Contents

一 Using Examples

```
<html>
    <head>
        <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
        <title>Virtual Tour Created By Easypano</title>
        <script type="text/javascript">
                var jsReady = true;
                function isJsReady(){
                        return jsReady;
                }

                function pageInit() {
                        jsReady = true;
                }

                function thisMovie(movieName) {
                        return document.getElementById(movieName);
                }
                function left(){

                         <!- the way to allocate Easypano Tourweaver API

function    left( panRate:int=20, id:String=""):Boolean; pan left scene ->

                        thisMovie("sotester").twAPI("left", "1");
                }
                function right() {

                        <!- the way to allocate Easypano Tourweaver API

function    right( panRate:int=20, id:String=""):Boolean; pan right scene ->

                        thisMovie("sotester").twAPI("right", "1");
                }
        </script>
    </head>
    <body leftMargin="0" topMargin="0" rightMargin="0" bottomMargin="0">
    <script type="text/javascript" src="swfobject.js"></script>
    <div id="flashcontent">
            To view virtual tour properly, Flash Player 9.0.28 or later version is needed.
            Please download the latest version of <a href="http://www.adobe.com/go/getflashplayer"
target="_blank">Flash Player</a> and install it on your computer.
```

```
    </div>

  <!- add flash player to page ->
    <script type="text/javascript">
            var so = new SWFObject("TWViewerp_g.swf", "sotester", "800", "600", "9.0.0", "#000000");
            so.addParam("allowNetworking", "all");
            so.addParam("allowScriptAccess", "always");
            so.addParam("allowFullScreen", "true");
            so.addParam("scale", "noscale");

            so.addVariable("lwImg", "");
            so.addVariable("lwBgColor", "255,255,255,255");
            so.addVariable("lwBarBgColor", "255,255,255,255");
            so.addVariable("lwBarColor", "255,0,255,0");
            so.addVariable("lwBarBounds", "-240,-120,480,8");
            so.addVariable("lwlocation", "7");
            so.addVariable("lwShowLoadingPercent", "true");
            so.addVariable("lwTextColor", "255,0,0,0");
            so.addVariable("iniFile", "config_Project3.bin");
            so.addVariable("progressType", "0");
            so.addVariable("swfFile", "");
            so.addVariable("percentType", "0");
            so.addVariable("sizeFile", "filesize.txt");
            so.addVariable("href", location.href);
            so.write("flashcontent");
   </script>
      <!- add a button to make the scene pan left

  ->
      <input id="Button2"type="button" value="left" onclick="left()" />
      <!- add a button to make the scene pan right

  ->
      <input id="Button1"type="button" value="right" onclick="right()" />
    </body>
</html>
```

In the codes of the above-mentioned examples of using API, the blue codes are the main part. It mainly function is to display how to allocate Easypano Tourweaver API to realize the specific functions. twAPI is the inherent property of Easypano Tourweaver API. You can use it to allocate all the internal functional functions. The usage is: **thisMovie("SWFObject ID").twAPI** （ “functionName”，“parameter1”，“parameter2”,…）;

二 Description of the interface

2.1 switchToScene

Allocate the name, fov and transition effect of the switching scene, rotate automatically after switching.

```
function  switchToScene(
    sceneName:String,
    pan:Number = NaN,
    tilt:Number = NaN,
    fov:Number = NaN,
    transitonEffect:String = null,
    transitionTime:Number = NaN,
    srcPan1:Number = NaN,
    srcTilt1:Number = NaN,
    srcFov1:Number = NaN,
    srcPan2:Number = NaN,
    srcTilt2:Number=NaN,
    srcFov2:Number=NaN,
    sizeMode1:String=″ActualSize″,
    sizeMode2:String=″ActualSize″,
    sizeMode3:String=″ActualSize″,
    isChangeState:Boolean=true
):Boolean;
```

Parameters

sceneName

[String]   Scene name

Pan

[Number] As to panoramic images, it refers to rotating to the Pan angle of the current scene.

As to plan images, it refers to switching to the pixel point of the horizontal direction.

