

# CImageViewer Module

Published 2014-10-28 | (Compatible with SDK 2.5,3.5,4.5,5.0,5.1 and 2011,2012,2013,2014 models)

Enables JPEG files to be played in SAMSUNG DTV.

## Contents

### Specifications

### Basic Instructions

### EVENTS

### Playing Local Files

#### Playing Images in an Application

#### Playing Images from USB

### Reference

#### CImageViewer Class

#### Methods

CImageViewer is a common module that enables JPEG files to be displayed on Samsung DTV. With this module, images can be displayed on the DTV screen with a high resolution of 1920x1080 pixels. In addition, the size, location and transition effects of the images are also displayed.

To use CImageViewer, ensure that the TV source is designated as the media source. Source transition can be made by setting <srcctl> item to y or through SetMediaSource() method of TVMW Plugin.

For more information about CImageViewer module, see [Basic Instructions](#), [Events](#), [Playing Local Files](#), and [Reference](#).

## Specifications

Features	Description
<b>Resolution</b>	1920x1080 pixels  Images are displayed in Media source. The display pixel of the application is marked as 960x540 pixels, but that of the displaying image in media source is marked as 1920x1080 pixels.
<b>Supported file format</b>	JPEG only
<b>Scaling range</b>	Designate the area where an image should be displayed. The image is scaled according to the area you set. Ensure that the resolution is set over 300 pixels. If it is lesser than 300 pixels, images can appear distorted.
<b>Transition Effects</b>	Total 10 modes are provided:  NONE, FADE1, FADE2, BLIND, SPIRAL, CHECKER, LINEAR, STAIRS, WIPE, RANDOM

## Basic Instructions

The basic instructions depict setting the location of the images, and scaled display of the images:

Include the common module

The common module is included by putting the code shown below in the <head> of index.html file.

```
<script type="text/javascript" language="javascript" src="$MANAGER_WIDGET/Common/API/CImageViewer.js"></script>
```

Create instance

Create ImageViewer instance

```
var ImageViewer = new CImageViewer('Common ImageViewer');
```

Initialize ImageViewer

Set the area where images are displayed and functions which will execute each event. The show() method shows the area managed by ImageViewer on screen.

Area designation via setFrameArea() method is executed based on basic resolution (960x540 pixel). The appointed area operates at 1920x1080 pixel resolution.

```
ImageViewer.setFrameArea(100, 100, 300, 300);
ImageViewer.setOnNetworkError(function() {
    alert('Network error occurs!');
});
```

```
ImageViewer.show();
```

Play images

Play images to ensure that they retain original proportion at the designated areas and they are indicated by scaling.

```
ImageViewer.play('$WIDGET/picture.jpg', 640, 480);
```

Close the application

When the application is closed, ImageViewer should be initialized by calling destroy() method.

#### Note

In case the application is closed without initialization while playing images in ImageViewer, fatal problems such as TV system pause might occur, and it could result in issues for using ImageViewer in other applications.

```
// Run ImageViewer.destroy() method on <body>'s onunload function
```

```
ImageViewer.destroy();
```

Events

ImageViewer loads various events for precise controlling of image display. You can operate events after registering a function to the corresponding event.

## EVENTS

Event	Description	Function
Buffering start	This event occurs during the initiation of image loading (after successful connection with server), if the image you intend to play exists on the Internet.	<a href="#">setOnBufferingStart()</a>
Buffering complete	This event occurs when the image is ready for display after loading.	<a href="#">setOnBufferingComplete()</a>
Rendering complete	This occurs when the image is completely displayed on the screen. In case of using a transition effect, this happens when the effect animation has been completed.	<a href="#">setOnRenderingComplete()</a>
Network error	This event occurs if the image URL cannot be accessed.	<a href="#">setOnNetworkError()</a>
Render error	This event occurs when image decoding and rendering fails.	<a href="#">setOnRenderError()</a>

## Playing Local Files

ImageViewer displays images in application or USB as well as on the Internet. A special path starting with

\$ is necessary to display images from application (files included in application) or USB.

