

MICROPHONE.MICROPHONE

This class provides some functions to use a microphone such as information of a microphone, enable/disable, available effects, volume control, etc.

Add the following line for microphone.Microphone class into a html file your own :

```
<script type="text/javascript" src="$MANAGER_WIDGET/Common/webapi/1.0/deviceapis.js"></script>
```

You can declare microphone.Microphone class like this :

```
ex) var microphone = deviceapis.microphone;
```

Contents

Methods

[disableDevice](#)
[enableDevice](#)
[getDeviceID](#)
[getEnabledEffects](#)
[getName](#)
[getSupportedEffects](#)
[getUniqueID](#)
[getVolumeLevel](#)
[play](#)
[setEffect](#)
[setVolumeLevel](#)
[stop](#)

Methods

disableDevice	
Description	
disable the microphone device	
Parameters	none
Return	■Boolean - true : call is successful - false : call is not successful
Emulator Support	Yes
SDK Constraint	None
Example	

```

var microphone = window.deviceapis.microphone || {};
var mic = null;

Main.keyDown = function()
{
    microphone.getMicrophones(Main.onMicrophoneObtained);
    ....
}

Main.onMicrophoneObtained = function(mics)
{
    mic = mics[0];
}

bool fResult = mic.disableDevice();

```

enableDevice

Description

enable the microphone device

Parameters	<ul style="list-style-type: none"> ■format <ul style="list-style-type: none"> - Unsigned short - audio data format from microphone(MICROPHONE_FORMAT). * MICROPHONE_FORMAT_SIGNED_16BIT_LITTLE_ENDIAN ■framerate <ul style="list-style-type: none"> - Unsigned short - framerate of retrieved audio data(MICROPHONE_FRAMERATE). * MICROPHONE_FRAMERATE_48000
Return	<ul style="list-style-type: none"> ■Boolean <ul style="list-style-type: none"> - true : call is successful - false : call is not successful
Emulator Support	Yes
SDK Constraint	None

Example

```

var microphone = window.deviceapis.microphone || {};
var mic = null;

Main.keyDown = function()
{
    microphone.getMicrophones(Main.onMicrophoneObtained);
    ....
}

Main.onMicrophoneObtained = function(mics)
{
    mic = mics[0];
}

bool fResult = mic.enableDevice(MICROPHONE_FORMAT_SIGNED_16BIT_LITTLE_ENDIAN, MICROPHONE_FRAMERATE_48000)

```

getDeviceID

Description

get device ID of the device.

Parameters	none
Return	<ul style="list-style-type: none"> ■DOMString <ul style="list-style-type: none"> - device ID
Emulator Support	Yes
SDK Constraint	None

Example

```
var microphone = window.deviceapis.microphone || {};  
var mic = null;  
  
Main.keyDown = function()  
{  
    microphone.getMicrophones(Main.onMicrophoneObtained);  
    ....  
}  
  
Main.onMicrophoneObtained = function(mics)  
{  
    mic = mics[0];  
}  
  
string strDeviceID = mic.getDeviceID();
```

getEnabledEffects

Description

get all enabled effects of the microphone

Parameters	none
Return	■Unsigned short - all enabled effects of the microphone.
Emulator Support	Yes
SDK Constraint	None

Example

```
var microphone = window.deviceapis.microphone || {};  
var mic = null;  
  
Main.keyDown = function()  
{  
    microphone.getMicrophones(Main.onMicrophoneObtained);  
    ....  
}  
  
Main.onMicrophoneObtained = function(mics)  
{  
    mic = mics[0];  
}  
  
unsigned short nEffects = mic.getEnabledEffects();
```

getName

Description

get name of the device.

Parameters	none
Return	■DOMString - name of the device
Emulator Support	Yes
SDK Constraint	None

Example

```

var microphone = window.deviceapis.microphone || {};
var mic = null;

Main.keyDown = function()
{
    microphone.getMicrophones(Main.onMicrophoneObtained);
    ....
}

Main.onMicrophoneObtained = function(mics)
{
    mic = mics[0];
}

string strName = mic.getName();

```

getSupportedEffects

Description

get all effects that can be supported by the microphone

Parameters	none
Return	■Number - Unsigned short - audio effect(MICROPHONE_EFFECT) * MIC_EFFECT_REVERB
Emulator Support	Yes
SDK Constraint	None

Example

```

var microphone = window.deviceapis.microphone || {};
var mic = null;

Main.keyDown = function()
{
    microphone.getMicrophones(Main.onMicrophoneObtained);
    ....
}

Main.onMicrophoneObtained = function(mics)
{
    mic = mics[0];
}

unsigned short nEffects = mic.getSupportedEffects();

```

getUniqueID

Description

get unique ID of the device.

