library.

**Syntax**
```
media.next();
```

**Parameters**
None

**Return Value**
None

**Example**
```
var media = document.getElementById("media");
media.next();
```

<table>
<thead>
<tr>
<th>Supported SDK / Emulator Version</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>SDK Version</strong></td>
</tr>
<tr>
<td>1.5 or higher</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
</tbody>
</table>

### previous
Description
The NetCast Platform supports playlist (ASX), and developers can play previous media by calling `media.previous()` API.

Refer to section 'userAgent String' in Developing > Developing Web App > App Development Guide section in this Library.

Syntax
```
media.previous();
```

Parameters
None

Return Value
None

Example
```
var media = document.getElementById("media");
media.previous();
```

Supported SDK / Emulator Version
<table>
<thead>
<tr>
<th>SDK Version</th>
<th>Emulator Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.5 or higher</td>
<td>LG Smart TV Emulator 2011 : SDK 1.5 or higher</td>
</tr>
<tr>
<td></td>
<td>LG Smart TV Emulator 2012 : SDK 2.0 or higher</td>
</tr>
<tr>
<td></td>
<td>LG Smart TV Emulator 2013 : SDK 3.0 or higher</td>
</tr>
</tbody>
</table>

seek

Description
This seek API will be supported for HTTP streaming only if the server supports the HTTP range header and MMS. Developers can set the time position of playback using `media.seek(position)` API, and the position value has millisecond precision.

Syntax
```
media.seek(position);
```

Parameters
```
position [in] Position value. It must have millisecond precision.
```

Return Value
None

Example
```
var media = document.getElementById("media");
media.seek(30000);
```

Supported SDK / Emulator Version
<table>
<thead>
<tr>
<th>SDK Version</th>
<th>Emulator Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.5 or higher</td>
<td>LG Smart TV Emulator 2011 : SDK 1.5 or higher</td>
</tr>
<tr>
<td></td>
<td>LG Smart TV Emulator 2012 : SDK 2.0 or higher</td>
</tr>
<tr>
<td></td>
<td>LG Smart TV Emulator 2013 : SDK 3.0 or higher</td>
</tr>
</tbody>
</table>

mediaPlayInfo
Description
This API gets media playback related information.

Syntax
```
media.mediaPlayInfo();
```

Parameters
None

Return Value
This API returns an object which contains values for duration, current position, remaining amount of buffer, instant bitrate and target bitrate. Property names and meanings are listed in the following table. The duration variable has same meaning and value with the property named as playTime. The currentPosition has same meaning and value as the property named as playPosition.

<table>
<thead>
<tr>
<th>Variable</th>
<th>Meaning</th>
<th>Unit</th>
</tr>
</thead>
<tbody>
<tr>
<td>duration</td>
<td>Duration of media file (same as playTime property).</td>
<td>millisecond</td>
</tr>
<tr>
<td>currentPosition</td>
<td>Current play position (same as playPosition property).</td>
<td>millisecond</td>
</tr>
<tr>
<td>buffRemain</td>
<td>Remaining amount of buffer (if the buffer reaches the end of stream then the value will be -1)</td>
<td>millisecond</td>
</tr>
<tr>
<td>bitrateInstant</td>
<td>Instant stream input bitrate</td>
<td>bit per second</td>
</tr>
<tr>
<td>bitrateTarget</td>
<td>Target bitrate for stream playback</td>
<td>bit per second</td>
</tr>
</tbody>
</table>

Developers can use this information for drawing a progress bar and its associated buffering status. Moreover, developers can use bitrateInstant and bitrateTarget for drawing a streaming speed level meter. The implementation of a streaming speed level meter is strongly required by the NetCast Platform specification, in order that the user can better understand the state of his network connection.

The following figure graphically illustrates the buffer related variables.

[Figure] Graphical explanation of buffer related variables

The following figure is a reference graphical implementation of the streaming speed level meter.

[Figure] Graphical implementation example of streaming speed level meter

Example
```
// Example of 'get playback time position'
```
```javascript
var playInfo = document.media.mediaPlayInfo();
duration = playInfo.duration;
currentPosition = playInfo.currentPosition;
bufBegin = playInfo.bufBegin;
bufEnd = playInfo.bufEnd;
bufRemain = playInfo.bufRemain;
bitrateInstant = playInfo.bitrateInstant;
bitrateTarget = playInfo.bitrateTarget;
```

**See Also**

playTime
playPosition

**Supported SDK / Emulator Version**

<table>
<thead>
<tr>
<th>SDK Version</th>
<th>Emulator Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.5 or higher</td>
<td>LG Smart TV Emulator 2011 : SDK 1.5 or higher</td>
</tr>
<tr>
<td></td>
<td>LG Smart TV Emulator 2012 : Partially Supported</td>
</tr>
<tr>
<td></td>
<td>LG Smart TV Emulator 2013 : SDK 3.0 or higher</td>
</tr>
</tbody>
</table>

### setWidevineXXX

**Description**

Since NetCast 2.0, setWidevineXXX APIs for setting Credential Information are provided as shown in the following table. All these values are set at Runtime. Developers must send the credential information through these APIs. If the value is not set through the API, the default value will be empty (null) or zero (0). All setWidevineXXX API arguments are string types.

Developers should call these APIs before playing media content (i.e. before the play() API is called).

**Syntax**

```javascript
media.setWidevineDrmURL(DrmServerURL);
media.setWidevineDeviceID(DeviceID);
media.setWidevineStreamID(StreamID);
media.setWidevineClientIP(ClientIP);
media.setWidevineUserData(UserData);
media.setWidevineDrmAckURL(DrmAckServerURL);
media.setWidevineHeartbeatURL(HeartbeatURL);
media.setWidevineHeartbeatPeriod(HeartbeatPeriod);
media.setWidevineDeviceType(DeviceType);
```

**Parameters**

Credential Information will be set by shown API methods. This table describes each API method’s parameter.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Type</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>DrmServerURL</td>
<td>String</td>
<td>URL for Widevine DRM key server</td>
</tr>
<tr>
<td>DeviceID</td>
<td>String</td>
<td>Unique player device ID</td>
</tr>
<tr>
<td>StreamID</td>
<td>String</td>
<td>Unique stream ID</td>
</tr>
<tr>
<td>ClientIP</td>
<td>String</td>
<td>IP address of client</td>
</tr>
<tr>
<td>UserData</td>
<td>String</td>
<td>Additional optional user data</td>
</tr>
<tr>
<td>DrmAckServerURL</td>
<td>String</td>
<td>URL for server that receives entitlement confirmations</td>
</tr>
<tr>
<td>HeartbeatURL</td>
<td>String</td>
<td>URL to receive client heartbeats</td>
</tr>
<tr>
<td>Parameter</td>
<td>Type</td>
<td>Meaning</td>
</tr>
<tr>
<td>------------------</td>
<td>---------</td>
<td>----------------------------------------------------------------</td>
</tr>
<tr>
<td>HeartbeatPeriod</td>
<td>String</td>
<td>Duration between consecutive heartbeats in seconds</td>
</tr>
<tr>
<td>DeviceType</td>
<td>String</td>
<td>Device type (default value: 0)</td>
</tr>
</tbody>
</table>

**Note**

Set Credential Information which is given by Content Provider or Security Service. Some of Credential Information are required and the others are optional. It depends on Content Provider or Security Service.

**Return Value**

None

**Example**

```javascript
// Example of 'set Widevine Credential information'
var media = document.getElementById("media");

media.setWidevineStreamID('123');
media.setWidevineDrmURL('https://drmser.cgi');
media.setWidevineDeviceID('abcd');
media.setWidevineUserData('239084');
media.setWidevineDrmAckURL('');
media.setWidevineHeartbeatURL('');
media.setWidevineHeartbeatPeriod('');
media.setWidevineDeviceType('TV');

media.play(1);
```

**Supported SDK / Emulator Version**

<table>
<thead>
<tr>
<th>SDK Version</th>
<th>Emulator Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.5 or higher</td>
<td>LG Smart TV Emulator 2011: Not Supported</td>
</tr>
<tr>
<td></td>
<td>LG Smart TV Emulator 2012: Not Supported</td>
</tr>
<tr>
<td></td>
<td>LG Smart TV Emulator 2013: Not Supported</td>
</tr>
</tbody>
</table>
Properties

Properties of Media Player Plugin and API are as follows:

version

Description
The NetCast Platform provides a version read-only property in the Media Player plugin object. Developers can get version of Media Player Plugin using this property. It will return the Media Player Plugin version information as a string type.

Syntax
media.version;

Example
var media = document.getElementById("media");
mediaVersion = media.version;

Supported SDK / Emulator Version

<table>
<thead>
<tr>
<th>SDK Version</th>
<th>Emulator Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.5 or higher</td>
<td>LG Smart TV Emulator 2011 : SDK 1.5 or higher</td>
</tr>
<tr>
<td></td>
<td>LG Smart TV Emulator 2012 : SDK 2.0 or higher</td>
</tr>
<tr>
<td></td>
<td>LG Smart TV Emulator 2013 : SDK 3.0 or higher</td>
</tr>
</tbody>
</table>

type

Description
The NetCast Platform provides a type read-write property in the Media Player plugin object. Developers can get media type (MIME type) using this property. It will return the MIME type information as a string type.

Syntax
media.type;

Example
var media = document.getElementById("media");
mimeType = media.type;

Supported SDK / Emulator Version

<table>
<thead>
<tr>
<th>SDK Version</th>
<th>Emulator Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.5 or higher</td>
<td>LG Smart TV Emulator 2011 : SDK 1.5 or higher</td>
</tr>
<tr>
<td></td>
<td>LG Smart TV Emulator 2012 : SDK 2.0 or higher</td>
</tr>
<tr>
<td></td>
<td>LG Smart TV Emulator 2013 : SDK 3.0 or higher</td>
</tr>
</tbody>
</table>

data

Description
The NetCast Platform provides a data read-write property in the Media Player plugin object. Developers can get media URL information using this property. It will return the media URL information as a string type.

Syntax
media.data;

Example
var media = document.getElementById("media");
Supported SDK / Emulator Version

<table>
<thead>
<tr>
<th>SDK Version</th>
<th>Emulator Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.5 or higher</td>
<td>LG Smart TV Emulator 2011 : SDK 1.5 or higher</td>
</tr>
<tr>
<td></td>
<td>LG Smart TV Emulator 2012 : SDK 2.0 or higher</td>
</tr>
<tr>
<td></td>
<td>LG Smart TV Emulator 2013 : SDK 3.0 or higher</td>
</tr>
</tbody>
</table>

**width / height**

**Description**
The NetCast Platform provides ‘width’ and ‘height’ read-write properties in the Media Player plugin object. It will return the width and height information of the media object as string types.

Developers can set the size of video using ‘media.width = value; media.height = value;’ API.

Developers can also make a full-screen video by specifying the media object size.

**Syntax**

```
media.width;
media.height;
```

**Example**

```javascript
// Example of 'get media size information'
var media = document.getElementById("media");
width = media.width;
height = media.height;

// Example of 'set size'
var media = document.getElementById("media");
media.width = 1280;
media.height = 720;

// Example of 'setting video full screen'
<body style='margin:0'>
<script language="javascript">
  ...
  var media = document.getElementById("media");
  media.width = 1280;
  media.height = 720;
  ...
</script>
  ...
</body>
```

Supported SDK / Emulator Version

<table>
<thead>
<tr>
<th>SDK Version</th>
<th>Emulator Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.5 or higher</td>
<td>LG Smart TV Emulator 2011 : SDK 1.5 or higher</td>
</tr>
<tr>
<td></td>
<td>LG Smart TV Emulator 2012 : SDK 2.0 or higher</td>
</tr>
<tr>
<td></td>
<td>LG Smart TV Emulator 2013 : SDK 3.0 or higher</td>
</tr>
</tbody>
</table>

**playTime**

**Description**
The NetCast Platform provides playtime read-only property in the Media Player plugin object. Developers can get play time using this property. It will return the duration of the currently playing media item as a string type in milliseconds.

**Syntax**

```
media.playTime;
```

**Example**

```
var media = document.getElementById("media");
playTime = media.playTime;
```

**Supported SDK / Emulator Version**

<table>
<thead>
<tr>
<th>SDK Version</th>
<th>Emulator Version</th>
</tr>
</thead>
</table>
| 1.5 or higher | LG Smart TV Emulator 2011 : SDK 1.5 or higher  
|             | LG Smart TV Emulator 2012 : SDK 2.0 or higher  
|             | LG Smart TV Emulator 2013 : SDK 3.0 or higher  |

**playPosition**

**Description**

The NetCast Platform provides playPosition read-only property in the Media Player plugin object. Developers can get play position using this property. It will return the play position of the currently playing media item as a string type in milliseconds.

**Syntax**

```
media.playPosition;
```

**Example**

```
var media = document.getElementById("media");
playPosition = media.playPosition;
```

**Supported SDK / Emulator Version**

<table>
<thead>
<tr>
<th>SDK Version</th>
<th>Emulator Version</th>
</tr>
</thead>
</table>
| 1.5 or higher | LG Smart TV Emulator 2011 : SDK 1.5 or higher  
|             | LG Smart TV Emulator 2012 : SDK 2.0 or higher  
|             | LG Smart TV Emulator 2013 : SDK 3.0 or higher  |

**playState**

**Description**

The NetCast Platform provides playState read-only property in the Media Player plugin object. Developers can get play state using this property. It will return the play state of the currently playing media item as an enumerated number. See the following table for the mapping rule between the resolutions and enumerated return values.

[Table] The enumerated value of play state

<table>
<thead>
<tr>
<th>Play State</th>
<th>Enumerated Return Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Stopped</td>
</tr>
<tr>
<td>1</td>
<td>Playing</td>
</tr>
<tr>
<td>2</td>
<td>Paused</td>
</tr>
<tr>
<td>3</td>
<td>Connecting</td>
</tr>
</tbody>
</table>
Play State | Enumerated Return Value
---|---
4 | Buffering
5 | Finished
6 | Error

**Syntax**

```javascript
media.playState;
```

**Example**

```javascript
var media = document.getElementById("media");
playState = media.playState;
```

**Supported SDK / Emulator Version**

<table>
<thead>
<tr>
<th>SDK Version</th>
<th>Emulator Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.5 or higher</td>
<td>LG Smart TV Emulator 2011 : SDK 1.5 or higher</td>
</tr>
<tr>
<td></td>
<td>LG Smart TV Emulator 2012 : SDK 2.0 or higher</td>
</tr>
<tr>
<td></td>
<td>LG Smart TV Emulator 2013 : SDK 3.0 or higher</td>
</tr>
</tbody>
</table>

**error**

**Description**
The NetCast Platform provides error read-only property in the Media Player plugin object. Developers can get error code using this property. Developers can get the error code using the API if the Media Player plugin meets an error while the current media file is playing.

<table>
<thead>
<tr>
<th>Error code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>A/V format not supported</td>
</tr>
<tr>
<td>1</td>
<td>Cannot connect to server or connection lost</td>
</tr>
<tr>
<td>2</td>
<td>Unidentified error</td>
</tr>
<tr>
<td>1000</td>
<td>File is not found</td>
</tr>
<tr>
<td>1001</td>
<td>Invalid protocol</td>
</tr>
<tr>
<td>1002</td>
<td>DRM failure</td>
</tr>
<tr>
<td>1003</td>
<td>Play list is empty</td>
</tr>
<tr>
<td>1004</td>
<td>Unrecognized play list</td>
</tr>
<tr>
<td>1005</td>
<td>Invalid ASX format</td>
</tr>
<tr>
<td>1006</td>
<td>Error in downloading play list</td>
</tr>
<tr>
<td>1007</td>
<td>Out of memory</td>
</tr>
<tr>
<td>1008</td>
<td>Invalid URL list format</td>
</tr>
<tr>
<td>1009</td>
<td>Not playable in play list</td>
</tr>
<tr>
<td>1100</td>
<td>Unidentified WM-DRM error</td>
</tr>
<tr>
<td>1101</td>
<td>Incorrect license in local license store</td>
</tr>
</tbody>
</table>
### Error code

<table>
<thead>
<tr>
<th>Error code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1102</td>
<td>Fail in receiving correct license from server</td>
</tr>
<tr>
<td>1103</td>
<td>Stored license is expired</td>
</tr>
</tbody>
</table>

#### Syntax

```javascript
media.error;
```

#### Example

```javascript
var media = document.getElementById('media');
errorCode = media.error;
```

**See Also**

[onError](#)

### Supported SDK / Emulator Version

<table>
<thead>
<tr>
<th>SDK Version</th>
<th>Emulator Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.5 or higher</td>
<td>LG Smart TV Emulator 2011 : SDK 1.5 or higher</td>
</tr>
<tr>
<td></td>
<td>LG Smart TV Emulator 2012 : SDK 2.0 or higher</td>
</tr>
<tr>
<td></td>
<td>LG Smart TV Emulator 2013 : SDK 3.0 or higher</td>
</tr>
</tbody>
</table>

#### autoStart

**Description**

The NetCast Platform provides `autoStart` read-write property in the Media Player plugin object. Developers can get and set `autoStart` property. Developers should set the value to `false` if the media file playout is not to be started automatically.

#### Syntax

```javascript
media.autoStart;
```

#### Example

```javascript
var media = document.getElementById('media');
media.autoStart = true; // write
autoStart = media.autoStart; // read
```

// Example of 'set autoStart of media'
```html
<object type="application/x-netcast-av"
    data="http://192.168.1.50/example.wmv"
    width="1280"
    height="720"
    autoStart = "true"
    id="media">
</object> //write
```

### Supported SDK / Emulator Version

<table>
<thead>
<tr>
<th>SDK Version</th>
<th>Emulator Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.5 or higher</td>
<td>LG Smart TV Emulator 2011 : SDK 1.5 or higher</td>
</tr>
<tr>
<td></td>
<td>LG Smart TV Emulator 2012 : SDK 2.0 or higher</td>
</tr>
<tr>
<td></td>
<td>LG Smart TV Emulator 2013 : SDK 3.0 or higher</td>
</tr>
</tbody>
</table>

#### isScannable

**Description**
The NetCast Platform provides isScannable read-only property in the Media Player plugin object. Developers can get ‘isScannable’ property. If the value of this property is true, the current media can be scanned (fastforward or rewind). If a media file has not been opened, the value of this property will be false. The media can be scanned only if media is indexed and delivered via the MMS protocol.

Syntax
media.isScannable;

Example
function processPlayChangeFunction() {
    var media = document.getElementById("media");
    if (media.playState == 1) {
        isScanable = media.isScannable;
        ...
    }
}

function init() {
    var media = document.getElementById("media");
    media.onPlayStateChange = processPlayStateChangeFunction();
    ...
}

See Also
play (Trick mode play)

Supported SDK / Emulator Version

<table>
<thead>
<tr>
<th>SDK Version</th>
<th>Emulator Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.5 or higher</td>
<td>LG Smart TV Emulator 2011: Not Supported</td>
</tr>
<tr>
<td></td>
<td>LG Smart TV Emulator 2012: SDK 2.4 or higher</td>
</tr>
<tr>
<td></td>
<td>LG Smart TV Emulator 2013: SDK 3.0 or higher</td>
</tr>
</tbody>
</table>

Note
The ‘isScannable’ property can be used after the media is played.

speed

Description
The NetCast Platform provides ‘speed’ read-only property in the Media Player plugin object. Developers can get ‘speed’ property. The value of this property is the relative playback speed of the media file currently being played. Its value is 0 if the playState is not 1 (playing). The speed can differ from 1 only if the media is indexed and delivered via the MMS protocol.

Syntax
media.speed;

Example
var media = document.getElementById("media");
speed = media.speed;

See Also
isScannable
playState

Supported SDK / Emulator Version
bufferingProgress

Description
The NetCast Platform provides 'bufferingProgress' read-only property in the Media Player plugin object. Developers can get 'bufferingProgress' property. The value of this property is the percentage of buffering complete. Each time playback stops and restarts, this property may decrease or increase. It does not vary if playback is paused. This property returns a valid value only after a media file has been opened and decoding starts.

Note
The value of this property is not very accurate. It is just informative. Therefore, application authors should not use the value of this property for logical decisions in an application. For example, an application should not enable and disable a loading message using the value of this property. In this example, it would be better for application authors to use the onBuffering event instead of this property. See onBuffering.

Syntax
media.bufferingProgress;

Example
var media = document.getElementById("media");
bufferingProgress = media.bufferingProgress;

See Also
onBuffering
mediaPlayInfo

subtitleOn

Description
The NetCast Platform provides 'subtitleOn' read-write property in the Media Player plugin object. Developers can get and set 'subtitleOn' property. The value of this property is the status of subtitle decoder and is a Boolean type. LG Smart TV applications can turn the subtitle decoder on and off by setting this property with "true" or "false" values respectively.

The subtitle must be applied when a full size video is being played.

Syntax
media.subtitleOn;

Example
var media = document.getElementById("media");

//get subtitleOn property
subtitleOn = media.subtitleOn;
//set subtitleOn property
media.subtitleOn = newSubtitleOn;

**Supported SDK / Emulator Version**

<table>
<thead>
<tr>
<th>SDK Version</th>
<th>Emulator Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.5 or higher</td>
<td>LG Smart TV Emulator 2011 : Not Supported</td>
</tr>
<tr>
<td></td>
<td>LG Smart TV Emulator 2012 : Not Supported</td>
</tr>
<tr>
<td></td>
<td>LG Smart TV Emulator 2013 : Not Supported</td>
</tr>
</tbody>
</table>

**subtitle**

**Description**
The NetCast Platform provides ‘subtitle’ read-write property in the Media Player plugin object. Developers can get and set ‘subtitle’ property. The value of this property is the URL of the subtitle file. The media player retrieves the subtitle file before decoding the media file. LG Smart TV applications set this property every time a new subtitle file is required.

The subtitle must be applied when a full size video is being played.

**Note**

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

**Syntax**

media.subtitle;

**Example**

```javascript
var media = document.getElementById("media");

//get subtitle property
subtitleURL = media.subtitle;

//set subtitle property
media.subtitle = newSubtitleURL;
```

**Supported SDK / Emulator Version**

<table>
<thead>
<tr>
<th>SDK Version</th>
<th>Emulator Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.5 or higher</td>
<td>LG Smart TV Emulator 2011 : Not Supported</td>
</tr>
<tr>
<td></td>
<td>LG Smart TV Emulator 2012 : Not Supported</td>
</tr>
<tr>
<td></td>
<td>LG Smart TV Emulator 2013 : Not Supported</td>
</tr>
</tbody>
</table>

**mode3D**

**Description**
The NetCast Platform provides ‘mode3D’ read-only property in the Device Information / Media Player plugin object. Developers can get a 3D mode. It will return the current 3D format for 3D mode. Developers should get the value when TVs are presently in 3D mode. See the following table for available values about 3D format.

[Table] Available values for "mode3D" property

<table>
<thead>
<tr>
<th>Variable</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>off</td>
<td>Original 2D Format</td>
</tr>
<tr>
<td>from_2d_to_3d</td>
<td>2D-to-3D Conversion Format</td>
</tr>
</tbody>
</table>
### Variable

<table>
<thead>
<tr>
<th>Variable</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>side_by_side</td>
<td>Side-by-Side Frame Compatible 3D (Left / Right)</td>
</tr>
<tr>
<td>side_by_side_rl</td>
<td>Side-by-Side Frame Compatible 3D (Right / Left)</td>
</tr>
<tr>
<td>top_bottom</td>
<td>Top-and-Bottom (or Over-Under) Frame Compatible 3D</td>
</tr>
<tr>
<td>checker_bd</td>
<td>Checker board Frame Compatible 3D only available for HD format (1080p @ 30Hz)</td>
</tr>
</tbody>
</table>

### Syntax

```javascript
media.mode3D;
```

### Example

```javascript
var media = document.getElementById("media");
triMode = media.mode3D;
```

### Supported SDK / Emulator Version

<table>
<thead>
<tr>
<th>SDK Version</th>
<th>Emulator Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.5 or higher</td>
<td>LG Smart TV Emulator 2011 : Not Supported</td>
</tr>
<tr>
<td></td>
<td>LG Smart TV Emulator 2012 : Not Supported</td>
</tr>
<tr>
<td></td>
<td>LG Smart TV Emulator 2013 : Not Supported</td>
</tr>
</tbody>
</table>

### audioLanguage

#### Description
The NetCast Platform provides ‘audioLanguage’ write-only property that is used for selecting audio language when media content has the multi-audio language.

#### Note
The value of this property shall be exactly matched with the language code of audio track in media container. Matching of the two language codes is developer’s own responsibility. NetCast Platform does only search the matched audio track in media container.

#### Note
When video is paused, this API does not work normally.

#### Note
After the audio is changed, audio mute may occur for a few seconds due to the synchronization of video and audio.

#### Syntax

```javascript
media.audioLanguage;
```

#### Example

```html
// Example of ‘set audio language information of media’
<object type="video/x-ms-wmv"
    data="http://192.168.1.50/example.wmv"
    width="1280"
    height="720"
    audioLanguage="en"
    id="media">
</object>
```
// Example of 'set audio language information of media'
<script>
    var media = document.getElementById("media");

    // set audioLanguage property
    media.audioLanguage = newAudioLanguage
</script>

### Supported SDK / Emulator Version

<table>
<thead>
<tr>
<th>SDK Version</th>
<th>Emulator Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.5 or higher</td>
<td>LG Smart TV Emulator 2011: Not Supported</td>
</tr>
<tr>
<td></td>
<td>LG Smart TV Emulator 2012: Not Supported</td>
</tr>
<tr>
<td></td>
<td>LG Smart TV Emulator 2013: Not Supported</td>
</tr>
</tbody>
</table>
Events

Events of Media Player Plugin and API are as follows:

onPlayStateChange

Description
The NetCast Platform provides an onPlayStateChange event in the Media Player plugin object. Developers can receive play state change event. Developer can receive a play state change event when the play state of currently playing media item is changed.

To refer to the values of the playState property, see playState.

Syntax
```
media.onPlayStateChange = processPlayStateChangeFunction;
```

Example
```
<script language='javascript'>
function processPlayStateChangeFunction() {
    . . .
    // read and process playState property
    . . .
}
</script>

<object type="video/x-ms-wmv"
    data="http://192.168.1.50/example.wmv"
    width="1280"
    height="720"
    id="media">
</object>

<script>
    var media = document.getElementById('media');
    media.onPlayStateChange = processPlayStateChangeFunction;
</script>
```

See Also
playState

Supported SDK / Emulator Version

<table>
<thead>
<tr>
<th>SDK Version</th>
<th>Emulator Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.5 or higher</td>
<td>LG Smart TV Emulator 2011 : SDK 1.5 or higher</td>
</tr>
<tr>
<td></td>
<td>LG Smart TV Emulator 2012 : SDK 2.0 or higher</td>
</tr>
<tr>
<td></td>
<td>LG Smart TV Emulator 2013 : SDK 3.0 or higher</td>
</tr>
</tbody>
</table>

onBuffering

Description
The NetCast Platform provides an onBuffering event in the Media Player plugin object. Developers can receive buffering event. Developers can receive a buffering event when the media player begins and ends buffering. A Boolean type parameter specifies whether data buffering has started or finished. A value of true indicates that the data buffering has started. Buffering also occurs whenever playback stops and then restarts (either from calls to play() and stop() methods or when network congestion occurs during playing streamed media.

Syntax
media.onBuffering = processBufferingFunction;

Example

```javascript
function processBufferingFunction(isStarted)
{
    // process buffering
    ... 
}
</script>

<object type="video/x-ms-wmv"
    data="http://192.168.1.50/example.wmv"
    width="1280"
    height="720"
    id="media">
</object>

<script>
    var media = document.getElementById('media');
    media.onBuffering = processReadyStateChangeFunction;
</script>

See Also

bufferingProgress

Supported SDK / Emulator Version

<table>
<thead>
<tr>
<th>SDK Version</th>
<th>Emulator Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.5 or higher</td>
<td>LG Smart TV Emulator 2011 : SDK 1.5 or higher</td>
</tr>
<tr>
<td></td>
<td>LG Smart TV Emulator 2012 : SDK 2.0 or higher</td>
</tr>
<tr>
<td></td>
<td>LG Smart TV Emulator 2013 : SDK 3.0 or higher</td>
</tr>
</tbody>
</table>

onError

Description
The NetCast Platform provides an onError event in the Media Player plugin object. Developers can receive error event. Developers can receive an error event when the media player encounters an error while playing.

Syntax

```javascript
media.onError = processErrorFunction;
```

Example

```javascript
function processBufferingFunction(isStarted)
{
    // process buffering
    ...
}
</script>

<object type="video/x-ms-wmv"
    data="http://192.168.1.50/example.wmv"
    width="1280"
    height="720"
    id="media">
</object>
onDRMRightsError

Description
The script function that is called when a DRM licensing error occurs during playback, recording, or timeshifting of
DRM-protected AV content inside the embedded object.

Syntax
media.onDRMRightsError = HandleOnDRMRightsError;

Parameters
None

Return Value
errorState  [in] Error code detailing the type of error (0 : no license, 1 : invalid license)
contentID   [in] Unique ID of the content in the scope of DRM system that raises the error
DRMSystemID [in] For PlayReady, the value is “urn:dvb:casystemid:19219”.
rightsIssuerURL [in] Optional element indicating the value of the rightsIssuerURL that can be used to
non-silently obtain the rights for the content item currently being played for which this
DRM error is generated, in cases whereby the rightsIssuerURL is known. If different
URLs are retrieved from the stream and the metadata, then the conflict resolution is
implementation-dependent.

Example
// handle DRM errors
media.onDRMRightsError = function(errorState, contentID, DRMSystemID, rightsIssuerURL) {
  if (errorState == 0) {
    console.log('no license');
  }
  else if (errorState == 1) {
    log('invalid license');
  }
  log("HandleOnDRMRightsError errorState:" + errorState + "
    contentID:" + contentID + " DRMSystemID:" + DRMSystemID + "
    rightsIssuerURL:" + rightsIssuerURL);
}
Device Info Plugin and API

The NetCast Platform supports device information plugin. The following sub-sections describe the API of the device information plugin. See also section ‘userAgent String’ in Developing > Developing Web App > App Development Guide section in this Library for supporting feature information.

- Device Object
- Methods/Properties

Device Object

An example MIME type of the device information object is application/x-netcast-info for the NetCast Platform device information plugin. The device information object supports a single property called ‘id’, ‘width’, and ‘height’. Note that ‘width’ and ‘height’ must be always set to 0.

The NetCast Platform supports only one instance of the device information object at the same time so developers must not use more than one device information object.

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

Methods/Properties

Methods and properties are listed in the following sections.

- Methods
- Properties
Methods

Methods of Device Info Plugin and API are as follows:

getLocalTime

Description
The ‘getLocalTime’ API is a read-only API that gets the time setting of TV.

Syntax
device.getLocalTime();

Parameters
None

Return Value
This API returns the object that includes local time. It has the following properties.

<table>
<thead>
<tr>
<th>Variable</th>
<th>Meaning</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>year</td>
<td>year</td>
<td>int</td>
</tr>
<tr>
<td>month</td>
<td>month (1~12)</td>
<td>int</td>
</tr>
<tr>
<td>date</td>
<td>date (1~31)</td>
<td>int</td>
</tr>
<tr>
<td>hour</td>
<td>hour (0~23)</td>
<td>int</td>
</tr>
<tr>
<td>minute</td>
<td>minute (0~59)</td>
<td>int</td>
</tr>
<tr>
<td>second</td>
<td>second (0~59)</td>
<td>int</td>
</tr>
</tbody>
</table>

* It returns the number from 1 to 12, which is different from ‘month’ of the ‘getSystemTime API’.

Example
var device = document.getElementById("device");
var sysTime = device.getLocalTime();
var hour = sysTime.hour;
var min = sysTime.minute;
var sec = sysTime.second;
...

Supported SDK / Emulator Version

<table>
<thead>
<tr>
<th>SDK Version</th>
<th>Emulator Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.2 or higher</td>
<td>LG Smart TV Emulator 2012 : SDK 2.2 or higher</td>
</tr>
</tbody>
</table>

getSystemTime

Description
The ‘getSystemTime’ API is a read-only API that gets time setting of Linux system. The time setting value can be configured by TV broadcasting or user.

Syntax
device.getSystemTime();

**Parameters**
None

**Return Value**
This API returns the object that includes system time. It has the following properties.

<table>
<thead>
<tr>
<th>Table</th>
<th>Variables of getSystemTime()'s return object</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Variable</strong></td>
<td><strong>Meaning</strong></td>
</tr>
<tr>
<td>year</td>
<td>year</td>
</tr>
<tr>
<td>month</td>
<td>month (0~11)</td>
</tr>
<tr>
<td>date</td>
<td>date (1~31)</td>
</tr>
<tr>
<td>day</td>
<td>day (0~6)</td>
</tr>
<tr>
<td>dayOfYear</td>
<td>ordinal date (1~365)</td>
</tr>
<tr>
<td>hour</td>
<td>hour (0~23)</td>
</tr>
<tr>
<td>minute</td>
<td>minute (0~59)</td>
</tr>
<tr>
<td>second</td>
<td>second (0~59)</td>
</tr>
<tr>
<td>isDST</td>
<td>Indicates if Daylight Saving Time (DST) is effective at current system date/time in the country set by user.</td>
</tr>
<tr>
<td>time</td>
<td>String consist of “Day MM DD hh:mm:ss YYYY”</td>
</tr>
</tbody>
</table>

**Example**

```javascript
var device = document.getElementById("device");
var sysTime = device.getSystemTime();
var hour = sysTime.hour;
var min = sysTime.minute;
var sec = sysTime.second;
...
```

**Supported SDK / Emulator Version**

<table>
<thead>
<tr>
<th>SDK Version</th>
<th>Emulator Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.2 or higher</td>
<td>LG Smart TV Emulator 2011: Not Supported</td>
</tr>
<tr>
<td></td>
<td>LG Smart TV Emulator 2012: SDK 2.2 or higher</td>
</tr>
<tr>
<td></td>
<td>LG Smart TV Emulator 2013: SDK 3.0 or higher</td>
</tr>
</tbody>
</table>

**pentouchInfo**

**Description**
The ‘pentouchInfo()’ API gets the information of touch pen (power, battery life) in PDP.

**Note**
This API works exactly only when used in models that PDP pentouch is available.

**Syntax**

```javascript
device.pentouchInfo(penNumber);
```
### Parameters

- **penNumber**
  - [in] Pen number of two touch pens (0, 1)

### Return Value

This API returns object that includes the information of touch pen. It has the following properties.

<table>
<thead>
<tr>
<th>Variable</th>
<th>Meaning</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>powerOn</td>
<td>Power on/off status</td>
<td>Boolean</td>
</tr>
<tr>
<td>battery</td>
<td>Battery life (0~100)</td>
<td>int</td>
</tr>
</tbody>
</table>

#### Example

```javascript
var device = document.getElementById("device");
var penInfo1 = device.pentouchInfo(0);
var isPowerOn1 = penInfo1.powerOn;
var battery1 = penInfo1.battery;
...
```

#### See Also

- [supportPentouch](#)

---

### Supported SDK / Emulator Version

<table>
<thead>
<tr>
<th>SDK Version</th>
<th>Emulator Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.2 or higher</td>
<td>LG Smart TV Emulator 2011 : Not Supported</td>
</tr>
<tr>
<td></td>
<td>LG Smart TV Emulator 2012 : Not Supported</td>
</tr>
<tr>
<td></td>
<td>LG Smart TV Emulator 2013 : Not Supported</td>
</tr>
</tbody>
</table>

---

### setDrmLicenseInfo

#### Description

The NetCast Platform provides a ‘setDrmLicenseInfo()’ API in the Device Info plugin object. Developers can set DRM license information. It will return ‘1’ if the call is processed without error or ‘0’ if there is an error in processing the call or license setup is not completed. Therefore, applications should wait until they receive a ‘1’ before proceeding to the next step.

#### Syntax

```javascript
device.setDrmLicenseInfo(licenseInfo);
```

#### Parameters

- **licenseInfo**
  - [in] License information

#### Return Value

Returns ‘1’ if the call is processed without error or ‘0’ if there is an error in processing the call or license setup is not completed.

#### Example

```javascript
var device = document.getElementById("device");
callSuccess = device.setDrmLicenseInfo("<LICENSERESPONSE><LICENSE version='2.0.0.0'>
```
setVMConfigData

Description
The NetCast Platform provides a 'setVMConfigData()' API in the Device Info plugin object. Developers can set Verimatrix DRM certification information. It will return ‘1’ if Verimatrix initialization is successful or ‘0’ if Verimatrix initialization fails. Therefore, applications should wait until they receive a ‘1’ before proceeding to the next step.

Note
This API method should be called before playing the media. When using this API, the drmType property of Media object type should be set to “vermatrix”.

Syntax
device.setVMConfigData(serviceType, companyName, drmServerUrl);

Parameters
- serviceType [in] Service type must be 1
- companyName [in] Company name
- drmServerUrl [in] DRM Server URL

Return Value
Returns ‘1’ if Verimatrix initialization is successful or ‘0’ if Verimatrix initialization fails.

Example
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!");

Properties
Properties of Device Info Plugin and API are as follows:

version

Description
The NetCast Platform provides a ‘version’ read-only property in the Device Info Plugin object. Developers can get version of Device Info Plugin. It will return the string type of Device Info Plugin version information as a string type.

Syntax
device.version;

Example
var device = document.getElementById("device");
deviceVersion = device.version;

Supported SDK / Emulator Version

<table>
<thead>
<tr>
<th>SDK Version</th>
<th>Emulator Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.5 or higher</td>
<td>LG Smart TV Emulator 2011 : SDK 1.5 or higher</td>
</tr>
<tr>
<td></td>
<td>LG Smart TV Emulator 2012 : SDK 2.0 or higher</td>
</tr>
<tr>
<td></td>
<td>LG Smart TV Emulator 2013 : SDK 3.0 or higher</td>
</tr>
</tbody>
</table>

manufacturer

Description
The NetCast Platform provides a ‘manufacturer’ read-only property in the Device Information plugin object. Developers can get manufacturer ID using this property. It will return “LGE” as a string.

Syntax
device.manufacturer;

Example
var device = document.getElementById("device");
manufacturerId = device.manufacturer;

Supported SDK / Emulator Version

<table>
<thead>
<tr>
<th>SDK Version</th>
<th>Emulator Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.5 or higher</td>
<td>LG Smart TV Emulator 2011 : SDK 1.5 or higher</td>
</tr>
<tr>
<td></td>
<td>LG Smart TV Emulator 2012 : SDK 2.0 or higher</td>
</tr>
<tr>
<td></td>
<td>LG Smart TV Emulator 2013 : SDK 3.0 or higher</td>
</tr>
</tbody>
</table>

modelName

Description
The NetCast Platform provides a ‘modelName’ read-only property in the Device Information plugin object. Developers can get model name using this property. It will return the model name as a string type.

Syntax
device.modelName;

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

For example, the model name is retrieved in the following format: “55LM6700-NC”.

### Supported SDK / Emulator Version

<table>
<thead>
<tr>
<th>SDK Version</th>
<th>Emulator Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.5 or higher</td>
<td>LG Smart TV Emulator 2011 : SDK 1.5 or higher</td>
</tr>
<tr>
<td></td>
<td>LG Smart TV Emulator 2012 : SDK 2.0 or higher</td>
</tr>
<tr>
<td></td>
<td>LG Smart TV Emulator 2013 : SDK 3.0 or higher</td>
</tr>
</tbody>
</table>

### serialNumber

**Description**
The NetCast Platform provides a ‘serialNumber’ read-only property in the Device Information plugin object. Developers can get serial number using this property. It will return the serial number of the product as a string type.

**Caution**
Do not use serialNumber as a unique identifier because the serial number may not be returned with unique value in some cases.

**Syntax**

device.serialNumber;

**Example**

```javascript
var device = document.getElementById("device");
serialNumber = device.serialNumber;
```

### Supported SDK / Emulator Version

<table>
<thead>
<tr>
<th>SDK Version</th>
<th>Emulator Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.5 or higher</td>
<td>LG Smart TV Emulator 2011 : SDK 1.5 or higher</td>
</tr>
<tr>
<td></td>
<td>LG Smart TV Emulator 2012 : SDK 2.0 or higher</td>
</tr>
<tr>
<td></td>
<td>LG Smart TV Emulator 2013 : SDK 3.0 or higher</td>
</tr>
</tbody>
</table>

### swVersion

**Description**
The NetCast Platform provides a ‘swVersion’ read-only property in the Device Information plugin object. Developers can get software version using this property. It will return the software version of the product as a string type.

**Syntax**

device.swVersion;

**Example**

```javascript
var device = document.getElementById("device");
swVersion = device.swVersion;
```

For example, the software version is retrieved in the following format: “03.21.20”.

### Supported SDK / Emulator Version

<table>
<thead>
<tr>
<th>SDK Version</th>
<th>Emulator Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.5 or higher</td>
<td>LG Smart TV Emulator 2011 : SDK 1.5 or higher</td>
</tr>
<tr>
<td></td>
<td>LG Smart TV Emulator 2012 : SDK 2.0 or higher</td>
</tr>
<tr>
<td></td>
<td>LG Smart TV Emulator 2013 : SDK 3.0 or higher</td>
</tr>
</tbody>
</table>
**hwVersion**

**Description**
The NetCast Platform provides a 'hwVersion' read-only property in the Device Information plugin object. Developers can get hardware version using this property. It will return the hardware version of the product as a string type.

**Syntax**
```javascript
device.hwVersion;
```

**Example**
```javascript
var device = document.getElementById("device");
hwVersion = device.hwVersion;
```

**Supported SDK / Emulator Version**

<table>
<thead>
<tr>
<th>SDK Version</th>
<th>Emulator Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.5 or higher</td>
<td>LG Smart TV Emulator 2011 : SDK 1.5 or higher</td>
</tr>
<tr>
<td></td>
<td>LG Smart TV Emulator 2012 : SDK 2.0 or higher</td>
</tr>
<tr>
<td></td>
<td>LG Smart TV Emulator 2013 : SDK 3.0 or higher</td>
</tr>
</tbody>
</table>

**SDKVersion**

**Description**
The NetCast Platform provides a 'SDKVersion' read-only property. Developers can get SDK version supported in the current TV. It returns the String type of value (00.00.00).