Tilt

[Number] As to panoramic images, it refers to rotating to the Tilt angle of the current scene.

As to plan images, it refers to switching to the pixel point of the vertical direction.

FOV

[Number] As to panoramic images, it refers to rotating to the Fov angle of the current scene.As to plan images, it refers to how many pixels of the original image are placed in the horizontal direction of the player.

transitonEffect

[String] Transition effect type: default value: null

transitionTime

[Number] It refers to the duration of the transition effect when switching to next scene.

srcPan1

[Number] transition Pan.

srcTilt1

[Number] transition Tilt.

srcFov1

[Number] transition Fov.

srcPan2

[Number] transition Pan.

srcTilt2

[Number] transition Tilt.

srcFov2

    [Number]   transition Fov.

sizeMode1

    [String]   Size model,default:″ActualSize″.

sizeMode2

    [String]   Size model,default:″ActualSize″.

sizeMode3

    [String]   Size model,default:″ActualSize″.

isChangeState

    [Boolean]

Return：

true	success
false	fail

2.2 previousScene

Back to the previous scene of the current scene in the slideshow list

```
function  previousScene():Boolean;
```

Return

true	success
false	fail

2.3 nextScene

Go forward to the next scene of the current scene in the slideshow list

```
function  nextScene():Boolean;
```

Return

true	success
false	fail

2.4 backward

Backward to previous scene of the current scene in the playing history

```
function  backward():Boolean;
```

Return

true	success
false	fail

2.5 forward

Forward to next scene of the current scene in the playing history

```
function  forward():Boolean;
```

Return

true	success
false	fail

2.6 continuousForward

Whether to play the previous scene by continuous trigger (mainly used in prevFrame)

```
function  get continuousForward():Boolean;
```

Return

true	success
false	fail

Set the function of continuous trigger to play the previous scene(mainly used in prevFrame)

```
function  set continuousForward(value:Boolean):void;
```

Parameters

value

[Boolean]	true	continuation / false discontinuation

2.7 continuousBackward

Whether to play the next scene by continuous trigger (mainly used in prevFrame)

```
function  get continuousBackward():Boolean;
```

Return

true	success
false	fail

Set the function of continuous trigger to play the next scene (mainly used in prevFrame)

```
function  set continuousBackward(value:Boolean):void;
```

Parameters

value

[Boolean]	true	continuation / false discontinuation

2.8 up

Tilt up the scene automatically   tilt up automatically by clicking

```
function  up(
    tiltRate:int = 20,
    id:String = ""
):Boolean;
```

Parameters

tiltRate

[int]	Tilt up rate, default 20

id

[String]

Return

true	success
false	fail

2.9 down

Tilt down the scene automatically   tilt down automatically by clicking

```
function  down(
    tiltRate:int = 20,
    id:String = ""
):Boolean;
```

Parameters

Tilt down rate, default 20.

id

Return

true	success
false	fail

2.10 left

pan left automatically, pan left automatically by clicking

```
function  left(
    panRate:int=20,
    id:String=""
):Boolean;
```

Parameters

panRate

[int]	pan left rate, default

id

Return

True	success
false	fail

2.11 right

pan right automatically, pan left automatically by clicking

```
function  right(
    panRate:int = 20,
    id:String = ""
 ):Boolean;
```

Parameters

pan right rate default

id

Return

True	success
false	fail

2.12 zoomIn

Zoom in scene, zoom in by clicking, zoom in continuously by holding on clicking,

Release to stop zooming in

```
function  zoomIn(
    FOVRate:int = 20,
    id:String = ""
):Boolean;
```

Parameters

FOVRate

[int]	zoom in rate, default 20

id

[String]

Return

true	success

false	fail

2.13 zoomOut

Zoom out scene, zoom out by clicking, zoom out continuously by holding on clicking,

Release to stop zooming out

```
function  zoomOut(
    FOVRate:int=20,
    id:String=""
):Boolean;
```

Parameters

FOVRate

[int]	zoom out rate , default20.

