

# Using a Camera Application

Published 2014-10-28 | (Compatible with SDK 5.0,5.1 and 2014 models)

This tutorial describes how to use Camera APIs. Camera Application implements to display camera video on TV when play button is clicked, to undisplay camera video on TV when stop button is clicked.

Contents

## Source Files

### Camera Application

#### Setting Up Initial App Configuration

#### How to start and stop the Camera

### How to use at SDK emulator

## Source Files

Note

The files needed for the sample application are [here](#).

## Camera Application

This tutorial task consists of the following parts:

### Setting Up Initial App Configuration

### How to start and stop the Camera

## Setting Up Initial App Configuration

This section demonstrates the initial configuration of the App. It assumes that basic App creation is already understood by the reader.

1. Start the SDK for Samsung TV Apps.
2. Create a new App. The App is launched on the browser running on the SDK. All the JavaScript files are also included providing the dynamic behaviour to the App.
3. This GetCameraState() function checks the current camera state. And The RegisterEventCallback function registers a callback to receive camera related event. When the event arrives, the registered callback function is called with event value. The supported events are as follows, By registering event callback, application can know the change of camera status without polling the camera status.

```

camera.handleShow = function () {
    var state = deviceapis.camera.GetCameraState();

    if(state == "2" || state == "3"){

    }else if(state == "0" || state == "1"){
        deviceapis.camera.RegisterEventCallback(initCallback);
    }

    function initCallback(state){
        if(state == "2" || state == "3"){

        }else if(state == "0"){
            deviceapis.camera.StopCamVideo();
        }
    }
}

```

## How to start and stop the Camera

under a provision, the camera can be turned on or off.