**Syntax**
```javascript
device.SDKVersion;
```

**Example**
```javascript
var device = document.getElementById("device");
SDKVersion = device.SDKVersion;
```

**Supported SDK / Emulator Version**

<table>
<thead>
<tr>
<th>SDK Version</th>
<th>Emulator Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.2 or higher</td>
<td>LG Smart TV Emulator 2011 : Not Supported</td>
</tr>
<tr>
<td></td>
<td>LG Smart TV Emulator 2012 : SDK 2.2 or higher</td>
</tr>
<tr>
<td></td>
<td>LG Smart TV Emulator 2013 : SDK 3.0 or higher</td>
</tr>
</tbody>
</table>

**osdResolution**

**Description**
The NetCast Platform provides a 'osdResolution' read-only property in the Device Information plugin object. Developers can get OSD resolution which LG web engine uses for rendering web content using this property. It will return the OSD resolution as an enumerated number. The following table shows the mapping rules between the resolutions and enumerated return values.

[Table] Mapping table between the resolutions and enumerated return values

<table>
<thead>
<tr>
<th>OSD resolution</th>
<th>Enumerated return value</th>
</tr>
</thead>
<tbody>
<tr>
<td>640x480</td>
<td>0</td>
</tr>
<tr>
<td>720x576</td>
<td>1</td>
</tr>
<tr>
<td>1280x720</td>
<td>2</td>
</tr>
</tbody>
</table>
**OSD resolution**

<table>
<thead>
<tr>
<th>Resolution</th>
<th>Enumerated return value</th>
</tr>
</thead>
<tbody>
<tr>
<td>1920x1080</td>
<td>3</td>
</tr>
<tr>
<td>1366x768</td>
<td>4</td>
</tr>
</tbody>
</table>

**Syntax**

device.osdResolution;

**Example**

```javascript
var device = document.getElementById("device");
osdResolution = device.osdResolution;
```

---

**Supported SDK / Emulator Version**

<table>
<thead>
<tr>
<th>SDK Version</th>
<th>Emulator Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.5 or higher</td>
<td>LG Smart TV Emulator 2011 : SDK 1.5 or higher</td>
</tr>
<tr>
<td></td>
<td>LG Smart TV Emulator 2012 : SDK 2.0 or higher</td>
</tr>
<tr>
<td></td>
<td>LG Smart TV Emulator 2013 : SDK 3.0 or higher</td>
</tr>
</tbody>
</table>

---

**networkType**

**Description**

The NetCast Platform provides a `networkType` read-only property in the Device Information plugin object. Developers can get network type using this property. It will return the network type as an enumerated number. The following table shows the mapping rules between network types and enumerated return values.

<table>
<thead>
<tr>
<th>Network type</th>
<th>Numeric type of return value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Wired network</td>
<td>0</td>
</tr>
<tr>
<td>Wireless network</td>
<td>1</td>
</tr>
</tbody>
</table>

**Syntax**

device.networkType;

**Example**

```javascript
var device = document.getElementById("device");
networkType = device.networkType;
```

**Supported SDK / Emulator Version**

<table>
<thead>
<tr>
<th>SDK Version</th>
<th>Emulator Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.5 or higher</td>
<td>LG Smart TV Emulator 2011 : Not Supported</td>
</tr>
<tr>
<td></td>
<td>LG Smart TV Emulator 2012 : SDK 2.2 or higher</td>
</tr>
<tr>
<td></td>
<td>LG Smart TV Emulator 2013 : SDK 3.0 or higher</td>
</tr>
</tbody>
</table>

---

**net_macAddress**

**Description**

The NetCast Platform provides a `net_macAddress` read-only property in the Device Information plugin object. Developers can get MAC address using this property. It will return MAC address as a string type.

**Syntax**

```javascript
device.net_macAddress;
```
Example

```javascript
var device = document.getElementById("device");
macAddress = device.net_macAddress;
```

### Supported SDK / Emulator Version

<table>
<thead>
<tr>
<th>SDK Version</th>
<th>Emulator Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.5 or higher</td>
<td>LG Smart TV Emulator 2011: Not Supported</td>
</tr>
<tr>
<td></td>
<td>LG Smart TV Emulator 2012: SDK 2.2 or higher</td>
</tr>
<tr>
<td></td>
<td>LG Smart TV Emulator 2013: SDK 3.0 or higher</td>
</tr>
</tbody>
</table>

### drmClientInfo

**Description**
The NetCast Platform provides a 'drmClientInfo' read-only property in the Device Information plugin object. Developers can get DRM client information using this property. It will return the DRM client information as a string type.

**Note**
This API is only applicable for the pre-delivery method of WMDRM.

**Syntax**

```javascript
device.drmClientInfo;
```

**Example**

```javascript
var device = document.getElementById("device");
drmClientInfo = device.drmClientInfo;
```

**See Also**

setDrmLicenseInfo

### Supported SDK / Emulator Version

<table>
<thead>
<tr>
<th>SDK Version</th>
<th>Emulator Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.5 or higher</td>
<td>LG Smart TV Emulator 2011: Not Supported</td>
</tr>
<tr>
<td></td>
<td>LG Smart TV Emulator 2012: Not Supported</td>
</tr>
<tr>
<td></td>
<td>LG Smart TV Emulator 2013: Not Supported</td>
</tr>
</tbody>
</table>

### net_dhcp

**Description**
The NetCast Platform provides a 'net_dhcp' read-only property in the Device Information plugin object. Developers can get DHCP information using this property. It returns true if the system uses DHCP to access the network.

**Syntax**

```javascript
device.net_dhcp;
```

**Example**

```javascript
var device = document.getElementById("device");
useDHCP = device.net_dhcp;
```

### Supported SDK / Emulator Version

<table>
<thead>
<tr>
<th>SDK Version</th>
<th>Emulator Version</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>LG Smart TV Emulator 2011: Not Supported</td>
</tr>
<tr>
<td></td>
<td>LG Smart TV Emulator 2012: Not Supported</td>
</tr>
<tr>
<td></td>
<td>LG Smart TV Emulator 2013: Not Supported</td>
</tr>
</tbody>
</table>
### net_isConnected

**Description**
The NetCast Platform provides a `net_isConnected` read-only property in the Device Information plugin object. Developers can get network connection information using this property. It returns true when the system is connected to the network. “Connected” means the status when an ethernet cable is connected and Internet is available.

**Syntax**
```
device.net_isConnected;
```

**Example**
```
var device = document.getElementById("device");
isConnected = device.net_isConnected;
```

<table>
<thead>
<tr>
<th>Supported SDK / Emulator Version</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>SDK Version</strong></td>
</tr>
<tr>
<td>---</td>
</tr>
<tr>
<td>1.5 or higher</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
</tbody>
</table>

### net_hasIP

**Description**
The NetCast Platform provides a `net_hasIP` read-only property in the Device Information plugin object. Developers can get IP information using this property. It returns true if the system has valid IP address. It returns false if the IP address has been set by DHCP.

**Syntax**
```
device.net_hasIP;
```

**Example**
```
var device = document.getElementById("device");
hasIP = device.net_hasIP;
```

<table>
<thead>
<tr>
<th>Supported SDK / Emulator Version</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>SDK Version</strong></td>
</tr>
<tr>
<td>---</td>
</tr>
<tr>
<td>1.5 or higher</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
</tbody>
</table>

### net_ipAddress

**Description**
The NetCast Platform provides a `net_ipAddress` read-only property in the Device Information plugin object. Developers can get IP address using this property. It returns true if the system has valid IP address. It returns the IP address of the system as a string type.

**Syntax**
device.net_ipAddress;

**Example**
```
var device = document.getElementById("device");
ipAddress = device.net_ipAddress;
```

<table>
<thead>
<tr>
<th>Supported SDK / Emulator Version</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>SDK Version</strong></td>
</tr>
<tr>
<td>------------------</td>
</tr>
<tr>
<td>1.5 or higher</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
</tbody>
</table>

**net_netmask**

**Description**
The NetCast Platform provides a 'net_netmask' read-only property in the Device Information plugin object. Developers can get netmask using this property. It returns the netmask value of the system as a string type.

**Syntax**
```
device.net_netmask;
```

**Example**
```
var device = document.getElementById("device");
netmask = device.net_netmask;
```

<table>
<thead>
<tr>
<th>Supported SDK / Emulator Version</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>SDK Version</strong></td>
</tr>
<tr>
<td>------------------</td>
</tr>
<tr>
<td>1.5 or higher</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
</tbody>
</table>

**net_gateway**

**Description**
The NetCast Platform provides a 'net_gateway' read-only property in the Device Information plugin object. Developers can get gateway address using this property. It returns the gateway address value of the system as a string type.

**Syntax**
```
device.net_gateway;
```

**Example**
```
var device = document.getElementById("device");
gateway = device.net_gateway;
```

<table>
<thead>
<tr>
<th>Supported SDK / Emulator Version</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>SDK Version</strong></td>
</tr>
<tr>
<td>------------------</td>
</tr>
<tr>
<td>1.5 or higher</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
</tbody>
</table>

**net_dns1**
Description
The NetCast Platform provides a ‘net_dns1’ read-only property in the Device Information plugin object. Developers can get DNS1 address using this property. It returns the DNS1 address value of the system as a string type.

Syntax
device.net_dns1;

Example
var device = document.getElementById("device");
dns1 = device.net_dns1;

Supported SDK / Emulator Version

<table>
<thead>
<tr>
<th>SDK Version</th>
<th>Emulator Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.5 or higher</td>
<td>LG Smart TV Emulator 2011 : Not Supported</td>
</tr>
<tr>
<td></td>
<td>LG Smart TV Emulator 2012 : SDK 2.2 or higher</td>
</tr>
<tr>
<td></td>
<td>LG Smart TV Emulator 2013 : SDK 3.0 or higher</td>
</tr>
</tbody>
</table>

net_dns2

Description
The NetCast Platform provides a ‘net_dns2’ read-only property in the Device Information plugin object. Developers can get DNS2 address using this property. It returns the DNS2 address value of the system as a string type.

Syntax
device.net_dns2;

Example
var device = document.getElementById("device");
dnss = device.net_dns2;

Supported SDK / Emulator Version

<table>
<thead>
<tr>
<th>SDK Version</th>
<th>Emulator Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.5 or higher</td>
<td>LG Smart TV Emulator 2011 : Not Supported</td>
</tr>
<tr>
<td></td>
<td>LG Smart TV Emulator 2012 : SDK 2.2 or higher</td>
</tr>
<tr>
<td></td>
<td>LG Smart TV Emulator 2013 : SDK 3.0 or higher</td>
</tr>
</tbody>
</table>

supportMouse

Description
Some LG Smart TVs support a pointing remote control unit, called the Magic Remote. Developers can check whether or not supports the Magic Remote in TV by using the ‘supportMouse’ read-only property in the Device Information plugin object. It returns true if LG Smart TV supports Magic Remote, otherwise, it returns false.

See also section ‘Input Device’ in Developing > Developing Web App > App Development Guide section in this Library for detail information about Magic Remote.
See also section ‘userAgent String’ for optional feature supporting.

The following example shows how application authors can get ‘supportMouse’ information.

Syntax
device.supportMouse;

Example
var device = document.getElementById("device");
if(device.supportMouse) {
    supportMouse = device.supportMouse;
Supported SDK / Emulator Version

<table>
<thead>
<tr>
<th>SDK Version</th>
<th>Emulator Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.5 or higher</td>
<td>LG Smart TV Emulator 2011 : Not Supported</td>
</tr>
<tr>
<td></td>
<td>LG Smart TV Emulator 2012 : SDK 2.2 or higher</td>
</tr>
<tr>
<td></td>
<td>LG Smart TV Emulator 2013 : SDK 3.0 or higher</td>
</tr>
</tbody>
</table>

**supportVoiceRecog**

**Description**
The ‘supportVoiceRecog’ API is a read-only API that decides whether this LG Smart TV model supports voice recognition function or not.

**Syntax**
```
device.supportVoiceRecog;
```

**Example**
```
var device = document.getElementById("device");
if(device.supportVoiceRecog) {
    ...
}
```

See Also
- [Voice Recognition Plugin and API](#)

Supported SDK / Emulator Version

<table>
<thead>
<tr>
<th>SDK Version</th>
<th>Emulator Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.2 or higher</td>
<td>LG Smart TV Emulator 2011 : Not Supported</td>
</tr>
<tr>
<td></td>
<td>LG Smart TV Emulator 2012 : SDK 2.2 or higher</td>
</tr>
<tr>
<td></td>
<td>LG Smart TV Emulator 2013 : SDK 3.0 or higher</td>
</tr>
</tbody>
</table>

**supportPentouch**

**Description**
Some PDP TV models support two touch pens. The ‘supportPentouch’ API is a read-only API that gets whether this TV supports touch pen function or not.

**Syntax**
```
device.supportPentouch;
```

**Example**
```
var device = document.getElementById("device");
if(device.supportPentouch) {
    var penInfo1 = device.pentouchInfo(0);
    ...
}
```

See Also
- [pentouchInfo](#)
### support3D

**Description**

Some LG Smart TVs support 3D technology. For 3D videos, it is required that developers check whether the TV has a 3D rendering ability. The NetCast Platform provides a ‘support3D’ read-only property in the Device Information plugin object. It returns true if LG Smart TV supports 3D rendering ability, otherwise, it returns false.

See also section ‘userAgent String’ in Developing > Developing Web App > App Development Guide section in this Library for optional feature support.

**Syntax**

```javascript
device.support3D;
```

**Example**

```javascript
var device = document.getElementById("device");
if(device.support3D) {
    // can do something for 3D video
}
```

#### Supported SDK / Emulator Version

<table>
<thead>
<tr>
<th>SDK Version</th>
<th>Emulator Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.2 or higher</td>
<td>LG Smart TV Emulator 2011 : Not Supported</td>
</tr>
<tr>
<td></td>
<td>LG Smart TV Emulator 2012 : SDK 2.2 or higher</td>
</tr>
<tr>
<td></td>
<td>LG Smart TV Emulator 2013 : SDK 3.0 or higher</td>
</tr>
</tbody>
</table>

### support3DMode

**Description**

The NetCast Platform provides ‘support3DMode’ read-only property in the Device Information plugin object. Developers can check the compatibility of 3D format using this property. It will return ‘true’ if NetCast Platform supports a 3D format for 3D mode. Otherwise, it returns ‘false’.

**Syntax**

```javascript
device.support3DMode;
```

**Parameters**

None

**Return Value**

This API returns the object that includes supported 3d mode. It has the following properties.

[Table] Variables of support3DMode's return object

<table>
<thead>
<tr>
<th>Variable</th>
<th>Meaning</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>from_2d_to_3d</td>
<td>forced change mode of 2d to 3d</td>
<td>boolean</td>
</tr>
<tr>
<td>side_by_side</td>
<td>side by side left / right mode</td>
<td>boolean</td>
</tr>
<tr>
<td>side_by_side_rl</td>
<td>side by side right / left mode</td>
<td>boolean</td>
</tr>
</tbody>
</table>
Variable | Meaning | Type  
---|---|---
`top_bottom` | top_bottom mode | boolean  
`checker_bd` | checker board mode | boolean  

**Example**

```javascript
var device = document.getElementById("device");
var support3DMode = device.support3DMode;
var b2dto3d = support3DMode.from_2d_to_3d;
var bSideBySide = support3DMode.side_by_side;
var bSideBySideRL = support3DMode.side_by_side_rl;
var bTopBottom = support3DMode.top_bottom;
var bCheckerBD = support3DMode.checker_bd;
```

**Supported SDK / Emulator Version**

<table>
<thead>
<tr>
<th>SDK Version</th>
<th>Emulator Version</th>
</tr>
</thead>
</table>
| 1.5 or higher | LG Smart TV Emulator 2011 : Not Supported  
|             | LG Smart TV Emulator 2012 : SDK 2.0 or higher  
|             | LG Smart TV Emulator 2013 : SDK 3.0 or higher  

**preferredSubtitleLanguage**

**Description**

This read-only property returns the value of subtitle, which has been set through TV setting menu. The return value is based on ISO 639-2 language code.

**Note**

In some region, the TV setting menu does not have subtitle language setting menu. Therefore, the property may not have the value.

**Syntax**

```javascript
device.preferredSubtitleLanguage;
```

**Example**

```javascript
var device = document.getElementById("device");
preferredSubtitleLanguage = device.preferredSubtitleLanguage;
```

**Supported SDK / Emulator Version**

<table>
<thead>
<tr>
<th>SDK Version</th>
<th>Emulator Version</th>
</tr>
</thead>
</table>
| 1.5 or higher | LG Smart TV Emulator 2011 : Not Supported  
|             | LG Smart TV Emulator 2012 : SDK 2.0 or higher  
|             | LG Smart TV Emulator 2013 : SDK 3.0 or higher  

**preferredAudioLanguage**

**Description**

This read-only property returns the value of audio language, which has been set through TV setting menu by user. The return value is based on ISO 639-2 language code.

**Note**
In some region, the TV setting menu does not have audio language setting menu. Therefore, the property may not have the value.

Syntax

```javascript
device.preferredAudioLanguage;
```

Example

```javascript
var device = document.getElementById("device");
preferredAudioLanguage = device.preferredAudioLanguage;
```

**Supported SDK / Emulator Version**

<table>
<thead>
<tr>
<th>SDK Version</th>
<th>Emulator Version</th>
</tr>
</thead>
</table>
| 1.5 or higher | LG Smart TV Emulator 2011 : Not Supported  
                  LG Smart TV Emulator 2012 : SDK 2.0 or higher  
                  LG Smart TV Emulator 2013 : SDK 3.0 or higher |

**preferredSubtitleStatus**

**Description**

This read-only property returns the on/off status of subtitle, which has been set through TV setting menu.

**Note**

In some region, the TV setting menu does not have subtitle on/off setting menu. Therefore, the property may not have the value.

Syntax

```javascript
device.preferredSubtitleStatus;
```

Example

```javascript
var device = document.getElementById("device");
isSubtitleOn = device.preferredSubtitleStatus;
```

**Supported SDK / Emulator Version**

<table>
<thead>
<tr>
<th>SDK Version</th>
<th>Emulator Version</th>
</tr>
</thead>
</table>
| 1.5 or higher | LG Smart TV Emulator 2011 : Not Supported  
                  LG Smart TV Emulator 2012 : SDK 2.0 or higher  
                  LG Smart TV Emulator 2013 : SDK 3.0 or higher |

**tvLanguage2**

**Description**

This read-only property returns the value of language, which has been set through TV setting menu by user. The value will be returned in maximum 2 bytes (e.g. en) and is based on ISO 639-1.

Syntax

```javascript
device.tvLanguage2;
```

Example

```javascript
var device = document.getElementById("device");
tvLanguage = device.tvLanguage2;
```

**Supported SDK / Emulator Version**
### tvCountry2

**Description**
This read-only property returns the value of country, which has been set through TV setting menu by user. The value will be returned in maximum 2 bytes (e.g. en) and is based on ISO 3166.

**Syntax**
```
device.tvCountry2;
```

**Example**
```
var device = document.getElementById("device");
tvCountry= device.tvCountry2;
```

**Supported SDK / Emulator Version**

<table>
<thead>
<tr>
<th>SDK Version</th>
<th>Emulator Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.5 or higher</td>
<td>LG Smart TV Emulator 2011 : SDK 2.0 or higher</td>
</tr>
<tr>
<td></td>
<td>LG Smart TV Emulator 2012 : SDK 2.0 or higher</td>
</tr>
<tr>
<td></td>
<td>LG Smart TV Emulator 2013 : SDK 3.0 or higher</td>
</tr>
</tbody>
</table>

### timeZone

**Description**
The ‘timeZone’ API is a read-only API that gets time zone where the current device is located. (e.g. Korea 9)

**Syntax**
```
device.timeZone;
```

**Return Value**
This API returns time zone where the current device is located. (-12 ~ 12)

**Example**
```
var device = document.getElementById("device");
var timeZone = device.timeZone;
```

**Supported SDK / Emulator Version**

<table>
<thead>
<tr>
<th>SDK Version</th>
<th>Emulator Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.2 or higher</td>
<td>LG Smart TV Emulator 2011 : Not Supported</td>
</tr>
<tr>
<td></td>
<td>LG Smart TV Emulator 2012 : Not Supported</td>
</tr>
<tr>
<td></td>
<td>LG Smart TV Emulator 2013 : Not Supported</td>
</tr>
</tbody>
</table>

### platform

**Description**
The ‘platform’ API is a read-only API that gets the platform name of the current device.

**Syntax**
```
device.platform;
```
**Return Value**
This API returns the platform name in String type such as "NetCast 3.0".

**Example**
```javascript
var device = document.getElementById("device");
var platformName = device.platform;
```

**Supported SDK / Emulator Version**

<table>
<thead>
<tr>
<th>SDK Version</th>
<th>Emulator Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.2 or higher</td>
<td>LG Smart TV Emulator 2011 : Not Supported</td>
</tr>
<tr>
<td></td>
<td>LG Smart TV Emulator 2012 : SDK 2.2 or higher</td>
</tr>
<tr>
<td></td>
<td>LG Smart TV Emulator 2013 : SDK 3.0 or higher</td>
</tr>
</tbody>
</table>

**chipset**

**Description**
The 'chipset' API is a read-only API that gets if the current device is high-end or middle-end model.

**Syntax**
```javascript
device.chipset;
```

**Return Value**
*"H12": high-end*
*"M12": middle-end*

**Example**
```javascript
var device = document.getElementById("device");
var chipset = device.chipset;
```

**Supported SDK / Emulator Version**

<table>
<thead>
<tr>
<th>SDK Version</th>
<th>Emulator Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.2 or higher</td>
<td>LG Smart TV Emulator 2011 : SDK 2.2 or higher</td>
</tr>
<tr>
<td></td>
<td>LG Smart TV Emulator 2012 : SDK 2.2 or higher</td>
</tr>
<tr>
<td></td>
<td>LG Smart TV Emulator 2013 : SDK 3.0 or higher</td>
</tr>
</tbody>
</table>
Voice Recognition Plugin and API

The voice recognition plugin is used to use the voice recognition function of the Magic Remote on web applications. The plugin provides the API for voice recognition (converting voice into text). There are two modes for the voice recognition function: dictation and word mode. The language of the voice recognition function can be set under [OPTION > Language > Voice Search Language].

- Word mode: Keyword level of voice recognition. Maximum voice input time of 8 seconds allowed.
- Dictation mode: Sentence level of voice recognition. Maximum voice input time of 30 seconds allowed.

Based on the UX scenario of NetCast 3.0, word mode displays maximum 3 words for voice recognition result and users can select among them. In dictation mode, only one voice recognition result is provided.

The voice recognition plugin and API are supported since NetCast 3.0. The supported languages for the voice recognition are listed in the below tables.

For each region, the following languages are supported for voice recognition. Users can select the language from the voice recognition setting menu.

[Table] Supported languages for voice recognition per region for H12/M12

<table>
<thead>
<tr>
<th>Region</th>
<th>Country</th>
<th>Language</th>
<th>Supported languages for voice recognition</th>
</tr>
</thead>
<tbody>
<tr>
<td>North America</td>
<td>USA</td>
<td>US English, Spanish</td>
<td>US English, Spanish, Canadian French</td>
</tr>
<tr>
<td></td>
<td>Canada</td>
<td>Canadian French</td>
<td></td>
</tr>
<tr>
<td>Korea</td>
<td>Korea</td>
<td>Korean</td>
<td>Korean, US English</td>
</tr>
<tr>
<td>CIS</td>
<td>Russia</td>
<td>Russian</td>
<td>Russian, UK English</td>
</tr>
<tr>
<td>Latin America</td>
<td>Brazil</td>
<td>Portuguese</td>
<td>Portuguese, US English</td>
</tr>
<tr>
<td>Europe</td>
<td>UK</td>
<td>UK English</td>
<td>UK English</td>
</tr>
<tr>
<td></td>
<td>Germany</td>
<td>German</td>
<td>German, UK English</td>
</tr>
<tr>
<td></td>
<td>France</td>
<td>French</td>
<td>French, UK English</td>
</tr>
<tr>
<td></td>
<td>Italy</td>
<td>Italian</td>
<td>Italian, UK English</td>
</tr>
<tr>
<td></td>
<td>Spain</td>
<td>Spanish</td>
<td>Spanish, UK English</td>
</tr>
<tr>
<td></td>
<td>Sweden</td>
<td>Swedish</td>
<td>Swedish, UK English</td>
</tr>
<tr>
<td></td>
<td>Norway</td>
<td>Norwegian</td>
<td>Norwegian, UK English</td>
</tr>
<tr>
<td></td>
<td>Netherlands</td>
<td>Dutch</td>
<td>Dutch, UK English</td>
</tr>
<tr>
<td>Asia &amp; Pacific</td>
<td>Australia</td>
<td>Australian English</td>
<td>Australian English, US English</td>
</tr>
<tr>
<td></td>
<td>China</td>
<td>Chinese</td>
<td>Mandarin, UK English</td>
</tr>
<tr>
<td>Region</td>
<td>Country</td>
<td>Language</td>
<td>Supported languages for voice recognition</td>
</tr>
<tr>
<td>-----------------</td>
<td>-------------------</td>
<td>----------------</td>
<td>--------------------------------------------</td>
</tr>
<tr>
<td><strong>North America</strong></td>
<td><strong>USA, Canada</strong></td>
<td>US English</td>
<td>US English, UK English, Mexican Spanish, Canadian French</td>
</tr>
<tr>
<td>Central America</td>
<td>Mexico</td>
<td>Mexican Spanish</td>
<td>Mexican Spanish, US English, UK English, Canadian French</td>
</tr>
<tr>
<td></td>
<td>Honduras</td>
<td>Mexican Spanish</td>
<td>Mexican Spanish, US English, UK English, Brazilian Portuguese</td>
</tr>
<tr>
<td></td>
<td>Panama</td>
<td>Mexican Spanish</td>
<td></td>
</tr>
<tr>
<td><strong>Korea</strong></td>
<td>Korea</td>
<td>Korean</td>
<td>Korean, US English, UK English</td>
</tr>
<tr>
<td><strong>CIS</strong></td>
<td>Russia</td>
<td>Russian</td>
<td>Russian, US English, UK English</td>
</tr>
<tr>
<td>Latin America</td>
<td>Brazil</td>
<td>Portuguese</td>
<td>Brazilian Portuguese, US English, UK English, Mexican Spanish</td>
</tr>
<tr>
<td></td>
<td>Colombia</td>
<td>Mexican Spanish</td>
<td>Mexican Spanish, US English, UK English, Brazilian Portuguese</td>
</tr>
<tr>
<td><strong>Europe</strong></td>
<td><strong>UK</strong></td>
<td>UK English</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Germany</td>
<td>German</td>
<td></td>
</tr>
<tr>
<td></td>
<td>France</td>
<td>French</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Italy</td>
<td>Italian</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Spain</td>
<td>Spanish</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Sweden</td>
<td>Swedish</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Norway</td>
<td>Norwegian</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Netherlands</td>
<td>Dutch</td>
<td>German, EU French, Italian, EU Spanish, Swedish, Norwegian, Dutch, UK English, EU Portuguese, Danish, Finnish, Polish, Czech, Greek, Turkish, Russian, US English</td>
</tr>
<tr>
<td></td>
<td>Portugal</td>
<td>Portuguese</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Denmark</td>
<td>Danish</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Finland</td>
<td>Finnish</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Poland</td>
<td>Polish</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Czech</td>
<td>Czech</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Greece</td>
<td>Greek</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Turkey</td>
<td>Turkish</td>
<td></td>
</tr>
<tr>
<td><strong>Asia &amp; Pacific</strong></td>
<td><strong>Australia, New Zealand</strong></td>
<td><strong>Australian English</strong></td>
<td>Australian English, US English, UK English, Bahasa Indonesia, Bahasa</td>
</tr>
<tr>
<td>Region</td>
<td>Country</td>
<td>Language</td>
<td>Supported languages for voice recognition</td>
</tr>
<tr>
<td>------------</td>
<td>---------------</td>
<td>-------------------</td>
<td>--------------------------------------------------------</td>
</tr>
<tr>
<td>Indonesia</td>
<td>Bahasa Indonesia</td>
<td></td>
<td>Malaysia / Vietnamese</td>
</tr>
<tr>
<td>Malaysia</td>
<td>Bahasa Malaysia</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Vietnam</td>
<td>Vietnamese</td>
<td></td>
<td></td>
</tr>
<tr>
<td>China</td>
<td>China</td>
<td>Chinese</td>
<td>Mandarin</td>
</tr>
<tr>
<td>Taiwan, Hongkong</td>
<td>Taiwan Taiwanese Mandarin</td>
<td></td>
<td>US English / UK English / Chinese Mandarin / Taiwanese Mandarin / Hong Kong Cantonese</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Japan</td>
<td>Japan</td>
<td>Japanese</td>
<td>US English / UK English / Japanese</td>
</tr>
<tr>
<td>Israel</td>
<td>Israel</td>
<td>UK English</td>
<td>US English / UK English / EU French / UAE Arabic / Saudi Arabic</td>
</tr>
</tbody>
</table>

The following sections describe the API of the Voice Recognition plugin.
- Voice Object
- Properties/Methods/Events

### Voice Object

The MIME type of voice recognition object is ‘application/x-netcast-voice’. The properties of voice recognition object are ‘id’, ‘dictation’, ‘width’, and ‘height’. Note that ‘width’ and ‘height’ must be always set to 0.

The dictation property can be used when the voice recognition function is in dictation mode. The default is the word mode.

The following sample code is an example of Voice Recognition object in HTML.

```html
<object type="application/x-netcast-voice"
   id="voice"
   dictation="on"
   width="0"
   height="0">
</object>
```

### Properties/Methods/Events

Methods, properties, and events are listed in the following sections.
- Properties
- Methods
- Events
Properties

Properties of Voice Recognition Plugin and API are as follows:

isInitialized

Description
This property is used to determine whether the voice recognition function is initialized and returns the value as Boolean.

Syntax
voice.isInitialized;

Example
var voice = document.getElementById('voice');
var isInitialized = voice.isInitialized;

Supported SDK / Emulator Version

<table>
<thead>
<tr>
<th>SDK Version</th>
<th>Emulator Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.1 or higher</td>
<td>LG Smart TV Emulator 2011 : Not Supported</td>
</tr>
<tr>
<td></td>
<td>LG Smart TV Emulator 2012 : SDK 2.2 or higher</td>
</tr>
<tr>
<td></td>
<td>LG Smart TV Emulator 2013 : SDK 3.0 or higher</td>
</tr>
</tbody>
</table>

isEnable

Description
This property is used to determine whether the Magic Remote is paired (including its type) and whether the voice recognition function is enabled, and returns the value as Boolean.

Syntax
voice.isEnable;

Example
var voice = document.getElementById('voice');
var isEnable = voice.isEnable;

See Also
supportVoiceRecog

Supported SDK / Emulator Version

<table>
<thead>
<tr>
<th>SDK Version</th>
<th>Emulator Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.1 or higher</td>
<td>LG Smart TV Emulator 2011 : Not Supported</td>
</tr>
<tr>
<td></td>
<td>LG Smart TV Emulator 2012 : SDK 2.2 or higher</td>
</tr>
<tr>
<td></td>
<td>LG Smart TV Emulator 2013 : SDK 3.0 or higher</td>
</tr>
</tbody>
</table>

dictation

Description
This property is used to determine whether the voice recognition function is in dictation mode and returns the value as string (on/off). If the return value is "off", the function is in word mode. These methods can also be used to select the mode.

Syntax
voice.dictation;

Example

```javascript
var voice = document.getElementById('voice');
var dictationMode = voice.dictation;
voice.dictation = "on"; // dictation mode
voice.dictation = "off"; // word mode
```

Supported SDK / Emulator Version

<table>
<thead>
<tr>
<th>SDK Version</th>
<th>Emulator Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.1 or higher</td>
<td>LG Smart TV Emulator 2011 : Not Supported</td>
</tr>
<tr>
<td></td>
<td>LG Smart TV Emulator 2012 : SDK 2.2 or higher</td>
</tr>
<tr>
<td></td>
<td>LG Smart TV Emulator 2013 : SDK 3.0 or higher</td>
</tr>
</tbody>
</table>

language

Description
This property is used to return the language which is set by user from the setting menu.

Syntax

```javascript
voice.language;
```

Example

```javascript
var voice = document.getElementById('voice');
var voiceRecogLanguage = voice.language;
```

[Table] Language list

<table>
<thead>
<tr>
<th>Language</th>
<th>Value</th>
<th>Language</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>German</td>
<td>de</td>
<td>Swedish</td>
<td>sv</td>
</tr>
<tr>
<td>English</td>
<td>en</td>
<td>Norwegian</td>
<td>no</td>
</tr>
<tr>
<td>Spanish</td>
<td>es</td>
<td>Dutch</td>
<td>nl</td>
</tr>
<tr>
<td>French</td>
<td>fr</td>
<td>Russian</td>
<td>ru</td>
</tr>
<tr>
<td>Italian</td>
<td>it</td>
<td>Chinese</td>
<td>zh</td>
</tr>
<tr>
<td>Korean</td>
<td>ko</td>
<td>Portuguese</td>
<td>pt</td>
</tr>
</tbody>
</table>

Supported SDK / Emulator Version

<table>
<thead>
<tr>
<th>SDK Version</th>
<th>Emulator Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.2 or higher</td>
<td>LG Smart TV Emulator 2011 : Not Supported</td>
</tr>
<tr>
<td></td>
<td>LG Smart TV Emulator 2012 : SDK 2.4 or higher</td>
</tr>
<tr>
<td></td>
<td>LG Smart TV Emulator 2013 : SDK 3.0 or higher</td>
</tr>
</tbody>
</table>
Methods
Methods of Voice Recognition Plugin and API are as follows:

startRecognition

Description
This method is used to call native UI of the voice recognition function and receive the result as an event.

Syntax
voice.startRecognition();

Parameters
None

Return Value
None

Example
function startVoiceRecognition()
{
    var voice = document.getElementById('voice');
    voice.startRecognition();
}

Supported SDK / Emulator Version

<table>
<thead>
<tr>
<th>SDK Version</th>
<th>Emulator Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.1 or higher</td>
<td>LG Smart TV Emulator 2011 : Not Supported</td>
</tr>
<tr>
<td></td>
<td>LG Smart TV Emulator 2012 : SDK 2.2 or higher</td>
</tr>
<tr>
<td></td>
<td>LG Smart TV Emulator 2013 : SDK 3.0 or higher</td>
</tr>
</tbody>
</table>
Events

Events of Voice Recognition Plugin and API are as follows:

onrecogniz voz

Description
This event is added in order to receive the voice recognition result from the TV.

Syntax
voice.onrecognizevoice;

Example
function initPage()
{
  var voice = document.getElementById('voice');
  voice.onrecognizevoice = function(e){
    document.write(e);
  };
}

Supported SDK / Emulator Version

<table>
<thead>
<tr>
<th>SDK Version</th>
<th>Emulator Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.1 or higher</td>
<td>LG Smart TV Emulator 2011 : Not Supported</td>
</tr>
<tr>
<td></td>
<td>LG Smart TV Emulator 2012 : SDK 2.2 or higher</td>
</tr>
<tr>
<td></td>
<td>LG Smart TV Emulator 2013 : SDK 3.0 or higher</td>
</tr>
</tbody>
</table>

onbuttonenable

Description
The event is added 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 TV, false is returned; otherwise, true. The voice recognition button is enabled or disabled based on the value returned by this event.

Syntax
voice.onbuttonenable;

Example
function initPage()
{
  var voice = document.getElementById('voice');
  var button = document.getElementById('button');
  voice.onbuttonenable = function(e){
    button.disabled = !e;
  };
}

Supported SDK / Emulator Version

<table>
<thead>
<tr>
<th>SDK Version</th>
<th>Emulator Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.1 or higher</td>
<td>LG Smart TV Emulator 2011 : Not Supported</td>
</tr>
<tr>
<td></td>
<td>LG Smart TV Emulator 2012 : SDK 2.2 or higher</td>
</tr>
<tr>
<td></td>
<td>LG Smart TV Emulator 2013 : SDK 3.0 or higher</td>
</tr>
</tbody>
</table>
AppToApp Plugin and API

AppToApp plugin is used for communication with mobile device (like Android, iOS) in the application.

AppToApp Plugin and API is supported on NetCast 3.0 since LG Smart TV SDK 2.2.

The following sections describe the API of the AppToApp plugin.
  • AppToApp Object  
  • Methods/Events

AppToApp Object

The MIME type of apptoapp object is ‘application/x-netcast-apptoapp’. Properties of apptoapp object are ‘id’, ‘width’, and ‘height’. Note that ‘width’ and ‘height’ must be always set to 0.

The following sample code is an example of apptoapp object in HTML.

```html
<object type="application/x-netcast-apptoapp"
   id="apptoapp"
   width="0"
   height="0">
</object>
```

Methods/Events

Methods and events are listed in the following sections.
  • Methods  
  • Events
Methods

Methods of AppToApp Plugin and API are as follows:

sendMessage

Description
The 'sendMessage' API sends the message to all devices (Android, iOS) those are connected to LG Smart TV. For detailed information, refer to the References > UDAP Specifications section in this Library.

Syntax
```
apptoapp.sendMessage(type, message);
```

Parameters
- **type**
  
  [in] Custom type that indicates the message type (32-bit unsigned integer)

- **message**
  
  [in] Message (String type within 512 bytes)

Return Value
None

Example
```
apptoapp.sendMessage(12, "my message");
```

Supported SDK / Emulator Version

<table>
<thead>
<tr>
<th>SDK Version</th>
<th>Emulator Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.2 or higher</td>
<td>LG Smart TV Emulator 2011 : Not Supported</td>
</tr>
<tr>
<td></td>
<td>LG Smart TV Emulator 2012 : SDK 2.2 or higher</td>
</tr>
<tr>
<td></td>
<td>LG Smart TV Emulator 2013 : SDK 3.0 or higher</td>
</tr>
</tbody>
</table>

sendMessageTo

Description
The 'sendMessageTo' API sends the message to a specified connected device (Android, iOS). For detailed information, refer to the References > UDAP Specifications section in this Library.

Syntax
```
apptoapp.sendMessageTo(address, type, message);
```

Parameters
- **address**
  
  [in] Client address (32-bit integer address)
  
  (Available only if the address value has been returned at least once by 'onreceiveMessage' API from the connected device.

- **type**
  
  [in] Custom type (32-bit unsigned integer)

- **message**
  
  [in] Message (String type within 512 bytes)

Return Value
None

Example
```
apptoapp.sendMessageTo(3232235530, 12, "my message");
```

Supported SDK / Emulator Version

<table>
<thead>
<tr>
<th>SDK Version</th>
<th>Emulator Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.2 or higher</td>
<td>LG Smart TV Emulator 2011 : Not Supported</td>
</tr>
<tr>
<td></td>
<td>LG Smart TV Emulator 2012 : SDK 2.2 or higher</td>
</tr>
<tr>
<td></td>
<td>LG Smart TV Emulator 2013 : SDK 3.0 or higher</td>
</tr>
</tbody>
</table>
Events

Events of AppToApp Plugin and API are as follows:

onreceivingmessage

Description
This event handles the received message which is sent from the mobile device (Android, iOS).
(For detailed information, refer to the References > UDAP Specifications section in this Library.

Syntax

```javascript
apptoapp.onreceivingmessage = processCustomMessageFunction;
```

Example

```javascript
function processCustomMessageFunction(addr, msgType, msg)
{
    var last_client = addr; // message send device id
    if(msgType == 100) {
        // play music or something with msg
    }
    ...
}
</script>
```

Supported SDK / Emulator Version

<table>
<thead>
<tr>
<th>SDK Version</th>
<th>Emulator Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.2 or higher</td>
<td>LG Smart TV Emulator 2011: Not Supported</td>
</tr>
<tr>
<td></td>
<td>LG Smart TV Emulator 2012: SDK 2.2 or higher</td>
</tr>
<tr>
<td></td>
<td>LG Smart TV Emulator 2013: SDK 3.0 or higher</td>
</tr>
</tbody>
</table>
Sound Plugin and API

Sound Plugin and API are used for playing sound effect, a short wav file, while background music is being played using Media Player plugin. (It is possible to use Sound plugin with no background music being played.)

A wav file settings should be as below:
- Sampling rate : 44.1 K
- Audio bps: 16 bit
- Audio channel : mono

Multiple sound plugins can be created at the same time.

The following sections describe the API of the Sound plugin.
• Sound Object
• Methods

Sound Object

The MIME type of sound object is ‘audio/x-wav’. The properties of sound object are ‘id’, ‘data’, ‘width’, and ‘height’. Note that ‘width’ and ‘height’ must be always set to 0.

The following sample code is an example of Sound object in HTML.

```html
<object type="audio/x-wav"
    id="sound"
data=SOUND_URL
    width="0"
    height="0">
</object>

Methods

Method are listed in the following sections.
• Method
Method
Methods of Sound Plugin and API are as follows:

play

Description
This method is used to play a sound.

Syntax
sound.play();

Example
var sound = document.getElementById('sound');
sound.play();

Supported SDK / Emulator Version

<table>
<thead>
<tr>
<th>SDK Version</th>
<th>Emulator Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.2 or higher</td>
<td>LG Smart TV Emulator 2011 : Not Supported</td>
</tr>
<tr>
<td></td>
<td>LG Smart TV Emulator 2012 : SDK 2.2 or higher</td>
</tr>
<tr>
<td></td>
<td>LG Smart TV Emulator 2013 : SDK 3.0 or higher</td>
</tr>
</tbody>
</table>
DRMAgent Plugin and API

DRMAgent plugin is used for the generic playout of PlayReady content using the NetCast media object and using the drmAgent object.

DRMAgent Plugin and API is supported on NetCast 3.0 since LG Smart TV SDK 2.2.

The following sections describe the API of the DRMAgent plugin.
- DRMAgent Object
- Methods/Events

**DRMAgent Object**

The MIME type of DRMAgent object is ‘application/oipfDrmAgent’. Property of DRMAgent object is ‘id’, ‘width’, and ‘height’.

The following sample code is an example of DRMAgent object in HTML.