## Playing Images in an Application

The \$WIDGET string indicates a directory containing an application. Images in the application can be played by sending strings as shown below to `ImageViewer.play()`.

```
ImageViewer.play('$WIDGET/picture_2.jpg', 640, 480);
```

## Playing Images from USB

The \$USB\_DIR string indicates USB. The mounting path of USB refers to the USB port at which the USB is mounted in the TV. You need the USB mounting path to display images in USB, because the TV has plural USB ports, and mounting path varies for each USB connection. The location can be found through Storage plugin.

Figuring out the mount route

The Storage plugin finds out the path to mount USB. Complete the root directory path of USB by adding the path to the \$USB\_DIR string which indicates USB.

```
// Plugin object Designated
var eStoragePlugin = document.getElementById('pluginObjectStorage');

// Provided that one USB is connected
var deviceId = eStoragePlugin.GetUSBDeviceID(0);
```

```
// Provided that USB has one partition
var mountPath = eStoragePlugin.GetUSBMountPath(deviceId, 0);
var USBRootPath = '$USB_DIR/' + mountPath; // USB root directory path
```

Your TV has more than two USB ports. You need other functions in the Storage plugin to control more than two USBs or use USB with divided partitions.

Creating file routes

The `readDir()` API of the browser gives the file information in USB. The `readDir()` API puts file information on the path in an array and returns it. An example is shown below to refer whether the `readDir()` returns information of a directory or of a file. For more information about File API, see [File API](#).

```
var fileSystemObj = new FileSystem(); // Browser API
var dirs = new Array();
var files = new Array();
var data = fileSystemObj.readDir(USBRootPath); // readDir() returns the array of file information

for (var i = 0; i < data.length; i++) {
    if (data[i].isDir) {
        dirs.push(data[i].name);
    } else {
        files.push(data[i].name);
    }
}
```

The whole path of image files in USB is created by the file information given above.

```
var USBImageFile = USBRootPath + '/' + directory + '/' + file;
```

The actual variable of `USBImageFile` could be the same as below.

```
var USBImageFile = '$USB_DIR/sda1/myphotos/picture_2.jpg'
```

Displaying image files in USB

```
ImageViewer.play(USBImageFile);
```

## Reference

### CImageViewer Class

The `CImageViewer` is a class to play images.

CImageViewer	
<b>Library path</b>	\$MANAGER_WIDGET/Common/API/CImageViewer.js
<b>Syntax</b>	new CImageViewer(objectName:String)
<b>Methods</b>	<a href="#">clearScreen()</a> <a href="#">destroy()</a> <a href="#">endSlideshow()</a> <a href="#">getStopFlag()</a> <a href="#">hide()</a> <a href="#">isEffectAvailable()</a> <a href="#">play()</a> <a href="#">prepareNext()</a> <a href="#">setFrameArea()</a> <a href="#">setOnBufferingComplete()</a> <a href="#">setOnBufferingStart()</a> <a href="#">setOnNetworkError()</a> <a href="#">setOnRenderError()</a> <a href="#">setOnRenderingComplete()</a> <a href="#">show()</a> <a href="#">showNow()</a> <a href="#">startSlideshow()</a> <a href="#">stop()</a>
<b>Parameter</b>	objectName:String Includes names to distinguish generated objects. The debug message from ImageViewer has the name on it.
<b>Return Value</b>	CImageViewer object
<b>Remarks</b>	In reality, the TV provides only one ImageViewer. Therefore, it is necessary to create one object of CImageViewer. If an image is played via ImageViewer, call <a href="#">destroy()</a> while closing the application. Since Application Manager 2.269
<b>Example</b>	<pre>var ImageViewer = new CImageViewer('Common ImageViewer');</pre>
<b>Members</b>	<a href="#">Effect</a>

## Methods

clearScreen	
The clearScreen() method deletes images being displayed on the page.	
<b>Syntax</b>	ImageViewer.clearScreen();
<b>Parameter</b>	None
<b>Return Value</b>	None

clearScreen	
<b>Enumeration</b>	None
<b>Remarks</b>	stop() does not remove the playing image. To delete the image, call clearScreen().  Since Application Manager 2.269
<b>Example</b>	<pre>var ImageViewer = new CImageViewer('Common ImageViewer');     ImageViewer.clearScreen();</pre>