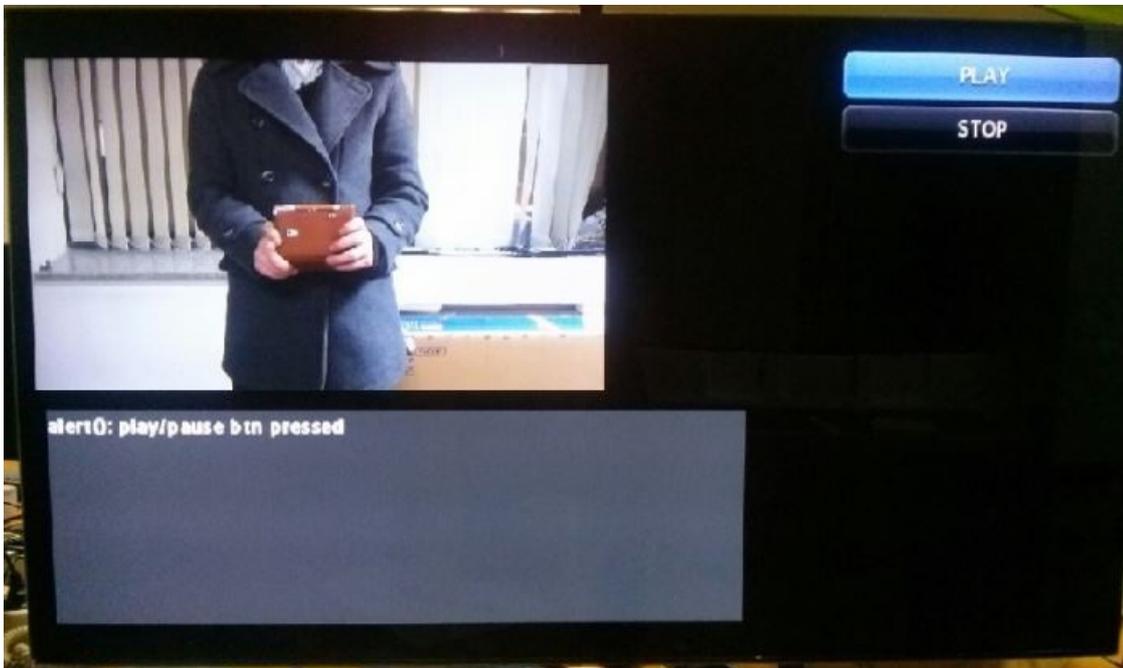
```

var Button = caph.wui.widget.Button;
var uicontext = new UIContext();
var option1 = { width:360, height:100, id:'playBtn', text:'play'};
var option2 = { width:360, height:100, id:'stopBtn', text:'stop'};

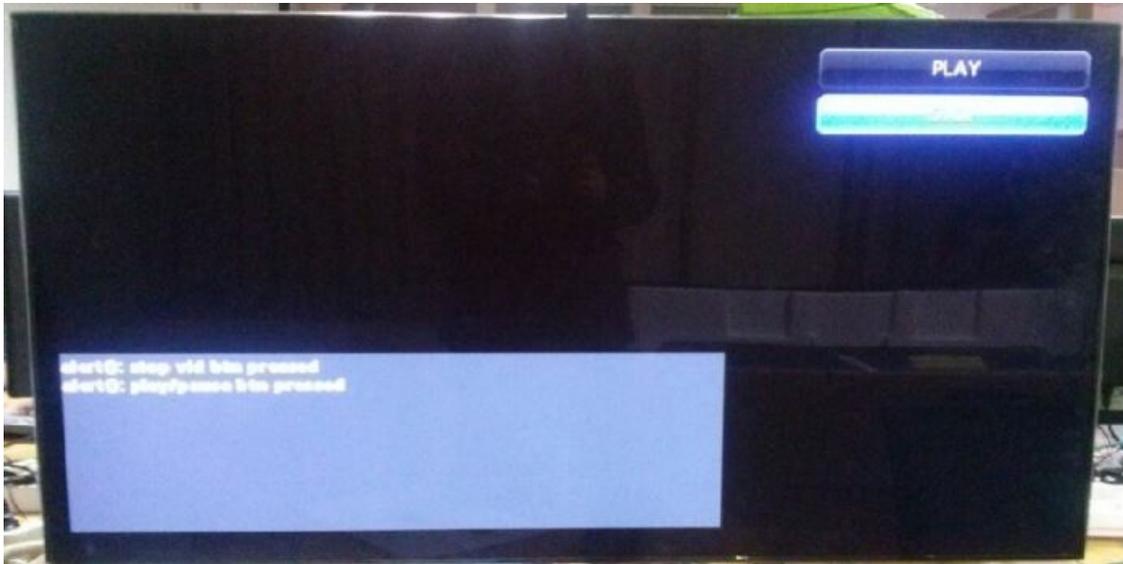
var playBtn = new Button( option1 );
playBtn.setAbsolutePosition(1050, 100, 0);
playBtn.addEventListener('click', function() {
    deviceapis.camera.StartCamVideo(21,0,968,680,deviceapis.camera.PL_CAMERA_RESOLUTION_VGA,devicea
pis.camera.PL_CAMERA_QUALITY_HIGH);
})
playBtn.render(uicontext);

var stopBtn = new Button( option2 );
stopBtn.setAbsolutePosition(1050, 200, 0);
stopBtn.addEventListener('click', function() {
    deviceapis.camera.StopCamVideo();
})
playBtn.render(uicontext);

```



**Figure 1** TV view: start camera



**Figure 2** TV view: stop camera

## How to use at SDK emulator

1. Press the Video Camera button in Virtual input.
2. Press the ... button for opening a video file instead of a real camera show.
3. If you press the play button using remote control at emulator, Camera Application implements to display camera video.  
And if you press the stop button using remote control at emulator, this app implements to undisplay camera video.

The directory structure of the application:

File/Directory	Description
Config.xml	Contains the environment to run the application on TV.
Index.html	Description of the elements display.
widget.info	Used for the App body opacity adjustment.
js/Camera.js	App sample file.