```
<object type="application/oipfDrmAgent"
   id="drmAgent">
   width="0"
   height="0">
</object>
```

**Methods/Events**

Methods and events are listed in the following sections.
- Methods
- Events
Methods

Methods of DRMAgent Plugin and API are as follows:

sendDRMMessage

Description
This API is used for sending message to DRM agent, using a message type as defined by the DRM system.

Syntax
```
drmAgent.sendDRMMessage(msgType, msg, DRMSystemID);
```

Parameters

- **msgType**
  
  [in] A globally unique message type as defined by the DRM system, for example, “application/vnd.ms-playready.initiator+xml” (i.e. MIME-type of PlayReady messages)

- **msg**
  
  [in] The message to be provided to the underlying DRM agent formatted according to the message type as indicated by attribute msgType. Valid format for the msg parameter are message formats described in DRMAgent Specific.

- **DRMSystemID**
  
  [in] For PlayReady, the DRMSystemID value is “urn:dvb:casystemid:19219”.

Return Value

A unique ID to identify the message

Example

```
//Smoother Streaming object
<object id="video" type="application/vnd.ms-sstr+xml"></object>

//Play Video
function playVideo()
{
  var video = document.getElementById("video");
  video.data = "http://content.contoso.com/smoothstreamingcontent.ism/manifest";
  video.play(1);
}

// License Pre-acquisition
// send licence request using sendDRMMessage
function sendLicenceRequest()
{
  var msgType = "application/vnd.ms-playready.initiator+xml";
  var xmlLicenceAcquisition =
      '<?xml version="1.0" encoding="utf-8"?>' +
      '<PlayReadyInitiator
        xmlns="http://schemas.microsoft.com/DRM/2007/03/protocols/">' +
        '<LicenseAcquisition>' +
        '<Header>' +
        '<WRMHEADER version="4.0.0.0">' +
        '<DATA>' +
        '<LA_URL>http://playready.contoso.com/rightsmanager.asmx</LA_URL>' +
        '<KID>mQxQH65mRkWJpAsdAqEDlQ==</KID>' +
        '<CHECKSUM>fsSmPwHRHgM=</CHECKSUM>' +
        '</DATA>' +
        '</WRMHEADER>' +
        '</Header>' +
        '</LicenseAcquisition>' +
      '</PlayReadyInitiator>';
  var DRMSysID = "urn:dvb:casystemid:19219";
  var oipfDrm = document.getElementById('oipfDrm');
```
oipfDrm.onDRMMessageResult = HandleOnDRMMessageResult;
oipfDrm.onDRMRightsError = HandleOnDRMRightsError;
oipfDrm.sendDRMMessage(msgType, xmlLicenceAcquisition, DRMSysID);

**Supported SDK / Emulator Version**

<table>
<thead>
<tr>
<th>SDK Version</th>
<th>Emulator Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.2 or higher</td>
<td>LG Smart TV Emulator 2011: Not Supported</td>
</tr>
<tr>
<td></td>
<td>LG Smart TV Emulator 2012: Not Supported</td>
</tr>
<tr>
<td></td>
<td>LG Smart TV Emulator 2013: Not Supported</td>
</tr>
</tbody>
</table>
Events

Events of DRMAgent Plugin and API are as follows:

onDRMMessageResult

Description
The script function that is called when the underlying DRM agent has a result message to report to the current HTML page as a consequence of a call to sendDRMMessage. The specified script function is called with three arguments msgID, resultMsg, and resultCode which are defined as follows.

Syntax

```javascript
drmAgent.onDRMMessageResult = HandleOnDRMMessageResult;
```

Parameters
None

Return Value

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Description</th>
<th>Semantics</th>
</tr>
</thead>
<tbody>
<tr>
<td>msgID</td>
<td>[in] Identifies the original message which has led to this resulting message.</td>
<td></td>
</tr>
<tr>
<td>resultMsg</td>
<td>[in] DRM system specific result message. Also see Valid Responses of Message Result in DRMAgent Specific.</td>
<td></td>
</tr>
<tr>
<td>resultCode</td>
<td>[in] Result code</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Result Code</th>
<th>Description</th>
<th>Semantics</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Successful</td>
<td>The action(s) requested by SendDRMMessage() completed successfully</td>
</tr>
<tr>
<td>1</td>
<td>Unknown Error</td>
<td>SendDRMMessage() failed because an unspecified error occurred.</td>
</tr>
<tr>
<td>2</td>
<td>Cannot Process Request</td>
<td>SendDRMMessage() failed because the DRM agent was unable to complete the necessary computations in the time allotted.</td>
</tr>
<tr>
<td>3</td>
<td>Unknown MIME Type</td>
<td>SendDRMMessage() failed, because the specified Mime Type is unknown for the specified DRM system indicated in the MIME type</td>
</tr>
<tr>
<td>4</td>
<td>User Consent Needed</td>
<td>SendDRMMessage() failed because user consent is needed for that action</td>
</tr>
</tbody>
</table>

Example

```javascript
drmAgent.onDRMMessageResult = function(msgId, resultMsg, resultCode)
{
    if (resultCode == 0)
    {
        var videoPlayer = document.getElementById("VideoPlayer");
        videoPlayer.data = "http://content.contoso.com/drmed/smoothstreamingcontent.ism/manifest";
        videoPlayer.play(1);
    }
    else{
        console.log("download failed. error:" + resultCode);
    }
}
```

Supported SDK / Emulator Version

<table>
<thead>
<tr>
<th>SDK Version</th>
<th>Emulator Version</th>
</tr>
</thead>
</table>
### SDK Version | Emulator Version
---|---
2.2 or higher | LG Smart TV Emulator 2011: Not Supported
 | LG Smart TV Emulator 2012: Not Supported
 | LG Smart TV Emulator 2013: Not Supported

### DRMAgent Specific

#### Pre Acquisition

```xml
<?xml version="1.0" encoding="utf-8"?>
  <LicenseAcquisition>
    <WRMHEADER xmlns=http://schemas.microsoft.com/DRM/2007/03/PlayReadyHeader version="4.0.0.0">
      <DATA>
        <PROTECTINFO>
          <KEYLEN>16</KEYLEN>
          <ALGID>AESCTR</ALGID>
        </PROTECTINFO>
        <LA_URL>http://rm.contoso.com/rightsmanager.asmx</LA_URL>
        <KID>lFmb2gxg0Cr5bfEnJXgJeA==</KID>
        <CHECKSUM>P7ORpD2IpA==</CHECKSUM>
      </DATA>
    </WRMHEADER>
    <CustomData>AuthZToken XYZ</CustomData>
  </LicenseAcquisition>
</PlayReadyInitiator>
```

<table>
<thead>
<tr>
<th>Element</th>
<th>Support by Device</th>
<th>Include by Service</th>
</tr>
</thead>
<tbody>
<tr>
<td>Header</td>
<td>Mandatory</td>
<td>Mandatory</td>
</tr>
<tr>
<td>CustomData</td>
<td>Mandatory</td>
<td>Optional</td>
</tr>
</tbody>
</table>

#### Authentication with Custom Data for Post-Acquisition

```xml
<?xml version="1.0" encoding="utf-8"?>
  <LicenseServerUriOverride>
    <LA_URL>http://rm.contoso.com/rightsmanager.asmx</LA_URL>
  </LicenseServerUriOverride>
</PlayReadyInitiator>
```

After this message, all further license post-acquisitions send the license challenge to the specified LA_URL, not to the default LA_URL contained in the WRMHEADER of the content (application persistence).

**Note**

License pre-acquisitions are not affected by this command.

```xml
<?xml version="1.0" encoding="utf-8"?>
  <LicenseServerUriOverride>
    <LA_URL></LA_URL>
  </LicenseServerUriOverride>
</PlayReadyInitiator>
```
After this message, all further license post acquisitions send the license challenge to the default LA_URL contained in the WRMHEADER of the content (URL overriding cancelled).

```xml
<Set Challenge CustomData for License Post-acquisition>
<?xml version="1.0" encoding="utf-8"?>
    <SetCustomData>
        <CustomData>AuthZToken XYZ</CustomData>
    </SetCustomData>
</PlayReadyInitiator>
```

After this message, all further license post-acquisitions use the specified value, as the CustomData value of the generated license challenges (application persistence).

Note
- License pre-acquisitions are not affected by this command.
- Other operations like JoinDomain, LeaveDomain and Metering are not affected by this command.

```xml
    <SetCustomData>
        <CustomData></CustomData>
    </SetCustomData>
</PlayReadyInitiator>
```

After this message, all further license post-acquisitions will use no CustomData value in the generated license challenges.

Valid Responses of Message Result

- License acquisition successful, with no CustomData in the license response >
  resultMsg:
  ```xml
  <?xml version="1.0" encoding="utf-8"?>
      <DRM_RESULT>0</DRM_RESULT>
  </PlayReadyResponse>
  ```
  resultCode : 0

- License acquisition successful, with some CustomData in the license response >
  resultMsg:
  ```xml
  <?xml version="1.0" encoding="utf-8"?>
      <DRM_RESULT>0</DRM_RESULT>
      <CustomData>CID=abc</CustomData>
  </PlayReadyResponse>
  ```
  resultCode : 0

- License acquisition failed, server could not deliver a license (server internal error, 0x8004C600) >
  resultMsg:
  ```xml
  <?xml version="1.0" encoding="utf-8"?>
      <DRM_RESULT>2147796480</DRM_RESULT>
  </PlayReadyResponse>
  ```
  resultCode : 1

Persistence of License

PlayReady allows services to deliver a device two types of licenses, whether the trigger of the license acquisition is by the application sendDRMMessage API or by the device media player:

- **Persistent** : A persistent license is stored in the device’s general non-volatile license store (typically a mspr.hds file stored on a hard disk or in a non-volatile memory).
- **Non Persistent** (also known as transient): A non persistent license is stored in the device's nonce-store which is volatile and cleaned every time PlayReady DRM sessions are initialized.

<table>
<thead>
<tr>
<th>License Type</th>
<th>Pre-acquisition triggered by the application (sendDRMMessage)</th>
<th>Automatic post-acquisition triggered by the media player</th>
</tr>
</thead>
<tbody>
<tr>
<td>Persistent License</td>
<td>License is stored in the device’s general purpose permanent license store (mspr.hds). License persists over application restart, across multiple applications, and over a device restart. Device must be able to store at least 16 licenses. Device should clean up the license store from expired licenses on a regular basis.</td>
<td></td>
</tr>
<tr>
<td>Non-Persistent License</td>
<td>Not supported.</td>
<td>License stored in the device’s nonce-store (in ram only) License must have the same persistence scope as the current media player context, to allow playback after license acquisition and until the media player is destroyed (see (*))</td>
</tr>
</tbody>
</table>
Media Device NetCast API

This API is supported only for PDP TV and Media device, not for LCD/LED TV device.

window.NetCastSetScreenSaver

Description
This API is only available in LG media devices.

This API can be used for NetCast application author to control the function of ‘Screen Saver’ in LG device. The life cycle of ‘Screen Saver’ operation is same as the application’s life cycle. The value will be set as the default value, enable ‘Screen Saver’, when users exit the application. However, the application author should enable the operation explicitly in their implementation.

The operations have distinctive screen saving animation of their own. The animation starts after a defined time passes. An application cannot be shown after the operation starts, because ‘Screen Saver’ animation is displayed on the top layer of all OSDs.
For an example of operation, Media device’s ‘Screen Saver’ appears when you leave the player in Stop mode for about five minutes.

If application authors do not want to use the function of ‘screen saver’ while application is running, application authors can use this API. Application authors can use this API, window.NetCastSetScreenSaver(), to disable the function of ‘screen saver’ while application is running.

Syntax
window.NetCastSetScreenSaver(control)

Parameters
control [in] ‘enabled’ or ‘disabled’

Return Value
None

Example
// Enabling the function of Screen Saver
function enableDeviceScreenSaver()
{
    window.NetCastSetScreenSaver('enabled');
}

// Disabling the function of Screen Saver
disableDeviceScreenSaver()
{
    window.NetCastSetScreenSaver('disabled');
}
Image Viewer Framework API

Image Viewer Framework provides the basic functionalities of a photo gallery and image viewer. This framework can be used to include complete image viewer inside any application. Even the APIs of the framework can be used for functionalities like adding image, removing image, going to a particular image, first, last, previous, next, flip, zoom, rotate, slideshow, random show and full screen. Developer can add their own controls and functionalities according to their preference.

This framework also provides IR Remote key navigation to control elements inside the framework. It will always focus on the currently selected control/button.

![Main Screen of Image Viewer Framework](image)

The following sections describe the API of the Image Viewer framework.

- How to Use Image Viewer Framework
- Markup Interface
- JS Interface
- Methods

How to Use Image Viewer Framework

To use this Image Viewer Framework in your application, you must include the framework(api.js) and key codes(keycode.js) in your application (index.html) as below.

```html
<script language="javascript" type="text/javascript"
    src="../lge/framework/ImageViewer/api/api.js">
</script>
<script language="javascript" type="text/javascript"
    src="../lge/framework/ImageViewer/util/keycode.js">
</script>
```

Note

For sample codes of Image Viewer Framework, refer to DISCOVER > Legacy Platform (NetCast) > Tools & Samples > Web Quick Start Sample App & Framework (ImageViewer) in Developer website.
Markup Interface

Create a container div of specified size (width and height) and include the images you want to use inside this div. Developer can set the width and height value as per their requirement and framework will adjust its control size accordingly. We recommend minimum width and height to be 610 x 384 pixels for clarity of controls.

```html
<div id="myGallery" class="myGallery" style="width:610px; height:384px">
  <ul>
    <li><img src="images/airCondition.jpg"></li>
    <li><img src="images/blueRayt.jpg"></li>
    <li><img src="images/carMusicSystem.jpg"></li>
    <li><img src="images/dishWash.jpg"></li>
    <li><img src="images/externalDrive.jpg"></li>
  </ul>
</div>
```

JS Interface

Create the object of API class and call createImgViewer method. It will add the controls and key navigation based on the parameter values.

```javascript
var app = new lge();
app.createImgViewer($("#myGallery"), true, null);
```

Methods

Methods are listed in the following sections.

- Methods
Methods

Methods of Image Viewer Framework API are as follows:

createImgViewer

Description
This API is used to create complete image viewer.
It will add all controls and key navigation among controls based on value control parameter. Key navigation with application controls (outside framework) will be handled based on callBack parameter value.

Syntax
createImgViewer(container, controls, callBack);

Parameters
- container: Container div of application in which image viewer need to be included.
- controls: Boolean. If true, it will add controls and key navigation to the framework. If false, no framework control will be added.
- callBack: The callback function of the application for resetting the application focus.

Return Value
None

Example
var app = new lge();
app.createImgViewer($("#myGallery"), true, null);

Note
The application has its own controls, and the framework also has its own controls. When the focus moves from application to the framework, passed callBack parameter will be called to clear the focus on application.

If developer wants to add own controls, it needs to pass controls parameter as ‘false’ and callBack parameter as ‘null’. In this case developer can use below APIs to add their own controls.
addImage

Description
This API is used to add new image at the end of the existing image list.

Syntax
addImage(url);

Parameters
url The full path of url of the new image to be added.

Return Value
None

Example
var app = new lge();
app.addImage("images/television.jpg");

removeImage

Description
This API is used to remove the currently shown image from the existing image list and show the next image. It shows previous image if the last image is removed. The one remaining image which is the last in the list cannot be removed and a message will show that the one remaining image cannot be removed from the image list.

Syntax
removeImage();

Parameters
None

Return Value
None

Example
var app = new lge();
app.removeImage();
### Supported SDK / Emulator Version

<table>
<thead>
<tr>
<th>SDK Version</th>
<th>Emulator Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.2 or higher</td>
<td>LG Smart TV Emulator 2011 : SDK 2.2 or higher</td>
</tr>
<tr>
<td></td>
<td>LG Smart TV Emulator 2012 : SDK 2.2 or higher</td>
</tr>
<tr>
<td></td>
<td>LG Smart TV Emulator 2013 : SDK 3.0 or higher</td>
</tr>
</tbody>
</table>

### gotoImage

**Description**
This API is used to display the image with the given index of the image list.

**Syntax**
gotoImage(index);

**Parameters**
- index: Image number. It starts with 1 (1,2,3,...n).

**Return Value**
None

**Example**
```javascript
var app = new lge();
app.gotoImage(4);
```

### Supported SDK / Emulator Version

<table>
<thead>
<tr>
<th>SDK Version</th>
<th>Emulator Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.2 or higher</td>
<td>LG Smart TV Emulator 2011 : SDK 2.2 or higher</td>
</tr>
<tr>
<td></td>
<td>LG Smart TV Emulator 2012 : SDK 2.2 or higher</td>
</tr>
<tr>
<td></td>
<td>LG Smart TV Emulator 2013 : SDK 3.0 or higher</td>
</tr>
</tbody>
</table>

### zoom

**Description**
This API is used to show all the zoom controls. This will perform all the zoom operations based on the value of the parameter.

**Syntax**
zoom(str);

**Parameters**
- show: If controls parameter is set to 'true' in createImgViewer API, it will display all the controls related to zoom.
- in/out: in and out are used to perform Zoom In and Zoom Out functionalities.
- left/right/up/down: left, right, up and down are used to pan in respective direction.
- back: If controls parameter is set to 'true' in createImgViewer API, it will redirect the controls to the main controls of image viewer.

**Return Value**
None

**Example**
```javascript
var app = new lge();
app.zoom("show");
```
Note
If developer wants to add own controls, it needs to pass controls parameter as ‘false’ in createImgViewer API. In this case, if developer uses zoom API with ‘show’ or ‘back’ parameter, nothing will happen. Developer can use this API with other parameters - in|out|left|right|up|down.

![Zoom controls of Image Viewer Framework](image)

**Supported SDK / Emulator Version**

<table>
<thead>
<tr>
<th>SDK Version</th>
<th>Emulator Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.2 or higher</td>
<td>LG Smart TV Emulator 2011: SDK 2.2 or higher</td>
</tr>
<tr>
<td></td>
<td>LG Smart TV Emulator 2012: SDK 2.2 or higher</td>
</tr>
<tr>
<td></td>
<td>LG Smart TV Emulator 2013: SDK 3.0 or higher</td>
</tr>
</tbody>
</table>

**rotate**

**Description**
This API is used to rotate the current image by 90 degrees clockwise or counterclockwise based on the value of the parameter.

**Syntax**
```
rotate(str);
```

**Parameters**
- `str` The value of the parameter should be `left` | `right`.

**Return Value**
None

**Example**
```javascript
var app = new lge();
app.rotate("left");
```
### Supported SDK / Emulator Version

<table>
<thead>
<tr>
<th>SDK Version</th>
<th>Emulator Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.2 or higher</td>
<td>LG Smart TV Emulator 2011 : SDK 2.2 or higher</td>
</tr>
<tr>
<td></td>
<td>LG Smart TV Emulator 2012 : SDK 2.2 or higher</td>
</tr>
<tr>
<td></td>
<td>LG Smart TV Emulator 2013 : SDK 3.0 or higher</td>
</tr>
</tbody>
</table>

### flip

**Description**
This API is used to mirror the current image horizontally or vertically based on the value of the parameter.

**Syntax**
```javascript
flip(str);
```

**Parameters**
- `str`: The value of the parameter should be `horizontal` | `vertical`.

**Return Value**
None

**Example**
```javascript
var app = new lge();
app.flip("horizontal");
```

### startSlideShow

**Description**
This API is used to start the slideshow of images on the list. It will also show slideshow controls of image viewer. Default slideshow speed is set to 4 seconds.

**Syntax**
```javascript
startSlideShow();
```

**Parameters**
None

**Return Value**
None

**Example**
```javascript
var app = new lge();
app.startSlideShow();
```
stopSlideShow

Description
This API is used to stop the current slideshow and show the main controls of image viewer.

Syntax
stopSlideShow();

Parameters
None

Return Value
None

Example
var app = new lge();
app.stopSlideShow();

Supported SDK / Emulator Version

<table>
<thead>
<tr>
<th>SDK Version</th>
<th>Emulator Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.2 or higher</td>
<td>LG Smart TV Emulator 2011 : SDK 2.2 or higher</td>
</tr>
<tr>
<td></td>
<td>LG Smart TV Emulator 2012 : SDK 2.2 or higher</td>
</tr>
<tr>
<td></td>
<td>LG Smart TV Emulator 2013 : SDK 3.0 or higher</td>
</tr>
</tbody>
</table>
**setSlideShowSpeed**

**Description**
This API is used to set the slideshow speed. It will also update the speed for currently running slideshow. It will set the speed to slow (6 seconds) or medium (4 seconds) or fast (2 seconds) based on the value of the parameter.

**Syntax**

```javascript
setSlideShowSpeed(str);
```

**Parameters**

- **str**
  The value of the parameter should be **slow** | **medium** | **fast**.

**Return Value**

None

**Example**

```javascript
var app = new lge();
app.setSlideShowSpeed("slow");
```

**Supported SDK / Emulator Version**

<table>
<thead>
<tr>
<th>SDK Version</th>
<th>Emulator Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.2 or higher</td>
<td>LG Smart TV Emulator 2011 : SDK 2.2 or higher</td>
</tr>
<tr>
<td></td>
<td>LG Smart TV Emulator 2012 : SDK 2.2 or higher</td>
</tr>
<tr>
<td></td>
<td>LG Smart TV Emulator 2013 : SDK 3.0 or higher</td>
</tr>
</tbody>
</table>

**getSlideShowSpeed**

**Description**
This API is used to return slideshow speed.

**Syntax**

```javascript
getSlideShowSpeed(slideShowSpeed);
```

**Parameters**

None

**Return Value**

- **slideShowSpeed**
  (string) If ‘slow’ is returned, the speed of slideshow is 6 seconds.
  (string) If ‘medium’ is returned, the speed of slideshow is 4 seconds.
  (string) If ‘fast’ is returned, the speed of slideshow is 2 seconds.

**Example**

```javascript
var app = new lge();
app.getSlideShowSpeed();
```

**Supported SDK / Emulator Version**

<table>
<thead>
<tr>
<th>SDK Version</th>
<th>Emulator Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.2 or higher</td>
<td>LG Smart TV Emulator 2011 : SDK 2.2 or higher</td>
</tr>
<tr>
<td></td>
<td>LG Smart TV Emulator 2012 : SDK 2.2 or higher</td>
</tr>
<tr>
<td></td>
<td>LG Smart TV Emulator 2013 : SDK 3.0 or higher</td>
</tr>
</tbody>
</table>
setRandomSlideShow

Description
This API is used to stop the current slideshow, shuffle image list and start the slideshow in random.

Syntax
setRandomSlideShow();

Parameters
None

Return Value
None

Example
var app = new lge();
app.setRandomSlideShow();

Supported SDK / Emulator Version

<table>
<thead>
<tr>
<th>SDK Version</th>
<th>Emulator Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.2 or higher</td>
<td>LG Smart TV Emulator 2011 : SDK 2.2 or higher</td>
</tr>
<tr>
<td></td>
<td>LG Smart TV Emulator 2012 : SDK 2.2 or higher</td>
</tr>
<tr>
<td></td>
<td>LG Smart TV Emulator 2013 : SDK 3.0 or higher</td>
</tr>
</tbody>
</table>

getRandomSlideShow

Description
This API is used to return boolean value based on whether the slideshow is running in random or not.

Syntax
g getRandomSlideShow();

Parameters
None

Return Value
randomSlideShow (boolean) If `true` is returned, random slideshow is running.
(boolean) If `false` is returned, random slideshow is stopped.

Example
var app = new lge();
app.getRandomSlideShow();

Supported SDK / Emulator Version

<table>
<thead>
<tr>
<th>SDK Version</th>
<th>Emulator Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.2 or higher</td>
<td>LG Smart TV Emulator 2011 : SDK 2.2 or higher</td>
</tr>
<tr>
<td></td>
<td>LG Smart TV Emulator 2012 : SDK 2.2 or higher</td>
</tr>
<tr>
<td></td>
<td>LG Smart TV Emulator 2013 : SDK 3.0 or higher</td>
</tr>
</tbody>
</table>

next

Description
This API is used to show the next image of the image list.
Syntax
next();

Parameters
None

Return Value
None

Example
var app = new lge();
app.next();

Supported SDK / Emulator Version

<table>
<thead>
<tr>
<th>SDK Version</th>
<th>Emulator Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.2 or higher</td>
<td>LG Smart TV Emulator 2011 : SDK 2.2 or higher</td>
</tr>
<tr>
<td></td>
<td>LG Smart TV Emulator 2012 : SDK 2.2 or higher</td>
</tr>
<tr>
<td></td>
<td>LG Smart TV Emulator 2013 : SDK 3.0 or higher</td>
</tr>
</tbody>
</table>

`previous`

Description
This API is used to show the previous image of the image list.

Syntax
previous();

Parameters
None

Return Value
None

Example
var app = new lge();
app.previous();

Supported SDK / Emulator Version

<table>
<thead>
<tr>
<th>SDK Version</th>
<th>Emulator Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.2 or higher</td>
<td>LG Smart TV Emulator 2011 : SDK 2.2 or higher</td>
</tr>
<tr>
<td></td>
<td>LG Smart TV Emulator 2012 : SDK 2.2 or higher</td>
</tr>
<tr>
<td></td>
<td>LG Smart TV Emulator 2013 : SDK 3.0 or higher</td>
</tr>
</tbody>
</table>

`first`

Description
This API is used to show the first image of the image list.

Syntax
first();

Parameters
None

Return Value
None

Example
var app = new lge();
app.first();

Supported SDK / Emulator Version

<table>
<thead>
<tr>
<th>SDK Version</th>
<th>Emulator Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.2 or higher</td>
<td>LG Smart TV Emulator 2011: SDK 2.2 or higher</td>
</tr>
<tr>
<td></td>
<td>LG Smart TV Emulator 2012: SDK 2.2 or higher</td>
</tr>
<tr>
<td></td>
<td>LG Smart TV Emulator 2013: SDK 3.0 or higher</td>
</tr>
</tbody>
</table>

last

Description
This API is used to show the last image of the image list.

Syntax
last();

Parameters
None

Return Value
None

Example
var app = new lge();
app.last();

Supported SDK / Emulator Version

<table>
<thead>
<tr>
<th>SDK Version</th>
<th>Emulator Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.2 or higher</td>
<td>LG Smart TV Emulator 2011: SDK 2.2 or higher</td>
</tr>
<tr>
<td></td>
<td>LG Smart TV Emulator 2012: SDK 2.2 or higher</td>
</tr>
<tr>
<td></td>
<td>LG Smart TV Emulator 2013: SDK 3.0 or higher</td>
</tr>
</tbody>
</table>

setDisplayMode

Description
This API is used to set the display mode to normal or fullscreen.

Syntax
setDisplayMode (str);

Parameters
str
The display mode should be normal | fullscreen based on its value.

Return Value
None
Example

```javascript
var app = new lge();
app.setDisplayMode("normal");
```

![Figure] Full Screen display mode of Image Viewer Framework

**Supported SDK / Emulator Version**

<table>
<thead>
<tr>
<th>SDK Version</th>
<th>Emulator Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.2 or higher</td>
<td>LG Smart TV Emulator 2011 : SDK 2.2 or higher</td>
</tr>
<tr>
<td></td>
<td>LG Smart TV Emulator 2012 : SDK 2.2 or higher</td>
</tr>
<tr>
<td></td>
<td>LG Smart TV Emulator 2013 : SDK 3.0 or higher</td>
</tr>
</tbody>
</table>

**getDisplayMode**

**Description**

This API is used to return the display mode.

**Syntax**

```javascript
getDisplayMode();
```

**Parameters**

None

**Return Value**

- `mode` (string) It should be **Normal** | **Fullscreen**.

**Example**

```javascript
var app = new lge();
app.getDisplayMode();
```

**Supported SDK / Emulator Version**

<table>
<thead>
<tr>
<th>SDK Version</th>
<th>Emulator Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.2 or higher</td>
<td>LG Smart TV Emulator 2011 : SDK 2.2 or higher</td>
</tr>
<tr>
<td></td>
<td>LG Smart TV Emulator 2012 : SDK 2.2 or higher</td>
</tr>
<tr>
<td></td>
<td>LG Smart TV Emulator 2013 : SDK 3.0 or higher</td>
</tr>
</tbody>
</table>
setFrameworkFocus

Description
This API is used to set the key navigation and the focus of the appropriate framework controls based on the position of framework controls with respect to application control. Position of framework with respect to application controls is passed as parameter.

Syntax
setFrameworkFocus(str);

Parameters
str: It should be left | right | up | down.

Return Value
None

Example
var app = new lge();
app.setFrameworkFocus("down");

Supported SDK / Emulator Version
<table>
<thead>
<tr>
<th>SDK Version</th>
<th>Emulator Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.2 or higher</td>
<td>LG Smart TV Emulator 2011: SDK 2.2 or higher</td>
</tr>
<tr>
<td></td>
<td>LG Smart TV Emulator 2012: SDK 2.2 or higher</td>
</tr>
<tr>
<td></td>
<td>LG Smart TV Emulator 2013: SDK 3.0 or higher</td>
</tr>
</tbody>
</table>

removeFrameworkFocus

Description
This API is used to clear the focus of framework controls.

Syntax
removeFrameworkFocus();

Parameters
None

Return Value
None

Example
var app = new lge();
app.removeFrameworkFocus();

Supported SDK / Emulator Version
<table>
<thead>
<tr>
<th>SDK Version</th>
<th>Emulator Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.2 or higher</td>
<td>LG Smart TV Emulator 2011: SDK 2.2 or higher</td>
</tr>
<tr>
<td></td>
<td>LG Smart TV Emulator 2012: SDK 2.2 or higher</td>
</tr>
<tr>
<td></td>
<td>LG Smart TV Emulator 2013: SDK 3.0 or higher</td>
</tr>
</tbody>
</table>
mouseWheel

Description
This API is used to change the index of currently shown image based on the direction of mouse wheel rolled. If wheel of mouse is rolled in forward direction, it will show the next image of the list. If it is rolled in backward direction, then it will show the previous image of the list.

Syntax
mouseWheel();

Parameters
None

Return Value
None

Example
var app = new lge();
app.mouseWheel();

Supported SDK / Emulator Version

<table>
<thead>
<tr>
<th>SDK Version</th>
<th>Emulator Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.2 or higher</td>
<td>LG Smart TV Emulator 2011 : SDK 2.2 or higher</td>
</tr>
<tr>
<td></td>
<td>LG Smart TV Emulator 2012 : SDK 2.2 or higher</td>
</tr>
<tr>
<td></td>
<td>LG Smart TV Emulator 2013 : SDK 3.0 or higher</td>
</tr>
</tbody>
</table>

focusNext

Description
This API is used to return boolean value based on whether the current focus is moved to the next image in same row or not.

Syntax
focusNext();

Parameters
None

Return Value
returnFocus
(booleange) If ‘true’ is returned, there is an image on the right of the currently focused image in same row.
(booleange) If ‘false’ is returned, there is no image on the right of the currently focused image in same row (i.e. last image).

Example
var app = new lge();
app.focusNext();

Supported SDK / Emulator Version

<table>
<thead>
<tr>
<th>SDK Version</th>
<th>Emulator Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.2 or higher</td>
<td>LG Smart TV Emulator 2011 : SDK 2.2 or higher</td>
</tr>
<tr>
<td></td>
<td>LG Smart TV Emulator 2012 : SDK 2.2 or higher</td>
</tr>
<tr>
<td></td>
<td>LG Smart TV Emulator 2013 : SDK 3.0 or higher</td>
</tr>
</tbody>
</table>
**focusPrevious**

**Description**
This API is used to return boolean value based on whether the current focus is moved to the previous image in same row or not.

**Syntax**

```
focusPrevious();
```

**Parameters**
None

**Return Value**

*returnFocus*

(\text{boolean}) If ‘true’ is returned, there is an image on the left of the currently focused image in same row.

(\text{boolean}) If ‘false’ is returned, there is no image on the left of the currently focused image (i.e. last image).

**Example**

```javascript
var app = new lge();
app.focusPrevious();
```

**Supported SDK / Emulator Version**

<table>
<thead>
<tr>
<th>SDK Version</th>
<th>Emulator Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.2 or higher</td>
<td>LG Smart TV Emulator 2011 : SDK 2.2 or higher</td>
</tr>
<tr>
<td></td>
<td>LG Smart TV Emulator 2012 : SDK 2.2 or higher</td>
</tr>
<tr>
<td></td>
<td>LG Smart TV Emulator 2013 : SDK 3.0 or higher</td>
</tr>
</tbody>
</table>

**focusUp**

**Description**
This API is used to return boolean value based on whether the current focus is moved to the one step up or not.

**Syntax**

```
focusUp();
```

**Parameters**
None

**Return Value**

*returnFocus*

(\text{boolean}) If ‘true’ is returned, there is an image above of the currently focused image.

(\text{boolean}) If ‘false’ is returned, there is no image above of the currently focused image.

**Example**

```javascript
var app = new lge();
app.focusUp();
```

**Supported SDK / Emulator Version**

<table>
<thead>
<tr>
<th>SDK Version</th>
<th>Emulator Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.2 or higher</td>
<td>LG Smart TV Emulator 2011 : SDK 2.2 or higher</td>
</tr>
<tr>
<td></td>
<td>LG Smart TV Emulator 2012 : SDK 2.2 or higher</td>
</tr>
<tr>
<td></td>
<td>LG Smart TV Emulator 2013 : SDK 3.0 or higher</td>
</tr>
</tbody>
</table>
**focusDown**

**Description**
This API is used to return boolean value based on whether the current focus is moved to the one step down or not.

**Syntax**
focusDown();

**Parameters**
None

**Return Value**
returnFocus (boolean) If 'true' is returned, there is an image below of the currently focused image. (boolean) If 'false' is returned, there is no image below of the currently focused image.

**Example**
var app = new lge();
app.focusDown();

**Supported SDK / Emulator Version**

<table>
<thead>
<tr>
<th>SDK Version</th>
<th>Emulator Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.2 or higher</td>
<td>LG Smart TV Emulator 2011 : SDK 2.2 or higher</td>
</tr>
<tr>
<td></td>
<td>LG Smart TV Emulator 2012 : SDK 2.2 or higher</td>
</tr>
<tr>
<td></td>
<td>LG Smart TV Emulator 2013 : SDK 3.0 or higher</td>
</tr>
</tbody>
</table>

**doSelection**

**Description**
This API is used to dispatch the click event of the corresponding framework element.

**Syntax**
doSelection();

**Parameters**
None

**Return Value**
None

**Example**
var app = new lge();
app.doSelection();

**Supported SDK / Emulator Version**

<table>
<thead>
<tr>
<th>SDK Version</th>
<th>Emulator Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.2 or higher</td>
<td>LG Smart TV Emulator 2011 : SDK 2.2 or higher</td>
</tr>
<tr>
<td></td>
<td>LG Smart TV Emulator 2012 : SDK 2.2 or higher</td>
</tr>
<tr>
<td></td>
<td>LG Smart TV Emulator 2013 : SDK 3.0 or higher</td>
</tr>
</tbody>
</table>
Media Plugin Video Player Framework API

Media Plugin Video Player Framework API is supported on NetCast 3.0 since LG Smart TV SDK 2.2.

Media Plugin Video Player Framework provides the basic functionalities of the video player. This framework uses the media object to display the video. This framework can be used to include complete Media Plugin Video Player inside any application. Even the APIs of the framework can be used for functionalities like creating video player, stop, play, pause, forward, rewind, toggle full screen and option media. The seek bar or the status bar comes by default. Developer can add their own controls and functionalities according to their preference.

This framework also provides IR Remote key navigation to control elements inside the framework. It will always focus on the currently selected control/button.

![Demo Screen of Video Player Framework in Normal Screen Mode without option button](image1)

![Demo Screen of Video Player Framework in Full Screen Mode with option button](image2)

The following sections describe the API of the Media Plugin Video Framework.

- How to Use Image Viewer Framework
- Markup Interface
- JS Interface
- Methods
How to Use Media Plugin Video Player Framework

To use this Media Plugin Video Player Framework in your application, you must include the framework(api.js) and key codes(keycode.js) in your application (index.html) as below.

```html
<script language="javascript" type="text/javascript"
src="../lge/framework/VideoPlayer/api/api.js">
</script>

<script language="javascript" type="text/javascript"
src="../lge/framework/VideoPlayer/util/keycode.js">
</script>
```

Note

For sample codes of Media Plugin Video Player Framework, refer to DISCOVER > Legacy Platform (NetCast) > Tools & Samples > Web Quick Start Sample App & Framework (MediaPluginVideoPlayer) in LG Developer website.

Markup Interface

Create a container div of specified size (width and height). Developer can set the width and height value as per their requirement. But if the width and height of video player provided by user is less than 320 x 180 pixels, player will automatically reset its size to 320 x 180 pixels (minimum size).

```html
<div id="myVideoPlayer" style="width:640px; height:320px">
</div>
```

JS Interface

Create the object of API class and call createVideoPlayer method. It will add the controls and key navigation based on the parameter values.

```javascript
var app = new lge();
app.createVideoPlayer($("#myVideoPlayer"), true, videoPath);
```
VideoPlayer Framework Implementation

1. The size and position of the video player can be set in the HTML page. If the width and height of the video player provided by user is less than 320 x 180 pixels, player will automatically reset its size to 320 x 180 pixels (minimum size).

2. If the size of the video player is specified as 1280 x 720 pixels, the player will display the following buttons.
   - Stop
   - Play/Pause
   - Forward
   - Rewind
   - Options

3. If the size of the video player is specified less than 1280 x 720 pixels, the player will display the following buttons.
   - Stop
   - Play/Pause
   - Forward
   - Rewind
   - Toggle full screen

4. If the video player toggle in full screen (1280 x 720 pixels), the player will display the following buttons.
   - Stop
   - Play/Pause
   - Forward
   - Rewind
   - Toggle full screen
   - Options

5. If any video player size specified is not the 16:9 aspect ratio, the video player will resize its size to 16:9 keeping the width constant.

Methods

Methods are listed in the following sections.
- Methods
Methods

Methods of Media Plugin Video Player Framework API are as follows:

createVideoPlayer

Description
This API is used to create the video player.

Syntax
createVideoPlayer(container, controls, path);

Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>container</td>
<td>Container div of application in which video player needs to be included.</td>
</tr>
<tr>
<td>controls</td>
<td>Boolean. If true, it will add controls and key navigation to the framework. If false, no framework control will be added.</td>
</tr>
<tr>
<td>path</td>
<td>The path of the video source.</td>
</tr>
</tbody>
</table>

Return Value
None

Example
var app = new lge();
app.createVideoPlayer($("#myVideoPlayer"), false, videoPath);

Note
If developer wants to add own controls, it needs to pass controls parameter as ‘false’. In this case, developer can use the below APIs to add their own controls.

Supported SDK / Emulator Version

<table>
<thead>
<tr>
<th>SDK Version</th>
<th>Emulator Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.2 or higher</td>
<td>LG Smart TV Emulator 2011 : SDK 2.2 or higher</td>
</tr>
<tr>
<td></td>
<td>LG Smart TV Emulator 2012 : SDK 2.2 or higher</td>
</tr>
<tr>
<td></td>
<td>LG Smart TV Emulator 2013 : SDK 3.0 or higher</td>
</tr>
</tbody>
</table>

play

Description
This API is used to play or pause the video based on the state of the video.
If the currently playing video is in pause state, the video is played. And if the currently playing video is in play state, the video is paused.

Syntax
play();

Parameters
None

Return Value
None

Example
var app = new lge();
app.play();
stop

Description
This API is used to stop the currently playing video. All the controls are reset when the video is stopped or finished.

Syntax
stop();

Parameters
None

Return Value
None

Example
var app = new lge();
app.stop();

Supported SDK / Emulator Version
<table>
<thead>
<tr>
<th>SDK Version</th>
<th>Emulator Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.2 or higher</td>
<td>LG Smart TV Emulator 2011 : SDK 2.2 or higher</td>
</tr>
<tr>
<td></td>
<td>LG Smart TV Emulator 2012 : SDK 2.2 or higher</td>
</tr>
<tr>
<td></td>
<td>LG Smart TV Emulator 2013 : SDK 3.0 or higher</td>
</tr>
</tbody>
</table>

rewind

Description
This API is used to seek the video backward by 10 seconds. If the video is in pause state, state is changed to play.

Syntax
rewind();

Parameters
None

Return Value
None

Example
var app = new lge();
app.rewind();

Supported SDK / Emulator Version
<table>
<thead>
<tr>
<th>SDK Version</th>
<th>Emulator Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.2 or higher</td>
<td>LG Smart TV Emulator 2011 : SDK 2.2 or higher</td>
</tr>
<tr>
<td></td>
<td>LG Smart TV Emulator 2012 : SDK 2.2 or higher</td>
</tr>
</tbody>
</table>

|             | LG Smart TV Emulator 2013 : SDK 3.0 or higher |
**fastFoward**

**Description**
This API is used to seek the video forward by 10 seconds. If the seek time is not more than duration of the video, the video is seeked. If the video is in pause state, state is changed to play.

**Syntax**
```javascript
fastFoward();
```

**Parameters**
None

**Return Value**
None

**Example**
```javascript
var app = new lge();
app.fastFoward();
```

**Supported SDK / Emulator Version**

<table>
<thead>
<tr>
<th>SDK Version</th>
<th>Emulator Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.2 or higher</td>
<td>LG Smart TV Emulator 2011 : SDK 2.2 or higher</td>
</tr>
<tr>
<td></td>
<td>LG Smart TV Emulator 2012 : SDK 2.2 or higher</td>
</tr>
<tr>
<td></td>
<td>LG Smart TV Emulator 2013 : SDK 3.0 or higher</td>
</tr>
</tbody>
</table>

**switchToFullScreenMode**

**Description**
This API is used to switch from normal screen mode to full screen mode or vice versa. Full screen mode will set the size to 1280 x 720 pixels.

**Syntax**
```javascript
switchToFullScreenMode();
```

**Parameters**
None

**Return Value**
None

**Example**
```javascript
var app = new lge();
app.switchToFullScreenMode();
```

**Supported SDK / Emulator Version**

<table>
<thead>
<tr>
<th>SDK Version</th>
<th>Emulator Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.2 or higher</td>
<td>LG Smart TV Emulator 2011 : SDK 2.2 or higher</td>
</tr>
<tr>
<td></td>
<td>LG Smart TV Emulator 2012 : SDK 2.2 or higher</td>
</tr>
<tr>
<td></td>
<td>LG Smart TV Emulator 2013 : SDK 3.0 or higher</td>
</tr>
</tbody>
</table>
optionMedia

Description
This API is used to open the quick menu on full screen. MENU(quick menu) can be launched by the user when the video is playing in full screen mode.

Syntax
```java
optionMedia();
```

Parameters
None

Return Value
None

Example
```javascript
var app = new lge();
app.optionMedia();
```

[Figure] Demo Screen of Video Player showing media options

### Supported SDK / Emulator Version

<table>
<thead>
<tr>
<th>SDK Version</th>
<th>Emulator Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.2 or higher</td>
<td>LG Smart TV Emulator 2011 : SDK 2.2 or higher</td>
</tr>
<tr>
<td></td>
<td>LG Smart TV Emulator 2012 : SDK 2.2 or higher</td>
</tr>
<tr>
<td></td>
<td>LG Smart TV Emulator 2013 : SDK 3.0 or higher</td>
</tr>
</tbody>
</table>
HTML5 Video Player Framework API

HTML5 Video Player Framework API is supported on NetCast 3.0 since LG Smart TV SDK 2.2

HTML5 Video Player Framework provides the basic functionalities of the video player. This framework uses the HTML5 Video element to display the video. This framework can be used to include complete HTML5 Video Player inside any application. Even the APIs of the framework can be used for functionalities like creating Video Player, stop, play, pause, forward, rewind, toggle full screen and option media. The seek bar or the status bar comes by default. Developer can add their own controls and functionalities according to their preference.

This framework also provides IR Remote key navigation to control elements inside the framework. It will always focus the currently selected control/button.

The following sections describe the API of the HTML5 Video Player framework.
How to Use HTML5 Video Player Framework

To use this HTML5 Video Player Framework in your application, you must include the framework(api.js) and key codes(keycode.js) in your application (index.html) as below.

