

AUDIO

This class manipulates volume with deviceapis.audiocontrol object and defines functions for volume control.

Add the following line for Audio class into a html file your own :

```
<object id="plugin" classid="clsid:SAMSUNG-INFOLINK-SEF" style="width:0px; height:0px;"></object>
```

You can declare Audio class like this :

```
ex) plugin = document.getElementById("plugin");

    plugin.Open('Audio', '1.000', 'Audio');

    plugin.Execute('method', 'parm1', 'parm2');
```

Contents

Constants

Methods

[CheckExternalOutMode](#)

[GetExternalOutMode](#)

[GetMute](#)

[GetOutputDevice](#)

[GetUserMute](#)

[GetVolume](#)

[SetRelativeVolume](#)

[SetUserMute](#)

[SetVolume](#)

Constants

Name	Value	Description
PL_AUDIO_AUDIO_OUT_MODE_PCM	0	The PCM mode
PL_AUDIO_AUDIO_OUT_MODE_DOLBY	1	The Dolby mode
PL_AUDIO_AUDIO_OUT_MODE_DTS	2	The DTS mode
PL_AUDIO_OUTPUT_DEVICE_MAIN_SPEAKER	0	The main speaker
PL_AUDIO_OUTPUT_DEVICE_EARPHONE	1	The earphone
PL_AUDIO_OUTPUT_DEVICE_SUBWOOFER	2	The subwoofer
PL_AUDIO_OUTPUT_DEVICE_EXTERNAL	3	The external speaker such as component audio, composite audio and so on
PL_AUDIO_OUTPUT_DEVICE_RECEIVER	4	The receiver such as HDMI, SPDIF and so on

Methods

CheckExternalOutMode

Description

The CheckExternalOutMode function checks whether external out mode can be changed to input value.

Parameters	■audioOutMode - PL_AUDIO_AUDIO_OUT_MODE - PCM, DOLBY, DTS, and so on
Return	■Success - 1 : the mode can be changed, ■Fail - negative number : the mode cannot be changed.
Emulator Support	Y
SDK Constraint	None

Example

```
var audioOutMode = SefPlugin.Execute("GetExternalOutMode");
alert("AudioPlugin.GetExternalOutMode() = " + audioOutMode);

if ( audioOutMode == PL_AUDIO_AUDIO_OUT_MODE_PCM ) {

    var checkAudioOutMode = SefPlugin.Execute ("CheckExternalOutMode", PL_AUDIO_AUDIO_OUT_MODE_DOLBY);
    if ( 0 == checkAudioOutMode ) {

        var result = SefPlugin.Execute ("SetExternalOutMode", PL_AUDIO_AUDIO_OUT_MODE_DOLBY);
    }
}

alert("SefPlugin.Execute GetOutputDevice() = " + SefPlugin.Execute("GetOutputDevice"));
```

GetExternalOutMode

Description

The GetExternalOutMode function returns external out mode (PCM, DOLBY, DTS, and so on).

Parameters	none
Return	■Success - PL_AUDIO_AUDIO_OUT_MODE ■Fail - negative number
Emulator Support	Y
SDK Constraint	None

Example

```
var audioOutMode = SefPlugin.Execute("GetExternalOutMode");

if ( audioOutMode == PL_AUDIO_AUDIO_OUT_MODE_PCM ) {

    var checkAudioOutMode = SefPlugin.Execute("CheckExternalOutMode", PL_AUDIO_AUDIO_OUT_MODE_DOLBY);

    if ( 0 == checkAudioOutMode ) {

        var result = SefPlugin.Execute("SetExternalOutMode", PL_AUDIO_AUDIO_OUT_MODE_DOLBY);
    }
}

alert("SefPlugin.GetOutputDevice = " + SefPlugin.Execute("GetOutputDevice"));
```

GetMute

Description

This API is used to returns the muting state of device

Parameters	none
Return	■Number - success : returns 1 - fail : returns negative number
Emulator Support	Yes
SDK Constraint	None

Example

```
var ret = SelfPlugin.Execute("GetMute");
```

```
console.log("Result = " + ret);
```

GetOutputDevice

Description

The GetOutputDevice function gets the audio device information used for audio output.

Parameters	none
Return	<div>■Success</div> <div>- PL_AUDIO_OUTPUT_DEVICE</div> <div>■Fail</div> <div>- error code</div>
Emulator Support	Y
SDK Constraint	None

Example

```
var outputDevice;
outputDevice = SelfPlugin.Execute("GetOutputDevice");

if( outputDevice == PL_AUDIO_OUTPUT_DEVICE_MAIN_SPEAKER )
{
    ...
}
else if( outputDevice == PL_AUDIO_OUTPUT_DEVICE_EARPHONE )
{
    ...
}
else if( outputDevice == PL_AUDIO_OUTPUT_DEVICE_SUBWOOFER )
{
    ...
}
else if( outputDevice == PL_AUDIO_OUTPUT_DEVICE_EXTERNAL )
{
    ...
}
else if( outputDevice == PL_AUDIO_OUTPUT_DEVICE_RECEIVER )
{
    ...
}
else
{
    ...
}
```

GetUserMute

Description

The GetUserMute function gets information about the DTV platform's UserMute state.

Parameters	none
Return	<div>■Success</div> <div>- PLR_TRUE : UserMute is on,</div> <div>- PLR_FALSE : UserMute is off</div> <div>■Fail</div> <div>- negative number</div>
Emulator Support	Y
SDK Constraint	None

Example

```

var userMute = SelfPlugin.Execute("GetUserMute");
if( PLR_TRUE == userMute ){

    ... UserMute On ...
}
else if( PLR_FALSE == userMute ){

    ... UserMute Off ...
}
else {

    ... Error ...
}

```

GetVolume

Description

The GetVolume function gets the volume level of the DTV platform. The range of volume is 0 ~ 100.

Parameters	none
Return	<ul style="list-style-type: none"> ■Success <ul style="list-style-type: none"> - value between 0 and 100 ■Fail <ul style="list-style-type: none"> - negative number
Emulator Support	Y
SDK Constraint	None

Example

```
var volume;
```

```

volume = SelfPlugin.Execute("GetVolume");
alert(volume);

```

SetRelativeVolume

Description

This API is used to sets the volume with given level based on the current level

Parameters	<ul style="list-style-type: none"> ■level_of_vol <ul style="list-style-type: none"> - Number - the level of volume
Return	<ul style="list-style-type: none"> ■Number <ul style="list-style-type: none"> - success : returns 1 - fail : returns negative number
Emulator Support	Yes
SDK Constraint	None

Example

```

var ret = SelfPlugin.Execute("SetRelativeVolume","10");
If(ret != true) { console.log("Failed! Result = " + ret); }

```

SetUserMute

Description

The SetUserMute plugin function sets the state of UserMute in DTV platform.

Parameters	<ul style="list-style-type: none"> ■userMute <ul style="list-style-type: none"> - int - 1 : UserMute state 0 : Non-UserMute state
Return	<ul style="list-style-type: none"> ■Success <ul style="list-style-type: none"> - PLR_TRUE ■Fail <ul style="list-style-type: none"> - negative number
Emulator Support	Y
SDK Constraint	None

Example	
SefPlugin.Execute("SetUserMute", 1);	
SetVolume	
Description	
Set Volume level (0~100)	
Parameters	■volume level - Number - 0 ~ 100
Return	■Void
Emulator Support	Y
SDK Constraint	None
Example	
.	