destroy	
The destroy() method pauses the playing image and initializes the ImageViewer.	
<b>Syntax</b>	ImageViewer.destroy();
<b>Parameter</b>	None
<b>Return Value</b>	None
<b>Enumeration</b>	None
<b>Remarks</b>	If you have used ImageViewer, this function should be called while closing the application. It is recommended that the function registered onunload function of <body> element call this method. The function is called when the application ends.  Since Application Manager 2.269
<b>Example</b>	<pre>var ImageViewer = new CImageViewer('Common ImageViewer');     ImageViewer.destroy();</pre>

endSlideshow	
The endSlideshow() method ends the slideshow mode.	
<b>Syntax</b>	ImageViewer.endSlideshow();
<b>Parameter</b>	None
<b>Return Value</b>	None
<b>Enumeration</b>	None
<b>Remarks</b>	Call the startSlideshow() method to enter slideshow mode which includes transition effect. The prepareNext() and showNow() methods work only in slideshow mode.  Since Application Manager 2.269
<b>Example</b>	<pre>var ImageViewer = new CImageViewer('Common ImageViewer');     ImageViewer.startSlideshow();     // Use transition effect     ImageViewer.endSlideshow();</pre>

getStopFlag	
The getStopFlag() method checks whether the ImageViewer is stopped or not.	
<b>Syntax</b>	ImageViewer.getStopFlag();
<b>Parameter</b>	None
<b>Return Value</b>	Boolean If the ImageViewer is stopped, return value is true, if it is operating, return value is false.
<b>Enumeration</b>	None

getStopFlag	
<b>Remarks</b>	Since Application Manager 2.269
<b>Example</b>	<pre>var ImageViewer = new CImageViewer('Common ImageViewer'); ImageViewer.getStopFlag();</pre>

hide	
The hide() method hides areas managed by the ImageViewer.	
<b>Syntax</b>	ImageViewer.hide();
<b>Parameter</b>	None
<b>Return Value</b>	None
<b>Enumeration</b>	None
<b>Remarks</b>	Since Application Manager 2.269
<b>Example</b>	<pre>var ImageViewer = new CImageViewer('Common ImageViewer'); ImageViewer.show(); // Play images ImageViewer.stop(); ImageViewer.hide();</pre>

isEffectAvailable	
The isEffectAvailable() method checks whether a transition effect is available.	
<b>Syntax</b>	ImageViewer.isEffectAvailable();
<b>Parameter</b>	None
<b>Return Value</b>	<p>Boolean</p> <p>If user settings are feasible for the transition effect, return value is true`, if it is operating, return value is ``false.</p> <p>Since Application Manager 2.269</p>
<b>Enumeration</b>	None
<b>Remarks</b>	Some models do not support transition effects due to hardware restrictions. In these models, effect animation is not possible even in the slideshow mode.
<b>Example</b>	<pre>var ImageViewer = new CImageViewer('Common ImageViewer'); ImageViewer.isEffectAvailable();</pre>

play	
The play() method plays images.	
<b>Syntax</b>	ImageViewer.play(url:String, width:Number, height:Number);
<b>Parameter</b>	<p>url:String</p> <p>Available to display image URL, images in Application and USB.</p> <p>width:Number</p> <p>Width unit of an image is pixel. The image is scaled based on the pixels, and displayed in the designated area by setFrameArea(). Slideshow mode does not need this parameter.</p> <p>height:Number</p> <p>Height unit of an image is pixel. The image is scaled based on the pixels and displayed in the designated area by setFrameArea(). Slideshow mode does not need this parameter.</p>

play	
<b>Return Value</b>	Since Application Manager 2.269
<b>Enumeration</b>	None
<b>Remarks</b>	Playable image type is JPEG only.
<b>Example</b>	<pre>var ImageViewer = new CImageViewer('Common ImageViewer'); ImageViewer.setFrameArea(50,50,480,320); ImageViewer.play(image_url, image_width, image_height);</pre>