```html
<script language="javascript" type="text/javascript" src="../lge/framework/VideoPlayer/api/api.js">
</script>
<script language="javascript" type="text/javascript" src="../lge/framework/VideoPlayer/util/keycode.js">
</script>
```

**Note**

For sample codes of HTML5 Video Player Framework, refer to DISCOVER > Legacy Platform (NetCast) > Tools & Samples > Web Quick Start Sample App & Framework (HTML5VideoPlayer) in LG Developer website.

### Markup Interface

Create a container div of specified size (width and height). If the width and height of video player provided by user is less than 320 x 180 pixels, player will automatically reset its size to 320 x 180 pixels (minimum size).

```html
<div id="myVideoPlayer" style="width:640px; height:320px">
</div>
```

### JS Interface

Create the object of API class and call createVideoPlayer method. It will add the controls and key navigation based on the parameter values.

```javascript
var app = new lge();
app.createVideoPlayer($("#myVideoPlayer"), true, videoPath);
```
HTML5 Video Player Framework Implementation

1. The size and position of the video player can be set in the HTML page. If the width and height of the video player provided by user is less than 320 x 180 pixels, player will automatically reset its size to 320 x 180 pixels (minimum size).

2. If the size of the video player is specified as 1280 x 720 pixels, the player will display the following buttons.
   - Stop
   - Play/Pause
   - Forward
   - Rewind
   - Options

3. If the size of the video player is specified less than 1280 x 720 pixels, the player will display the following buttons.
   - Stop
   - Play/Pause
   - Forward
   - Rewind
   - Toggle full screen

4. If the video player toggle in full screen (1280 x 720 pixels), the player will display the following buttons.
   - Stop
   - Play/Pause
   - Forward
   - Rewind
   - Toggle full screen
   - Options

5. If any video player size specified is not the 16:9 aspect ratio, the video player will resize its size to 16:9 keeping the width constant.

Methods

Methods are listed in the following sections.
- Methods
Methods
Methods of HTML5 Video Player Framework API are as follows:

createVideoPlayer

Description
This API is used to create the video player.

Syntax
createVideoPlayer(container, controls, path);

Parameters
- **container**: Container div of application in which video player needs to be included.
- **controls**: Boolean. If true, it will add controls and key navigation to the framework. If false, no framework control will be added.
- **path**: The path of the video source.

Return Value
None

Example
```javascript
var app = new lge();
app.createVideoPlayer($("#myVideoPlayer"), false, videoPath);
```

Note
If developer wants to add own controls, it needs to pass controls parameter as 'false'. In this case, developer can use the below APIs to add their own controls.

Supported SDK / Emulator Version

<table>
<thead>
<tr>
<th>SDK Version</th>
<th>Emulator Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.2 or higher</td>
<td>LG Smart TV Emulator 2011 : Not Supported</td>
</tr>
<tr>
<td></td>
<td>LG Smart TV Emulator 2012 : SDK 2.2 or higher</td>
</tr>
<tr>
<td></td>
<td>LG Smart TV Emulator 2013 : SDK 3.0 or higher</td>
</tr>
</tbody>
</table>

play

Description
This API is used to play or pause the video based on the state of the video.
If the currently playing video is in pause state, the video is played. And if the currently playing video is in play state, the video is paused.

Syntax
play();

Parameters
None

Return Value
None
Example
```javascript
var app = new lge();
app.play();
```

Supported SDK / Emulator Version

<table>
<thead>
<tr>
<th>SDK Version</th>
<th>Emulator Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.2 or higher</td>
<td>LG Smart TV Emulator 2011 : Not Supported</td>
</tr>
<tr>
<td></td>
<td>LG Smart TV Emulator 2012 : SDK 2.2 or higher</td>
</tr>
<tr>
<td></td>
<td>LG Smart TV Emulator 2013 : SDK 3.0 or higher</td>
</tr>
</tbody>
</table>

**stop**

**Description**
This API is used to stop the currently playing video. All the controls are reset when the video is stopped or finished.

**Syntax**
```javascript
stop();
```

**Parameters**
None

**Return Value**
None

**Example**
```javascript
var app = new lge();
app.stop();
```

Supported SDK / Emulator Version

<table>
<thead>
<tr>
<th>SDK Version</th>
<th>Emulator Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.2 or higher</td>
<td>LG Smart TV Emulator 2011 : Not Supported</td>
</tr>
<tr>
<td></td>
<td>LG Smart TV Emulator 2012 : SDK 2.2 or higher</td>
</tr>
<tr>
<td></td>
<td>LG Smart TV Emulator 2013 : SDK 3.0 or higher</td>
</tr>
</tbody>
</table>

**rewind**

**Description**
This API is used to seek the video backward by 10 seconds. If the video is in pause state, state is changed to play.

**Syntax**
```javascript
rewind();
```

**Parameters**
None

**Return Value**
None

**Example**
```javascript
var app = new lge();
app.rewind();
```
### fastFoward

**Description**
This API is used to seek the video forward by 10 seconds. If the seek time is not more than duration of the video, the video is seeked. If the video is in pause state, state is changed to play.

**Syntax**
```javascript
fastFoward();
```

**Parameters**
None

**Return Value**
None

**Example**
```javascript
var app = new lge();
app.fastFoward();
```

### switchToFullScreenMode

**Description**
This API is used to switch from normal screen mode to full screen mode or vice versa. Full screen mode will set the size to 1280 x 720 pixels.

**Syntax**
```javascript
switchToFullScreenMode();
```

**Parameters**
None

**Return Value**
None

**Example**
```javascript
var app = new lge();
app.switchToFullScreenMode();
```

### Supported SDK / Emulator Version

<table>
<thead>
<tr>
<th>SDK Version</th>
<th>Emulator Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.2 or higher</td>
<td>LG Smart TV Emulator 2011 : Not Supported</td>
</tr>
<tr>
<td></td>
<td>LG Smart TV Emulator 2012 : SDK 2.2 or higher</td>
</tr>
<tr>
<td></td>
<td>LG Smart TV Emulator 2013 : SDK 3.0 or higher</td>
</tr>
</tbody>
</table>
optionMedia

Description
This API is used to open the quick menu on full screen. MENU(quick menu) can be launched by the user when the video is playing in full screen mode.

Syntax
optionMedia();

Parameters
None

Return Value
None

Example
var app = new lge();
app.optionMedia();

[Figure] Demo Screen of HTML5 Video Player showing media options

Supported SDK / Emulator Version

<table>
<thead>
<tr>
<th>SDK Version</th>
<th>Emulator Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.2 or higher</td>
<td>LG Smart TV Emulator 2011 : Not Supported</td>
</tr>
<tr>
<td></td>
<td>LG Smart TV Emulator 2012 : SDK 2.2 or higher</td>
</tr>
<tr>
<td></td>
<td>LG Smart TV Emulator 2013 : SDK 3.0 or higher</td>
</tr>
</tbody>
</table>
Web UI Component API

This document introduces LG Web UI Component and describes attributes, APIs, and examples of each component to LG Smart TV web applications developers. Also, supported page effects are described.

Caution

Web UI Component API requires LG Smart TV SDK 2.2 or higher.

What is LG Web UI Component?
LG Smart TV SDK provides LG Web UI Components for developing LG Smart TV web applications. The LG Web UI components are designed to comply with the look and feel of LG Smart TV’s UI/UX. LG provides various components including Button, CheckBox, Focus and more. Also, 4 types of page effects are provided.

The UI components can be made using standard HTML markup or LG JavaScript Interface described in this document.

[Figure] Demo Screen of LG Web API Components

How to Use LG Web UI Components
To use the LG Web UI Components in your web application, you must use the WYSIWYG Editor of LG IDE provided with LG Smart TV SDK. Using the WYSIWYG Editor, you can easily use LG UI components in your web project. For detailed information of how to use it, refer to the Developing > Using SDK section in this Library.

[Figure] LG WYSIWYG Editor
The following sections describe the API of the UI component plugin.

- **AnimatedImage**
- **Button**
- **CheckBox**
- **Focus**
- **GroupButton**
- **GroupCheckBox**
- **GroupRadio**
- **ImageList**
- **ImageTile**
- **Label**
- **MsgBox**
- **Picker**
- **Progress**
- **Radio**
- **Rating**
- **Scroller**
- **Slider**
- **Tab**
- **TextInput**
- **ToggleSwitch**
- **TvHelp**
- **PageEffects**

### AnimatedImage

The **AnimatedImage** component is image reaction which occurs when user clicks or moves the mouse onto it.

![AnimatedImage Component](image)

**Figure**  AnimatedImage Component

Markup Interface, JS Interface (Constructor Parameters), and JS Interface (Methods) are listed in the following sections.

- **Markup Interface**
- **JS Interface (Constructor Parameters)**
- **JS Interface (Methods)**

### Inheritance Hierarchy

Object >> Component >> Container >> Animatemg
Markup Interface

Example

```html
<img src='./image.png' width='200' height='200' lge-type='AnimateImg' lge-attr='animation-hover:Bounce' id='animateImage1'>
```

Attributes Summary

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>lge-attr</td>
<td>Attributes of the LGE component</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>animation-hover</td>
<td>Type of the animation when mouse is over</td>
</tr>
<tr>
<td>animation-click</td>
<td>Type of the animation when click</td>
</tr>
</tbody>
</table>

**animation-hover**

**Description**

Type of animation occurs when mouse is over.
There are four types of animation. (Bounce | Drop | Pop | Rotate)
The first letter should be upper case.

**Example**

```html
<img src='./image.jpg' lge-type='AnimateImg' lge-attr='animation-hover:Bounce' id='animateImage1'>
```

**animation-click**

**Description**

Type of animation occurs when mouse is clicked.
There are four types of animation. (Bounce | Drop | Pop | Rotate)
The first letter should be upper case.

**Example**

```html
<img src='./image.jpg' lge-type='AnimateImg' lge-attr='animation-click:Bounce' id='animateImage1'>
```
JS Interface (Constructor Parameters)

Example
var ai1 = LGE.UI.AnimatedImage({
  src: './image.png',
  animate: {event: 'hover', action: 'Bounce'},
  style: 'width: 200px; height: 200px;',
  selector: ('td2')
});

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>animate</td>
<td>Type of event and action</td>
</tr>
<tr>
<td>id</td>
<td>Element ID of the component in document</td>
</tr>
<tr>
<td>selector</td>
<td>The element that becomes parent of the component</td>
</tr>
<tr>
<td>style</td>
<td>CSS Style of the component</td>
</tr>
<tr>
<td>src</td>
<td>Image URL of the component</td>
</tr>
</tbody>
</table>

**animate**

Description
Type of event and action

There are two parameters.
- **event**: Event type of the component. There are two event types. (hover | click)
- **action**: Action type of the component. There are four actions. (Bounce | Rotate | Drop | Pop)

Example
var ai1 = LGE.UI.AnimatedImage({
  src: './image.png',
  animate: {event: 'hover', action: 'Bounce'},
  style: 'width: 200px; height: 200px;',
  selector: ('td2')
});

**id**

Description
Element ID of the component in document.

Example
var ai1 = LGE.UI.AnimatedImage({
  id: 'animateimg1',
  src: './image.png',
  animate: {event: 'hover', action: 'Bounce'},
  style: 'width: 200px; height: 200px;',
  selector: ('td2')
});

**selector**

Description
The element that becomes parent of the component
Default is body.
**Example**

```javascript
var ai1 = LGE.UI.AnimatedImage({
    src: './image.png',
    animate: {event: 'hover', action: 'Bounce'},
    style: 'width:200px;height:200px;',
    selector: 'td2'
});
```

**style**

**Description**

CSS style of the component

**Example**

```javascript
var ai1 = LGE.UI.AnimatedImage({
    src: './image.png',
    animate: {event: 'hover', action: 'Bounce'},
    style: 'width:200px;height:200px;',
    selector: 'td2'
});
```

**src**

**Description**

Path of the image

**Example**

```javascript
var ai1 = LGE.UI.AnimatedImage({
    src: './image.png',
    animate: {event: 'hover', action: 'Bounce'},
    style: 'width:200px;height:200px;',
    selector: 'td2'
});
```
### JS Interface (Methods)

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>getAnimation</td>
<td>Gets the animation Object of the AnimatedImage.</td>
</tr>
<tr>
<td>getSrc</td>
<td>Gets the image src of the AnimatedImage.</td>
</tr>
<tr>
<td>getStyle</td>
<td>Gets CSS Style of the AnimatedImage.</td>
</tr>
<tr>
<td>setAnimation</td>
<td>Sets the animation Object of the AnimatedImage.</td>
</tr>
<tr>
<td>setSrc</td>
<td>Sets the image url of the AnimatedImage.</td>
</tr>
<tr>
<td>setStyle</td>
<td>Sets CSS Style of the AnimatedImage.</td>
</tr>
</tbody>
</table>

#### getAnimation

**Description**

Gets the animation Object of the AnimatedImage.

**Parameters**

None

**Return Value**

object (Object) The animation Object of AnimatedImage. {event, action}

**Example**

```javascript
var object = animateImg.getAnimation();
```

**Supported SDK / Emulator Version**

<table>
<thead>
<tr>
<th>SDK Version</th>
<th>Emulator Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.2 or higher</td>
<td>LG Smart TV Emulator 2011 : Not Supported</td>
</tr>
<tr>
<td></td>
<td>LG Smart TV Emulator 2012 : SDK 2.2 or higher</td>
</tr>
<tr>
<td></td>
<td>LG Smart TV Emulator 2013 : SDK 3.0 or higher</td>
</tr>
</tbody>
</table>

#### getSrc

**Description**

Gets the image src of the AnimatedImage.

**Parameters**

None

**Return Value**

src (String) The image src of the AnimatedImage

**Example**

```javascript
var src = animateImg.getSrc();
```

**Supported SDK / Emulator Version**

<table>
<thead>
<tr>
<th>SDK Version</th>
<th>Emulator Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.2 or higher</td>
<td>LG Smart TV Emulator 2011 : Not Supported</td>
</tr>
<tr>
<td></td>
<td>LG Smart TV Emulator 2012 : SDK 2.2 or higher</td>
</tr>
<tr>
<td></td>
<td>LG Smart TV Emulator 2013 : SDK 3.0 or higher</td>
</tr>
</tbody>
</table>
getStyle

**Description**

Gets CSS Style of the AnimatedImage.

**Parameters**

None

**Return Value**

style (String) CSS style of the AnimatedImage

**Example**

```javascript
var style = animateImg.getStyle();
```

**Supported SDK / Emulator Version**

<table>
<thead>
<tr>
<th>SDK Version</th>
<th>Emulator Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.2 or higher</td>
<td>LG Smart TV Emulator 2011 : Not Supported</td>
</tr>
<tr>
<td>2.2 or higher</td>
<td>LG Smart TV Emulator 2012 : SDK 2.2 or higher</td>
</tr>
<tr>
<td>3.0 or higher</td>
<td>LG Smart TV Emulator 2013 : SDK 3.0 or higher</td>
</tr>
</tbody>
</table>

setAnimation

**Description**

Sets the animation Object of the AnimatedImage.

**Parameters**

`event` (String) The event type of the AnimatedImage (hover | click)

`action` (String) The action type of the AnimatedImage (Bounce | Rotate | Pop | Drop)

**Return Value**

None

**Example**

```javascript
animateImg.setAnimation("click", "Bounce");
```

**Supported SDK / Emulator Version**

<table>
<thead>
<tr>
<th>SDK Version</th>
<th>Emulator Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.2 or higher</td>
<td>LG Smart TV Emulator 2011 : Not Supported</td>
</tr>
<tr>
<td>2.2 or higher</td>
<td>LG Smart TV Emulator 2012 : SDK 2.2 or higher</td>
</tr>
<tr>
<td>3.0 or higher</td>
<td>LG Smart TV Emulator 2013 : SDK 3.0 or higher</td>
</tr>
</tbody>
</table>

setSrc

**Description**

Sets the image url of the AnimatedImage.

**Parameters**

`src` (String) The image url of the AnimatedImage
Return Value
None

Example
animateImage.setSrc("./image2.jpg");

Supported SDK / Emulator Version

<table>
<thead>
<tr>
<th>SDK Version</th>
<th>Emulator Version</th>
</tr>
</thead>
</table>
| 2.2 or higher | LG Smart TV Emulator 2011 : Not Supported  
|             | LG Smart TV Emulator 2012 : SDK 2.2 or higher  
|             | LG Smart TV Emulator 2013 : SDK 3.0 or higher  |

**setStyle**

Description
Sets CSS Style of the AnimatedImage.

Parameters
- **style** (String) CSS style of the AnimatedImage

Return Value
None

Example
animateImage.setStyle("width:100px;height:100px;");

Supported SDK / Emulator Version

<table>
<thead>
<tr>
<th>SDK Version</th>
<th>Emulator Version</th>
</tr>
</thead>
</table>
| 2.2 or higher | LG Smart TV Emulator 2011 : Not Supported  
|             | LG Smart TV Emulator 2012 : SDK 2.2 or higher  
|             | LG Smart TV Emulator 2013 : SDK 3.0 or higher  |

**Button**

The Button component can have text and icon.

![Button Component]

Inheritance Hierarchy
Object >> Component >> Container >> Button

Markup Interface, JS Interface (Constructor Parameters), and JS Interface (Methods) are listed in the following sections.
- Markup Interface
- JS Interface (Constructor Parameters)
- JS Interface (Methods)
Markup Interface

Example

```html
<input lge-type='Button' type='button' value='text' />

<input lge-type='Button' type='submit' value='text2' />

<input lge-type='Button' type='reset' value='text3' />

<input lge-type='Button' type='image' value='text4' />

<a href='http://www.lge.com/' lge-type="Button">Button</a>
```

Attributes Summary

<table>
<thead>
<tr>
<th>lge-attr</th>
<th>Attributes of the LGE component</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>icon-position</td>
<td>Specifies the location of the icon in the button.</td>
</tr>
<tr>
<td>icon-type</td>
<td>Adds an icon to a button.</td>
</tr>
<tr>
<td>text-inline</td>
<td>More compact component that is only as wide as the text inside.</td>
</tr>
<tr>
<td>text-omit</td>
<td>Specifies whether text should be shown in the button or not.</td>
</tr>
</tbody>
</table>

icon-position

Description

Specifies the location of the icon. (Left | Right | Top | Bottom)

Example

```html
<input lge-type='button' lge-attr='icon-position:Left,icon-type:Plus' value='text' />
```

icon-type

Description

Adds an icon to the button by adding a 'icon-type' attribute on the button specifying the icon to display.

(Home | Delete | Plus | Arrowu | Arrowd | Arrowl | Arrowr | Star | Refresh | Back | Search | Setting)

Example

```html
<input lge-type='button' lge-attr='icon-type:Plus' value='text' />
```

text-inline

Description

More compact component that is only as wide as the text inside.

Default value is false. (true | false)

Example

```html
<input lge-type='button' lge-attr='text-inline:true' value='text' />
```
text-omit

Description
Specifies whether text should be shown in the button or not
Default value is false. (true | false)

Example
<input lge-type='button' lge-attr='text-omit:true' />
JS Interface (Constructor Parameters)

Example

```javascript
var button = LGE.UI.Button({
  selector:("td2"),
  type:'input',
  value:'text',
});

var abutton = LGE.UI.Button({
  selector:("td2"),
  type:'a',
  href:'http://www.lge.com/',
  value:'Button',
});
```

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>href</td>
<td>Hyperlink address (Only type=&quot;a&quot;)</td>
</tr>
<tr>
<td>id</td>
<td>Element ID of the component in document</td>
</tr>
<tr>
<td>iconPosition</td>
<td>Position of the icon in the button</td>
</tr>
<tr>
<td>iconType</td>
<td>Type of the icon to display in the button</td>
</tr>
<tr>
<td>selector</td>
<td>The element that becomes parent of the component</td>
</tr>
<tr>
<td>style</td>
<td>CSS Style of the component</td>
</tr>
<tr>
<td>textOmit</td>
<td>Text in the button is shown or not</td>
</tr>
<tr>
<td>textInline</td>
<td>More compact component that is only as wide as the text inside</td>
</tr>
<tr>
<td>type</td>
<td>The type of the Button.</td>
</tr>
<tr>
<td>value</td>
<td>Text in the button (Only type=&quot;button&quot;)</td>
</tr>
</tbody>
</table>

**href**

Description

Hyperlink Address of Button component. It is only for "A" tag.

Example

```javascript
var abutton = LGE.UI.Button({
  selector:("td2"),
  type:'a',
  href:'http://www.lge.com/',
  value:'Button',
});
```

**id**

Description

Element ID of the component in document.

Example

```javascript
var button = LGE.UI.Button({
  type:'input',
  value:'text',
  selector:("td2"),
});
```
iconPosition

Description
Specifies the location of the icon. (Left | Right | Top | Bottom)

Example
```javascript
var button = LGE.UI.Button({
  type:'input',
  value:'text',
  selector:('td2'),
  iconType:'Home',
  iconPosition:'Left'
});
```

iconType

Description
An icon can be added to a button by adding a data-icon attribute on the button specifying the icon to display. (Home | Delete | Plus | Arrowu | Arrowd | Arrowl | Star | Refresh | Back | Search | Setting)

Example
```javascript
var button = LGE.UI.Button({
  type:'input',
  value:'text',
  selector:('td2'),
  iconType:'Home',
  iconPosition:'Left'
});
```

selector

Description
The element that becomes parent of the component. Default is body.

Example
```javascript
var button = LGE.UI.Button({
  type:'input',
  value:'text',
  selector:('td2'),
  textInline:'true'
});
```

style

Description
CSS style of the component

Example
```javascript
var button = LGE.UI.Button({
  type:'input',
  value:'text',
```
textContent

Description
Specifies whether text should be shown in the button. Default value is false. (true | false)

Example
```javascript
var button = LGE.UI.Button({
  type:'input',
  selector:('td2'),
  iconType:'Home',
  iconPosition:'Left',
  textOmit:'true'
});
```

textInline

Description
More compact component that is only as wide as the text inside. Default value is false. (true | false)

Example
```javascript
var button = LGE.UI.Button({
  type:'input',
  selector:('td2'),
  iconType:'Home',
  iconPosition:'Left',
  textInline:'true'
});
```

type

Description
The tag type of the Button component. Default value is input. (a | input)

Example
```javascript
var button = LGE.UI.Button({
  type:'input',
  selector:('td2'),
  textInline:'true'
});
```

value

Description
Text of the component
Example

```javascript
var button = LGE.UI.Button(
    selector:("td2"),
    type:'input',
    value:'text',
});
```
## JS Interface (Methods)

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>getHref</td>
<td>Gets the URL of the Button (only A tag)</td>
</tr>
<tr>
<td>getIconPosition</td>
<td>Gets the icon position of the Button</td>
</tr>
<tr>
<td>getIconType</td>
<td>Gets the icon type of the Button</td>
</tr>
<tr>
<td>getText</td>
<td>Gets the text of the Button</td>
</tr>
<tr>
<td>getTextInline</td>
<td>Gets the text-inline of the Button</td>
</tr>
<tr>
<td>getTextOmit</td>
<td>Gets the text-omit of the Button</td>
</tr>
<tr>
<td>setHref</td>
<td>Sets the url link of the Button (only A tag)</td>
</tr>
<tr>
<td>setIconPosition</td>
<td>Sets the icon position of the Button</td>
</tr>
<tr>
<td>setIconType</td>
<td>Sets the icon type of the Button</td>
</tr>
<tr>
<td>setText</td>
<td>Sets the text of the Button</td>
</tr>
<tr>
<td>setTextInline</td>
<td>Sets the text-inline of the Button</td>
</tr>
<tr>
<td>setTextOmit</td>
<td>Sets the text-omit of the Button</td>
</tr>
</tbody>
</table>

### getHref

**Description**

Gets the URL of the Button. (only `<A>` tag)

**Parameters**

None

**Return Value**

url (String) The url link of Button

**Example**

```javascript
var url = button.getHref();
```

**Supported SDK / Emulator Version**

<table>
<thead>
<tr>
<th>SDK Version</th>
<th>Emulator Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.2 or higher</td>
<td>LG Smart TV Emulator 2011 : Not Supported</td>
</tr>
<tr>
<td></td>
<td>LG Smart TV Emulator 2012 : SDK 2.2 or higher</td>
</tr>
<tr>
<td></td>
<td>LG Smart TV Emulator 2013 : SDK 3.0 or higher</td>
</tr>
</tbody>
</table>

### getIconPosition

**Description**

Gets the icon position of the Button. (Left | Right | Top | Bottom)

**Parameters**

None

**Return Value**

iconPosition (String) The position type of Button
Example

```javascript
var iconPosition = button.getIconPosition();
```

### Supported SDK / Emulator Version

<table>
<thead>
<tr>
<th>SDK Version</th>
<th>Emulator Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.2 or higher</td>
<td>LG Smart TV Emulator 2011: Not Supported</td>
</tr>
<tr>
<td></td>
<td>LG Smart TV Emulator 2012: SDK 2.2 or higher</td>
</tr>
<tr>
<td></td>
<td>LG Smart TV Emulator 2013: SDK 3.0 or higher</td>
</tr>
</tbody>
</table>

### getIconType

**Description**

Gets the icon type of the Button. Here is icon type of the Button. 
(Home | Delete | Plus | Arrowu | Arrowd | Arrowr | Arrowl | Star | Refresh | Back | Search | Setting)

**Parameters**

None

**Return Value**

- `iconType` (String) The icon type of Button

**Example**

```javascript
var iconType = button.getIconType();
```

### Supported SDK / Emulator Version

<table>
<thead>
<tr>
<th>SDK Version</th>
<th>Emulator Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.2 or higher</td>
<td>LG Smart TV Emulator 2011: Not Supported</td>
</tr>
<tr>
<td></td>
<td>LG Smart TV Emulator 2012: SDK 2.2 or higher</td>
</tr>
<tr>
<td></td>
<td>LG Smart TV Emulator 2013: SDK 3.0 or higher</td>
</tr>
</tbody>
</table>

### getText

**Description**

Gets the text of the Button component.

**Parameters**

None

**Return Value**

- `text` (String) The text of the Button component

**Example**

```javascript
var text = button.getText();
```

### Supported SDK / Emulator Version

<table>
<thead>
<tr>
<th>SDK Version</th>
<th>Emulator Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.2 or higher</td>
<td>LG Smart TV Emulator 2011: Not Supported</td>
</tr>
<tr>
<td></td>
<td>LG Smart TV Emulator 2012: SDK 2.2 or higher</td>
</tr>
<tr>
<td></td>
<td>LG Smart TV Emulator 2013: SDK 3.0 or higher</td>
</tr>
</tbody>
</table>
getTextInline

Description
Gets the text-inline of the Button.

Parameters
None

Return Value
book (Boolean) The text-inline of the Button

Example
var textinline = button.getTextInline();

Supported SDK / Emulator Version

<table>
<thead>
<tr>
<th>SDK Version</th>
<th>Emulator Version</th>
</tr>
</thead>
</table>
| 2.2 or higher | LG Smart TV Emulator 2011: Not Supported  
LG Smart TV Emulator 2012: SDK 2.2 or higher  
LG Smart TV Emulator 2013: SDK 3.0 or higher |

getTextOmit

Description
 Gets the text-omit of the Button.

Parameters
None

Return Value
bool (Boolean) The text-omit of the Button

Example
var textomit = button.getTextOmit();

Supported SDK / Emulator Version

<table>
<thead>
<tr>
<th>SDK Version</th>
<th>Emulator Version</th>
</tr>
</thead>
</table>
| 2.2 or higher | LG Smart TV Emulator 2011: Not Supported  
LG Smart TV Emulator 2012: SDK 2.2 or higher  
LG Smart TV Emulator 2013: SDK 3.0 or higher |

setHref

Description
Sets the URL of the Button. (only A tag)

Parameters
url (String) The url link of Button

Return Value
None

Example
button.setHref("http://www.lge.com");
setIconPosition

**Description**
Sets the icon position of the Button. (Left | Right | Top | Bottom)

**Parameters**
- **iconPosition** (String) The icon position of Button

**Return Value**
None

**Example**

```java
button.setIconPosition("Top");
```

**Supported SDK / Emulator Version**

<table>
<thead>
<tr>
<th>SDK Version</th>
<th>Emulator Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.2 or higher</td>
<td>LG Smart TV Emulator 2011 : Not Supported</td>
</tr>
<tr>
<td></td>
<td>LG Smart TV Emulator 2012 : SDK 2.2 or higher</td>
</tr>
<tr>
<td></td>
<td>LG Smart TV Emulator 2013 : SDK 3.0 or higher</td>
</tr>
</tbody>
</table>

setIconType

**Description**
Sets the icon type of the Button.
(Home | Delete | Plus | Arrowu | Arrowd | Arrowr | Arrowl | Star | Refresh | Back | Search | Setting)

**Parameters**
None

**Return Value**
- **iconType** (String) The icon type of Button

**Example**

```java
button.setIconType("Home");
```

**Supported SDK / Emulator Version**

<table>
<thead>
<tr>
<th>SDK Version</th>
<th>Emulator Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.2 or higher</td>
<td>LG Smart TV Emulator 2011 : Not Supported</td>
</tr>
<tr>
<td></td>
<td>LG Smart TV Emulator 2012 : SDK 2.2 or higher</td>
</tr>
<tr>
<td></td>
<td>LG Smart TV Emulator 2013 : SDK 3.0 or higher</td>
</tr>
</tbody>
</table>

setText
**Description**
Sets the text of the Button component.

**Parameters**
- `text` (String) The text of the Button component

**Return Value**
None

**Example**
```java
button.setText("Hello world");
```

**Supported SDK / Emulator Version**

<table>
<thead>
<tr>
<th>SDK Version</th>
<th>Emulator Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.2 or higher</td>
<td>LG Smart TV Emulator 2011: Not Supported</td>
</tr>
<tr>
<td></td>
<td>LG Smart TV Emulator 2012: SDK 2.2 or higher</td>
</tr>
<tr>
<td></td>
<td>LG Smart TV Emulator 2013: SDK 3.0 or higher</td>
</tr>
</tbody>
</table>

**setTextInline**

**Description**
Sets the text-inline of the Button.

**Parameters**
- `bool` (Boolean) The text-inline of the Button

**Return Value**
None

**Example**
```java
button.setTextInline(true);
```

**Supported SDK / Emulator Version**

<table>
<thead>
<tr>
<th>SDK Version</th>
<th>Emulator Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.2 or higher</td>
<td>LG Smart TV Emulator 2011: Not Supported</td>
</tr>
<tr>
<td></td>
<td>LG Smart TV Emulator 2012: SDK 2.2 or higher</td>
</tr>
<tr>
<td></td>
<td>LG Smart TV Emulator 2013: SDK 3.0 or higher</td>
</tr>
</tbody>
</table>

**setTextOmit**

**Description**
Set the text-omit of the Button.

**Parameters**
- `bool` (Boolean) The text-omit of the Button

**Return Value**
None

**Example**
```java
button.setTextOmit(true);
```
Supported SDK / Emulator Version

<table>
<thead>
<tr>
<th>SDK Version</th>
<th>Emulator Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.2 or higher</td>
<td>LG Smart TV Emulator 2011 : Not Supported</td>
</tr>
<tr>
<td></td>
<td>LG Smart TV Emulator 2012 : SDK 2.2 or higher</td>
</tr>
<tr>
<td></td>
<td>LG Smart TV Emulator 2013 : SDK 3.0 or higher</td>
</tr>
</tbody>
</table>

**CheckBox**

The CheckBox component is an element that permits the user to make multiple selections.

[Figure] CheckBox Component

**Inheritance Hierarchy**

Object >> Component >> Container >> ListItemComponent >> CheckBoxRadioList >> CheckBox

Markup Interface, JS Interface (Constructor Parameters), and JS Interface (Methods) are listed in the following sections.

- Markup Interface
- JS Interface (Constructor Parameters)
- JS Interface (Methods)
**Markup Interface**

**Example**

```html
<input type="checkbox" lge-type="CheckBox" checked value="Checkbox" />
```

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>lge-attr</td>
<td>Attributes of the LGE component</td>
</tr>
</tbody>
</table>

**text-inline**

**Description**

More compact component that is only as wide as the text inside.
Default value is false. (true | false)

**Example**

```html
<input type="checkbox" lge-type="CheckBox" checked lge-attr='text-inline:true' value='text' />
```
JS Interface (Constructor Parameters)

**Example**

```javascript
var Checkbox1 = LGE.UI.CheckBox({
    selector:("td2"),
    checked:'true',
    id:'checkbox1',
    text:'Basic checkbox1',
});
```

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>checked</td>
<td>Specifies whether the component should be checked or not.</td>
</tr>
<tr>
<td>id</td>
<td>Element ID of the component in document.</td>
</tr>
<tr>
<td>selector</td>
<td>The element ID that becomes parent of the component.</td>
</tr>
<tr>
<td>style</td>
<td>CSS Style of the component.</td>
</tr>
<tr>
<td>text</td>
<td>Text in the component.</td>
</tr>
<tr>
<td>textInline</td>
<td>More compact component that is only as wide as the text inside.</td>
</tr>
</tbody>
</table>

**checked**

**Description**

Specifies whether the component should be checked or not. Default value is false. (true | false)

**Example**

```javascript
var Checkbox1 = LGE.UI.CheckBox({
    selector:("id1"),
    checked:'true',
    id:'checkbox1',
    text:'Basic checkbox1',
});
```

**id**

**Description**

Element ID of the component in document.

**Example**

```javascript
var Checkbox1 = LGE.UI.CheckBox({
    selector:("id1"),
    checked:'true',
    id:'checkbox1',
    text:'Basic checkbox1',
});
```

**selector**

**Description**

The element that becomes parent of the component. Default is body.
Example

```javascript
var Checkbox1 = LGE.UI.CheckBox({
    selector:("id1"),
    checked:'true',
    id:'checkbox1',
    text:'Basic checkbox1',
});
```

**style**

**Description**
CSS style of the component

**Example**

```javascript
var Checkbox1 = LGE.UI.CheckBox({
    selector:("id1"),
    checked:'true',
    id:'checkbox1',
    text:'Basic checkbox1',
    style:'width:200px',
});
```

**text**

**Description**
Text of the component

**Example**

```javascript
var Checkbox1 = LGE.UI.CheckBox({
    selector:("id1"),
    checked:'true',
    id:'checkbox1',
    text:'Basic checkbox1',
});
```

**textInline**

**Description**
More compact component that is only as wide as the text inside. Default value is false. (true | false)

**Example**

```javascript
var Checkbox1 = LGE.UI.CheckBox({
    selector:("id1"),
    id:'checkbox1',
    text:'Basic checkbox1',
    textInline:'true'
});
```
JS Interface (Methods)

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>getCheck</td>
<td>Gets check state of the CheckBox.</td>
</tr>
<tr>
<td>getText</td>
<td>Gets the text of the CheckBox.</td>
</tr>
<tr>
<td>getTextInline</td>
<td>Gets the text-inline of the CheckBox.</td>
</tr>
<tr>
<td>setCheck</td>
<td>Sets check state of the CheckBox.</td>
</tr>
<tr>
<td>setText</td>
<td>Sets the text of the CheckBox.</td>
</tr>
<tr>
<td>setTextInline</td>
<td>Sets the text-inline of the CheckBox.</td>
</tr>
</tbody>
</table>

**getCheck**

**Description**
Gets check state of the CheckBox.

**Parameters**
None

**Return Value**
`bool` (Boolean) The check state of the CheckBox

**Example**

```javascript
var checked = checkbox.getCheck();
```

**Supported SDK / Emulator Version**

<table>
<thead>
<tr>
<th>SDK Version</th>
<th>Emulator Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.2 or higher</td>
<td>LG Smart TV Emulator 2011 : Not Supported</td>
</tr>
<tr>
<td></td>
<td>LG Smart TV Emulator 2012 : SDK 2.2 or higher</td>
</tr>
<tr>
<td></td>
<td>LG Smart TV Emulator 2013 : SDK 3.0 or higher</td>
</tr>
</tbody>
</table>

**getText**

**Description**
Gets the text of the CheckBox.

**Parameters**
None

**Return Value**
`text` (String) The text of the Button component

**Example**

```javascript
var text = checkbox.getText();
```

**Supported SDK / Emulator Version**

<table>
<thead>
<tr>
<th>SDK Version</th>
<th>Emulator Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.2 or higher</td>
<td>LG Smart TV Emulator 2011 : Not Supported</td>
</tr>
<tr>
<td></td>
<td>LG Smart TV Emulator 2012 : SDK 2.2 or higher</td>
</tr>
<tr>
<td></td>
<td>LG Smart TV Emulator 2013 : SDK 3.0 or higher</td>
</tr>
</tbody>
</table>
**getTextInline**

**Description**
Gets the text-inline of the CheckBox.

**Parameters**
None

**Return Value**
book (Boolean) The text-inline of the CheckBox

**Example**
```javascript
var textinline = checkbox.getTextInline();
```

**Supported SDK / Emulator Version**

<table>
<thead>
<tr>
<th>SDK Version</th>
<th>Emulator Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.2 or higher</td>
<td>LG Smart TV Emulator 2011 : Not Supported</td>
</tr>
<tr>
<td></td>
<td>LG Smart TV Emulator 2012 : SDK 2.2 or higher</td>
</tr>
<tr>
<td></td>
<td>LG Smart TV Emulator 2013 : SDK 3.0 or higher</td>
</tr>
</tbody>
</table>

**setCheck**

**Description**
Sets check state of the CheckBox.

**Parameters**
bool (Boolean) The text-omit of the CheckBox

**Return Value**
None

**Example**
```javascript
checkbox.setCheck(true);
```

**Supported SDK / Emulator Version**

<table>
<thead>
<tr>
<th>SDK Version</th>
<th>Emulator Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.2 or higher</td>
<td>LG Smart TV Emulator 2011 : Not Supported</td>
</tr>
<tr>
<td></td>
<td>LG Smart TV Emulator 2012 : SDK 2.2 or higher</td>
</tr>
<tr>
<td></td>
<td>LG Smart TV Emulator 2013 : SDK 3.0 or higher</td>
</tr>
</tbody>
</table>

**setText**

**Description**
Sets the text of the CheckBox.

**Parameters**
text (String) The text of CheckBox

**Return Value**
Example

```java
checkbox.setText("checkbox text");
```

### Supported SDK / Emulator Version

<table>
<thead>
<tr>
<th>SDK Version</th>
<th>Emulator Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.2 or higher</td>
<td>LG Smart TV Emulator 2011 : Not Supported</td>
</tr>
<tr>
<td></td>
<td>LG Smart TV Emulator 2012 : SDK 2.2 or higher</td>
</tr>
<tr>
<td></td>
<td>LG Smart TV Emulator 2013 : SDK 3.0 or higher</td>
</tr>
</tbody>
</table>

#### setTextInline

**Description**
Sets the text-inline of the CheckBox.

**Parameters**

```java
bool (Boolean) The text-inline of the CheckBox
```

**Return Value**

None

**Example**

```java
checkbox.setTextInline(true);
```

### Supported SDK / Emulator Version

<table>
<thead>
<tr>
<th>SDK Version</th>
<th>Emulator Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.2 or higher</td>
<td>LG Smart TV Emulator 2011 : Not Supported</td>
</tr>
<tr>
<td></td>
<td>LG Smart TV Emulator 2012 : SDK 2.2 or higher</td>
</tr>
<tr>
<td></td>
<td>LG Smart TV Emulator 2013 : SDK 3.0 or higher</td>
</tr>
</tbody>
</table>

#### Focus

Focus component is a container that can have any element developer want. This component has focus so that user can focus this component by using direction key.

![Focus Component](image)

[Figure] Focus Component

### Inheritance Hierarchy
Object >> Component >> Container >> Focus

Markup Interface, JS Interface (Constructor Parameters), and JS Interface (Methods) are listed in the following sections.

- Markup Interface
- JS Interface (Constructor Parameters)
- JS Interface (Methods)
Markup Interface

Example

```html
<div lge-type="Focus">
  ...
</div>
```

Attributes Summary

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>lge-attr</td>
<td>Attributes of the LGE component</td>
</tr>
<tr>
<td>option</td>
<td>The option for not drawing focus border</td>
</tr>
</tbody>
</table>

**option**

**Description**
Focus component has a focus border. For not drawing focus border, use option:"NoFocus".

Example

```html
<div lge-type="Focus" lge-attr="option:'NoFocus';">
  ...
</div>
```
JS Interface (Constructor Parameters)

Example

```javascript
var focus1 = LGE.UI.Focus({
  selector:("id1"),
  id:"focus1",
});
```

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>id</td>
<td>Element ID of the component in document</td>
</tr>
<tr>
<td>option</td>
<td>The option for not drawing focus border</td>
</tr>
<tr>
<td>selector</td>
<td>The element ID that becomes parent of the component</td>
</tr>
</tbody>
</table>

**id**

**Description**
Element ID of the component in document.

**Example**

```javascript
var focus1 = LGE.UI.Focus({
  selector:("id1"),
  id:"focus1",
});
```

**option**

**Description**
Focus component has a focus border. For not drawing focus border, use option:"NoFocus".

**Example**

```javascript
var focus1= LGE.UI.Focus({
  selector:("id1"),
  id:"focus1",
  option:"NoFocus",
});
```

**selector**

**Description**
The element that becomes parent of the component
Default is body.

**Example**

```javascript
var focus1 = LGE.UI.Focus({
  selector:("id1"),
  id:"focus1",
});
```
**JS Interface (Methods)**

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>setOption</td>
<td>Sets the option of Focus.</td>
</tr>
</tbody>
</table>

**setOption**

**Description**
Sets the option of Focus. Focus component has a border. For not drawing border, use option = "noFocus".

**Parameters**
None

**Return Value**
- `option` (String) The option of Focus (Focus | NoFocus)

**Example**
```
focus.setOption("NoFocus");
```

**Supported SDK / Emulator Version**

<table>
<thead>
<tr>
<th>SDK Version</th>
<th>Emulator Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.2 or higher</td>
<td>LG Smart TV Emulator 2011 : Not Supported</td>
</tr>
<tr>
<td></td>
<td>LG Smart TV Emulator 2012 : SDK 2.2 or higher</td>
</tr>
<tr>
<td></td>
<td>LG Smart TV Emulator 2013 : SDK 3.0 or higher</td>
</tr>
</tbody>
</table>

**GroupButton**

The GroupButton component makes a group of buttons.

[Figure] GroupButton Component

**Inheritance Hierarchy**
Object >> Component >> Container >> GroupButton

Markup Interface, JS Interface (Constructor Parameters), and JS Interface (Methods) are listed in the following sections.
- **Markup Interface**
- **JS Interface (Constructor Parameters)**
- **JS Interface (Methods)**
### Markup Interface

**Example**