Id

[String]

Return

true	success
false	fail

2.14 reset

Recover to the initial position of the current scene

```
function  reset():Boolean;
```

Return

True	success
false	fail

2.15 stop

Stop rotating/ zooming in/zooming out of the current scene

```
function  stop():Boolean;
```

Return

true	success
false	fail

2.16 playPauseMovie

Play or pause the movie named Name in the configuration files

```
function  playPauseMovie(
    name:String = null,
    loop:String = "true"
):Boolean;
```

Parameters

Name

[String]   name of the movie, if null, means the current movie

Loop

[String]   Whether to loop the movie, if true, the movie will play from start when it plays at the end, otherwise, the movie will stop when it plays at the end. Default:true

Return

true	success
false	fail

2.17 stopMovie

Stop playing the current movie

```
function  stopMovie():Boolean;
```

Return

true	success
false	fail

2.18 switchToMap

Switch the map in Map Viewer to the map image named Name

```
function  switchToMap(
    name:String,
    transitionEffect:String = null,
    transitionEffectTime:Number = NaN,
    id:String = ””
):Boolean;
```

Parameters

name

[String] Refers to the name of the switched map

transitionEffect

[String] Refers to the transition effect of switching to the next map. Default null

transitionEffectTime

[Number] Refers to the transition duration of switching to the next map.

_mode Show the mode of current operating：DEFAULT,SHOW,HIDE.

Return

    true   success

    false   fail

2.19 mUp

Tilt up the map automatically

```
function  mUp(
    tiltRate:int = 20
):Boolean;
```

Parameters

    tiltRate

        [int]   Tilt up rate, default

Return

    true   success

    false   fail

2.20 mDown

Tilt down the map automatically

```
function  mDown(
    tiltRate:int=20
):Boolean;
```

Parameters

    tiltRate

        [int]   tilt down rate, default 20.

Return

    True   success

false	fail

2.21 mLeft

Pan left the map automatically

```
function  mLeft(
    panRate:int = 20
):Boolean;
```

Parameters

panRate

[int]	pan left rate, default 20

Return

True	success
false	fail

2.22 mRight

Pan right the map automatically

```
function  mRight(
    panRate:int = 20
):Boolean;
```

Parameters

panRate

[int]	pan right rate, default 20

Return

true	success
false	fail

2.23 mZoomOut

Zoom out the map

```
function  mZoomOut(
    FOVRate:int=20
):Boolean;
```

Parameters

FOVRate

[int]	zoom out rate , default20.
	Zoom out rate, default

Return

true	success
false	fail

2.24 mZoomIn

Zoom in the map

```
function  mZoomIn(
    FOVRate:int=20
):Boolean;
```

Parameters parameters

FOVRate

[int]   zoom out rate , default20.

Zoom in rate, default

Return

true	success
false	fail

2.25 fullScreen

Pup up a full screen window. If it has really been full screen, call the API again, then close the full screen window, and return to the normal window

```
function  fullScreen(
    viewername:String
):Boolean;
```

Parameters

viewername

[String]   Refers to control the viewer to full screen. The value are sviewer | mviewer | mviewer, Sviewer means Scene Viewer component, mviewer means Map Viewer component, dviewer means Detail Image Viewer component

Return

true	success
false	fail

2.26 closeWindow

Close the current window

```
function  closeWindow():Boolean;
```

Return

true	success

false fail

2.27 executeScript

Execute the script

```
function  executeScript(
    script:String
):Boolean;
```

Parameters

Script

[String]	Script contents

Return

true	success
false	fail

2.28 linkURL

Open the allocate URl

```
function  linkURL(
    url:String,
    des:String="_blank"
):Boolean;
```

Parameters

url

[String]	URL

Des

[String]	open method, default blank.

Return

true	success
false	fail

2.29 emailTo

send email, pop up the default mail server, fill the URL of the current virtual tour to the text area automatically

```
function  emailTo(
    mailXML:String = “”
):Boolean;
```

Parameters

mailXML

[String]	Email address

Return

True	success
false	fail