prepareNext	
The prepareNext() method loads the next image to execute a transition effect in slideshow mode.	
<b>Syntax</b>	ImageViewer.prepareNext(url:String, effect:ImageViewer.Effect);
<b>Parameter</b>	<p>url:String</p> <p>Available to display image URL, images in Application and USB.</p> <p>effect:ImageViewer.Effect</p> <p>Refer to the transition effect you intend to apply and effect variables in the instance of CImageViewer.</p>
<b>Return Value</b>	None
<b>Enumeration</b>	None
<b>Remarks</b>	<p>When this method is called, an image starts to load. Effect animation is shown by calling showNow() after the Buffering Complete event.</p> <p>Some models do not support transition effects due to hardware restrictions. Use <a href="#">isEffectAvailable()</a> to find out if the transition effect is available in your TV.</p> <p>Since Application Manager 2.269</p>
<b>Example</b>	<pre>var ImageViewer = new CImageViewer('Common ImageViewer'); ImageViewer.setFrameArea(0, 0, 960, 540); ImageViewer.startSlideshow(); ImageViewer.play(firstImage, 300, 400); // after Rendering complete event ImageViewer.prepareNext(NextImage, ImageViewer.Effect.FADE1); // after Buffering complete event ImageViewer.shwoNow(); // ... // after finishing slideshow ImageViewer.endSlideshow();</pre>

setFrameArea	
The setFrameArea() method sets the area within which images are displayed.	
<b>Syntax</b>	ImageViewer.setFrameArea(left:Number, top:Number, width:Number, height:Number);
<b>Parameter</b>	<p>left:Number - left coordinate of the area</p> <p>top:Number - top coordinate of the area</p> <p>width:Number - width of the area</p> <p>height:Number - height of the area</p> <p>The unit is pixel.</p>
<b>Return Value</b>	None

setFrameArea	
<b>Enumeration</b>	None
<b>Remarks</b>	<p>For example, if the image is set as setFrameArea(0, 0, 225, 200) and the upper left area of the TV screen is transferred to Media source, images at 450x400 pixels are displayed intact. If an image you intend to display is larger than the standard, it is scaled in accordance with the aspect ratio and displayed in the area.</p> <p>Since Application Manager 2.269</p>
<b>Example</b>	<pre>var ImageViewer = new CImageViewer('Common ImageViewer'); ImageViewer.setFrameArea(0, 0, 480, 320); ImageViewer.play(image_url, image_width, image_height);</pre>

setOnBufferingComplete	
The setOnBufferingComplete() method registers a function executed in case of Buffering complete event while loading an image.	
<b>Syntax</b>	ImageViewer.setOnBufferingComplete(callback:function);
<b>Parameter</b>	<p>callback:function</p> <p>The function executed in the event of Buffering complete event.</p>
<b>Return Value</b>	None
<b>Enumeration</b>	None
<b>Remarks</b>	<p>Buffering complete event occurs after image loading. If this event takes places after calling prepareNext(), it means that the image is ready for the transition effect. At this stage, calling showNow() begins effect animation.</p> <p>Since Application Manager 2.269</p>
<b>Example</b>	<pre>var ImageViewer = new CImageViewer('Common ImageViewer'); ImageViewer.setOnBufferingComplete(function() {     // Add your code here });</pre>

setOnBufferingStart	
The setOnBufferingStart() method registers a function executed in case of Buffering start event.	
<b>Syntax</b>	ImageViewer.setOnBufferingStart(callback:function);
<b>Parameter</b>	<p>callback:function</p> <p>The function executed in the event of Buffering start event.</p>
<b>Return Value</b>	None
<b>Enumeration</b>	None
<b>Remarks</b>	<p>If the image you intend to play is on the Internet, this event happens when starting to load actual image data after connecting to the server successfully.</p> <p>Since Application Manager 2.269</p>
<b>Example</b>	<pre>var ImageViewer = new CImageViewer('Common ImageViewer'); ImageViewer.setOnBufferingStart(function() {     // Add your code here });</pre>

setOnNetworkError	
The setOnNetworkError() method registers a function executed in case of Network error event.	
<b>Syntax</b>	ImageViewer.setOnNetworkError(callback:function);



setOnNetworkError	
<b>Parameter</b>	callback:function  The function executed in the event of Network error event.
<b>Return Value</b>	None
<b>Enumeration</b>	None
<b>Remarks</b>	This event occurs in case of network errors while loading images.  Since Application Manager 2.269
<b>Example</b>	<pre>var ImageViewer = new CImageViewer('Common ImageViewer'); ImageViewer.setOnNetworkError(function() {     // Add your code here });</pre>