```html
<div lge-type='GroupButton' lge-attr='box-orient:Vertical'>
    <input type="button" value="Button1" lge-attr="icon-type:plus"/>
    <input type="button" value="Button2" lge-attr="icon-type:home"/>
    <input type="button" value="Button3" lge-attr="icon-type:arrowd;"/>
</div>
```

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>lge-attr</td>
<td>Attributes of the LGE component</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>box-orient</td>
<td>Aligns the parts of component vertically or horizontally.</td>
</tr>
</tbody>
</table>

### box-orient

**Description**

Aligns the parts of GroupButton vertically or horizontally.  
Default value is Vertical. (Vertical | Horizontal)

**Example**

```html
<div lge-type='GroupButton' lge-attr='box-orient:Horizontal'>
    <input type="button" value="Button1" lge-attr="icon-type:Plus"/>
    <input type="button" value="Button2" lge-attr="icon-type:Arrowd;"/>
</div>
```
JS Interface (Constructor Parameters)

Example

```javascript
var gbtn1 = LGE.UI.GroupButton({
  selector: "id1",
  items: [
    {type: 'input', value: 'btn1', iconType: 'Home'},
    {type: 'input', value: 'btn2', iconType: 'Plus'},
  ]
});
```

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>boxOrient</td>
<td>Aligns the parts of GroupButton vertically or horizontally.</td>
</tr>
<tr>
<td>id</td>
<td>Element ID of the component in document</td>
</tr>
<tr>
<td>items</td>
<td>A list of buttons</td>
</tr>
<tr>
<td>selector</td>
<td>The element ID that becomes parent of the component</td>
</tr>
</tbody>
</table>

**boxOrient**

Description
Aligns the parts of GroupButton vertically or horizontally. Default value is Vertical. (Vertical | Horizontal)

Example

```javascript
var gbtn1 = LGE.UI.GroupButton({
  selector: "id1",
  boxOrient: 'Horizontal',
  items: [
    {type: 'input', value: 'btn1', iconType: 'Home'},
    {type: 'input', value: 'btn2', iconType: 'Plus'},
  ]
});
```

**id**

Description
Element ID of the component in document.

Example

```javascript
var gbtn1 = LGE.UI.GroupButton({
  selector: "id1",
  boxOrient: 'Horizontal',
  id: 'groupbutton',
  items: [
    {type: 'input', value: 'btn1', iconType: 'Home'},
    {type: 'input', value: 'btn2', iconType: 'Plus'},
  ]
});
```

**items**
Description
A list of Buttons. It is array and has buttons.

Example
```javascript
var gbtn1 = LGE.UI.GroupButton({
    selector: "id1",
    boxOrient: 'Horizontal',
    items: [
        {type: 'input', value: 'btn1', iconType: 'Home'},
        {type: 'input', value: 'btn2', iconType: 'Plus'},
    ]
});
```

selector

Description
The element that becomes parent of the component
Default is body.

Example
```javascript
var gbtn1 = LGE.UI.GroupButton({
    selector: "id1",
    boxOrient: 'Horizontal',
    items: [
        {type: 'input', value: 'btn1', iconType: 'Home'},
        {type: 'input', value: 'btn2', iconType: 'Plus'},
    ]
});
```
## JS Interface (Methods)

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>addItem</td>
<td>Adds item to the GroupButton.</td>
</tr>
<tr>
<td>getBoxOrient</td>
<td>Gets the order direction of the GroupButton.</td>
</tr>
<tr>
<td>getItem</td>
<td>Gets the item object of the GroupButton.</td>
</tr>
<tr>
<td>removeItem</td>
<td>Removes item of the GroupButton.</td>
</tr>
<tr>
<td>setBoxOrient</td>
<td>Sets the order direction of the GroupButton.</td>
</tr>
<tr>
<td>setItem</td>
<td>Sets the item object of the GroupButton.</td>
</tr>
</tbody>
</table>

### addItem

**Description**

Adds item to the GroupButton.

**Parameters**

- **option** *(Object)* The options of CheckBox. For more information of this parameter, see JavaScript Constructor Paramter in Button component.

**Return Value**

None

**Example**

```javascript
groupBy.addItem({
  type: 'input',
  value: 'text',
  iconType: 'Home',
  iconPosition: 'Left',
});
```

**Supported SDK / Emulator Version**

<table>
<thead>
<tr>
<th>SDK Version</th>
<th>Emulator Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.2 or higher</td>
<td>LG Smart TV Emulator 2011 : Not Supported</td>
</tr>
<tr>
<td></td>
<td>LG Smart TV Emulator 2012 : SDK 2.2 or higher</td>
</tr>
<tr>
<td></td>
<td>LG Smart TV Emulator 2013 : SDK 3.0 or higher</td>
</tr>
</tbody>
</table>

### getBoxOrient

**Description**

Gets the order direction of the GroupButton.

**Parameters**

None

**Return Value**

- **BoxOrient** *(String)* The order direction of the GroupButton

**Example**

```javascript
alert( groupButton.getBoxOrient() ); // “Vertical” or “Horizontal”
```
### Supported SDK / Emulator Version

<table>
<thead>
<tr>
<th>SDK Version</th>
<th>Emulator Version</th>
</tr>
</thead>
</table>
| 2.2 or higher | LG Smart TV Emulator 2011: Not Supported  
                LG Smart TV Emulator 2012: SDK 2.2 or higher  
                LG Smart TV Emulator 2013: SDK 3.0 or higher |

### getItem

**Description**

Gets the item object of the GroupButton.

**Parameters**

**Return Value**

*Item* *(Object)* The item object of GroupButton. The item object has only getXXX method. For more information of this object, see JavaScript API in Button component.

**Example**

```javascript
var iconPosition = groupButton.getItem(0).getIconPosition();
```

### Supported SDK / Emulator Version

<table>
<thead>
<tr>
<th>SDK Version</th>
<th>Emulator Version</th>
</tr>
</thead>
</table>
| 2.2 or higher | LG Smart TV Emulator 2011: Not Supported  
                LG Smart TV Emulator 2012: SDK 2.2 or higher  
                LG Smart TV Emulator 2013: SDK 3.0 or higher |

### removeItem

**Description**

Removes item of the GroupButton.

**Parameters**

*Index* *(Number)* The index of the Button

**Return Value**

None

**Example**

```javascript
groupButton.removeItem(1);
```

### Supported SDK / Emulator Version

<table>
<thead>
<tr>
<th>SDK Version</th>
<th>Emulator Version</th>
</tr>
</thead>
</table>
| 2.2 or higher | LG Smart TV Emulator 2011: Not Supported  
                LG Smart TV Emulator 2012: SDK 2.2 or higher  
                LG Smart TV Emulator 2013: SDK 3.0 or higher |

### setBoxOrient

**Description**

Sets the order direction of the GroupButton.
Parameters

**BoxOrient**

(String) The order direction of the GroupButton (Vertical | Horizontal)

Return Value

None

Example

```javascript
groupBy.setBoxOrient("Vertical");
```

<table>
<thead>
<tr>
<th>Supported SDK / Emulator Version</th>
<th>SDK Version</th>
<th>Emulator Version</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>2.2 or higher</td>
<td>LG Smart TV Emulator 2011 : Not Supported</td>
</tr>
<tr>
<td></td>
<td></td>
<td>LG Smart TV Emulator 2012 : SDK 2.2 or higher</td>
</tr>
<tr>
<td></td>
<td></td>
<td>LG Smart TV Emulator 2013 : SDK 3.0 or higher</td>
</tr>
</tbody>
</table>

**setItem**

Description

Sets the item object of the GroupButton.

Parameters

**Index**

(Number) The index of the Button

Return Value

**Item**

(Object) The item object of GroupButton. The item object has only setXXX method. For more information of this object, see JavaScript API in Button component.

Example

```javascript
groupBy.setItem(0).setIconPosition("Horizontal");
```

<table>
<thead>
<tr>
<th>Supported SDK / Emulator Version</th>
<th>SDK Version</th>
<th>Emulator Version</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>2.2 or higher</td>
<td>LG Smart TV Emulator 2011 : Not Supported</td>
</tr>
<tr>
<td></td>
<td></td>
<td>LG Smart TV Emulator 2012 : SDK 2.2 or higher</td>
</tr>
<tr>
<td></td>
<td></td>
<td>LG Smart TV Emulator 2013 : SDK 3.0 or higher</td>
</tr>
</tbody>
</table>

**GroupCheckBox**

The GroupCheckBox component makes a group of Checkboxes.

[Figure] GroupCheckBox Component

Inheritance Hierarchy

Object >> Component >> Container >> GroupCheckBoxRadioList >> GroupCheckBox
Markup Interface, JS Interface (Constructor Parameters), and JS Interface (Methods) are listed in the following sections.

- Markup Interface
- JS Interface (Constructor Parameters)
- JS Interface (Methods)
**Markup Interface**

**Example**

```html
<div lge-type="GroupCheckBox">
  <fieldset>
    <legend>Default Group CheckBox</legend>
    <label><input type="checkbox" lge-type="CheckBox" />I agree</label>
    <label><input type="checkbox" lge-type="CheckBox" />I agree2</label>
    <label><input type="checkbox" lge-type="CheckBox" />I agree3</label>
  </fieldset>
</div>
```

**Attributes Summary**

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>lge-attrib</td>
<td>Attributes of the LGE component</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>box-orient</td>
<td>Aligns the parts of component vertically or horizontally.</td>
</tr>
</tbody>
</table>

**box-orient**

**Description**

Aligns the parts of GroupButton vertically or horizontally.

Default value is Vertical. (Vertical | Horizontal)

**Example**

```html
<div lge-type="GroupCheckBox" lge-attrib="box-orient:'Vertical';">
  <fieldset>
    <legend>Default Group CheckBox</legend>
    <label><input type="checkbox" lge-type="CheckBox" />I agree</label>
    <label><input type="checkbox" lge-type="CheckBox" />I agree2</label>
    <label><input type="checkbox" lge-type="CheckBox" />I agree3</label>
  </fieldset>
</div>
```
JS Interface (Constructor Parameters)

Example

```javascript
var Checkbox2 = LGE.UI.GroupCheckBox({
    selector:("td2"),
    legend:'Default Group CheckBox',
    items:[
        {tabRight:'check1',checked:'checked',id:'1',text:'I agree'},
        {id:'2', text:'I agree2'},
        {id:'3', text:'I agree3'},
    ],
});
```

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>boxOrient</td>
<td>Aligns the parts of component vertically or horizontally.</td>
</tr>
<tr>
<td>id</td>
<td>Element ID of the component in document</td>
</tr>
<tr>
<td>items</td>
<td>A list of checkboxes</td>
</tr>
<tr>
<td>legend</td>
<td>Text of legend tag in the component</td>
</tr>
<tr>
<td>selector</td>
<td>The element that becomes parent of the component</td>
</tr>
</tbody>
</table>

**boxOrient**

**Description**
Aligns the parts of CheckBoxes vertically or horizontally.
Default value is Vertical. (Vertical | Horizontal)

**Example**

```javascript
var Checkbox2 = LGE.UI.GroupCheckBox({
    selector:("td2"),
    boxOrient:'Horizontal',
    legend:'Default Group CheckBox',
    items:[
        {checked:'checked', id:'1', text:'I agree'},
        {id:'2', text:'I agree2'},
    ],
});
```

**id**

**Description**
Element ID of the component in document.

**Example**

```javascript
var Checkbox2 = LGE.UI.GroupCheckBox({
    selector:("td2"),
    id:'groupcheckbox',
    boxOrient:'Horizontal',
    legend:'Default Group CheckBox',
    items:[
        {checked:'checked', id:'1', text:'I agree'},
        {id:'2', text:'I agree2'},
    ],
});
```
items

Description
A list of CheckBox Components

Example
```javascript
var Checkbox2 = LGE.UI.GroupCheckBox({
  selector:('td2'),
  boxOrient:'Horizontal',
  legend:'Default Group CheckBox',
  items:[
    {checked:'checked', id:'1', text:'I agree'},
    {id:'2', text:'I agree2'},
  ],
});
```

legend

Description
Text of legend element with GroupCheckBox Components.

Example
```javascript
var Checkbox2 = LGE.UI.GroupCheckBox({
  selector:('td2'),
  boxOrient:'Horizontal',
  legend:'Default Group CheckBox',
  items:[
    {checked:'checked', id:'1', text:'I agree'},
    {id:'2', text:'I agree2'},
  ],
});
```

selector

Description
The element that becomes parent of the component
Default is body.

Example
```javascript
var Checkbox2 = LGE.UI.GroupCheckBox({
  selector:('td2'),
  boxOrient:'Horizontal',
  legend:'Default Group CheckBox',
  items:[
    {checked:'checked', id:'1', text:'I agree'},
    {id:'2', text:'I agree2'},
  ],
});
```
JS Interface (Methods)

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>addItem</td>
<td>Adds item to the GroupCheckBox.</td>
</tr>
<tr>
<td>getBoxOrient</td>
<td>Gets the order direction of the GroupCheckBox.</td>
</tr>
<tr>
<td>getItem</td>
<td>Gets the item of the GroupCheckBox.</td>
</tr>
<tr>
<td>getLegend</td>
<td>Gets the legend of the GroupCheckBox.</td>
</tr>
<tr>
<td>removeItem</td>
<td>Removes the item of the GroupCheckBox.</td>
</tr>
<tr>
<td>setBoxOrient</td>
<td>Sets the order direction of the GroupCheckBox.</td>
</tr>
<tr>
<td>setItem</td>
<td>Sets the item of the GroupCheckBox.</td>
</tr>
<tr>
<td>setLegend</td>
<td>Sets the order direction of the GroupCheckBox.</td>
</tr>
</tbody>
</table>

**addItem**

**Description**
Sets the order direction of the GroupCheckBox.

**Parameters**
- check (Boolean) Whether the CheckBox should be checked or not
- text (String) The text of CheckBox

**Return Value**
None

**Example**
```javascript
groupCheckBox.addItem(true, “CheckBoxText”);
```

**Supported SDK / Emulator Version**

<table>
<thead>
<tr>
<th>SDK Version</th>
<th>Emulator Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.2 or higher</td>
<td>LG Smart TV Emulator 2011 : Not Supported</td>
</tr>
<tr>
<td></td>
<td>LG Smart TV Emulator 2012 : SDK 2.2 or higher</td>
</tr>
<tr>
<td></td>
<td>LG Smart TV Emulator 2013 : SDK 3.0 or higher</td>
</tr>
</tbody>
</table>

**getBoxOrient**

**Description**
Gets the order direction of the GroupCheckBox.

**Parameters**
None

**Return Value**
BoxOrient (String) The order direction of the GroupCheckBox

**Example**
```javascript
alert( groupCheckBox.getBoxOrient() ); // “Vertical” or “Horizontal”
```

**Supported SDK / Emulator Version**

<table>
<thead>
<tr>
<th>SDK Version</th>
<th>Emulator Version</th>
</tr>
</thead>
</table>
### getItem

**Description**

 Gets the item of the GroupCheckBox.

**Parameters**

- `index` (Number) The index of CheckBox in GroupCheckBox

**Return Value**

- `item` (Object) The item of Checkbox

For more information of this object, see JavaScript API in CheckBox component.

**Example**

```javascript
var text = groupCheckBox.getItem(1).getText();
groupCheckBox.getItem(0).setText("test");
```

### getLegend

**Description**

 Gets the legend of the GroupCheckBox.

**Parameters**

None

**Return Value**

- `legend` (String) The legend of GroupCheckBox

**Example**

```javascript
var legend = groupCheckBox.getLegend();
```

### removeItem

**Description**
Remove the item of the GroupCheckBox.

**Parameters**
- **index** *(Number)* The index of CheckBox in GroupCheckBox

**Return Value**
None

**Example**
```javascript
groupCheckBox.removeItem(0);
```

**Supported SDK / Emulator Version**

<table>
<thead>
<tr>
<th>SDK Version</th>
<th>Emulator Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.2 or higher</td>
<td>LG Smart TV Emulator 2011: Not Supported, LG Smart TV Emulator 2012: SDK 2.2 or higher, LG Smart TV Emulator 2013: SDK 3.0 or higher</td>
</tr>
</tbody>
</table>

**setItem**

**Description**
Sets the item of the GroupCheckBox.

**Parameters**
- **index** *(Number)* The index of CheckBox in GroupCheckBox
- **check** Whether the state of CheckBox should be checked or not.
- **text** The text of CheckBox

**Return Value**
None

**Example**
```javascript
groupCheckBox.setItem(1, true, 'Checkbox1');
```

**Supported SDK / Emulator Version**

<table>
<thead>
<tr>
<th>SDK Version</th>
<th>Emulator Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.2 or higher</td>
<td>LG Smart TV Emulator 2011: Not Supported, LG Smart TV Emulator 2012: SDK 2.2 or higher, LG Smart TV Emulator 2013: SDK 3.0 or higher</td>
</tr>
</tbody>
</table>

**setBoxOrient**

**Description**
Sets the order direction of the GroupCheckBox.

**Parameters**
- **direction** *(String)* The order direction of the GroupCheckBox ('Vertical' | 'Horizontal')

**Return Value**
None

**Example**
```javascript
groupCheckBox.setBoxOrient('Horizontal');
```
**Supported SDK / Emulator Version**

<table>
<thead>
<tr>
<th>SDK Version</th>
<th>Emulator Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.2 or higher</td>
<td>LG Smart TV Emulator 2011 : Not Supported</td>
</tr>
<tr>
<td></td>
<td>LG Smart TV Emulator 2012 : SDK 2.2 or higher</td>
</tr>
<tr>
<td></td>
<td>LG Smart TV Emulator 2013 : SDK 3.0 or higher</td>
</tr>
</tbody>
</table>

**setLegend**

**Description**
Sets the order direction of the GroupCheckBox.

**Parameters**

| legend | (String) The legend of the GroupCheckBox |

**Return Value**

None

**Example**

```javascript
groupCheckBox.setLegend("Group Legend 1");
```

**Supported SDK / Emulator Version**

<table>
<thead>
<tr>
<th>SDK Version</th>
<th>Emulator Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.2 or higher</td>
<td>LG Smart TV Emulator 2011 : NotSupported</td>
</tr>
<tr>
<td></td>
<td>LG Smart TV Emulator 2012 : SDK 2.2 or higher</td>
</tr>
<tr>
<td></td>
<td>LG Smart TV Emulator 2013 : SDK 3.0 or higher</td>
</tr>
</tbody>
</table>

**GroupRadio**

The GroupRadio component makes a group of Radios.

![GroupRadio Component](image)

**Inheritance Hierarchy**

Object >> Component >> Container >> GroupCheckBoxRadioList >> GroupCheckBox

Markup Interface, JS Interface (Constructor Parameters), and JS Interface (Methods) are listed in the following sections.

- [Markup Interface](#)
- [JS Interface (Constructor Parameters)](#)
- [JS Interface (Methods)](#)
Markup Interface

Example

```html
<div lge-type="GroupRadio">
  <fieldset>
    <legend>Horizontal option Group Radio</legend>
    <label><input type="radio" lge-type=" Radio" />I agree</label>
    <label><input type="radio" lge-type=" Radio" />I agree2</label>
    <label><input type="radio" lge-type=" Radio" checked/>I agree3</label>
  </fieldset>
</div>
```

Attributes Summary

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>lge-attr</td>
<td>Attributes of the LGE component</td>
</tr>
<tr>
<td>box-orient</td>
<td>Aligns the parts of component vertically or horizontally.</td>
</tr>
</tbody>
</table>

**box-orient**

Description

Aligns the parts of Radios vertically or horizontally.
Default value is Vertical. (Vertical | Horizontal)

Example

```html
<div lge-type="GroupRadio" lge-attr="box-orient:'Horizontal';">
  <fieldset>
    <legend>Horizontal option Group Radio</legend>
    <label><input type="radio" lge-type="Radio" />I agree</label>
    <label><input type="radio" lge-type="Radio" />I agree2</label>
    <label><input type="radio" lge-type="Radio" checked/>I agree3</label>
  </fieldset>
</div>
```
JS Interface (Constructor Parameters)

Example

```javascript
var Radio2 = LGE.UI.GroupRadio({
  selector:("td2"),
  boxOrient:'Vertical',
  legend:'Default Group Radio',
  items:[
    {checked:'checked',id:'1', text:'I agree'},
    {id:'2', text:'I agree2'},
    {id:'3', text:'I agree3'},
  ],
});
```

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>boxOrient</td>
<td>Aligns the parts of component vertically or horizontally.</td>
</tr>
<tr>
<td>Legend</td>
<td>Text of legend tag in the component</td>
</tr>
<tr>
<td>Id</td>
<td>Element ID of the component in document</td>
</tr>
<tr>
<td>Items</td>
<td>A list of radios</td>
</tr>
<tr>
<td>Selector</td>
<td>The element that becomes parent of the component</td>
</tr>
</tbody>
</table>

**boxOrient**

**Description**

Aligns the parts of Radio vertically or horizontally. Default value is Vertical. (Vertical | Horizontal)

**Example**

```javascript
var Radio2 = LGE.UI.GroupRadio({
  selector:("td2"),
  boxOrient:'Vertical',
  legend:'Default Group Radio',
  items:[
    {checked:'checked',id:'1', text:'I agree'},
    {id:'2', text:'I agree2'},
  ],
});
```

**legend**

**Description**

Text of legend element with GroupRadio Components.

**Example**

```javascript
var Radio2 = LGE.UI.GroupRadio({
  selector:("td2"),
  boxOrient:'Vertical',
  legend:'Default Group Radio',
  items:[
    {checked:'checked',id:'1', text:'I agree'},
    {id:'2', text:'I agree2'},
  ],
});
```
id

Description
Element ID of the component in document.

Example
```javascript
var Radio2 = LGE.UI.GroupRadio({
  id:'groupradio',
  selector:("td2"),
  boxOrient:'Vertical',
  legend:'Default Group Radio',
  items:[
    {checked:'checked',id:'1', text:'I agree'},
    {id:'2', text:'I agree2'},
    {id:'3', text:'I agree3'},
  ],
});
```

items

Description
A list of Radio Components

Example
```javascript
var Radio2 = LGE.UI.GroupRadio({
  selector:("td2"),
  boxOrient:'Vertical',
  legend:'Default Group Radio',
  items:[
    {checked:'checked',id:'1', text:'I agree'},
    {id:'2', text:'I agree2'},
    {id:'3', text:'I agree3'},
  ],
});
```

selector

Description
The element that becomes parent of the component
Default is body.

Example
```javascript
var Radio2 = LGE.UI.GroupRadio({
  selector:("td2"),
  boxOrient:'Vertical',
  legend:'Default Group Radio',
  items:[
    {checked:'checked',id:'1', text:'I agree'},
    {id:'2', text:'I agree2'},
    {id:'3', text:'I agree3'},
  ],
});
```
JS Interface (Methods)

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>addItem</code></td>
<td>Adds item to the GroupCheckBox.</td>
</tr>
<tr>
<td><code>getBoxOrient</code></td>
<td>Gets the order direction of the GroupCheckBox.</td>
</tr>
<tr>
<td><code>getItem</code></td>
<td>Gets the item of the GroupCheckBox.</td>
</tr>
<tr>
<td><code>getLegend</code></td>
<td>Gets the legend of the GroupCheckBox.</td>
</tr>
<tr>
<td><code>removeItem</code></td>
<td>Removes the item of the GroupCheckBox.</td>
</tr>
<tr>
<td><code>setBoxOrient</code></td>
<td>Sets the order direction of the GroupCheckBox.</td>
</tr>
<tr>
<td><code>setItem</code></td>
<td>Sets the item of the GroupCheckBox.</td>
</tr>
<tr>
<td><code>setLegend</code></td>
<td>Sets the order direction of the GroupCheckBox.</td>
</tr>
</tbody>
</table>

**addItem**

**Description**
Sets the order direction of the GroupRadio.

**Parameters**
- `check` (Boolean) Whether the status of Radio should be checked or not
- `text` (String) The text of Radio

**Return Value**
None

**Example**
```
groupRadio.addItem(true, "Text");
```

**Supported SDK / Emulator Version**

<table>
<thead>
<tr>
<th>SDK Version</th>
<th>Emulator Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.2 or higher</td>
<td>LG Smart TV Emulator 2011 : Not Supported</td>
</tr>
<tr>
<td></td>
<td>LG Smart TV Emulator 2012 : SDK 2.2 or higher</td>
</tr>
<tr>
<td></td>
<td>LG Smart TV Emulator 2013 : SDK 3.0 or higher</td>
</tr>
</tbody>
</table>

**getBoxOrient**

**Description**
Gets the order direction of the GroupRadio.

**Parameters**
None

**Return Value**
`BoxOrient` (String) The order direction of the GroupRadio

**Example**
```
alert( groupRadio.getBoxOrient() ); // “Vertical” or “Horizontal”
```

**Supported SDK / Emulator Version**

<table>
<thead>
<tr>
<th>SDK Version</th>
<th>Emulator Version</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### SDK Version | Emulator Version
--- | ---
2.2 or higher | LG Smart TV Emulator 2011 : Not Supported
| LG Smart TV Emulator 2012 : SDK 2.2 or higher
| LG Smart TV Emulator 2013 : SDK 3.0 or higher

#### getItem

**Description**
Gets the item of the GroupRadio.

**Parameters**
- `index` (Number) The index of Radio in GroupRadio

**Return Value**
- `item` (Object) The item of Radio
  For more information of this object, see JavaScript API in Radio component.

**Example**
```javascript
var text = groupRadio.getItem(1).getText();
groupRadio.getItem(0).setText("test");
```

#### getLegend

**Description**
 Gets the legend of the GroupRadio.

**Parameters**
None

**Return Value**
- `Legend` (String) The legend of GroupRadio

**Example**
```javascript
var legend = groupRadio.getLegend();
```

#### removeItem

**Description**
Removes the item of the GroupRadio.

**Parameters**

- **index** (Number) The index of Radio in GroupRadio

**Return Value**

None

**Example**

```javascript
groupRadio.removeItem(0);
```

**Supported SDK / Emulator Version**

<table>
<thead>
<tr>
<th>SDK Version</th>
<th>Emulator Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.2 or higher</td>
<td>LG Smart TV Emulator 2011 : Not Supported</td>
</tr>
<tr>
<td></td>
<td>LG Smart TV Emulator 2012 : SDK 2.2 or higher</td>
</tr>
<tr>
<td></td>
<td>LG Smart TV Emulator 2013 : SDK 3.0 or higher</td>
</tr>
</tbody>
</table>

**setBoxOrient**

**Description**

Sets the order direction of the GroupRadio.

**Parameters**

- **direction** (String) The order direction of the GroupRadio ("Vertical" | "Horizontal")

**Return Value**

None

**Example**

```javascript
groupRadio.setBoxOrient('Horizontal');
```

**Supported SDK / Emulator Version**

<table>
<thead>
<tr>
<th>SDK Version</th>
<th>Emulator Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.2 or higher</td>
<td>LG Smart TV Emulator 2011 : Not Supported</td>
</tr>
<tr>
<td></td>
<td>LG Smart TV Emulator 2012 : SDK 2.2 or higher</td>
</tr>
<tr>
<td></td>
<td>LG Smart TV Emulator 2013 : SDK 3.0 or higher</td>
</tr>
</tbody>
</table>

**setItem**

**Description**

Sets the item of the GroupRadio.

**Parameters**

- **index** (Number) The index of Radio in GroupRadio
- **check** Whether the state of Radio should be checked or not.
- **text** The text of Radio

**Return Value**

None

**Example**

```javascript
groupRadio.setItem(1, true, 'Radio1');
```
<table>
<thead>
<tr>
<th>Supported SDK / Emulator Version</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>SDK Version</strong></td>
</tr>
<tr>
<td>-----------------</td>
</tr>
<tr>
<td>2.2 or higher</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
</tbody>
</table>

**setLegend**

**Description**
Sets the order direction of the GroupRadio.

**Parameters**

`legend` *(String)* The legend of the GroupRadio

**Return Value**
None

**Example**

```javascript
groupRadio.setLegend("Group Legend 1");
```

<table>
<thead>
<tr>
<th>Supported SDK / Emulator Version</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>SDK Version</strong></td>
</tr>
<tr>
<td>-----------------</td>
</tr>
<tr>
<td>2.2 or higher</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
</tbody>
</table>

**ImageList**

The ImageList component is a group of images and used for showing like flow list.

![ImageList Component](image)

**Inheritance Hierarchy**

Object >> Component >> Container >> ImageList

Markup Interface, JS Interface (Constructor Parameters), and JS Interface (Methods) are listed in the following sections.

- **Markup Interface**
- **JS Interface (Constructor Parameters)**
- **JS Interface (Methods)**
Markup Interface

Example

```html
<div lge-type="ImageList" id="imgList" style='width:1000px;height:400px'>
  <li lge-type="ImageListItem" value="1">
    <img src='./image/imagelist_0.jpg'>
  </li>
  <li lge-type="ImageListItem" value="2">
    <img src='./image/imagelist_1.jpg'>
  </li>
  <li lge-type="ImageListItem" value="3">
    <img src='./image/imagelist_2.jpg'>
  </li>
  <li lge-type="ImageListItem" value="4">
    <img src='./image/imagelist_3.jpg'>
  </li>
  <li lge-type="ImageListItem" value="5">
    <img src='./image/imagelist_4.jpg'>
  </li>
</div>
```

Attributes Summary

None
JS Interface (Constructor Parameters)

Example

```javascript
var ImageList = LGE.UI.ImageList({
    image:
        ['./image/imagelist_0.jpg',
        './image/imagelist_1.jpg',
        './image/imagelist_2.jpg',
        './image/imagelist_3.jpg',
        './image/imagelist_4.jpg',
        './image/imagelist_5.jpg',
    },
    selector:('td2'),
    style:'width:1000px;height:400px;',
});
```

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>id</td>
<td>Element ID of the component in document</td>
</tr>
<tr>
<td>image</td>
<td>A list of Images</td>
</tr>
<tr>
<td>selector</td>
<td>The element that becomes parent of the component</td>
</tr>
<tr>
<td>style</td>
<td>CSS Style of the component</td>
</tr>
</tbody>
</table>

id

Description
Element ID of the component in document.

Example
```
var ImageList = LGE.UI.ImageList({
    id: 'imglist',
    image:
        ['./image/imagelist_0.jpg',
        './image/imagelist_1.jpg',
    },
    selector:('td2'),
    style:'width:1000px;height:400px;',
});
```

image

Description
List of the images

Example
```
var ImageList = LGE.UI.ImageList({
    image:
        ['./image/imagelist_0.jpg',
        './image/imagelist_1.jpg',
        './image/imagelist_2.jpg',
        './image/imagelist_3.jpg',
        './image/imagelist_4.jpg',
        './image/imagelist_5.jpg',
    },
});
```
selector

Description
The element that becomes parent of the component
Default is body.

Example
```javascript
var ImageList = LGE.UI.ImageList({
  image:[
    './image/imagelist_0.jpg',
    './image/imagelist_1.jpg',
  ],
  selector:('td2'),
  style:'width:1000px;height:400px;',
});
```

style

Description
CSS style of the component

Example
```javascript
var ImageList = LGE.UI.ImageList({
  image:[
    './image/imagelist_0.jpg',
    './image/imagelist_1.jpg',
  ],
  selector:('td2'),
  style:'width:1000px;height:400px;',
});
```
JS Interface (Methods)

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>addImageItem</td>
<td>Adds an ImageItem of the ImageList.</td>
</tr>
<tr>
<td>getCurrentIndex</td>
<td>Gets the current index of the ImageItem.</td>
</tr>
<tr>
<td>moveImageItem</td>
<td>Moves to the ImageItem index of the ImageList.</td>
</tr>
<tr>
<td>removeImageItem</td>
<td>Removes the indexed Item of the ImageList.</td>
</tr>
</tbody>
</table>

**addImageItem**

**Description**
Adds an ImageItem of the ImageList.

**Parameters**
- index (Number) The index of the ImageItem
- path (String) The path of the ImageItem

**Return Value**
None

**Example**
```javascript
var imagelist = LGE.UI.getComp('imagelist');
imagelist.addImageItem(5, './image1.gif');
```

**Supported SDK / Emulator Version**

<table>
<thead>
<tr>
<th>SDK Version</th>
<th>Emulator Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.2 or higher</td>
<td>LG Smart TV Emulator 2011 : Not Supported</td>
</tr>
<tr>
<td></td>
<td>LG Smart TV Emulator 2012 : SDK 2.2 or higher</td>
</tr>
<tr>
<td></td>
<td>LG Smart TV Emulator 2013 : SDK 3.0 or higher</td>
</tr>
</tbody>
</table>

**getCurrentIndex**

**Description**
 Gets the current index of the ImageItem.

**Parameters**
None

**Return Value**
- index (Number) The current index of ImageItem

**Example**
```javascript
var imagelist = LGE.UI.getComp('imagelist');
var index = imagelist.getCurrentIndex();
```

**Supported SDK / Emulator Version**

<table>
<thead>
<tr>
<th>SDK Version</th>
<th>Emulator Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.2 or higher</td>
<td>LG Smart TV Emulator 2011 : Not Supported</td>
</tr>
<tr>
<td></td>
<td>LG Smart TV Emulator 2012 : SDK 2.2 or higher</td>
</tr>
<tr>
<td></td>
<td>LG Smart TV Emulator 2013 : SDK 3.0 or higher</td>
</tr>
</tbody>
</table>
**moveImageItem**

**Description**
Moves to the ImageItem of the ImageList.

**Parameters**
- `index` (Number) The index of the ImageItem

**Return Value**
None

**Example**
```javascript
var imagelist = LGE.UI.getComp('imagelist');
imagelist.moveImageItem(3);
```

**Supported SDK / Emulator Version**

<table>
<thead>
<tr>
<th>SDK Version</th>
<th>Emulator Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.2 or higher</td>
<td>LG Smart TV Emulator 2011: Not Supported</td>
</tr>
<tr>
<td></td>
<td>LG Smart TV Emulator 2012: SDK 2.2 or higher</td>
</tr>
<tr>
<td></td>
<td>LG Smart TV Emulator 2013: SDK 3.0 or higher</td>
</tr>
</tbody>
</table>

**removeImageItem**

**Description**
Removes the Item of the ImageList.

**Parameters**
- `index` (Number) The index of the ImageItem

**Return Value**
None

**Example**
```javascript
var imagelist = LGE.UI.getComp('imagelist');
imagelist.removeItem(2);
```

**Supported SDK / Emulator Version**

<table>
<thead>
<tr>
<th>SDK Version</th>
<th>Emulator Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.2 or higher</td>
<td>LG Smart TV Emulator 2011: Not Supported</td>
</tr>
<tr>
<td></td>
<td>LG Smart TV Emulator 2012: SDK 2.2 or higher</td>
</tr>
<tr>
<td></td>
<td>LG Smart TV Emulator 2013: SDK 3.0 or higher</td>
</tr>
</tbody>
</table>

**ImageTile**

The ImageTile component is a group of images and used for showing like a tile.
Inheritance Hierarchy
Object >> Component >> Container >> ImageTile

Markup Interface, JS Interface (Constructor Parameters), and JS Interface (Methods) are listed in the following sections.
• Markup Interface
• JS Interface (Constructor Parameters)
• JS Interface (Methods)
Markup Interface

Example

```html
<div lge-type="ImageTile" lge-attr="option:1;" style="width:700px;height:500px;"
 id="imgTile3">
 <table lge-type="ImageTileItem">
  <tr>
   <td>
    <img lge-type="ImageItem" lge-attr="tabRight:'img1';" src='images/21.jpg'>
   </td>
   <td>
    <img lge-type="ImageItem" id="img1" src='images/11.jpg'></td>
  </tr>
 </table>
 <table lge-type="ImageTileItem">
  <tr>
   <td>
    <img lge-type="ImageItem" src='images/12.jpg'>
   </td>
   <td>
    <img lge-type="ImageItem" src='images/11.jpg'></td>
  </tr>
 </table>
</div>

Attributes Summary

None
JS Interface (Constructor Parameters)

Example

```javascript
var ts = LGE.UI.ImageTile({
    selector:$("td2"),
    tables:[
        (table:[ // 1st table
            {images:[ // 1st table, 1st row
                {tabRight:'img1', src:'images/11.jpg', rowspan:'1', colspan:'2'}, // 1st row, 1st ~2nd column
                {src:'images/11.jpg'}, // 1st row, 3rd column
            ]},
            {images:[ // 2nd row
                {src:'images/11.jpg'}, // 2nd row, 1st column
                {src:'images/11.jpg'}, // 2nd row, 2nd column
                {src:'images/11.jpg'}, // 2nd row, 3rd column
            ]},
        ],
        (table:[ // 2nd table
            {images:[ // 2nd table, 1st row
                {src:'images/11.jpg'}, // 1st row, 1st column
                {src:'images/11.jpg'}, // 1st row, 2nd column
                {src:'images/11.jpg'}, // 1st row, 3rd column
            ]},
            {images:[ // 2nd row
                {src:'images/11.jpg'}, // 2nd row, 1st column
                {src:'images/11.jpg'}, // 2nd row, 2nd column
                {src:'images/11.jpg'}, // 2nd row, 3rd column
            ]},
        ]),
        option:'1',
        style:'width:700px;height:500px;',
        id:'imgTile3',
    ]
});
```

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>id</td>
<td>Element ID of the component in document</td>
</tr>
<tr>
<td>option</td>
<td>Table moves in one drag</td>
</tr>
<tr>
<td>selector</td>
<td>The element that becomes parent of the component</td>
</tr>
<tr>
<td>style</td>
<td>CSS Style of the component</td>
</tr>
<tr>
<td>tables</td>
<td>A list of tables that has images</td>
</tr>
</tbody>
</table>

**id**

**Description**
Element ID of the component in document.

**Example**

```javascript
var ts = LGE.UI.ImageTile({
    selector:$("td2"),
    tables:[
        (table:[ // 1st table
            {images:[ // 1st table, 1st row
                {src:'images/11.jpg'}, // 1st row, 1st column
            ]},
            {images:[ // 2nd row
                {src:'images/11.jpg'}, // 2nd row, 1st column
            ]},
        ])
```
option

Description
This option is times of table moves in one drag. For example, if option: 2 and one drag, the table moves two time.

Example

```javascript
var ts = LGE.UI.ImageTile({
  selector: $('td2'),
  tables:[
    {table:
      {images:[
        {src:'images/11.jpg'}, // 1st table, 1st row
          // 1st row, 1st column
        ],
        {images:[
          {src:'images/11.jpg'}, // 2nd row
            // 2nd row, 1st column
        ]},
        option:'1',
        style:'width:700px;height:500px;',
        id:'imgTile3',
      ]},
    ]},
  option:'1',
  style:'width:700px;height:500px;',
  id:'imgTile3',
});
```

selector

Description
The element that becomes parent of the component
Default is body.

Example

```javascript
var ts = LGE.UI.ImageTile({
  selector: $('td2'),
  tables:[
    {table:
      {images:[
        {src:'images/11.jpg'}, // 1st table, 1st row
          // 1st row, 1st column
        ],
        {images:[
          {src:'images/11.jpg'}, // 2nd row
            // 2nd row, 1st column
        ]},
        option:'1',
        style:'width:700px;height:500px;',
        id:'imgTile3',
      ]},
    ]},
  option:'1',
  style:'width:700px;height:500px;',
  id:'imgTile3',
});
```
style

Description
CSS style of the component

Example
```javascript
var ts = LGE.UI.ImageTile({
    selector: $('td2'),
    tables: [
        {table: { // 1st table
            images: [ // 1st table, 1st row
                {src: 'images/11.jpg'}, // 1st row, 1st column
            ],
            images: [ // 2nd row
                {src: 'images/11.jpg'}, // 2nd row, 1st column
            ],
        },
        option: '1',
        style: 'width: 700px; height: 500px;',
        id: 'imgTile3',
    ]});
```

tables

Description
tables: A list of tables that has images.
- table: A table of images. It has several rows.
  + images: A row of images. It has several images.
    > src: path of image
    > rowspan, colspan: how many cross or how many down cells
    > tabRight, tabLeft, tabUp, tabDown: Indicate next UI component by each key action.

Example
```javascript
var ts = LGE.UI.ImageTile({
    selector: $('td2'),
    tables: [
        {table: { // 1st table
            images: [ // 1st table, 1st row
                {src: 'images/11.jpg'}, // 1st row, 3rd column
            ],
            images: [ // 2nd row
                {src: 'images/11.jpg'}, // 2nd row, 1st column
                {src: 'images/11.jpg'}, // 2nd row, 2nd column
                {src: 'images/11.jpg'}, // 2nd row, 3rd column
            ],
        },
        {table: { // 2nd table
            images: [ // 2nd table, 1st row
                {src: 'images/11.jpg'}, // 1st row, 1st column
                {src: 'images/11.jpg'}, // 1st row, 2nd column
                {src: 'images/11.jpg'}, // 1st row, 3rd column
            ],
            images: [ // 2nd row
                {src: 'images/11.jpg'}, // 2nd row, 1st column
                {src: 'images/11.jpg'}, // 2nd row, 2nd column
                {src: 'images/11.jpg'}, // 2nd row, 3rd column
            ],
        }],
    ]});
```
}},

},

option:'1',

style:'width:700px;height:500px;',

id:'imgTile3',

});
JS Interface (Methods)

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>getBackground</td>
<td>Gets background color of ImageTile.</td>
</tr>
<tr>
<td>getBorderSpacing</td>
<td>Gets border spacing of ImageTile.</td>
</tr>
<tr>
<td>getMargin</td>
<td>Gets margin of ImageTile.</td>
</tr>
<tr>
<td>getNumToMove</td>
<td>Gets num to move of ImageTile.</td>
</tr>
<tr>
<td>setBackground</td>
<td>Sets background color of ImageTile.</td>
</tr>
<tr>
<td>setBorderSpacing</td>
<td>Sets border spacing of ImageTile.</td>
</tr>
<tr>
<td>setMargin</td>
<td>Sets margin of ImageTile.</td>
</tr>
<tr>
<td>setNumToMove</td>
<td>Sets num to move of ImageTile.</td>
</tr>
</tbody>
</table>

**getBackground**

**Description**

Gets background color of ImageTile.