Parameters	none
Return	■DOMString - uniqueID - address/uniqueID of the device
Emulator Support	Yes
SDK Constraint	None

Example

```

var microphone = window.deviceapis.microphone || {};
var mic = null;

Main.keyDown = function()
{
    microphone.getMicrophones(Main.onMicrophoneObtained);
    ....
}

Main.onMicrophoneObtained = function(mics)
{
    mic = mics[0];
}

string strUniqueID = mic.getUniqueID();

```

getVolumeLevel

Description

get the level of the microphone's volume Actually, this volume means a percentage of master volume which is controlled by volume key on the remote controller. For example, when the master volume is 80, if this API returns 50, the actual microphone volume is 40. $[80 * (50/100) = 40]$ The formula to get actual microphone volume is like this: $Master-Volume * (Microphone-volume/100) = Actual microphone-volume$.

Parameters	none
Return	■Number - Unsigned short - volume(from 0 to 100) is returned.
Emulator Support	Yes
SDK Constraint	None

Example

```

var microphone = window.deviceapis.microphone || {};
var mic = null;

Main.keyDown = function()
{
    microphone.getMicrophones(Main.onMicrophoneObtained);
    ....
}

Main.onMicrophoneObtained = function(mics)
{
    mic = mics[0];
}

unsigned short nVolume = mic.getVolumeLevel();

```

play

Description

start to play the microphone device, it means user starts to hear their voice from the TV speaker. If play() is called when there is already some background sound playing with video or image drawing, it may occur sync problem between the background sound and video or image drawing. * On windows PC, the most common reason for play() to return false is when none of the audio output devices such as speaker or earphone are connected.

Parameters	none
Return	■Boolean - result - true : call is successful, false : call is not successful
Emulator Support	Yes
SDK Constraint	None

Example
<pre> var microphone = window.deviceapis.microphone {}; var mic = null; Main.keyDown = function() { microphone.getMicrophones(Main.onMicrophoneObtained); } Main.onMicrophoneObtained = function(mics) { mic = mics[0]; } bool fResult = mic.play(); </pre>

setEffect

Description

enable/disable the effect.

Parameters	<ul style="list-style-type: none"> ■effect <ul style="list-style-type: none"> - Unsigned short - audio effect(MICROPHONE_EFFECT) <ul style="list-style-type: none"> * MIC_EFFECT_REVERB ■enable <ul style="list-style-type: none"> - Boolean - true : set the effect, false : release the effect ■param1 (Optional) <ul style="list-style-type: none"> - Float - this value is used to set a tempo value when MICROPHONE_EFFECT_FILTER effect used. The tempo value's range is - 95.0~5000.0 ■param2 (Optional) <ul style="list-style-type: none"> - Float - this value is used to set a pitch value when MICROPHONE_EFFECT_FILTER effect used. The pitch value's range is -60.0~60.0 ■param3 (Optional) <ul style="list-style-type: none"> - Float - this value is used to set a rate change value when MICROPHONE_EFFECT_FILTER effect used. The rate change value's range is - 95.0~5000.0
Return	<ul style="list-style-type: none"> ■Boolean <ul style="list-style-type: none"> - result - true : call is successful, false : call is failure.
Emulator Support	Yes
SDK Constraint	None

Example

```

var microphone = window.deviceapis.microphone || {};
var mic = null;

Main.keyDown = function()
{
    microphone.getMicrophones(Main.onMicrophoneObtained);
    ....
}

Main.onMicrophoneObtained = function(mics)
{
    mic = mics[0];
}

bool fResult = mic.setEffects(MIC_EFFECT_REVERB, true);
fResult = mic.setEffects(MIC_EFFECT_FILTER, true, 0 ,5, 50);

```

setVolumeLevel

Description

set the level of the microphone's volume. The range of level is from 0 to 100. Actually, this volume means a percentage of master volume which is controlled by volume key on the remote controller. The formula to get actual microphone volume is like this: $\text{Master-Volume} \times (\text{Microphone-volume} / 100) = \text{Actual microphone-volume}$. According to the formula, if you set the microphone volume as zero, the actual microphone volume would be zero.

Parameters

■volume
- Unsigned short
- volume level (Minimum: 0 to Maximum : 100)

Return

■Boolean
- result
- true : call is successful, false : failure

Emulator Support

Yes

SDK Constraint

None

Example

```
var microphone = window.deviceapis.microphone || {};  
var mic = null;  
  
Main.keyDown = function()  
{  
    microphone.getMicrophones(Main.onMicrophoneObtained);  
    ....  
}  
  
Main.onMicrophoneObtained = function(mics)  
{  
    mic = mics[0];  
}  
  
bool fResult = mic.setVolumeLevel(100);
```

stop

Description

stop playing the microphone device, it means user stops hearing their voice from the TV speaker. If stop() is called when there is already some background sound playing with video or image drawing, it may occur sync problem between the background sound and video or image drawing.

Parameters

none

Return

■Boolean
- result
- true : call is successful, false : failure

Emulator Support

Yes

SDK Constraint

None

Example

```
var microphone = window.deviceapis.microphone || {};  
var mic = null;  
  
Main.keyDown = function()  
{  
    microphone.getMicrophones(Main.onMicrophoneObtained);  
    ....  
}  
  
Main.onMicrophoneObtained = function(mics)  
{  
    mic = mics[0];  
}  
  
bool fResult = mic.stop();
```