setOnRenderError	
The setOnRenderError() method registers a function executed in case of Decoding or Rendering error.	
<b>Syntax</b>	ImageViewer.setOnRenderError(callback:function);
<b>Parameter</b>	callback:function  The function executed in case of Decoding or Rendering error.
<b>Return Value</b>	None
<b>Enumeration</b>	None
<b>Remarks</b>	This event takes place when an error occurs while decoding and rendering after buffering and while trying to play a format which is not basically supported (image files except for JPEG).  Since Application Manager 2.269
<b>Example</b>	<pre>var ImageViewer = new CImageViewer('Common ImageViewer'); ImageViewer.setOnRenderError(function() {     // Add your code here });</pre>

setOnRenderingComplete	
The setOnRenderingComplete() method registers a function executed in case of Rendering complete event.	
<b>Syntax</b>	ImageViewer.setOnRenderingComplete(callback:function);
<b>Parameter</b>	callback:function  The function executed in the event of Rendering complete event.
<b>Return Value</b>	None
<b>Enumeration</b>	None
<b>Remarks</b>	This event occurs when the image is displayed completely on the screen. If a transition effect is used, the event occurs when effect animation has ended.  Since Application Manager 2.269
<b>Example</b>	<pre>var ImageViewer = new CImageViewer('Common ImageViewer'); ImageViewer.setOnRenderingComplete(function() {     // Add your code here });</pre>

show	
The show() method shows the area managed by ImageViewer on screen.	
<b>Syntax</b>	ImageViewer.show();
<b>Parameter</b>	None
<b>Return Value</b>	None
<b>Enumeration</b>	None
<b>Remarks</b>	Since Application Manager 2.269
<b>Example</b>	<pre>var ImageViewer = new CImageViewer("Common ImageViewer"); ImageViewer.show(); // Play images ImageViewer.stop(); ImageViewer.hide();</pre>

showNow	
The showNow() method begins to play the transition effect. Loading the image should be ended with the prepareNext() method.	
<b>Syntax</b>	ImageViewer.showNow();
<b>Parameter</b>	None
<b>Return Value</b>	None
<b>Enumeration</b>	None
<b>Remarks</b>	<p>Some models do not support transition effects due to hardware restrictions. See <a href="#">isEffectAvailable()</a>.</p> <p>Calling prepareNext() starts loading of an image. Effect animation is shown by calling showNow() after Buffering complete event.</p> <p>Since Application Manager 2.269</p>
<b>Example</b>	<pre>var ImageViewer = new CImageViewer('Common ImageViewer'); ImageViewer.setFrameArea(0,0,960,540); ImageViewer.startSlideshow(); ImageViewer.play(firstImage, 300, 400); // after Rendering complete event ImageViewer.prepareNext(NextImage, ImageViewer.Effect.FADE1); // after Buffering complete event ImageViewer.shwoNow(); // ... // after finishing slideshow ImageViewer.endSlideshow();</pre>

startSlideshow	
The startSlideshow() method starts the slideshow mode.	
<b>Syntax</b>	ImageViewer.startSlideshow();
<b>Parameter</b>	None
<b>Return Value</b>	None
<b>Enumeration</b>	None
<b>Remarks</b>	<p>If you want to close slideshow mode, call endSlideshow(). Methods such as prepareNext() and showNow() are available only in slideshow mode.</p> <p>Since Application Manager 2.269</p>

startSlideshow	
<b>Example</b>	<pre>var ImageViewer = new CImageViewer('Common ImageViewer'); ImageViewer.startSlideshow(); // Use transition effect ImageViewer.endSlideshow();</pre>

stop	
The stop() method stops image processing.	
<b>Syntax</b>	ImageViewer.stop();
<b>Parameter</b>	None
<b>Return Value</b>	None
<b>Enumeration</b>	None
<b>Remarks</b>	<p>Stops image processing such as image loading and transition effect animation. To remove playing images from the screen, see clearScreen().</p> <p>Since Application Manager 2.269</p>
<b>Example</b>	<pre>var ImageViewer = new CImageViewer('Common ImageViewer'); ImageViewer.setFrameArea(50, 50, 480, 320); ImageViewer.play(image_url, image_width, image_height); ImageViewer.stop();</pre>