**Parameters**

None

**Return Value**

*Color* (String) The background color of Image

**Example**

```javascript
var imagetile = LGE.UI.getComp('imagetile');
var color = imagetile.getBackground();
```

**Supported SDK / Emulator Version**

<table>
<thead>
<tr>
<th>SDK Version</th>
<th>Emulator Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.2 or higher</td>
<td>LG Smart TV Emulator 2011: Not Supported</td>
</tr>
<tr>
<td></td>
<td>LG Smart TV Emulator 2012: SDK 2.2 or higher</td>
</tr>
<tr>
<td></td>
<td>LG Smart TV Emulator 2013: SDK 3.0 or higher</td>
</tr>
</tbody>
</table>

**getBorderSpacing**

**Description**

Gets border spacing of ImageTile.

**Parameters**

None

**Return Value**

*borderSpacing* (String) The border spacing of ImageTile with unit

**Example**

```javascript
var imagetile = LGE.UI.getComp('imagetile');
var space = imagetile.getBorderSpacing(); // ex) 10px
```

**Supported SDK / Emulator Version**

<table>
<thead>
<tr>
<th>SDK Version</th>
<th>Emulator Version</th>
</tr>
</thead>
</table>
**setMargin**

**Description**
Gets margin of ImageTile.

**Parameters**
None

**Return Value**
margin (String) The margin of ImageTile with unit

**Example**
```
var imagetile = LGE.UI.getComp('imagetile');
var margin = imagetile.getMargin(); // 10px
```

**Supported SDK / Emulator Version**

<table>
<thead>
<tr>
<th>SDK Version</th>
<th>Emulator Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.2 or higher</td>
<td>LG Smart TV Emulator 2011 : Not Supported</td>
</tr>
<tr>
<td></td>
<td>LG Smart TV Emulator 2012 : SDK 2.2 or higher</td>
</tr>
<tr>
<td></td>
<td>LG Smart TV Emulator 2013 : SDK 3.0 or higher</td>
</tr>
</tbody>
</table>

**getNumToMove**

**Description**
Gets number to move of ImageTile. This option is times of table moves in one drag. For example, if option:2 and one drag, the table moves two times.

**Parameters**
None

**Return Value**
index (Number) The index of ImageTile

**Example**
```
var imagetile = LGE.UI.getComp('imagetile');
var index = imagetile.getNumToMove();
```

**Supported SDK / Emulator Version**

<table>
<thead>
<tr>
<th>SDK Version</th>
<th>Emulator Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.2 or higher</td>
<td>LG Smart TV Emulator 2011 : Not Supported</td>
</tr>
<tr>
<td></td>
<td>LG Smart TV Emulator 2012 : SDK 2.2 or higher</td>
</tr>
<tr>
<td></td>
<td>LG Smart TV Emulator 2013 : SDK 3.0 or higher</td>
</tr>
</tbody>
</table>

**setBackground**
Description
Sets background color of ImageTile.

Parameters
Color                   (String) The background color of Image

Return Value
None

Example
var imagetile = LGE.UI.getComp('imagetile');
imagetile.setBackground("blue");

Supported SDK / Emulator Version

<table>
<thead>
<tr>
<th>SDK Version</th>
<th>Emulator Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.2 or higher</td>
<td>LG Smart TV Emulator 2011 : Not Supported</td>
</tr>
<tr>
<td></td>
<td>LG Smart TV Emulator 2012 : SDK 2.2 or higher</td>
</tr>
<tr>
<td></td>
<td>LG Smart TV Emulator 2013 : SDK 3.0 or higher</td>
</tr>
</tbody>
</table>

setBorderSpacing

Description
Sets border spacing of ImageTile.

Parameters
borderSpacing       (String) The border spacing of ImageTile with unit

Return Value
None

Example
var imagetile = LGE.UI.getComp('imagetile');
imagetile.setBorderSpacing("10px");

Supported SDK / Emulator Version

<table>
<thead>
<tr>
<th>SDK Version</th>
<th>Emulator Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.2 or higher</td>
<td>LG Smart TV Emulator 2011 : Not Supported</td>
</tr>
<tr>
<td></td>
<td>LG Smart TV Emulator 2012 : SDK 2.2 or higher</td>
</tr>
<tr>
<td></td>
<td>LG Smart TV Emulator 2013 : SDK 3.0 or higher</td>
</tr>
</tbody>
</table>

setMargin

Description
Sets margin of ImageTile.

Parameters
margin                  (String) the margin of ImageTile with unit

Return Value
None

Example
var imagetile = LGE.UI.getComp('imagetile');
imagetile.setMargin("10px");

### Supported SDK / Emulator Version

<table>
<thead>
<tr>
<th>SDK Version</th>
<th>Emulator Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.2 or higher</td>
<td>LG Smart TV Emulator 2011: Not Supported</td>
</tr>
<tr>
<td></td>
<td>LG Smart TV Emulator 2012: SDK 2.2 or higher</td>
</tr>
<tr>
<td></td>
<td>LG Smart TV Emulator 2013: SDK 3.0 or higher</td>
</tr>
</tbody>
</table>

### setNumToMove

**Description**

Sets number to move of ImageTile. This option is times of table moves in one drag. For example, if option:2 and one drag, table moves two times.

**Parameters**

- **index** (Number) The index of ImageItem

**Return Value**

None

**Example**

```javascript
var imagetile = LGE.UI.getComp('imagetile');
imagetile.setNumToMove(1);
```

### Supported SDK / Emulator Version

<table>
<thead>
<tr>
<th>SDK Version</th>
<th>Emulator Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.2 or higher</td>
<td>LG Smart TV Emulator 2011: Not Supported</td>
</tr>
<tr>
<td></td>
<td>LG Smart TV Emulator 2012: SDK 2.2 or higher</td>
</tr>
<tr>
<td></td>
<td>LG Smart TV Emulator 2013: SDK 3.0 or higher</td>
</tr>
</tbody>
</table>

### Label

The Label represents a caption in a user interface.

![label](image)

[Figure] Label Component

### Inheritance Hierarchy

Object >> Component >> Container >>ItemListComponent >> Label

#### Markup Interface, JS Interface (Constructor Parameters), and JS Interface (Methods)

- **Markup Interface**
- **JS Interface (Constructor Parameters)**
- **JS Interface (Methods)**
Markup Interface

Example
<label lge-type='Label'>label2</label>

Attributes Summary

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>lge-attr</td>
<td>Attributes of the LGE component</td>
</tr>
</tbody>
</table>

| text-inline      | More compact component that is only as wide as the text inside |
| text-scroll      | Specifies whether text should be moved in the label.         |
| text-scroll-direction | When you add lge-scroll='true' attribute in the label, specifies moving direction. |

**text-inline**

Description
More compact component that is only as wide as the text inside
Default value is false. (true | false)

Example
<label lge-type='Label' lge-attr="text-inline:true;">label2</label>

**text-scroll**

Description
Specifies whether text should be moved in the label.
Default value is false. (true | false)

Example
<label lge-type='Label' lge-attr='text-scroll:true;text-scroll-direction:up'>label2</label>

**text-scroll-direction**

Description
When you add lge-scroll='true' attribute in the label, specifies moving direction.
Default value is Left. (Left | Right | Up | Down)

Example
<label lge-type='Label' lge-attr='text-scroll-direction:Up';text-scroll:true;>label2</label>
JS Interface (Constructor Parameters)

Example
var label1 = LGE.UI.Label({
  selector:("td2"),
  text:'label3',
  id:'label1',
  style:'width:100%',
});

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>id</td>
<td>Element ID of the component in document</td>
</tr>
<tr>
<td>selector</td>
<td>The element that becomes parent of the component</td>
</tr>
<tr>
<td>style</td>
<td>CSS style of the component</td>
</tr>
<tr>
<td>text</td>
<td>Text of the component</td>
</tr>
<tr>
<td>textInline</td>
<td>More compact component that is only as wide as the text inside</td>
</tr>
<tr>
<td>textScroll</td>
<td>Specifies whether text should be moved in the label.</td>
</tr>
<tr>
<td>textScrollDirection</td>
<td>When you add lge-scroll='true' attribute in the label, specifies moving direction.</td>
</tr>
</tbody>
</table>

id

Description
Element ID of the component in document.

Example
var label1 = LGE.UI.Label({
  selector:("td2"),
  id:'label1',
  text:'label3',
});

selector

Description
The element that becomes parent of the component
Default is body.

Example
var label1 = LGE.UI.Label({
  selector:("td2"),
  id:'label1',
  text:'label3',
});

style

Description
CSS style of the component
Example

```javascript
var label1 = LGE.UI.Label({
    selector:("td2"),
    id:'label1',
    text:'label3',
    style:'width:200px',
});
```

**text**

Description
Text of the component

Example

```javascript
var label1 = LGE.UI.Label({
    selector:("td2"),
    id:'label1',
    text:'label3',
});
```

**textInline**

Description
More compact component that is only as wide as the text inside
Default value is false. (true | false)

Example

```javascript
var label1 = LGE.UI.Label({
    selector:("td2"),
    text:'label3',
    textInline: true,
});
```

**textScroll**

Description
Specifies whether text should be moved in the label.
Default value is false. (true | false)

Example

```javascript
var label1 = LGE.UI.Label({
    selector:("td2"),
    id:'label1',
    text:'label3',
    textScroll: true,
    textScrollDirection: 'up'
});
```

**textScrollDirection**

Description
When you add lge-scroll='true' attribute in the label, specifies moving direction.
Default value is Left. (Left | Right | Up | Down)

Example

```javascript
var label1 = LGE.UI.Label(
    selector:("td2"),
    id:'label1',
    text:'label3',
    textScroll: true,
    textScrollDirection:'Up'
));
```
### JS Interface (Methods)

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>getText</td>
<td>Gets the text of the Label.</td>
</tr>
<tr>
<td>getTextInline</td>
<td>Returns true if this label is text-inline label.</td>
</tr>
<tr>
<td>getTextScroll</td>
<td>Returns true if this label text is scrolled.</td>
</tr>
<tr>
<td>getTextScrollDirection</td>
<td>Gets the text scroll direction of the Label.</td>
</tr>
<tr>
<td>setText</td>
<td>Sets the text of the Label.</td>
</tr>
<tr>
<td>setTextInline</td>
<td>Sets the text-inline of this label.</td>
</tr>
<tr>
<td>setTextScroll</td>
<td>Sets the scroll of this label text.</td>
</tr>
<tr>
<td>setTextScrollDirection</td>
<td>Sets the text scroll direction of the Label.</td>
</tr>
</tbody>
</table>

#### getText

**Description**

Gets the text of the Label.

**Parameters**

None

**Return Value**

`text` (String) The text of Label

**Example**

```javascript
var text = label1.getText();
```

**Supported SDK / Emulator Version**

<table>
<thead>
<tr>
<th>SDK Version</th>
<th>Emulator Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.2 or higher</td>
<td>LG Smart TV Emulator 2011 : Not Supported</td>
</tr>
<tr>
<td></td>
<td>LG Smart TV Emulator 2012 : SDK 2.2 or higher</td>
</tr>
<tr>
<td></td>
<td>LG Smart TV Emulator 2013 : SDK 3.0 or higher</td>
</tr>
</tbody>
</table>

#### getTextInline

**Description**

Returns true if this label is text-inline label.

**Parameters**

None

**Return Value**

`bool` (Boolean) Whether this label is text-inline or not

**Example**

```javascript
var isInline = label1.getTextInline();
```

**Supported SDK / Emulator Version**

<table>
<thead>
<tr>
<th>SDK Version</th>
<th>Emulator Version</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>LG Smart TV Emulator 2011 : Not Supported</td>
</tr>
<tr>
<td></td>
<td>LG Smart TV Emulator 2012 : SDK 2.2 or higher</td>
</tr>
<tr>
<td></td>
<td>LG Smart TV Emulator 2013 : SDK 3.0 or higher</td>
</tr>
</tbody>
</table>
### getTextureScroll

**Description**

Returns true if this label text is scrolled.

**Parameters**

None

**Return Value**

`bool` (Boolean) Whether this label is scollable or not

**Example**

```
var isScrollable = label1.getTextScroll();
```

<table>
<thead>
<tr>
<th>SDK Version</th>
<th>Emulator Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.2 or higher</td>
<td>LG Smart TV Emulator 2011 : Not Supported, LG Smart TV Emulator 2012 : SDK 2.2 or higher, LG Smart TV Emulator 2013 : SDK 3.0 or higher</td>
</tr>
</tbody>
</table>

### getTextureScrollDirection

**Description**

Gets the text scroll direction of the Label.

**Parameters**

None

**Return Value**

`textScrollDirection` (String) The text scroll direction of the Label.

**Example**

```
var scrollDirection = label1.getTextScrollDirection();
```

<table>
<thead>
<tr>
<th>SDK Version</th>
<th>Emulator Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.2 or higher</td>
<td>LG Smart TV Emulator 2011 : Not Supported, LG Smart TV Emulator 2012 : SDK 2.2 or higher, LG Smart TV Emulator 2013 : SDK 3.0 or higher</td>
</tr>
</tbody>
</table>

### setText

**Description**

Sets the text of the Label.
Parameters
text (String) The text of Label

Return Value
None

Example
label1.setText("Hello world");

Supported SDK / Emulator Version

<table>
<thead>
<tr>
<th>SDK Version</th>
<th>Emulator Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.2 or higher</td>
<td>LG Smart TV Emulator 2011: Not Supported</td>
</tr>
<tr>
<td></td>
<td>LG Smart TV Emulator 2012: SDK 2.2 or higher</td>
</tr>
<tr>
<td></td>
<td>LG Smart TV Emulator 2013: SDK 3.0 or higher</td>
</tr>
</tbody>
</table>

setTextInline

Description
Sets the text-inline of this label.

Parameters
bool (Boolean) Whether this label is text-inline or not

Return Value
None

Example
label1.setTextInline(true);

Supported SDK / Emulator Version

<table>
<thead>
<tr>
<th>SDK Version</th>
<th>Emulator Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.2 or higher</td>
<td>LG Smart TV Emulator 2011: Not Supported</td>
</tr>
<tr>
<td></td>
<td>LG Smart TV Emulator 2012: SDK 2.2 or higher</td>
</tr>
<tr>
<td></td>
<td>LG Smart TV Emulator 2013: SDK 3.0 or higher</td>
</tr>
</tbody>
</table>

setTextScroll

Description
Sets the scroll of this label text.

Parameters
bool (Boolean) Whether this label is scrollable or not

Return Value
None

Example
label1.setTextScroll(true);

Supported SDK / Emulator Version

<table>
<thead>
<tr>
<th>SDK Version</th>
<th>Emulator Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.2 or higher</td>
<td>LG Smart TV Emulator 2011: Not Supported</td>
</tr>
<tr>
<td></td>
<td>LG Smart TV Emulator 2012: SDK 2.2 or higher</td>
</tr>
<tr>
<td></td>
<td>LG Smart TV Emulator 2013: SDK 3.0 or higher</td>
</tr>
</tbody>
</table>
setTextScrollDirection

Description
Sets the text scroll direction of the Label.

Parameters
textScrollDirection (String) The text scroll direction of the Label (Left | Right | Up | Down)

Return Value
None

Example
label1.setTextScrollDirection("Left");

Supported SDK / Emulator Version

<table>
<thead>
<tr>
<th>SDK Version</th>
<th>Emulator Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.2 or higher</td>
<td>LG Smart TV Emulator 2011: Not Supported</td>
</tr>
<tr>
<td></td>
<td>LG Smart TV Emulator 2012: SDK 2.2 or higher</td>
</tr>
<tr>
<td></td>
<td>LG Smart TV Emulator 2013: SDK 3.0 or higher</td>
</tr>
</tbody>
</table>

MsgBox

The MsgBox component is used to forward a popup message.

[Figure] MsgBox Component

Inheritance Hierarchy
Object >> Component >> Container >> Popup >> MsgBox

Markup Interface, JS Interface (Constructor Parameters), and JS Interface (Methods) are listed in the following sections.
- Markup Interface
- JS Interface (Constructor Parameters)
- JS Interface (Methods)
**Markup Interface**

**Example**

```html
<div lge-type="MsgBox" lge-attr="option:BtnYesNo" id="MsgBoxTest">
    <div>MessageBox Title</div>
    <div>Message Box Contents</div>
</div>
```

**Attributes Summary**

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>lge-attr</td>
<td>Attributes of the LGE component</td>
</tr>
<tr>
<td>option</td>
<td>Type of the button in message box</td>
</tr>
</tbody>
</table>

**option**

**Description**

Type of the button in message box.
There are four types of button. (BtnOK | BtnOKCancel | BtnYesNo | BtnYesNoCancel)

**Example**

```html
<div lge-type="MsgBox" lge-attr="option:BtnYesNo" id="MsgBoxTest">
    <div>MessageBox Title</div>
    <div>Message Box Contents</div>
</div>
```
JS Interface (Constructor Parameters)

Example
```javascript
var msg16 = LGE.UI.MsgBox({
  title:'Default String',
  message:'Default Messages',
  btnType:'BtnYesNoCancel'
});
```

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>btnType</td>
<td>Type of the button in message box</td>
</tr>
<tr>
<td>id</td>
<td>Element ID of the component in document</td>
</tr>
<tr>
<td>message</td>
<td>Contents of message box</td>
</tr>
<tr>
<td>messageRes</td>
<td>Contents of message box, It has key of resource.</td>
</tr>
<tr>
<td>title</td>
<td>Title of message box</td>
</tr>
<tr>
<td>titleRes</td>
<td>Title of message box, It has key of resource.</td>
</tr>
</tbody>
</table>

**btnType**

**Description**
Type of the button in message box.
There are four type of button. (BtnOK | BtnOKCancel | BtnYesNo | BtnYesNoCancel)

Example
```javascript
var msg16 = LGE.UI.MsgBox({
  title:'Default String',
  message:'Default Messages',
  btnType:'BtnYesNoCancel'
});
```

**id**

**Description**
Element ID of the component in document.

Example
```javascript
var msg16 = LGE.UI.MsgBox({
  id:'msgbox1',
  title:'Default String',
  message:'Default Messages',
  btnType:'BtnYesNoCancel'
});
```

**message**

**Description**
Contents of message box
Example
```
var msg16 = LGE.UI.MsgBox({
  title:'Default String',
  message:'Default Messages',
  btnType:'BtnYesNoCancel'
});
```

**messageRes**

**Description**
Contents of message box. The value is key of resource.

Example
```
var msg16 = LGE.UI.MsgBox({
  title:'Default String',
  messageRes:'key1',
  btnType:'BtnYesNoCancel'
});
```

**title**

**Description**
Title of message box

Example
```
var msg16 = LGE.UI.MsgBox({
  title:'Default String',
  message:'Default Messages',
  btnType:'BtnYesNoCancel'
});
```

**titleRes**

**Description**
Title of message box. The value is key of resource.

Example
```
var msg16 = LGE.UI.MsgBox({
  titleRes:'key2',
  message:'Default Messages',
  btnType:'BtnYesNoCancel'
});
```
JS Interface (Methods)

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>addEventListener</td>
<td>Adds the event listener.</td>
</tr>
<tr>
<td>close</td>
<td>Closes the popup.</td>
</tr>
<tr>
<td>doModal</td>
<td>Shows the popup of Modal type.</td>
</tr>
<tr>
<td>getBtnType</td>
<td>Gets the button type of the MsgBox.</td>
</tr>
<tr>
<td>getMessage</td>
<td>Gets the message in the MsgBox.</td>
</tr>
<tr>
<td>getTitle</td>
<td>Gets the title in the MsgBox.</td>
</tr>
<tr>
<td>setBtnType</td>
<td>Sets the button type of the MsgBox.</td>
</tr>
<tr>
<td>setMessage</td>
<td>Sets the message in the MsgBox.</td>
</tr>
<tr>
<td>setMessageRes</td>
<td>Sets the resource message in the MsgBox.</td>
</tr>
<tr>
<td>setTitle</td>
<td>Sets the title in the MsgBox.</td>
</tr>
<tr>
<td>setTitleRes</td>
<td>Sets the resource title in the MsgBox.</td>
</tr>
</tbody>
</table>

**addEventLister**

**Description**
Adds the event listener. There is only “closed” event. When happened “closed” event, it sends Button value as a parameter to target function.

**Parameters**
None

**Return Value**
- **event** (String) The type of event for listening event of the MsgBox. There is “closed” event if MsgBox is closed.
- **target** (String) The target function that processes event.

**Example**
```javascript
var msgBox = LGE.UI.getComp('msgBox');
msgBox.addEventListener('closed', function(aEvent) {
  if(aEvent.btn == 'Yes'|aEvent.btn == 'OK') {
    alert("OK or YES");
  }else{
    alert("others");
  }
});
```

**Supported SDK / Emulator Version**

<table>
<thead>
<tr>
<th>SDK Version</th>
<th>Emulator Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.2 or higher</td>
<td></td>
</tr>
<tr>
<td></td>
<td>LG Smart TV Emulator 2011 : Not Supported</td>
</tr>
<tr>
<td></td>
<td>LG Smart TV Emulator 2012 : SDK 2.2 or higher</td>
</tr>
<tr>
<td></td>
<td>LG Smart TV Emulator 2013 : SDK 3.0 or higher</td>
</tr>
</tbody>
</table>

**close**

**Description**
Closes the MsgBox.
### Parameters
None

### Return Value
None

### Example
```javascript
var msgBox = LGE.UI.getComp('msgBox');
msgBox.close();
```

#### Supported SDK / Emulator Version
<table>
<thead>
<tr>
<th>SDK Version</th>
<th>Emulator Version</th>
</tr>
</thead>
</table>
| 2.2 or higher | LG Smart TV Emulator 2011 : Not Supported  
                      LG Smart TV Emulator 2012 : SDK 2.2 or higher  
                      LG Smart TV Emulator 2013 : SDK 3.0 or higher |

#### doModal

### Description
Shows the popup of Modal type.

### Parameters
None

### Return Value
None

### Example
```javascript
var msgBox = LGE.UI.getComp('msgBox');
msgBox.doModal();
```

#### Supported SDK / Emulator Version
<table>
<thead>
<tr>
<th>SDK Version</th>
<th>Emulator Version</th>
</tr>
</thead>
</table>
| 2.2 or higher | LG Smart TV Emulator 2011 : Not Supported  
                      LG Smart TV Emulator 2012 : SDK 2.2 or higher  
                      LG Smart TV Emulator 2013 : SDK 3.0 or higher |

#### getBtnType

### Description
Gets the button type of the MsgBox.

### Parameters
None

### Return Value
btnType (String) The Button type of the MsgBox.

There are four types of button. (BtnOK | BtnOKCancel | BtnYesNo | BtnYesNoCancel)

### Example
```javascript
var msgBox = LGE.UI.getComp('msgBox');
var btnType = msgBox.getBtnType();
```
### getويرس

**Description**

Gets the message in the MsgBox.

**Parameters**

None

**Return Value**

`(String) The message in the MsgBox`

**Example**

```javascript
var msgBox = LGE.UI.getComp('msgBox');
var msg = msgBox.getMessage();
```

### getTitle

**Description**

Gets the title in the MsgBox.

**Parameters**

None

**Return Value**

`(String) The title of the MsgBox`

**Example**

```javascript
var msgBox = LGE.UI.getComp('msgBox');
var title = msgBox.getTitle();
```

### setBtnType

---

**Supported SDK / Emulator Version**

<table>
<thead>
<tr>
<th>SDK Version</th>
<th>Emulator Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.2 or higher</td>
<td>LG Smart TV Emulator 2011 : Not Supported</td>
</tr>
<tr>
<td></td>
<td>LG Smart TV Emulator 2012 : SDK 2.2 or higher</td>
</tr>
<tr>
<td></td>
<td>LG Smart TV Emulator 2013 : SDK 3.0 or higher</td>
</tr>
</tbody>
</table>

---

**Supported SDK / Emulator Version**

<table>
<thead>
<tr>
<th>SDK Version</th>
<th>Emulator Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.2 or higher</td>
<td>LG Smart TV Emulator 2011 : Not Supported</td>
</tr>
<tr>
<td></td>
<td>LG Smart TV Emulator 2012 : SDK 2.2 or higher</td>
</tr>
<tr>
<td></td>
<td>LG Smart TV Emulator 2013 : SDK 3.0 or higher</td>
</tr>
</tbody>
</table>
Description
Sets the button type of the MsgBox.

Parameters
- **btnType** (String) The Button type of the MsgBox.
  There are four types of button. (BtnOK | BtnOKCancel | BtnYesNo | BtnYesNoCancel)

Return Value
None

Example
```javascript
var msgBox = LGE.UI.getComp('msgBox');
msgBox.setBtnType(BtnOK);
```

---

**Supported SDK / Emulator Version**

<table>
<thead>
<tr>
<th>SDK Version</th>
<th>Emulator Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.2 or higher</td>
<td>LG Smart TV Emulator 2011 : Not Supported</td>
</tr>
<tr>
<td></td>
<td>LG Smart TV Emulator 2012 : SDK 2.2 or higher</td>
</tr>
<tr>
<td></td>
<td>LG Smart TV Emulator 2013 : SDK 3.0 or higher</td>
</tr>
</tbody>
</table>

---

**setMessage**

Description
Sets the message in the MsgBox.

Parameters
- **message** (String) The message in the MsgBox

Return Value
None

Example
```javascript
var msgBox = LGE.UI.getComp('msgBox');
msgBox.setMessage("message contents");
```

---

**Supported SDK / Emulator Version**

<table>
<thead>
<tr>
<th>SDK Version</th>
<th>Emulator Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.2 or higher</td>
<td>LG Smart TV Emulator 2011 : Not Supported</td>
</tr>
<tr>
<td></td>
<td>LG Smart TV Emulator 2012 : SDK 2.2 or higher</td>
</tr>
<tr>
<td></td>
<td>LG Smart TV Emulator 2013 : SDK 3.0 or higher</td>
</tr>
</tbody>
</table>

---

**setMessageRes**

Description
Sets the message in the MsgBox.

Parameters
- **value** (String) Contents of message box. The value is key of resource.

Return Value
None

Example
```javascript
```
var msgBox = LGE.UI.getComp('msgBox');
msgBox.setMessageRes("key1");

**Supported SDK / Emulator Version**

<table>
<thead>
<tr>
<th>SDK Version</th>
<th>Emulator Version</th>
</tr>
</thead>
</table>
| 2.2 or higher | LG Smart TV Emulator 2011 : Not Supported
| | LG Smart TV Emulator 2012 : SDK 2.2 or higher
| | LG Smart TV Emulator 2013 : SDK 3.0 or higher

**setTitle**

**Description**
Sets the title in the MsgBox.

**Parameters**

| title | (String) The title of the MsgBox |

**Return Value**
None

**Example**

var msgBox = LGE.UI.getComp('msgBox');
msgBox.setTitle("title1");

**Supported SDK / Emulator Version**

<table>
<thead>
<tr>
<th>SDK Version</th>
<th>Emulator Version</th>
</tr>
</thead>
</table>
| 2.2 or higher | LG Smart TV Emulator 2011 : Not Supported
| | LG Smart TV Emulator 2012 : SDK 2.2 or higher
| | LG Smart TV Emulator 2013 : SDK 3.0 or higher

**setTitleRes**

**Description**
Sets the title in the MsgBox.

**Parameters**

| title | (String) Title of message box. The value is key of resource. |

**Return Value**
None

**Example**

var msgBox = LGE.UI.getComp('msgBox');
msgBox.setTitleRes("key2");

**Supported SDK / Emulator Version**

<table>
<thead>
<tr>
<th>SDK Version</th>
<th>Emulator Version</th>
</tr>
</thead>
</table>
| 2.2 or higher | LG Smart TV Emulator 2011 : Not Supported
| | LG Smart TV Emulator 2012 : SDK 2.2 or higher
| | LG Smart TV Emulator 2013 : SDK 3.0 or higher

**Picker**
The Picker component is an element that allows the user to choose options of a predefined set.

[Figure] Picker Component

Inheritance Hierarchy
Object >> Component >> Container >> PickerHost
Object >> Component >> Container >> ItemListComponent >> Picker

Markup Interface, JS Interface (Constructor Parameters), and JS Interface (Methods) are listed in the following sections.
• Markup Interface
• JS Interface (Constructor Parameters)
• JS Interface (Methods)
### Markup Interface

**Example**

```html
<div lge-type="PickerHost" lge-attr="option:General" id="picker1" style="width:400px;">
  <select lge-attr="index:1">
    <option> La
ble 1 </option>
    <option> Lable 2 </option>
    <option> Lable 3 </option>
    <option> Lable 4 </option>
  </select>

  <select lge-attr="index:2">
    <option> Lable 1 </option>
    <option> Lable 2 </option>
    <option selected="selected"> Lable 3 </option>
    <option> Lable 4 </option>
  </select>
</div>
```

### Attributes Summary

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>lge-attr</td>
<td>Attributes of the LGE component</td>
</tr>
<tr>
<td>lge-res</td>
<td>Resource text of components</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>index</td>
<td>The index of the Picker in PickerHost</td>
</tr>
<tr>
<td>option</td>
<td>The type of the PickerHost</td>
</tr>
</tbody>
</table>

### index

**Description**

The index of the Picker in PickerHost. This attribute is only for `<Select>` tag.

**Example**

```html
<div lge-type="PickerHost" lge-attr="option:General" id="picker1" style="width:400px;">
  <select lge-attr="index:1">
    <option> Lable 1 </option>
    <option> Lable 2 </option>
    <option> Lable 3 </option>
    <option> Lable 4 </option>
  </select>

  <select lge-attr="index:2">
    <option> Lable 1 </option>
    <option> Lable 2 </option>
    <option selected="selected"> Lable 3 </option>
    <option> Lable 4 </option>
  </select>
</div>
```

### option

**Description**
The type of the PickerHost. It has three types.
- General: It is option for customizing. (default value)
- Date: It is option for date. PickerHost component has three Pickers: year, month, and day.
- Time: It is option for time. PickerHost component has three Pickers: AM/PM, hour, and minutes.

Example

```html
<div lge-type="PickerHost" lge-attr="option:General" id="picker1">
  <select lge-attr="index:1">
    <option>Lable 1</option>
    <option>Lable 2</option>
    <option>Lable 3</option>
  </select>
  <select lge-attr="index:2">
    <option>Lable 1</option>
    <option>Lable 2</option>
    <option selected="selected">Lable 3</option>
  </select>
</div>

<div lge-type="PickerHost" lge-attr="option:Date" id="picker2"></div>

<div lge-type="PickerHost" lge-attr="option:Time" id="picker3"></div>
```
JS Interface (Constructor Parameters)

Example
var picker = LGE.UI.Picker({
    type:'General',
    text:[
        ['Label 1', 'Label 2', 'Label 3'],
        ['Label 1', 'Label 2', 'Label 3']
    ],
    selector:('td2'),
    id:'picker1',
    style:'width:50%' 
});

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>id</td>
<td>Element ID of the component in document</td>
</tr>
<tr>
<td>selector</td>
<td>The element that becomes parent of the component</td>
</tr>
<tr>
<td>text</td>
<td>A list of text list of Pickers</td>
</tr>
<tr>
<td>textures</td>
<td>A list of text or resource list of Pickers</td>
</tr>
<tr>
<td>type</td>
<td>The type of the PickerHost</td>
</tr>
<tr>
<td>width</td>
<td>The width of the PickerHost</td>
</tr>
</tbody>
</table>

id

Description
Element ID of the component in document.

Example
var picker = LGE.UI.Picker({
    selector:('td2'),
    id:'picker',
    type:'General',
    text:[
        ['Label 1', 'Label 2', 'Label 3'],
        ['Label 1', 'Label 2', 'Label 3']
    ],
});

selector

Description
The element that becomes parent of the component
Default is body.

Example
var picker = LGE.UI.Picker({
    selector:('td2'),
    type:'General',
    text:[
        ['Label 1', 'Label 2', 'Label 3'],
        ['Label 1', 'Label 2', 'Label 3']
    ],
});
text

Description
A list of Picker. It consists of only text.

Example
```javascript
var picker = LGE.UI.Picker({
    selector:('td2'),
    type:'General',
    text:[
        ['Label 1', 'Label 2', 'Label 3'],
        ['Label 1', 'Label 2', 'Label 3']
    ],
});
```

textRes

Description
A list of Picker. It consists of text and resource text.

Example
```javascript
var picker = LGE.UI.Picker({
    selector:('td2'),
    type:'General',
    textRes:{
        text:[
            ['Text 1', 'Text 2', 'Text 3'],
        ],
        res:[
            ['Label 1', 'Label 2', 'Label 3'],
        ],
    },
});
```
type

Description
The type of the PickerHost. It has three types.
- General : It is option for customizing. (default value)
- Date : It is option for date. PickerHost component has three Pickers: year, month, and day.
- Time: It is option for time. PickerHost component has three Pickers: AM/PM, hour, and minutes.

Example
```javascript
var picker = LGE.UI.Picker({
    selector:('td2'),
    type:'General',
    text:[
        ['Label 1', 'Label 2', 'Label 3'],
        ['Label 1', 'Label 2', 'Label 3']
    ],
});
```
```javascript
var picker = LGE.UI.Picker({
    selector:('td2'),
    type:'Date',
});
```
width

Description
Width of the PickerHost. The unit is pixel.

Example
```javascript
var picker = LGE.UI.Picker({
  selector: ("td2"),
  type: 'Time',
});
```
JS Interface (Methods)

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>addItem</td>
<td>Adds an Item of the Picker.</td>
</tr>
<tr>
<td>addItemWithRes</td>
<td>Adds an Item of the Picker.</td>
</tr>
<tr>
<td>getOption</td>
<td>Gets option of Picker.</td>
</tr>
<tr>
<td>getSelected</td>
<td>Gets focused option of Picker.</td>
</tr>
<tr>
<td>removeItem</td>
<td>Removes the Item of the Picker.</td>
</tr>
<tr>
<td>setOption</td>
<td>Sets option of the Picker.</td>
</tr>
<tr>
<td>setPickerStyle</td>
<td>Sets the style of the Picker.</td>
</tr>
<tr>
<td>setSelect</td>
<td>Sets focus option of the Picker.</td>
</tr>
</tbody>
</table>

**addItem**

**Description**
Adds an Item of the Picker.

**Parameters**
- pickerId (Number) The index of picker
- text (String) The text of the item
- value (String) The value of the item

**Return Value**
None

**Example**
```javascript
picker.addItem(1, "text1", "value1");
```

**Supported SDK / Emulator Version**

<table>
<thead>
<tr>
<th>SDK Version</th>
<th>Emulator Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.2 or higher</td>
<td>LG Smart TV Emulator 2011 : Not Supported</td>
</tr>
<tr>
<td></td>
<td>LG Smart TV Emulator 2012 : SDK 2.2 or higher</td>
</tr>
<tr>
<td></td>
<td>LG Smart TV Emulator 2013 : SDK 3.0 or higher</td>
</tr>
</tbody>
</table>

**addItemWithRes**

**Description**
Adds an Item of the Picker with resource text.

**Parameters**
- pickerId (Number) The index of picker
- res (String) The resource text of the item
- value (String) The value of the item

**Return Value**
None

**Example**
```javascript
picker.addItemWithRes(1, "on", "value1");
```
### Supported SDK / Emulator Version

<table>
<thead>
<tr>
<th>SDK Version</th>
<th>Emulator Version</th>
</tr>
</thead>
</table>
| 2.2 or higher | LG Smart TV Emulator 2011 : Not Supported  
LG Smart TV Emulator 2012 : SDK 2.2 or higher  
LG Smart TV Emulator 2013 : SDK 3.0 or higher |

---

### getOption

**Description**

Gets the option of the picker.

**Parameters**

None

**Return Value**

- `option` : (String) The type of the picker (general | date | time)

**Example**

```javascript
var pickerType = picker.getOption();
```

---

### getSelected

**Description**

Gets focused option of Picker.

**Parameters**

- `pickerId` : (Number) The index of picker

**Return Value**

- `value` : (String) The value of focused option in Picker.

**Example**

```javascript
var text = picker.getSelected(0);
```

---

### removeItem

**Description**
Remove the Item of the Picker.

**Parameters**

- **pickerId** *(Number)* The index of picker
- **index** *(Number)* The index of the item

**Return Value**

None

**Example**

```javascript
picker.removeItem(0, 1);
```

**Supported SDK / Emulator Version**

<table>
<thead>
<tr>
<th>SDK Version</th>
<th>Emulator Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.2 or higher</td>
<td>LG Smart TV Emulator 2011 : Not Supported</td>
</tr>
<tr>
<td></td>
<td>LG Smart TV Emulator 2012 : SDK 2.2 or higher</td>
</tr>
<tr>
<td></td>
<td>LG Smart TV Emulator 2013 : SDK 3.0 or higher</td>
</tr>
</tbody>
</table>

**setOption**

**Description**

Sets option of the Picker.

**Parameters**

- **option** *(String)* The type of Picker (General | Date | Time)

**Return Value**

None

**Example**

```javascript
picker.setOption("Date");
```

**Supported SDK / Emulator Version**

<table>
<thead>
<tr>
<th>SDK Version</th>
<th>Emulator Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.2 or higher</td>
<td>LG Smart TV Emulator 2011 : Not Supported</td>
</tr>
<tr>
<td></td>
<td>LG Smart TV Emulator 2012 : SDK 2.2 or higher</td>
</tr>
<tr>
<td></td>
<td>LG Smart TV Emulator 2013 : SDK 3.0 or higher</td>
</tr>
</tbody>
</table>

**setPickerStyle**

**Description**

Sets CSS style of the picker.

**Parameters**

- **pickerId** *(Number)* The index of picker
- **style** *(String)* CSS style of the picker

**Return Value**

None

**Example**

```javascript
picker.setPickerStyle(1, "color:red");
```
### Supported SDK / Emulator Version

<table>
<thead>
<tr>
<th>SDK Version</th>
<th>Emulator Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.2 or higher</td>
<td>LG Smart TV Emulator 2011: Not Supported</td>
</tr>
<tr>
<td></td>
<td>LG Smart TV Emulator 2012: SDK 2.2 or higher</td>
</tr>
<tr>
<td></td>
<td>LG Smart TV Emulator 2013: SDK 3.0 or higher</td>
</tr>
</tbody>
</table>

### setSelect

**Description**
Sets focus of the picker.

**Parameters**
- pickerId (Number): The index of picker
- index (Number): The index of the picker item

**Return Value**
None

**Example**
```javascript
picker.setPickerStyle(1, "color:red");
```

### Progress

The Progress component shows the progress of external data loading.

[Figure] Progress Component

### Inheritance Hierarchy

Object >> Component >> Container >> ItemListComponent >> ProgressBar

Markup Interface, JS Interface (Constructor Parameters), and JS Interface (Methods) are listed in the following sections.
- Markup Interface
- JS Interface (Constructor Parameters)
- JS Interface (Methods)
Markup Interface

Example

```html
<input type="range" id='progress1' lge-type="ProgressBar" name="progress" value="1" min="0" max="100" /> 
```

Attributes Summary

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>min</td>
<td>Indicates the allowed range of values for the element.</td>
</tr>
<tr>
<td>max</td>
<td>Indicates the allowed range of values for the element. The default maximum is 100.</td>
</tr>
<tr>
<td>value</td>
<td>Gives the default value of the input element.</td>
</tr>
</tbody>
</table>
JS Interface (Constructor Parameters)

**Example**
```javascript
var progress1 = LGE.UI.Progress({
  id:'progress1',
  max:'100',
  min:'0',
  selector:('td2'),
  style:'width:100%;height:100%','
  value:'10',
});
```

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>id</td>
<td>Element ID of the component in document</td>
</tr>
<tr>
<td>max</td>
<td>Maximum value of the component</td>
</tr>
<tr>
<td>min</td>
<td>Minimum value of the component</td>
</tr>
<tr>
<td>style</td>
<td>CSS style of the component</td>
</tr>
<tr>
<td>selector</td>
<td>The element that becomes parent of the component</td>
</tr>
<tr>
<td>value</td>
<td>Value of the component</td>
</tr>
</tbody>
</table>

**id**

**Description**
Element ID of the component in document.

**Example**
```javascript
var progress1 = LGE.UI.Progress({
  id:'progress1'
  max:'100',
  min:'0',
  selector:('td2'),
  value:'10',
});
```

**max**

**Description**
Maximum value of the slider can have.

**Example**
```javascript
var progress1 = LGE.UI.Progress({
  id:'progress1'
  max:'100',
  min:'0',
  selector:('td2'),
  value:'10',
});
```

**min**

**Description**
Minimum value of the component.
Description
Minimum value of the slider can have.

Example
var progress1 = LGE.UI.Progress({
    id:'progress1'
    min:'0',
    max:'100',
    selector:('td2'),
    value:'10',
});

style

Description
CSS style of the component

Example
var progress1 = LGE.UI.Progress({
    id:'progress1'
    max:'100',
    min:'0',
    selector:('td2'),
    style:'width:100%;height:100%',
    value:'10',
});

selector

Description
The element that becomes parent of the component
Default is body.

Example
var progress1 = LGE.UI.Progress({
    id:'progress1'
    max:'100',
    min:'0',
    selector:('td2'),
    value:'10',
});

value

Description
Value of the component. The value of the component must be biggerer than the min value and less than max value.

Example
var progress1 = LGE.UI.Progress({
    id:'progress1'
    max:'100',
    min:'0',
    selector:('td2'),
    value:'10',
});
value:'10',
});
<table>
<thead>
<tr>
<th>JS Interface (Methods)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Method</td>
</tr>
<tr>
<td>getMax</td>
</tr>
<tr>
<td>getMin</td>
</tr>
<tr>
<td>getStep</td>
</tr>
<tr>
<td>getValue</td>
</tr>
<tr>
<td>resetProgress</td>
</tr>
<tr>
<td>setMax</td>
</tr>
<tr>
<td>setMin</td>
</tr>
<tr>
<td>setStep</td>
</tr>
<tr>
<td>setValue</td>
</tr>
<tr>
<td>startProgress</td>
</tr>
<tr>
<td>stopProgress</td>
</tr>
</tbody>
</table>

**getMax**

**Description**

Gets the max value of the Progress.

**Parameters**

None

**Return Value**

max (Number) The max value of the Progress

**Example**

```javascript
var max = progress.getMax();
```

**Supported SDK / Emulator Version**

<table>
<thead>
<tr>
<th>SDK Version</th>
<th>Emulator Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.2 or higher</td>
<td>LG Smart TV Emulator 2011 : Not Supported</td>
</tr>
<tr>
<td></td>
<td>LG Smart TV Emulator 2012 : SDK 2.2 or higher</td>
</tr>
<tr>
<td></td>
<td>LG Smart TV Emulator 2013 : SDK 3.0 or higher</td>
</tr>
</tbody>
</table>

**getMin**

**Description**

Gets the min value of Progress.

**Parameters**

None

**Return Value**

min (Number) The min value of the Progress

**Example**
```javascript
var max = progress.getMin();
```

## Supported SDK / Emulator Version

<table>
<thead>
<tr>
<th>SDK Version</th>
<th>Emulator Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.2 or higher</td>
<td>LG Smart TV Emulator 2011: Not Supported</td>
</tr>
<tr>
<td></td>
<td>LG Smart TV Emulator 2012: SDK 2.2 or higher</td>
</tr>
<tr>
<td></td>
<td>LG Smart TV Emulator 2013: SDK 3.0 or higher</td>
</tr>
</tbody>
</table>

### getStep

**Description**
Gets the step value of the Progress.

**Parameters**
None

**Return Value**

step (Number) The step value of the Progress

**Example**

```javascript
var step = progress.getStep();
```

## Supported SDK / Emulator Version

<table>
<thead>
<tr>
<th>SDK Version</th>
<th>Emulator Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.2 or higher</td>
<td>LG Smart TV Emulator 2011: Not Supported</td>
</tr>
<tr>
<td></td>
<td>LG Smart TV Emulator 2012: SDK 2.2 or higher</td>
</tr>
<tr>
<td></td>
<td>LG Smart TV Emulator 2013: SDK 3.0 or higher</td>
</tr>
</tbody>
</table>

### getValue

**Description**
Gets the value of the progress.

**Parameters**
None

**Return Value**

value (Number) The value of the progress

**Example**

```javascript
var value = progress.getValue();
```

## Supported SDK / Emulator Version

<table>
<thead>
<tr>
<th>SDK Version</th>
<th>Emulator Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.2 or higher</td>
<td>LG Smart TV Emulator 2011: Not Supported</td>
</tr>
<tr>
<td></td>
<td>LG Smart TV Emulator 2012: SDK 2.2 or higher</td>
</tr>
<tr>
<td></td>
<td>LG Smart TV Emulator 2013: SDK 3.0 or higher</td>
</tr>
</tbody>
</table>

### resetProgress

**Description**
Resets the progress.

**Parameters**
None

**Return Value**
None

**Example**
```
var progress = LGE.UI.getComp('progress');
$('#reset_btn').click(function(){
    progress.resetProgress();
});
```

**Supported SDK / Emulator Version**

<table>
<thead>
<tr>
<th>SDK Version</th>
<th>Emulator Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.2 or higher</td>
<td>LG Smart TV Emulator 2011 : Not Supported</td>
</tr>
<tr>
<td></td>
<td>LG Smart TV Emulator 2012 : SDK 2.2 or higher</td>
</tr>
<tr>
<td></td>
<td>LG Smart TV Emulator 2013 : SDK 3.0 or higher</td>
</tr>
</tbody>
</table>

### setMax

**Description**
Sets the max value of the Progress.

**Parameters**
- **max** (Number) The max value of the Progress

**Return Value**
None

**Example**
```
var progress = LGE.UI.getComp('progress');
progress.setMax(10);
```

**Supported SDK / Emulator Version**

<table>
<thead>
<tr>
<th>SDK Version</th>
<th>Emulator Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.2 or higher</td>
<td>LG Smart TV Emulator 2011 : Not Supported</td>
</tr>
<tr>
<td></td>
<td>LG Smart TV Emulator 2012 : SDK 2.2 or higher</td>
</tr>
<tr>
<td></td>
<td>LG Smart TV Emulator 2013 : SDK 3.0 or higher</td>
</tr>
</tbody>
</table>

### setMin

**Description**
Sets the min value of the Progress.

**Parameters**
- **min** (Number) The min value of the Progress

**Return Value**
None

**Example**
var progress = LGE.UI.getComp('progress');
progress.setMin(-10);

### Supported SDK / Emulator Version

<table>
<thead>
<tr>
<th>SDK Version</th>
<th>Emulator Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.2 or higher</td>
<td>LG Smart TV Emulator 2011 : Not Supported</td>
</tr>
<tr>
<td></td>
<td>LG Smart TV Emulator 2012 : SDK 2.2 or higher</td>
</tr>
<tr>
<td></td>
<td>LG Smart TV Emulator 2013 : SDK 3.0 or higher</td>
</tr>
</tbody>
</table>

#### setStep

**Description**
Sets the step value of the Progress.

**Parameters**
- `step` (Number) The step value of the Progress

**Return Value**
None

**Example**
```javascript
var progress = LGE.UI.getComp('progress');
progress.setStep(1);
```

### Supported SDK / Emulator Version

<table>
<thead>
<tr>
<th>SDK Version</th>
<th>Emulator Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.2 or higher</td>
<td>LG Smart TV Emulator 2011 : Not Supported</td>
</tr>
<tr>
<td></td>
<td>LG Smart TV Emulator 2012 : SDK 2.2 or higher</td>
</tr>
<tr>
<td></td>
<td>LG Smart TV Emulator 2013 : SDK 3.0 or higher</td>
</tr>
</tbody>
</table>

#### setValue

**Description**
Sets the value of the Progress.

**Parameters**
- `value` (Number) The value of the progress

**Return Value**
None

**Example**
```javascript
var progress = LGE.UI.getComp('progress');
progress.setValue(0);
```

### Supported SDK / Emulator Version

<table>
<thead>
<tr>
<th>SDK Version</th>
<th>Emulator Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.2 or higher</td>
<td>LG Smart TV Emulator 2011 : Not Supported</td>
</tr>
<tr>
<td></td>
<td>LG Smart TV Emulator 2012 : SDK 2.2 or higher</td>
</tr>
<tr>
<td></td>
<td>LG Smart TV Emulator 2013 : SDK 3.0 or higher</td>
</tr>
</tbody>
</table>
startProgress

**Description**
Starts the progress.

**Parameters**
None

**Return Value**
None

**Example**
```javascript
var progress = LGE.UI.getComp('progress');
$('#start_btn').click(function(){
    progress.startProgress();
});
```

**Supported SDK / Emulator Version**

<table>
<thead>
<tr>
<th>SDK Version</th>
<th>Emulator Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.2 or higher</td>
<td>LG Smart TV Emulator 2011 : Not Supported</td>
</tr>
<tr>
<td></td>
<td>LG Smart TV Emulator 2012 : SDK 2.2 or higher</td>
</tr>
<tr>
<td></td>
<td>LG Smart TV Emulator 2013 : SDK 3.0 or higher</td>
</tr>
</tbody>
</table>

stopProgress

**Description**
Stops the progress.

**Parameters**
None

**Return Value**
None

**Example**
```javascript
var progress = LGE.UI.getComp('progress');
$('#stop_btn').click(function(){
    progress.stopProgress();
});
```

**Supported SDK / Emulator Version**

<table>
<thead>
<tr>
<th>SDK Version</th>
<th>Emulator Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.2 or higher</td>
<td>LG Smart TV Emulator 2011 : Not Supported</td>
</tr>
<tr>
<td></td>
<td>LG Smart TV Emulator 2012 : SDK 2.2 or higher</td>
</tr>
<tr>
<td></td>
<td>LG Smart TV Emulator 2013 : SDK 3.0 or higher</td>
</tr>
</tbody>
</table>

Radio

The **Radio** component is an element that allows the user to choose only one of a predefined set of options.

![Radio Component](image)
Inheritance Hierarchy
Object >> Component >> Container >> ItemListComponent >> CheckBoxRadioList >> Radio

Markup Interface, JS Interface (Constructor Parameters), and JS Interface (Methods) are listed in the following sections.
• Markup Interface
• JS Interface (Constructor Parameters)
• JS Interface (Methods)
Markup Interface

Example
<input type="radio" lge-type="Radio" checked />

Attributes Summary

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>lge-attr</td>
<td>Attributes of the LGE component</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>text-inline</td>
<td>More compact component that is only as wide as the text inside.</td>
</tr>
</tbody>
</table>

text-inline

Description
More compact component that is only as wide as the text inside.
Default value is false. (true | false)

Example
<input type="radio" lge-type="Radio" checked lge-attr='text-inline: true' value='text'/>
JS Interface (Constructor Parameters)

Example

```javascript
var Radio1 = LGE.UI.Radio(
  selector:("td2"),
  checked:'true',
  id:'radio1',
  text:'Basic radio1',
});
```

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>checked</td>
<td>Whether the Radio component should be checked or not</td>
</tr>
<tr>
<td>id</td>
<td>Element ID of the component in document</td>
</tr>
<tr>
<td>selector</td>
<td>The element that becomes parent of the component</td>
</tr>
<tr>
<td>style</td>
<td>CSS style of the component</td>
</tr>
<tr>
<td>text</td>
<td>Text in the component</td>
</tr>
<tr>
<td>textInline</td>
<td>More compact component that is only as wide as the text inside</td>
</tr>
</tbody>
</table>

**checked**

**Description**

Whether the Radio component should be checked or not

Default value is false. (true | false)

**Example**

```javascript
var Radio1 = LGE.UI.Radio(
  selector:("td2"),
  checked:'true',
  id:'radio1',
  text:'Basic radio1',
});
```

**id**

**Description**

Element ID of the component in document.

**Example**

```javascript
var Radio1 = LGE.UI.Radio(
  selector:("td2"),
  id:'radio1',
  text:'Basic radio1',
});
```

**selector**

**Description**

The element that becomes parent of the component

Default is body.
Example

```javascript
var Radio1 = LGE.UI.Radio({
  selector:("td2"),
  checked:'true',
  id:'radio1',
  text:'Basic radio1',
  textInline:'true'
});
```

**style**

Description
CSS style of the component

Example

```javascript
var Radio1 = LGE.UI.Radio({
  selector:("td2"),
  id:'radio1',
  text:'Basic radio1',
  style:'width:200px',
});
```

**text**

Description
Text of the component

Example

```javascript
var Radio1 = LGE.UI.Radio({
  selector:("td2"),
  checked:'true',
  id:'radio1',
  text:'Basic radio1',
  textInline:'true'
});
```

**textInline**

Description
More compact component that is only as wide as the text inside.
Default value is false. (true | false)

Example

```javascript
var Radio1 = LGE.UI.Radio({
  selector:("td2"),
  id:'radio1',
  text:'Basic radio1',
  textInline:'true'
});
```
JS Interface (Methods)

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>getCheck</td>
<td>Gets check state of the Radio.</td>
</tr>
<tr>
<td>getText</td>
<td>Gets the text of the Radio.</td>
</tr>
<tr>
<td>getTextInline</td>
<td>Gets the text-inline of the Radio.</td>
</tr>
<tr>
<td>setCheck</td>
<td>Sets check state of the Radio.</td>
</tr>
<tr>
<td>setText</td>
<td>Sets the text of the Radio.</td>
</tr>
<tr>
<td>setTextInline</td>
<td>Sets the text-inline of the Radio.</td>
</tr>
</tbody>
</table>

**getCheck**

Description
Gets check state of the Radio.

Parameters
None

Return Value
`bool` (Boolean) The check state of the Radio

Example
```
var checked = radio1.getCheck();
```

**Supported SDK / Emulator Version**

<table>
<thead>
<tr>
<th>SDK Version</th>
<th>Emulator Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.2 or higher</td>
<td>LG Smart TV Emulator 2011 : Not Supported</td>
</tr>
<tr>
<td></td>
<td>LG Smart TV Emulator 2012 : SDK 2.2 or higher</td>
</tr>
<tr>
<td></td>
<td>LG Smart TV Emulator 2013 : SDK 3.0 or higher</td>
</tr>
</tbody>
</table>

**getText**

Description
Gets the text of the Radio.

Parameters
None

Return Value
`text` (String) The text of Radio

Example
```
var text = radio1.getText();
```

**Supported SDK / Emulator Version**

<table>
<thead>
<tr>
<th>SDK Version</th>
<th>Emulator Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.2 or higher</td>
<td>LG Smart TV Emulator 2011 : Not Supported</td>
</tr>
<tr>
<td></td>
<td>LG Smart TV Emulator 2012 : SDK 2.2 or higher</td>
</tr>
<tr>
<td></td>
<td>LG Smart TV Emulator 2013 : SDK 3.0 or higher</td>
</tr>
</tbody>
</table>
**getTextInline**

**Description**
Gets the text-inline of the Radio.

**Parameters**
None

**Return Value**
bool (Boolean) The text-inline of the Radio

**Example**
```
var textinline = radio1.getTextInline();
```

**Supported SDK / Emulator Version**

<table>
<thead>
<tr>
<th>SDK Version</th>
<th>Emulator Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.2 or higher</td>
<td>LG Smart TV Emulator 2011: Not Supported</td>
</tr>
<tr>
<td></td>
<td>LG Smart TV Emulator 2012: SDK 2.2 or higher</td>
</tr>
<tr>
<td></td>
<td>LG Smart TV Emulator 2013: SDK 3.0 or higher</td>
</tr>
</tbody>
</table>

**setCheck**

**Description**
Sets whether the state of the Radio should be checked or not.

**Parameters**
bool (Boolean) The check state of the Radio

**Return Value**
None

**Example**
```
radio1.setCheck(true);
```

**Supported SDK / Emulator Version**

<table>
<thead>
<tr>
<th>SDK Version</th>
<th>Emulator Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.2 or higher</td>
<td>LG Smart TV Emulator 2011: Not Supported</td>
</tr>
<tr>
<td></td>
<td>LG Smart TV Emulator 2012: SDK 2.2 or higher</td>
</tr>
<tr>
<td></td>
<td>LG Smart TV Emulator 2013: SDK 3.0 or higher</td>
</tr>
</tbody>
</table>

**setText**

**Description**
Sets the text of the Radio.

**Parameters**

| text (String) The text of Radio |

**Return Value**
None

**Example**
```
radio1.setText(“Radio Text”);
```
**Supported SDK / Emulator Version**

<table>
<thead>
<tr>
<th>SDK Version</th>
<th>Emulator Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.2 or higher</td>
<td>LG Smart TV Emulator 2011 : Not Supported</td>
</tr>
<tr>
<td></td>
<td>LG Smart TV Emulator 2012 : SDK 2.2 or higher</td>
</tr>
<tr>
<td></td>
<td>LG Smart TV Emulator 2013 : SDK 3.0 or higher</td>
</tr>
</tbody>
</table>

**setTextInline**

**Description**
Sets the text-inline of the Radio.

**Parameters**

```java
bool (Boolean) The text-inline of the Radio
```

**Return Value**
None

**Example**

```java
radio1.setTextInline(true);
```

**Supported SDK / Emulator Version**

<table>
<thead>
<tr>
<th>SDK Version</th>
<th>Emulator Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.2 or higher</td>
<td>LG Smart TV Emulator 2011 : Not Supported</td>
</tr>
<tr>
<td></td>
<td>LG Smart TV Emulator 2012 : SDK 2.2 or higher</td>
</tr>
<tr>
<td></td>
<td>LG Smart TV Emulator 2013 : SDK 3.0 or higher</td>
</tr>
</tbody>
</table>

**Rating**

The Rating component is used for rating something.

![Rating Component](image)

**[Figure] Rating Component**

**Inheritance Hierarchy**
Object >> Component >> Container >> Rating

Markup Interface, JS Interface (Constructor Parameters), and JS Interface (Methods) are listed in the following sections.

- [Markup Interface](#)
- [JS Interface (Constructor Parameters)](#)
- [JS Interface (Methods)](#)
## Markup Interface

### Example

```html
<div lge-type="Rating" id="rating1">
  <input type="range" min="0" max="100"/>
</div>
```

### Attributes Summary

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>lge-attr</td>
<td>Attributes of the LGE component</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Ige-attr</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>N/A</td>
<td>N/A</td>
</tr>
</tbody>
</table>
JS Interface (Constructor Parameters)

Example
```javascript
var ts7 = LGE.UI.Rating({
    min: 0,
    max: 10,
    value: 6,
    selector: "td2"
});
```

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>id</td>
<td>Element ID of the component in document</td>
</tr>
<tr>
<td>max</td>
<td>Max value of the component</td>
</tr>
<tr>
<td>min</td>
<td>Min value of the component</td>
</tr>
<tr>
<td>selector</td>
<td>The element that becomes parent of the component</td>
</tr>
<tr>
<td>style</td>
<td>CSS style of the component</td>
</tr>
<tr>
<td>value</td>
<td>Value of the component</td>
</tr>
</tbody>
</table>

**id**

Description
Element ID of the component in document.

Example
```javascript
var rating = LGE.UI.Rating({
    id: "rating1",
    selector: "td2"
});
```

**max**

Description
Maximum value of the Rating component
Default value is 100.

Example
```javascript
var ts8 = LGE.UI.Rating({
    id: "rating1",
    max: 100,
    selector: "td2"
});
```

**min**

Description
Minimum value of the Rating component.
Default value is 0.
Example

```javascript
var ts8 = LGE.UI.Rating({
  id:"rating1",
  max:100,
  min:0,
  selector:("td2")
});
```

**selector**

**Description**
The element that becomes parent of the component
Default is body.

**Example**

```javascript
var ts8 = LGE.UI.Rating({
  id:"rating1",
  selector:("td2")
});
```

**style**

**Description**
CSS style of the component

**Example**

```javascript
var ts8 = LGE.UI.Rating({
  id:"rating1",
  selector:("td2"),
  style:"width:500px",
});
```

**value**

**Description**
Value of the Rating component. It is bigger than the min value and less than max value.
Full star value has (Min-Max)/5.
For example, if max is 100, min is 50, and 3 stars and half, the value is 85.

**Example**

```javascript
var ts8 = LGE.UI.Rating({
  id:"rating1",
  selector:("td2"),
  value:0,
});
```
## JS Interface (Methods)

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>getMax</td>
<td>Gets the max value of the Rating.</td>
</tr>
<tr>
<td>getMin</td>
<td>Gets the min value of the Rating.</td>
</tr>
<tr>
<td>getValue</td>
<td>Gets the value of the Rating.</td>
</tr>
<tr>
<td>setMax</td>
<td>Sets the max value of the Rating.</td>
</tr>
<tr>
<td>setMin</td>
<td>Sets the min value of the Rating.</td>
</tr>
<tr>
<td>setValue</td>
<td>Sets the value of the Rating.</td>
</tr>
</tbody>
</table>

### getMax

**Description**

Gets the max value of the Rating.

**Parameters**

None

**Return Value**

\[ max \text{ (Number)} \text{ the max value of the Rating} \]

**Example**

```javascript
var rating = LGE.UI.getComp('rating');
var max = rating.getMax();
```

#### Supported SDK / Emulator Version

<table>
<thead>
<tr>
<th>SDK Version</th>
<th>Emulator Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.2 or higher</td>
<td>LG Smart TV Emulator 2011 : Not Supported</td>
</tr>
<tr>
<td></td>
<td>LG Smart TV Emulator 2012 : SDK 2.2 or higher</td>
</tr>
<tr>
<td></td>
<td>LG Smart TV Emulator 2013 : SDK 3.0 or higher</td>
</tr>
</tbody>
</table>

### getMin

**Description**

Gets the min value of Rating.

**Parameters**

None

**Return Value**

\[ min \text{ (Number)} \text{ The min value of the Rating} \]

**Example**

```javascript
var rating = LGE.UI.getComp('rating');
var min = rating.getMin();
```

#### Supported SDK / Emulator Version

<table>
<thead>
<tr>
<th>SDK Version</th>
<th>Emulator Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.2 or higher</td>
<td>LG Smart TV Emulator 2011 : Not Supported</td>
</tr>
<tr>
<td></td>
<td>LG Smart TV Emulator 2012 : SDK 2.2 or higher</td>
</tr>
</tbody>
</table>
getValue

Description
Gets the value of the Rating.

Parameters
None

Return Value

value (Number) The value of the Rating

Example

```js
var rating = LGE.UI.getComp('rating');
var max = rating.getValue();
```

Supported SDK / Emulator Version

<table>
<thead>
<tr>
<th>SDK Version</th>
<th>Emulator Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.2 or higher</td>
<td>LG Smart TV Emulator 2011 : Not Supported</td>
</tr>
<tr>
<td></td>
<td>LG Smart TV Emulator 2012 : SDK 2.2 or higher</td>
</tr>
<tr>
<td></td>
<td>LG Smart TV Emulator 2013 : SDK 3.0 or higher</td>
</tr>
</tbody>
</table>

setMax

Description
Sets the max value of the Rating.

Parameters

max (Number) The max value of the Rating

Return Value
None

Example

```js
var rating = LGE.UI.getComp('rating');
rating.setMax(10);
```

Supported SDK / Emulator Version

<table>
<thead>
<tr>
<th>SDK Version</th>
<th>Emulator Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.2 or higher</td>
<td>LG Smart TV Emulator 2011 : Not Supported</td>
</tr>
<tr>
<td></td>
<td>LG Smart TV Emulator 2012 : SDK 2.2 or higher</td>
</tr>
<tr>
<td></td>
<td>LG Smart TV Emulator 2013 : SDK 3.0 or higher</td>
</tr>
</tbody>
</table>

setMin

Description
Sets the min value of the Rating.
Parameters

min (Number) The min value of the Rating

Return Value

None

Example

```javascript
var rating = LGE.UI.getComp('rating');
rating.setMin(-10);
```

Supported SDK / Emulator Version

<table>
<thead>
<tr>
<th>SDK Version</th>
<th>Emulator Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.2 or higher</td>
<td>LG Smart TV Emulator 2011 : Not Supported</td>
</tr>
<tr>
<td></td>
<td>LG Smart TV Emulator 2012 : SDK 2.2 or higher</td>
</tr>
<tr>
<td></td>
<td>LG Smart TV Emulator 2013 : SDK 3.0 or higher</td>
</tr>
</tbody>
</table>

setValue

Description

Sets the value of the Rating.

Parameters

value (Number) The value of the Rating

Return Value

None

Example

```javascript
var rating = LGE.UI.getComp('rating');
rating.setValue(0);
```

Supported SDK / Emulator Version

<table>
<thead>
<tr>
<th>SDK Version</th>
<th>Emulator Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.2 or higher</td>
<td>LG Smart TV Emulator 2011 : Not Supported</td>
</tr>
<tr>
<td></td>
<td>LG Smart TV Emulator 2012 : SDK 2.2 or higher</td>
</tr>
<tr>
<td></td>
<td>LG Smart TV Emulator 2013 : SDK 3.0 or higher</td>
</tr>
</tbody>
</table>

Scroller

The Scroller component is scrollable container.
Inheritance Hierarchy
Object >> Component >> Container >> Scroller

Markup Interface, JS Interface (Constructor Parameters), and JS Interface (Methods) are listed in the following sections.
- Markup Interface
- JS Interface (Constructor Parameters)
- JS Interface (Methods)
Markup Interface

Example

```html
<div lge-type=" Scroller" lge-attr="img-scroll-direction:Both" id="Scroll1">
...
</div>
```

Attributes Summary

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>lge-attr</td>
<td>Attributes of the LGE component</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>img-scroll-direction</td>
<td>Direction of scroll</td>
</tr>
</tbody>
</table>

**img-scroll-direction**

**Description**
Direction of scroll
Default is Vertical. (Vertical | Horizontal | Both)

**Example**

```html
<div lge-type=" Scroller" lge-attr="img-scroll-direction:Both">
...
</div>
```
JS Interface (Constructor Parameters)

Example
var ts6 = LGE.UI.Scroller({
    id:'ScrollerTest',
    imgScrollDirection:'Both',
    selector:('td2'),
});
ts6.addContents("...");

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>id</td>
<td>Element ID of the component in document</td>
</tr>
<tr>
<td>imgScrollDirection</td>
<td>Direction of scroll</td>
</tr>
<tr>
<td>selector</td>
<td>The element that becomes parent of the component</td>
</tr>
<tr>
<td>style</td>
<td>CSS style of the component</td>
</tr>
</tbody>
</table>

id

Description
Element ID of the component in document.

Example
var ts6 = LGE.UI.Scroller({
    id:'ScrollerTest',
    imgScrollDirection:'Both',
    selector:('td2'),
});
ts6.addContents("...");

imgScrollDirection

Description
Direction of scroll
Default is Vertical. (Vertical | Horizontal | Both)

Example
var ts6 = LGE.UI.Scroller({
    id:'ScrollerTest',
    imgScrollDirection:'Both',
    selector:('td2'),
});
ts6.addContents("...");

selector

Description
The element that becomes parent of the component
Default is body.
Example

```javascript
var ts6 = LGE.UI.Scroller({
    id: 'ScrollerTest',
    imgScrollDirection: 'Both',
    selector: ('td2'),
});
ts6.addContents("...");
```

**style**

**Description**
CSS style of the component

**Example**

```javascript
var ts6 = LGE.UI.Scroller({
    id: 'ScrollerTest',
    imgScrollDirection: 'Both',
    selector: ('td2'),
    style: "width:100%",
});
ts6.addContents("...");
```
# JS Interface (Methods)

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>addContents</td>
<td>Adds contents in the Scroller.</td>
</tr>
<tr>
<td>getScrollDirection</td>
<td>Gets the scroll direction in the Scroller.</td>
</tr>
<tr>
<td>getScrollLock</td>
<td>Returns true if scroll is locked.</td>
</tr>
<tr>
<td>setScrollDirection</td>
<td>Sets the scroll direction in the Scroller.</td>
</tr>
<tr>
<td>setScrollLock</td>
<td>Locks the scroll.</td>
</tr>
</tbody>
</table>

## addContents

**Description**
Adds contents in the Scroller.

**Parameters**
- `contents` (String or Object) Contents in the Scroller

**Return Value**
None

**Example**
```javascript
var scroller = LGE.UI.getComp('scroller');
scroller.addContents("add contents");
scroller.addContents($("#otherElement");
scroller.addContents(document.createTextNode("Hello"));
```

**Supported SDK / Emulator Version**

<table>
<thead>
<tr>
<th>SDK Version</th>
<th>Emulator Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.2 or higher</td>
<td></td>
</tr>
<tr>
<td></td>
<td>LG Smart TV Emulator 2011 : Not Supported</td>
</tr>
<tr>
<td></td>
<td>LG Smart TV Emulator 2012 : SDK 2.2 or higher</td>
</tr>
<tr>
<td></td>
<td>LG Smart TV Emulator 2013 : SDK 3.0 or higher</td>
</tr>
</tbody>
</table>

## getScrollDirection

**Description**
Gets the scroll direction in the Scroller.

**Parameters**
None

**Return Value**
- `scrollDirection` (String) The scroll direction (Horizontal | Vertical | Both)

**Example**
```javascript
var scroller = LGE.UI.getComp('scroller');
var direction = scroller.getScrollDirection();
```

**Supported SDK / Emulator Version**

<table>
<thead>
<tr>
<th>SDK Version</th>
<th>Emulator Version</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>SDK Version</td>
<td>Emulator Version</td>
</tr>
<tr>
<td>-------------</td>
<td>------------------</td>
</tr>
</tbody>
</table>
| 2.2 or higher | LG Smart TV Emulator 2011 : Not Supported  
LG Smart TV Emulator 2012 : SDK 2.2 or higher  
LG Smart TV Emulator 2013 : SDK 3.0 or higher |

**getScrollLock**

**Description**

Returns true if scroll is locked.

**Parameters**

None

**Return Value**

`lock` (Boolean) Whether the scroll is locked or not

**Example**

```javascript
var scroller = LGE.UI.getComp('scroller');
if(scroller.getScrollLock())
    scroller.setScrollLock(false);
```

**setScrollDirection**

**Description**

Sets the scroll direction in the Scroller.

**Parameters**

`scrollDirection` (String) The scroll direction (Horizontal | Vertical | Both)

**Return Value**

None

**Example**

```javascript
var scroller = LGE.UI.getComp('scroller');
scroller.setScrollDirection("Horizontal");
```

**setScrollLock**

**Description**

Locks the scroll.
Parameters

lock (String) Whether the scroll is locked or not (On | Off)

Return Value
None

Example

```javascript
var scroller = LGE.UI.getComp('scroller');
if(scroller.getScrollLock() == 'On')
    scroller.setScrollLock('Off');
```

Supported SDK / Emulator Version

<table>
<thead>
<tr>
<th>SDK Version</th>
<th>Emulator Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.2 or higher</td>
<td>LG Smart TV Emulator 2011: Not Supported</td>
</tr>
<tr>
<td></td>
<td>LG Smart TV Emulator 2012: SDK 2.2 or higher</td>
</tr>
<tr>
<td></td>
<td>LG Smart TV Emulator 2013: SDK 3.0 or higher</td>
</tr>
</tbody>
</table>

Slider

The Slider is used for input fields that should contain a value from a range of numbers.

[Figure] Slider Component

Inheritance Hierarchy

Object >> Component >> Container >> ItemListComponent >> Slider

Markup Interface, JS Interface (Constructor Parameters), and JS Interface (Methods) are listed in the following sections.

- Markup Interface
- JS Interface (Constructor Parameters)
- JS Interface (Methods)
Markup Interface

Example
<input type="range" lge-type='Slider' value="0" min="0" max="100">

Attributes Summary

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>min</td>
<td>Indicates the allowed range of values for the element.</td>
</tr>
<tr>
<td>max</td>
<td>Indicates the allowed range of values for the element. The default maximum is 100.</td>
</tr>
<tr>
<td>value</td>
<td>Gives the default value of the input element.</td>
</tr>
</tbody>
</table>
JS Interface (Constructor Parameters)

Example
var slider1 = LGE.UI.Slider({
    selector:("id1"),
    value:'0',
    min:'0',
    max:'100',
    id:'slider1',
    style:'width:500px',
});

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>id</td>
<td>Element ID of the component in document</td>
</tr>
<tr>
<td>max</td>
<td>Maximum value of the component</td>
</tr>
<tr>
<td>min</td>
<td>Minimum value of the component</td>
</tr>
<tr>
<td>style</td>
<td>CSS style of the component</td>
</tr>
<tr>
<td>selector</td>
<td>The element that becomes parent of the component</td>
</tr>
<tr>
<td>value</td>
<td>Value of the component</td>
</tr>
</tbody>
</table>

id

Description
Element ID of the component in document.

Example
var slider1 = LGE.UI.Slider({
    id:'slider1',
    selector:("id1"),
    value:'0',
    min:'0',
    max:'100',
});

max

Description
Maximum value of the slider can have.

Example
var slider1 = LGE.UI.Slider({
    id:'slider1',
    selector:("id1"),
    value:'0',
    min:'0',
    max:'100',
});
min

Description
Minimum value of the slider can have.

Example
var slider1 = LGE.UI.Slider({
    id: 'slider1',
    selector: ('id1'),
    value: '0',
    min: '0',
    max: '100',
});

style

Description
CSS style of the component

Example
var slider1 = LGE.UI.Slider({
    id: 'slider1',
    selector: ('id1'),
    value: '0',
    min: '0',
    max: '100',
    style: 'width:500px',
 });

selector

Description
The element that becomes parent of the component
Default is body.

Example
var slider1 = LGE.UI.Slider({
    id: 'slider1',
    selector: ('id1'),
    value: '0',
    min: '0',
    max: '100',
    style: 'width:500px',
});

value

Description
Value of the component. The value of the component must be bigger than the min value and less than the max value.

Example
var slider1 = LGE.UI.Slider({
    id: 'slider1',
});
selector: ("id1"),
    value: '10',
    min: '0',
    max: '100',
    style: 'width:500px',
});
### JS Interface (Methods)

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>getMax</td>
<td>Gets the maximum value of the Slider.</td>
</tr>
<tr>
<td>getMin</td>
<td>Gets the minimum value of the Slider.</td>
</tr>
<tr>
<td>getValue</td>
<td>Gets the positioning value of the Slider.</td>
</tr>
<tr>
<td>setMax</td>
<td>Sets the maximum value of the Slider.</td>
</tr>
<tr>
<td>setMin</td>
<td>Sets the minimum value of the Slider.</td>
</tr>
<tr>
<td>setValue</td>
<td>Sets the positioning value of the Slider.</td>
</tr>
</tbody>
</table>

#### getMax

**Description**

Gets the max value of the Slider.

**Parameters**

None

**Return Value**

max (Number) The max value of the Slider

**Example**

```javascript
var slider = LGE.UI.getComp('slider');
var max = slider.getMax();
```

**Supported SDK / Emulator Version**

<table>
<thead>
<tr>
<th>SDK Version</th>
<th>Emulator Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.2 or higher</td>
<td>LG Smart TV Emulator 2011 : Not Supported</td>
</tr>
<tr>
<td></td>
<td>LG Smart TV Emulator 2012 : SDK 2.2 or higher</td>
</tr>
<tr>
<td></td>
<td>LG Smart TV Emulator 2013 : SDK 3.0 or higher</td>
</tr>
</tbody>
</table>

#### getMin

**Description**

Gets the min value of Slider.

**Parameters**

None

**Return Value**

min (Number) The min value of the Slider

**Example**

```javascript
var slider = LGE.UI.getComp('slider');
var min = slider.getMin();
```

**Supported SDK / Emulator Version**

<table>
<thead>
<tr>
<th>SDK Version</th>
<th>Emulator Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.2 or higher</td>
<td>LG Smart TV Emulator 2011 : Not Supported</td>
</tr>
<tr>
<td></td>
<td>LG Smart TV Emulator 2012 : SDK 2.2 or higher</td>
</tr>
</tbody>
</table>
### getValue

**Description**
Gets the value of the Slider.

**Parameters**
None

**Return Value**

```plaintext
value
(Number) The value of the Slider
```

**Example**

```javascript
var slider = LGE.UI.getComp('slider');
var max = slider.getValue();
```

---

### setMax

**Description**
Sets the max value of the Slider.

**Parameters**

```plaintext
max
(Number) The max value of the Slider
```

**Return Value**
None

**Example**

```javascript
var slider = LGE.UI.getComp('slider');
slider.setMax(10);
```

---

### setMin

**Description**

---

---
**Description**
Sets the min value of the Slider.

**Parameters**
- **min** (Number) The min value of the Slider

**Return Value**
None

**Example**
```
var slider = LGE.UI.getComp('slider');
slider.setMin(-10);
```

**Supported SDK / Emulator Version**

<table>
<thead>
<tr>
<th>SDK Version</th>
<th>Emulator Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.2 or higher</td>
<td>LG Smart TV Emulator 2011 : Not Supported</td>
</tr>
<tr>
<td></td>
<td>LG Smart TV Emulator 2012 : SDK 2.2 or higher</td>
</tr>
<tr>
<td></td>
<td>LG Smart TV Emulator 2013 : SDK 3.0 or higher</td>
</tr>
</tbody>
</table>

---

**setValue**

**Description**
Sets the value of the Slider.

**Parameters**
- **value** (Number) The value of the Slider

**Return Value**
None

**Example**
```
var slider = LGE.UI.getComp('slider');
slider.setValue(0);
```

**Supported SDK / Emulator Version**

<table>
<thead>
<tr>
<th>SDK Version</th>
<th>Emulator Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.2 or higher</td>
<td>LG Smart TV Emulator 2011 : Not Supported</td>
</tr>
<tr>
<td></td>
<td>LG Smart TV Emulator 2012 : SDK 2.2 or higher</td>
</tr>
<tr>
<td></td>
<td>LG Smart TV Emulator 2013 : SDK 3.0 or higher</td>
</tr>
</tbody>
</table>

---

**Tab**

The Tab component is container for group of tabs. This component consists of TabHeader and TabBody. TabHeader is a parent of TabInfo that is title of tab. TabBody is parent of the TabContents.

![Tab Component](image)

**[Figure] Tab Component**
Inheritance Hierarchy
Object >> Component >> Container >> TabHost

Markup Interface, JS Interface (Constructor Parameters), and JS Interface (Methods) are listed in the following sections.
- Markup Interface
- JS Interface (Constructor Parameters)
- JS Interface (Methods)
Markup Interface

Example

```xml
<div lge-type="TabHost" id="tab1">
  <div lge-type="TabHeader">
    <div lge-type="TabInfo" lge-attr="index:tab1;">
      <div>TabTitle1</div>
    </div>
  </div>
  <div lge-type="TabBody">
    <div lge-type="TabContents" lge-attr="index:tab1;">Tab1 Contents</div>
    <div lge-type="TabContents" lge-attr="index:tab2;">Tab2 Contents</div>
    <div lge-type="TabContents" lge-attr="index:tab3;">Tab3 Contents</div>
  </div>
</div>
```

Attributes Summary

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>lge-attr</td>
<td>Attributes of the LGE component</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>index</td>
<td>Index of each tab</td>
</tr>
</tbody>
</table>

index

Description

Index of each tab. This attribute connects TabInfo and TabContents.

Example

```xml
<div lge-type="TabHost" id="tab1">
  <div lge-type="TabHeader">
    <div lge-type="TabInfo" lge-attr="index:tab1;">
      <div>TabTitle1</div>
    </div>
  </div>
  <div lge-type="TabBody">
    <div lge-type="TabContents" lge-attr="index:tab1;">Tab1 Contents</div>
    <div lge-type="TabContents" lge-attr="index:tab2;">Tab2 Contents</div>
  </div>
</div>
```
JS Interface (Constructor Parameters)

Example
var tabhost = LGE.UI.Tab(
    text:[
        {title:'TabTitle1',content:'Tab 1 Content'},
        {title:'TabTitle2',content:'Tab 2 Content'},
        {title:'TabTitle3',content:'Tab 3 Content'}
    ]
    selector:("td2"),
    id:'tabhost',
});

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>id</td>
<td>Element ID of the component in document</td>
</tr>
<tr>
<td>selector</td>
<td>The element that becomes parent of the component</td>
</tr>
<tr>
<td>text</td>
<td>A list of tabs</td>
</tr>
</tbody>
</table>

id

Description
Element ID of the component in document.

Example
var tabhost = LGE.UI.Tab(
    text:[
        {title:'TabTitle1',content:'Tab 1 Content'},
        {title:'TabTitle2',content:'Tab 2 Content'},
        {title:'TabTitle3',content:'Tab 3 Content'}
    ]
    selector:("td2"),
    id:'tabhost',
});

selector

Description
The element that becomes parent of the component
Default is body.

Example
var tabhost = LGE.UI.Tab(
    text:[
        {title:'TabTitle1',content:'Tab 1 Content'},
        {title:'TabTitle2',content:'Tab 2 Content'},
        {title:'TabTitle3',content:'Tab 3 Content'}
    ]
    selector:("td2"),
});

text
Description
A list of tabs. Each tab has two parameters.
- title : Title of tab.
- content : Content of tab

Example
```javascript
var tabhost = LGE.UI.Tab({
  text:[
    {title:'TabTitle1',content:'Tab 1 Content'},
    {title:'TabTitle2',content:'Tab 2 Content'},
    {title:'TabTitle3',content:'Tab 3 Content'}
  ],
  selector:('td2'),
});
```
# JS Interface (Methods)

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>addTab</td>
<td>Adds an item of the Tab.</td>
</tr>
<tr>
<td>getSelected</td>
<td>Gets the current selected item of the Tab.</td>
</tr>
<tr>
<td>getTabHeight</td>
<td>Gets the height of the Tab</td>
</tr>
<tr>
<td>getTabWidth</td>
<td>Gets the width of the Tab</td>
</tr>
<tr>
<td>setSelected</td>
<td>Selects index item of the Tab.</td>
</tr>
<tr>
<td>setTab</td>
<td>Sets the item of the Tab.</td>
</tr>
<tr>
<td>setTabHeight</td>
<td>Sets the height of the Tab.</td>
</tr>
<tr>
<td>setTabWidth</td>
<td>Sets the width of the Tab.</td>
</tr>
</tbody>
</table>

## addTab

**Description**
Adds an item of the Tab.

**Parameters**
- **tabInfo** *(Object)* A set *(tabTitle, content)* of Tab information
  - tabTitle : The title of the tab
  - content : The content of the tab

**Return Value**
None

**Example**

```javascript
var tabhost = LGE.UI.getComp('tab');
tabhost.addTab({
    tabTitle: "title1",
    content: "<p>hellow world!</p>",
});
```

## getSelected

**Description**
Gets the current selected item of the Tab.

**Parameters**
None

**Return Value**
- **tabID** *(Number)* The index of the tab

## Supported SDK / Emulator Version

<table>
<thead>
<tr>
<th>SDK Version</th>
<th>Emulator Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.2 or higher</td>
<td>LG Smart TV Emulator 2011 : Not Supported</td>
</tr>
<tr>
<td></td>
<td>LG Smart TV Emulator 2012 : SDK 2.2 or higher</td>
</tr>
<tr>
<td></td>
<td>LG Smart TV Emulator 2013 : SDK 3.0 or higher</td>
</tr>
</tbody>
</table>
Example

```javascript
var tabhost = LGE/UI.getComp('tab');
var index = tabhost.getSelected();
```

### Supported SDK / Emulator Version

<table>
<thead>
<tr>
<th>SDK Version</th>
<th>Emulator Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.2 or higher</td>
<td>LG Smart TV Emulator 2011 : Not Supported</td>
</tr>
<tr>
<td></td>
<td>LG Smart TV Emulator 2012 : SDK 2.2 or higher</td>
</tr>
<tr>
<td></td>
<td>LG Smart TV Emulator 2013 : SDK 3.0 or higher</td>
</tr>
</tbody>
</table>

### getTabHeight

**Description**

Gets the height of the Tab.

**Parameters**

None

**Return Value**

```javascript
height (Number) The height of the Tab
```

**Example**

```javascript
var tabhost = LGE/UI.getComp('tab');
var height = tabhost.getTabHeight();
```

### Supported SDK / Emulator Version

<table>
<thead>
<tr>
<th>SDK Version</th>
<th>Emulator Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.2 or higher</td>
<td>LG Smart TV Emulator 2011 : Not Supported</td>
</tr>
<tr>
<td></td>
<td>LG Smart TV Emulator 2012 : SDK 2.2 or higher</td>
</tr>
<tr>
<td></td>
<td>LG Smart TV Emulator 2013 : SDK 3.0 or higher</td>
</tr>
</tbody>
</table>

### getTabWidth

**Description**

Gets the Width of the Tab.

**Parameters**

None

**Return Value**

```javascript
width (Number) The width of the Tab
```

**Example**

```javascript
var tabhost = LGE/UI.getComp('tab');
var width = tabhost.getTabWidth();
```

### Supported SDK / Emulator Version

<table>
<thead>
<tr>
<th>SDK Version</th>
<th>Emulator Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.2 or higher</td>
<td>LG Smart TV Emulator 2011 : Not Supported</td>
</tr>
<tr>
<td></td>
<td>LG Smart TV Emulator 2012 : SDK 2.2 or higher</td>
</tr>
<tr>
<td></td>
<td>LG Smart TV Emulator 2013 : SDK 3.0 or higher</td>
</tr>
</tbody>
</table>
**setSelected**

**Description**
Selects the current selected item of the Tab.

**Parameters**
- **tabID**  
  (Number) The index of the tab

**Return Value**
None

**Example**
```javascript
var tabhost = LGE.UI.getComp('tab');
tabhost setSelected(1);
```

**Supported SDK / Emulator Version**
<table>
<thead>
<tr>
<th>SDK Version</th>
<th>Emulator Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.2 or higher</td>
<td>LG Smart TV Emulator 2011 : Not Supported</td>
</tr>
<tr>
<td></td>
<td>LG Smart TV Emulator 2012 : SDK 2.2 or higher</td>
</tr>
<tr>
<td></td>
<td>LG Smart TV Emulator 2013 : SDK 3.0 or higher</td>
</tr>
</tbody>
</table>

**setTab**

**Description**
Sets the item of the Tab.

**Parameters**
- **tabID**  
  (Number) The index of the tab
- **tabInfo**  
  (Object) A set (tabTitle, content) of Tab information
  - tabTitle : The title of the tab
  - content : The content of the tab

**Return Value**
None

**Example**
```javascript
var tabhost = LGE.UI.getComp('tab');
tabhost.setTab(2, {
  tabTitle: "title2",
  content: "<p>hellow world!</p>",
});
```

**Supported SDK / Emulator Version**
<table>
<thead>
<tr>
<th>SDK Version</th>
<th>Emulator Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.2 or higher</td>
<td>LG Smart TV Emulator 2011 : Not Supported</td>
</tr>
<tr>
<td></td>
<td>LG Smart TV Emulator 2012 : SDK 2.2 or higher</td>
</tr>
<tr>
<td></td>
<td>LG Smart TV Emulator 2013 : SDK 3.0 or higher</td>
</tr>
</tbody>
</table>

**setTabHeight**
Description
Sets the height of the Tab.

Parameters
height (Number) The height of the Tab

Return Value
None

Example
var tabhost = LGE.UI.getComp('tab');
tabhost.setTabHeight(500);

Supported SDK / Emulator Version

<table>
<thead>
<tr>
<th>SDK Version</th>
<th>Emulator Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.2 or higher</td>
<td>LG Smart TV Emulator 2011: Not Supported</td>
</tr>
<tr>
<td></td>
<td>LG Smart TV Emulator 2012: SDK 2.2 or higher</td>
</tr>
<tr>
<td></td>
<td>LG Smart TV Emulator 2013: SDK 3.0 or higher</td>
</tr>
</tbody>
</table>

setTabWidth

Description
Sets the width of the Tab.

Parameters
width (Number) The width of the Tab

Return Value
None

Example
var tabhost = LGE.UI.getComp('tab');
tabhost.setTabWidth(800);

Supported SDK / Emulator Version

<table>
<thead>
<tr>
<th>SDK Version</th>
<th>Emulator Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.2 or higher</td>
<td>LG Smart TV Emulator 2011: Not Supported</td>
</tr>
<tr>
<td></td>
<td>LG Smart TV Emulator 2012: SDK 2.2 or higher</td>
</tr>
<tr>
<td></td>
<td>LG Smart TV Emulator 2013: SDK 3.0 or higher</td>
</tr>
</tbody>
</table>

TextInput

The TextInput component allows users to input data.

[Figure] TextInput Component

Inheritance Hierarchy
Object >> Component >> Container >> TextInput

Markup Interface, JS Interface (Constructor Parameters), and JS Interface (Methods) are listed in the following sections.
• Markup Interface
• JS Interface (Constructor Parameters)
• JS Interface (Methods)
Markup Interface

Example
<input type="text" lge-type="TextInput" value="value1" id='text1' />

<input type="password" lge-type="TextInput" value="value2" id='text1' />

<textarea lge-type="TextInput" name="textarea12" /></textarea>

Attributes Summary

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>lge-attr</td>
<td>Attributes of the LGE component</td>
</tr>
<tr>
<td>type</td>
<td>The type of Input</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>text-inline</td>
<td>More compact component that is only as wide as the text inside</td>
</tr>
</tbody>
</table>

**text-inline**

**Description**
More compact component that is only as wide as the text inside
Default value is false. (true | false)

**Example**
<input type="text" lge-type="TextInput" value="value2" id='text1' lge-attr="text-inline:true;" />
JS Interface (Constructor Parameters)

Example

```javascript
var ts8 = LGE.UI.TextInput({
  type:"text",
  selector:("td2"),
});
```

```javascript
var ts9 = LGE.UI.TextInput({
  type:"password",
  selector:("td2"),
});
```

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>id</td>
<td>Element ID of the component in document</td>
</tr>
<tr>
<td>selector</td>
<td>The element that becomes parent of the component</td>
</tr>
<tr>
<td>style</td>
<td>CSS style of the component</td>
</tr>
<tr>
<td>text</td>
<td>Text of the component</td>
</tr>
<tr>
<td>textInline</td>
<td>More compact component that is only as wide as the text inside</td>
</tr>
<tr>
<td>type</td>
<td>Type of the component</td>
</tr>
<tr>
<td>value</td>
<td>Value of the component</td>
</tr>
</tbody>
</table>

id

Description
Element ID of the component in document.

Example

```javascript
var ts8 = LGE.UI.TextInput({
  id:"textinput1",
  type:"text",
  selector:("td2"),
});
```

selector

Description
The element that becomes parent of the component
Default is body.

Example

```javascript
var ts8 = LGE.UI.TextInput({
  id:"textinput1",
  type:"text",
  selector:("td2"),
});
```

style
Description
CSS style of the component

Example
```javascript
var ts8 = LGE.UI.TextInput({
    type: "text",
    selector: "td2",
    style: "left:100px",
});
```

**text**

Description
The text of the component

Example
```javascript
var ts8 = LGE.UI.TextInput({
    type: "text",
    selector: "td2",
    style: "width:100px",
    text: "text ....",
});
```

**textinline**

Description
More compact component that is only as wide as the text inside
Default value is false. (true | false)

Example
```javascript
var ts8 = LGE.UI.TextInput({
    type: "text",
    selector: "td2",
    style: "width:100px",
    text: "text ....",
    textInline: 'true',
});
```

**type**

Description
The type of the component (text | password)

Example
```javascript
var ts8 = LGE.UI.TextInput({
    type: "text",
    selector: "td2",
    style: "width:100px",
    text: "text ....",
});
```
JS Interface (Methods)

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>getText</td>
<td>Gets the text in the TextInput.</td>
</tr>
<tr>
<td>getTextInline</td>
<td>Gets the text-inline the TextInput.</td>
</tr>
<tr>
<td>setText</td>
<td>Sets the text in the TextInput.</td>
</tr>
<tr>
<td>setTextInline</td>
<td>Sets the text-inline in the TextInput.</td>
</tr>
</tbody>
</table>

**getText**

**Description**

Gets the text in the TextInput.

**Parameters**

None

**Return Value**

`text` (String) The text in the TextInput

**Example**

```javascript
var text = textInput1.getText();
```

**Supported SDK / Emulator Version**

<table>
<thead>
<tr>
<th>SDK Version</th>
<th>Emulator Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.2 or higher</td>
<td>LG Smart TV Emulator 2011 : Not Supported</td>
</tr>
<tr>
<td></td>
<td>LG Smart TV Emulator 2012 : SDK 2.2 or higher</td>
</tr>
<tr>
<td></td>
<td>LG Smart TV Emulator 2013 : SDK 3.0 or higher</td>
</tr>
</tbody>
</table>

**getTextInline**

**Description**

Gets the text-inline of the TextInput.

**Parameters**

None

**Return Value**

`bool` (Boolean) The text-inline of the TextInput

**Example**

```javascript
var textinline = textInput1.getTextInline();
```

**Supported SDK / Emulator Version**

<table>
<thead>
<tr>
<th>SDK Version</th>
<th>Emulator Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.2 or higher</td>
<td>LG Smart TV Emulator 2011 : Not Supported</td>
</tr>
<tr>
<td></td>
<td>LG Smart TV Emulator 2012 : SDK 2.2 or higher</td>
</tr>
<tr>
<td></td>
<td>LG Smart TV Emulator 2013 : SDK 3.0 or higher</td>
</tr>
</tbody>
</table>
**setText**

**Description**
Gets the text in the TextInput.

**Parameters**
text (String) The text in the TextInput

**Return Value**
None

**Example**

```
textInput1.setText("text input test");
```

**Supported SDK / Emulator Version**

<table>
<thead>
<tr>
<th>SDK Version</th>
<th>Emulator Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.2 or higher</td>
<td>LG Smart TV Emulator 2011: Not Supported</td>
</tr>
<tr>
<td></td>
<td>LG Smart TV Emulator 2012: SDK 2.2 or higher</td>
</tr>
<tr>
<td></td>
<td>LG Smart TV Emulator 2013: SDK 3.0 or higher</td>
</tr>
</tbody>
</table>

**setTextInline**

**Description**
Sets the text-inline of the TextInput.

**Parameters**
bool (Boolean) The text-inline of the TextInput

**Return Value**
None

**Example**

```
textInput1.setTextInline(true);
```

**Supported SDK / Emulator Version**

<table>
<thead>
<tr>
<th>SDK Version</th>
<th>Emulator Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.2 or higher</td>
<td>LG Smart TV Emulator 2011: Not Supported</td>
</tr>
<tr>
<td></td>
<td>LG Smart TV Emulator 2012: SDK 2.2 or higher</td>
</tr>
<tr>
<td></td>
<td>LG Smart TV Emulator 2013: SDK 3.0 or higher</td>
</tr>
</tbody>
</table>

**ToggleSwitch**

The ToggleSwitch is used for selecting option.

![ToggleSwitch Component]

**Inheritance Hierarchy**

Object >> Component >> Container >> ItemListComponent >> SelectableList >> ToggleSwitch

Markup Interface, JS Interface (Constructor Parameters), and JS Interface (Methods) are listed in the following sections.

- **Markup Interface**
• JS Interface (Constructor Parameters)
• JS Interface (Methods)
**Markup Interface**

**Example**

```xml
<div lge-type="ToggleSwitch" id="toggleswitch1">
  <select>
    <option>On</option>
    <option selected>Off</option>
  </select>
</div>
```

**Attributes Summary**

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>lge-attr</td>
<td>Attributes of the LGE component</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>text-inline</td>
<td>More compact component that is only as wide as the text inside</td>
</tr>
</tbody>
</table>

**text-inline**

**Description**

More compact component that is only as wide as the text inside

Default value is false (true | false)

**Example**

```xml
<div lge-type="ToggleSwitch" id="toggleswitch1" lge-attr="text-inline:true;" >
  <select>
    <option>On</option>
    <option selected>Off</option>
  </select>
</div>
```
JS Interface (Constructor Parameters)

Example
```javascript
var toggleswitch = LGE.UI.ToggleSwitch({
  on:{text:"On"},
  off:{text:"Off"},
  selector:("td2")
});
```

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>id</td>
<td>Element ID of the component in document</td>
</tr>
<tr>
<td>on</td>
<td>Shows the text when the component is On.</td>
</tr>
<tr>
<td>off</td>
<td>Shows the text when the component is Off.</td>
</tr>
<tr>
<td>selector</td>
<td>The element that becomes parent of the component</td>
</tr>
<tr>
<td>textInline</td>
<td>More compact component that is only as wide as the text inside</td>
</tr>
</tbody>
</table>

id

Description
Element ID of the component in document.

Example
```javascript
var toggleswitch = LGE.UI.ToggleSwitch({
  id:'toggleswitch',
  on:{text:"On"},
  off:{text:"Off"},
  selector:("td2"),
  textInline:true,
});
```

on

Description
Shows the text when the component is On.

Example
```javascript
var toggleswitch = LGE.UI.ToggleSwitch({
  on:{text:"On"},
  off:{text:"Off"},
  selector:("td2"),
  textInline:true,
});
```

off

Description
Shows the text when the component is Off.

Example
var toggleswitch = LGE.UI.ToggleSwitch({
on:{text:"On"},
off:{text:"Off"},
selector:("td2"),
textInline:true,
});

**selector**

**Description**
The element that becomes parent of the component
Default is body.

**Example**

```javascript
var toggleswitch = LGE.UI.ToggleSwitch({
on:{text:"On"},
off:{text:"Off"},
selector:("td2"),
textInline:true,
});
```

textinline

**Description**
More compact component that is only as wide as the text inside
Default value is false. (true | false)

**Example**

```javascript
var ts8 = LGE.UI.TextInput({
type:"text",
selector:("td2"),
style:"width:100px",
text:"text . . .",
textInline:'true',
});
```
### JS Interface (Methods)

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>getSelected</td>
<td>Gets the selected index of ToggleSwitch.</td>
</tr>
<tr>
<td>getText</td>
<td>Gets the selected text of ToggleSwitch.</td>
</tr>
<tr>
<td>getTextInline</td>
<td>Gets the text-inline of ToggleSwitch.</td>
</tr>
<tr>
<td>setTextRes</td>
<td>Sets the resource text of the ToggleSwitch.</td>
</tr>
<tr>
<td>setSelected</td>
<td>Sets the selected index of the ToggleSwitch.</td>
</tr>
<tr>
<td>setText</td>
<td>Sets the text of the ToggleSwitch.</td>
</tr>
<tr>
<td>setTextInline</td>
<td>Sets the text-inline of ToggleSwitch.</td>
</tr>
</tbody>
</table>

#### getSelected

**Description**

Gets the selected index of ToggleSwitch.

**Parameters**

None

**Return Value**

- `index` (Number) The selected index of ToggleSwitch

**Example**

```javascript
var toggle = LGE.UI.getComp('toggle');
var index = toggle.getSelected();
```

**Supported SDK / Emulator Version**

<table>
<thead>
<tr>
<th>SDK Version</th>
<th>Emulator Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.2 or higher</td>
<td>LG Smart TV Emulator 2011: Not Supported</td>
</tr>
<tr>
<td></td>
<td>LG Smart TV Emulator 2012: SDK 2.2 or higher</td>
</tr>
<tr>
<td></td>
<td>LG Smart TV Emulator 2013: SDK 3.0 or higher</td>
</tr>
</tbody>
</table>

#### getText

**Description**

Gets the selected text of ToggleSwitch.

**Parameters**

None

**Return Value**

- `text` (String) The selected text of ToggleSwitch

**Example**

```javascript
var toggle = LGE.UI.getComp('toggle');
var text = toggle.getText();
```
### getTextureInline

**Description**
Gets the text-inline of the ToggleSwitch.

**Parameters**
None

**Return Value**

```
bool (Boolean) The text-inline of the ToggleSwitch
```

**Example**

```javascript
var toggle = LGE.UI.getComp('toggle');
var textinline = toggle.getTextInline();
```

### setTextRes

**Description**
Sets the text of the ToggleSwitch.

**Parameters**

- `index` (Number) The index of ToggleSwitch’s text
- `value` (String) The value of text resource

**Return Value**
None

**Example**

```javascript
var toggle = LGE.UI.getComp('toggle');
toggle.setTextRes(1, "Off");
```

### Supported SDK / Emulator Version

<table>
<thead>
<tr>
<th>SDK Version</th>
<th>Emulator Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.2 or higher</td>
<td>LG Smart TV Emulator 2011 : Not Supported</td>
</tr>
<tr>
<td></td>
<td>LG Smart TV Emulator 2012 : SDK 2.2 or higher</td>
</tr>
<tr>
<td></td>
<td>LG Smart TV Emulator 2013 : SDK 3.0 or higher</td>
</tr>
</tbody>
</table>

### setSelected

```javascript
```

### Supported SDK / Emulator Version

<table>
<thead>
<tr>
<th>SDK Version</th>
<th>Emulator Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.2 or higher</td>
<td>LG Smart TV Emulator 2011 : Not Supported</td>
</tr>
<tr>
<td></td>
<td>LG Smart TV Emulator 2012 : SDK 2.2 or higher</td>
</tr>
<tr>
<td></td>
<td>LG Smart TV Emulator 2013 : SDK 3.0 or higher</td>
</tr>
</tbody>
</table>
Description
Sets the selected index of the ToggleSwitch.

Parameters
index (Number) The index of ToggleSwitch's text

Return Value
None

Example
var toggle = LGE.UI.getComp('toggle');
toggle.setSelected(1);

Supported SDK / Emulator Version

<table>
<thead>
<tr>
<th>SDK Version</th>
<th>Emulator Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.2 or higher</td>
<td>LG Smart TV Emulator 2011 : Not Supported</td>
</tr>
<tr>
<td></td>
<td>LG Smart TV Emulator 2012 : SDK 2.2 or higher</td>
</tr>
<tr>
<td></td>
<td>LG Smart TV Emulator 2013 : SDK 3.0 or higher</td>
</tr>
</tbody>
</table>

setText

Description
Sets the text of the ToggleSwitch.

Parameters
index (Number) The index of ToggleSwitch's text
text (String) The text of the ToggleSwitch

Return Value
None

Example
var toggle = LGE.UI.getComp('toggle');
toggle.setText(1, "Off");

Supported SDK / Emulator Version

<table>
<thead>
<tr>
<th>SDK Version</th>
<th>Emulator Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.2 or higher</td>
<td>LG Smart TV Emulator 2011 : Not Supported</td>
</tr>
<tr>
<td></td>
<td>LG Smart TV Emulator 2012 : SDK 2.2 or higher</td>
</tr>
<tr>
<td></td>
<td>LG Smart TV Emulator 2013 : SDK 3.0 or higher</td>
</tr>
</tbody>
</table>

setTextInline

Description
Sets the text-inline of ToggleSwitch.

Parameters
bool (Boolean) Whether the text-inline is true or not.

Return Value
None

Example
Supported SDK / Emulator Version

<table>
<thead>
<tr>
<th>SDK Version</th>
<th>Emulator Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.2 or higher</td>
<td>LG Smart TV Emulator 2011 : Not Supported</td>
</tr>
<tr>
<td></td>
<td>LG Smart TV Emulator 2012 : SDK 2.2 or higher</td>
</tr>
<tr>
<td></td>
<td>LG Smart TV Emulator 2013 : SDK 3.0 or higher</td>
</tr>
</tbody>
</table>

**TvHelp**

The TvHelp component support users control the Button with the remote control.

![Figure] TvHelp Component

**Inheritance Hierarchy**
Object >> Component >> Container >> ListItemComponent >> TvHelp

Markup Interface, JS Interface (Constructor Parameters), and JS Interface (Methods) are listed in the following sections.
- [Markup Interface](#)
- [JS Interface (Constructor Parameters)](#)
- [JS Interface (Methods)](#)
Markup Interface

Example

```html
<div lge-type="TvHelp" id="TvHelpTest">
  <input type="button" lge-attr='option:Red' value="Red"/>
  <input type="button" lge-attr='option:Yellow' value="Yellow"/>
  <a href='http://www.lge.com' lge-attr='option:Green'>Green</a>
</div>
```

Attributes Summary

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>lge-attr</td>
<td>Attributes of the LGE component</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Ige-attr</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>option</td>
<td>Color of the TvHelp button</td>
</tr>
</tbody>
</table>

option

Description
This option is color of TvHelp Button. There are four colors. (Red, Yellow, Green, and Blue)

Example

```html
<div lge-type="TvHelp" id="TvHelpTest">
  <input type="button" lge-attr='option:Red' value="Red"/>
  <input type="button" lge-attr='option:Yellow' value="Yellow"/>
  <a href='http://www.lge.com' lge-attr='option:Green'>Green</a>
</div>
```
JS Interface (Constructor Parameters)

**Example**

```javascript
var tvhelp = LGE.UI.TvHelp({
  selector:("td2"),
  id:’tvhelp’
  items:[
    {tag:'input', type:’button’,option:'Red',text:'Red'},
    {tag:'input', type:’image’,option:'Blue',text:'Red'},
    {tag:'a',option:'Yellow',text:'Yellow',link:'http://lge.com'},
    {tag:'a',option:'Green',text:'Green',link:'http://lge.com'},
  ]
});
```

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>items</td>
<td>A list of buttons</td>
</tr>
<tr>
<td>id</td>
<td>Element ID of the component in document</td>
</tr>
<tr>
<td>selector</td>
<td>The element that becomes parent of the component</td>
</tr>
<tr>
<td>style</td>
<td>CSS style of the component</td>
</tr>
</tbody>
</table>

**id**

**Description**
Element ID of the component in document.

**Example**

```javascript
var tvhelp1 = LGE.UI.TvHelp({
  id:’tvhelp’,
  selector:("td2"),
  items:[
    {tag:'input', type:’button’,option:'Red',text:'Red'},
    {tag:'input', type:’image’,option:'Blue',text:'Red'},
    {tag:'a',option:'Yellow',text:'Yellow',link:'http://lge.com'},
    {tag:'a',option:'Green',text:'Green',link:'http://lge.com'},
  ]);```

**items**

**Description**
A list of buttons in TvHelp.

**Button**
- id : Element ID of the Button
- tag : tag of the button. There are two tags: <A>, <INPUT>
- type : type of the <INPUT> (if tag is <INPUT>) (button | reset | submit | image)
- option : Color of the button.
- text : text of the button.
- link : link of the button (if tag is <A>)
- style : CSS style of the component

**Example**

```javascript
var tvhelp1 = LGE.UI.TvHelp({
  selector:("td2"),
  items:[
```
selector

Description
The element that becomes parent of the component
Default is body.

Example
```javascript
var tvhelp1 = LGE.UI.TvHelp({
    id: "TvHelp1",
    selector: "td2",
    items: [
        {tag: 'input', type: 'button', option: 'Red', text: 'Red'},
    ],
});
```

style

Description
CSS style of the component

Example
```javascript
var tvhelp1 = LGE.UI.TvHelp({
    selector: "td2",
    items: [
        {tag: 'input', type: 'button', option: 'Red', text: 'Red'},
    ],
    style: "top: 100px",
});
```
### JS Interface (Methods)

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>addIt</td>
<td>Adds a TvhHelp Button in the TvhHelp.</td>
</tr>
<tr>
<td>getLink</td>
<td>Gets the link url of the TvhHelp Button in the TvhHelp.</td>
</tr>
<tr>
<td>getOption</td>
<td>Gets the option (Red,Blue,Yellow,Green) of the TvhHelp Button.</td>
</tr>
<tr>
<td>getText</td>
<td>Gets the text of the Button.</td>
</tr>
<tr>
<td>removeItem</td>
<td>Removes the TvhHelp Button in the TvhHelp.</td>
</tr>
<tr>
<td>setItem</td>
<td>Sets the TvhHelp Button.</td>
</tr>
</tbody>
</table>

### addItem

**Description**

Adds a TvhHelp Button in the TvhHelp.

**Parameters**

- **options** (Object) The sets of the TvhHelp Button Information.
  - tag : the element tag type of the Button.
  - type : the type of the Button (If tag is INPUT, required)
  - link : the url link of the Button (if tag is <A>)
  - text : the text of the Button
  - option : the color of the Button (Red | Blue | Green | Yellow)

**Return Value**

None

**Example**

```javascript
var tvhelp = LGE.UI.getComp('tvhelp');
tvhelp.addItem({
  tag: "a",
  link: "http://www.lge.com",
  text: "button",
  option: "Red"
});
```

### getLink

**Description**

Gets the link url of the TvhHelp Button in the TvhHelp.

**Parameters**

- **index** (Number) The index of the TvhHelp Button in the TvhHelp

**Return Value**

- **url** (String) The link url of this TvhHelp Button

### Supported SDK / Emulator Version

<table>
<thead>
<tr>
<th>SDK Version</th>
<th>Emulator Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.2 or higher</td>
<td>LG Smart TV Emulator 2011 : Not Supported</td>
</tr>
<tr>
<td></td>
<td>LG Smart TV Emulator 2012 : SDK 2.2 or higher</td>
</tr>
<tr>
<td></td>
<td>LG Smart TV Emulator 2013 : SDK 3.0 or higher</td>
</tr>
</tbody>
</table>
Example

```javascript
var tvhelp = LGE.UI.getComp('tvhelp');
var url = tvhelp.getLink();
```

**Supported SDK / Emulator Version**

<table>
<thead>
<tr>
<th>SDK Version</th>
<th>Emulator Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.2 or higher</td>
<td>LG Smart TV Emulator 2011: Not Supported</td>
</tr>
<tr>
<td></td>
<td>LG Smart TV Emulator 2012: SDK 2.2 or higher</td>
</tr>
<tr>
<td></td>
<td>LG Smart TV Emulator 2013: SDK 3.0 or higher</td>
</tr>
</tbody>
</table>

**getOption**

**Description**

Gets the option (Red, Blue, Yellow, Green) of the TvHelp Button.

**Parameters**

- `index` (Number) The index of the TvHelp Button in the TvHelp

**Return Value**

- `option` (String) The option (Red, Blue, Yellow, Green) of the TvHelp Button

Example

```javascript
var tvhelp = LGE.UI.getComp('tvhelp');
var colorOfButton = tvhelp.getOption(1);
```

**Supported SDK / Emulator Version**

<table>
<thead>
<tr>
<th>SDK Version</th>
<th>Emulator Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.2 or higher</td>
<td>LG Smart TV Emulator 2011: Not Supported</td>
</tr>
<tr>
<td></td>
<td>LG Smart TV Emulator 2012: SDK 2.2 or higher</td>
</tr>
<tr>
<td></td>
<td>LG Smart TV Emulator 2013: SDK 3.0 or higher</td>
</tr>
</tbody>
</table>

**getText**

**Description**

Gets the text of the Button.

**Parameters**

- `index` (Number) The index of the TvHelp Button in the TvHelp

**Return Value**

- `text` (String) The text of the Button

Example

```javascript
var tvhelp = LGE.UI.getComp('tvhelp');
var colorOfButton = tvhelp.getOption(1);
```

**Supported SDK / Emulator Version**

<table>
<thead>
<tr>
<th>SDK Version</th>
<th>Emulator Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.2 or higher</td>
<td>LG Smart TV Emulator 2011: Not Supported</td>
</tr>
<tr>
<td></td>
<td>LG Smart TV Emulator 2012: SDK 2.2 or higher</td>
</tr>
<tr>
<td></td>
<td>LG Smart TV Emulator 2013: SDK 3.0 or higher</td>
</tr>
</tbody>
</table>
**removeItem**

**Description**
Removes the TvHelp Button in the TvHelp.

**Parameters**
- **index** *(Number)* The index of the TvHelp Button in the TvHelp

**Return Value**
None

**Example**
```javascript
var tvhelp = LGE.UI.getComp('tvhelp');
tvhelp.removeItem(1);
```

**Supported SDK / Emulator Version**

<table>
<thead>
<tr>
<th>SDK Version</th>
<th>Emulator Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.2 or higher</td>
<td>LG Smart TV Emulator 2011 : Not Supported</td>
</tr>
<tr>
<td></td>
<td>LG Smart TV Emulator 2012 : SDK 2.2 or higher</td>
</tr>
<tr>
<td></td>
<td>LG Smart TV Emulator 2013 : SDK 3.0 or higher</td>
</tr>
</tbody>
</table>

**setItem**

**Description**
Gets the selected index of ToggleSwitch.

**Parameters**
- **index** *(Number)* The index of the TvHelp Button in the TvHelp
- **options** *(Object)* The set of the TvHelp Button Information
  - **link** : the url link of the Button (if tag is A)
  - **text** : the text of the Button
  - **option** : the color of the Button (Red | Blue | Green | Yellow)

**Return Value**
None

**Example**
```javascript
var tvhelp = LGE.UI.getComp('tvhelp');
tvhelp.setItem(0, {
    link: "http://www.lge.com",
    text: "button",
    option: "Red"
});
```

**Supported SDK / Emulator Version**

<table>
<thead>
<tr>
<th>SDK Version</th>
<th>Emulator Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.2 or higher</td>
<td>LG Smart TV Emulator 2011 : Not Supported</td>
</tr>
<tr>
<td></td>
<td>LG Smart TV Emulator 2012 : SDK 2.2 or higher</td>
</tr>
<tr>
<td></td>
<td>LG Smart TV Emulator 2013 : SDK 3.0 or higher</td>
</tr>
</tbody>
</table>

**PageEffects**
Page effect allows you to change page with effects. There are four types of page effect and methods. (Fade, Flip, Pop, and Slide)

JS Interface (Constructor Parameters), JS Interface (Methods), and Effects are listed in the following sections.

- [JS Interface (Constructor Parameters)]
- [JS Interface (Methods)]
- [FadeEffect](LGE.PageEffect.FadeEffect)
- [FlipEffect](LGE.PageEffect.FlipEffect)
- [PopEffect](LGE.PageEffect.PopEffect)
- [SlideEffect](LGE.PageEffect.SlideEffect)
<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>currentPage</td>
<td>The current page for effect</td>
</tr>
</tbody>
</table>

**currentPage**

**Description**
The current page for effect

**Example**

```javascript
const effect = LGE.PageEffect.FlipEffect({
  currentPage: $('#page1'),
})
```
execute

Description
Executes effect animation.

Parameters
- target (Object) The page will be on the display after effect.
- options (Object) The options for effect. (duration | timing | direction)
  - duration: Determines how long the animation will run. This unit is milliseconds.
  - timing: Determines how an animation progresses between keyframes. (ease | linear | ease-in | ease-out | ease-in-out)
  - direction: Determines animation direction. (left | right | up | down)
- commands (Object) The command that has ‘before’ and ‘after’. ‘before’ command is executed before effect. ‘after’ command is executed after effect.

Return Value
None

Example
```javascript
var cCmd = {
  before:function() {
    if(isCurrentPage) $page2.css('visibility','hidden');
    else $page1.css('visibility','hidden');
  },
  after:function() {
    if(isCurrentPage) {
      $page2.css('visibility','visible');
      $page1.css('visibility','hidden');
      isCurrentPage = false;
    }else {
      $page1.css('visibility','visible');
      $page2.css('visibility','hidden');
      isCurrentPage = true;
    }
  }
};
var opt = {
  duration: 500,
  timing:'linear',
  direction: 'left',
};
effect.execute($page2, opt, cCmd);
```

Supported SDK / Emulator Version

<table>
<thead>
<tr>
<th>SDK Version</th>
<th>Emulator Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.2 or higher</td>
<td>LG Smart TV Emulator 2011: Not Supported</td>
</tr>
<tr>
<td></td>
<td>LG Smart TV Emulator 2012: SDK 2.2 or higher</td>
</tr>
<tr>
<td></td>
<td>LG Smart TV Emulator 2013: SDK 3.0 or higher</td>
</tr>
</tbody>
</table>
FadeEffect

FadeEffect causes the current page to fade out while making the other page seems to be concealed.

Example

```javascript
var isCurrentPage = true;

$('#button').bind('click',function(){
    var $page1 = $('#page1'), $page2 = $('#page2');
    var cCmd = {
        before:function() {
            if(isCurrentPage) $page2.css('visibility','hidden');
            else $page1.css('visibility','hidden');
        },
        after:function() {
            if(isCurrentPage) {
                $page2.css('visibility','visible');
                $page1.css('visibility','hidden');
                isCurrentPage = false;
            }else {
                $page1.css('visibility','visible');
                $page2.css('visibility','hidden');
                isCurrentPage = true;
            }
        }
    };
    var opt = {
        duration: 500,
        timing: 'linear',
        direction: 'down',
    };
    var effect = {};
    if(isCurrentPage) {
        effect = LGE.PageEffect.FadeEffect({currentPage:$page1});
        effect.execute($page2,opt,cCmd);
    }else {
        effect = LGE.PageEffect.SlideEffect({currentPage:$page2});
        effect.execute($page1,opt,cCmd);
    }
```
### Supported SDK / Emulator Version

<table>
<thead>
<tr>
<th>SDK Version</th>
<th>Emulator Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.2 or higher</td>
<td>LG Smart TV Emulator 2011 : Not Supported</td>
</tr>
<tr>
<td></td>
<td>LG Smart TV Emulator 2012 : SDK 2.2 or higher</td>
</tr>
<tr>
<td></td>
<td>LG Smart TV Emulator 2013 : SDK 3.0 or higher</td>
</tr>
</tbody>
</table>
FlipEffect

FlipEffect causes the current page to flip while making the other page seems to be concealed.

Example

```javascript
var isCurrentPage = true;

$('button1').bind('click',function(){
  var $page1 = $('#page1'), $page2 = $('#page2');

  var cCmd = {
    before:function() {
      if(isCurrentPage) $page2.css('visibility','hidden');
      else $page1.css('visibility','hidden');
    },
    after:function() {
      if(isCurrentPage) {
        $page2.css('visibility','visible');
        $page1.css('visibility','hidden');
        isCurrentPage = false;
      }else {
        $page1.css('visibility','visible');
        $page2.css('visibility','hidden');
        isCurrentPage = true;
      }
    }
  };

  var opt = {
    duration: 500,
    timing:'linear',
    direction: 'down',
  };

  var effect = {};
  if(isCurrentPage) {
    effect = LGE.PageEffect.FadeEffect({currentPage:$page1});
    effect.execute($page2,opt,cCmd);
  }else {
```

[Figure] FlipEffect
```javascript
    effect = LGE.PageEffect.SlideEffect({currentPage:$page2});
    effect.execute($page1,opt,cCmd);
});
</script>

<input id="button1" type="button" lge-type="Button" value="Change" />

<div id="page1"> page1 </div>
<div id="page2"> page2 </div>

### Supported SDK / Emulator Version

<table>
<thead>
<tr>
<th>SDK Version</th>
<th>Emulator Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.2 or higher</td>
<td>LG Smart TV Emulator 2011 : Not Supported</td>
</tr>
<tr>
<td></td>
<td>LG Smart TV Emulator 2012 : SDK 2.2 or higher</td>
</tr>
<tr>
<td></td>
<td>LG Smart TV Emulator 2013 : SDK 3.0 or higher</td>
</tr>
</tbody>
</table>
```
PopEffect

PopEffect causes the current page to pop up while making the other page seems to be concealed.

Example

```javascript
var isCurrentPage = true;

$('#button1').bind('click', function(){
    var $page1 = $('#page1'), $page2 = $('#page2');

    var cCmd = {
        before: function() {
            if(isCurrentPage) $page2.css('visibility','hidden');
            else $page1.css('visibility','hidden');
        },
        after: function() {
            if(isCurrentPage) {
                $page2.css('visibility','visible');
                $page1.css('visibility','hidden');
                isCurrentPage = false;
            } else {
                $page1.css('visibility','visible');
                $page2.css('visibility','hidden');
                isCurrentPage = true;
            }
        }
    };

    var opt = {
        duration: 500,
        timing: 'linear',
        direction: 'down',
    };

    var effect = {};
    if(isCurrentPage) {
        effect = LGE.PageEffect.PopEffect({currentPage:$page1});
        effect.execute($page2, opt, cCmd);
    } else {
        effect = LGE.PageEffect.PopEffect({currentPage:$page2});
        effect.execute($page1, opt, cCmd);
    }
});
```
Supported SDK / Emulator Version

<table>
<thead>
<tr>
<th>SDK Version</th>
<th>Emulator Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.2 or higher</td>
<td>LG Smart TV Emulator 2011 : Not Supported</td>
</tr>
<tr>
<td></td>
<td>LG Smart TV Emulator 2012 : SDK 2.2 or higher</td>
</tr>
<tr>
<td></td>
<td>LG Smart TV Emulator 2013 : SDK 3.0 or higher</td>
</tr>
</tbody>
</table>
SlideEffect

SlideEffect causes the current page to slide up while making the other page seems to be concealed.

Example

```javascript
var isCurrentPage = true;

$('#button1').bind('click', function(){
    var $page1 = $('#page1'), $page2 = $('#page2');

    var cCmd = {
        before: function() {
            if(isCurrentPage) $page2.css('visibility','hidden');
            else $page1.css('visibility','hidden');
        },
        after: function() {
            if(isCurrentPage) {
                $page2.css('visibility','visible');
                $page1.css('visibility','hidden');
                isCurrentPage = false;
            }else {
                $page1.css('visibility','visible');
                $page2.css('visibility','hidden');
                isCurrentPage = true;
            }
        }
    }

    var opt = {
        duration: 500,
        timing: 'linear',
        direction: 'down',
    }

    var effect = {};
    if(isCurrentPage) {
        effect = LGE.PageEffect.SlideEffect({currentPage:$page1});
    }
```
effect.execute($page2, opt, cCmd);
}
else {
    effect = LGE.PageEffect.SlideEffect({currentPage:$page2});
    effect.execute($page1, opt, cCmd);
}

</script>

<input id="button1" type="button" lge-type="Button" value="Change" />

<div id="page1"> page1 </div>
<div id="page2"> page2 </div>

### Supported SDK / Emulator Version

<table>
<thead>
<tr>
<th>SDK Version</th>
<th>Emulator Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.2 or higher</td>
<td>LG Smart TV Emulator 2011 : Not Supported</td>
</tr>
<tr>
<td></td>
<td>LG Smart TV Emulator 2012 : SDK 2.2 or higher</td>
</tr>
<tr>
<td></td>
<td>LG Smart TV Emulator 2013 : SDK 3.0 or higher</td>
</tr>
</tbody>
</table>
# Annex A Differences in Media Devices

Media products partly support NetCast 3.0. Please see the detailed information below for Media products development.

<table>
<thead>
<tr>
<th>Chapter of this Document</th>
<th>Constraints and difference of Media Devices compared with TV</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>NetCast API</strong></td>
<td></td>
</tr>
<tr>
<td>A. window.NetCastSetDefaultAspectRatio</td>
<td>Not applicable (Media products set Aspect Ratio at Device Setup menu)</td>
</tr>
<tr>
<td>B. Media specific API (window.NetCastSetScreenSaver(control)):</td>
<td>Refer to section Media Device NetCast API.</td>
</tr>
<tr>
<td>C. window.NetCastLaunchQMENU(), window.NetCastLaunchRATIO()</td>
<td>Not applicable.</td>
</tr>
<tr>
<td><strong>Media Object</strong></td>
<td></td>
</tr>
<tr>
<td>A. mode3D</td>
<td>from [Table] Available values for “mode3D” property 3, Media products do not support variable “off” and “from_2d_to_3d”.</td>
</tr>
<tr>
<td><strong>Media Player Plugin (Methods)</strong></td>
<td></td>
</tr>
<tr>
<td>A. In “Set Widevine Credential Information”, the following are added.</td>
<td></td>
</tr>
<tr>
<td>i. media.setWidevinePortalID(&quot;Portal&quot;);</td>
<td></td>
</tr>
<tr>
<td>ii. media.setWidevineStoreFront(&quot;StoreFront&quot;);</td>
<td></td>
</tr>
<tr>
<td>B. In example, the following are added.</td>
<td></td>
</tr>
<tr>
<td>i. UserData (Portal), additional optional user data (Identifies the operator)</td>
<td></td>
</tr>
<tr>
<td>ii. UserData (Storefront), additional optional user data (Identifies store run by operator)</td>
<td></td>
</tr>
<tr>
<td><strong>Media Player API (Properties)</strong></td>
<td></td>
</tr>
<tr>
<td>A. error</td>
<td>ErrorCode (1200) : Verimatrix failure</td>
</tr>
<tr>
<td>B. mode3D</td>
<td>Media products do not support variable “off” and “from_2d_to_3d”.</td>
</tr>
<tr>
<td></td>
<td>Media products only support “checker_bd” for HD format (1080p @ 24Hz).</td>
</tr>
<tr>
<td><strong>Device Info Plugin (Methods)</strong></td>
<td></td>
</tr>
<tr>
<td>A. getResponseFailMsg() is added.</td>
<td></td>
</tr>
<tr>
<td>B. The following APIs are not supported.</td>
<td></td>
</tr>
<tr>
<td>- getLocalTime</td>
<td></td>
</tr>
<tr>
<td>- getTime</td>
<td></td>
</tr>
<tr>
<td>- pentouchInfo</td>
<td></td>
</tr>
<tr>
<td><strong>Voice Recognition Plugin and API</strong></td>
<td></td>
</tr>
<tr>
<td><strong>AppToApp Plugin and API</strong></td>
<td></td>
</tr>
<tr>
<td><strong>Sound Plugin and API</strong></td>
<td></td>
</tr>
<tr>
<td>A. Media devices do not use these Plugin and APIs.</td>
<td></td>
</tr>
</tbody>
</table>
Annex B Utilities for Using Web UI Components

This section describes utilities for using LG Web UI components.

- Converting HTML Components into LG Components
- JavaScript Handler for UI Components
- Defining the Next Focus

Converting HTML Components into LG Components

If you call 'LGE.initialize()' method in HTML file, web engine searches markup interface and converts HTML UI components into LG Web UI components with LG look and feel GUI. Then, it generates LG Web UI Component APIs that a web developer can use.

For example, there is a button in HTML file before calling the 'LGE.initialize()' method.

![HTML UI component - button](image)

```
<input type="button" lge-type="Button" value="This is Button">
```

After calling 'LGE.initialize()' method, the button component is converted into LG UI component in HTML file and LG Web UI Component APIs are generated by web engine.

This function must be called after the DOM is fully loaded. If you want to use a JavaScript Open API, you have to call 'LGE.initialize()' before using it. It is recommended to call 'LGE.initialize()' in the beginning of the JavaScript code.

**Example**

```
LGE.initialize();
```

```
LGE.initialize('#contents');
```

![LG UI component - button](image)

JavaScript Handler for UI Components

Developers can handle all UI components. Also, developers can access APIs.

**Example**

```javascript
var button = LGE.UI.getComp("button");
button.setText("Hello");
```
Defining the Next Focus

Every UI component gets focus by using direction key. When pressing the direction key, focus is moved to the next UI component. By adding some attributes, the next focus to the UI component can be defined.

There are following three ways to define the next focus.

- Adding attribute in lge-attr
- Adding parameters in JavaScript constructor
- Calling direction key event
### Adding attribute in lge-attr

Developers can define the next focus to UI component by adding attribute in lge-attr.

There are four types of attributes: (tab-right, tab-left, tab-up, and tab-down) Each attribute has value that is Element ID of the component in document.

**Example**

```html
<input lge-type='Button'
lge-attr="tab-right:'otherButtonID1' tab-left:'otherButtonID2';"
type='button' value='button' />`
```
Adding parameters in JavaScript constructor

Developers can define the next focus to UI component by adding parameters in constructor.

There are four types of parameters. (tabRight, tabLeft, tabUp, and tabDown) Each parameter has value that is Element ID of the component in document.

**Example**

```javascript
var btn1 = LGE.UI.Button({
  type:'input',
  value:'text1',
  iconType:'plus',
  selector:("td2"),
  tabRight:'otherButtonID1',
  tabDown:'otherButtonID2',
});
```
Calling direction key event

Also, developers can call Direction Key event using JavaScript methods.

There are four types of methods. (tabRight, tabLeft, tabUp, and tabDown)

**Example**

```javascript
$('#button').bind('click', function()
    button.tabRight();
});
```