# **Advance Video Tracking did not fire on Roman Israel Site**

## Causes

1. YouTube API ‘stateChange’ event tracking function (function onPlayerStateChange(event)) which is used by us to actually track advanced events were already being used by the developer for other purposes
2. YouTube Player was being destroyed before stop/pause of video

## Remediations

1. Developer and Tracking Code had to be merged in YouTube API ‘stateChange’ event tracking function (function onPlayerStateChange(event)).

Following is how it looked after we merged code. The Section highlighted in blue is developer code and that in yellow is tracking code:

function onPlayerStateChange(event){

if(event.data == 0){

// auto close only on first auto play video that is launched after preloader

if(\_autoPlay){

$.fancybox.close(true);

\_autoPlay = false;

}

}

else if(event.data == 1){

//initPageBlurListener();

}

// DTM tracking

console.log('\*\*\* iFrame embed onPlayerStateChange \*\*\*',event.target);

video\_name = event.target.getVideoData();

video\_name = video\_name.title;

video\_length = event.target.getDuration();

//Getting the value of s from siteCatalyst loaded by DTM on the page.

//If more than one siteCatalyst is present, pick the one which has 's' defined

var s = null;

if(\_satellite != null){

for (i = 0; i < \_satellite.getToolsByType('sc').length && s == null; i++) {

s = \_satellite.getToolsByType('sc')[i].getS();

}

}else{

console.log('\*\*\* \_satellite is null \*\*\*');

}

if(s!=null){

if ((event.data === 1) && YT.PlayerState.PLAYING === 1) {

//\*-\* PLAY

console.log("\*-\* Player is on play mode " + event.data + ' ' + event.target.getCurrentTime(), s);

if (Math.floor(event.target.getCurrentTime()) === 0) {

//if (event.target.getCurrentTime() === 0) {

s.Media.open(video\_name, video\_length, 'Youtube Object Embed');

s.Media.play(video\_name, event.target.getCurrentTime());

} else {

s.Media.play(video\_name, event.target.getCurrentTime());

}

} else if (event.data === 2) {

//\*-\* PAUSE --- CAN USE THIS FOR ENDING TOO =-- check on time -5 sec!!

console.log("\*-\* Player is on pause mode " + event.data + ' ' + event.target.getCurrentTime());

s.Media.stop(video\_name, event.target.getCurrentTime()); //this will cause the monitor to have media.event='STOP'

} else if (event.data === 3) {

//\*-\* SKIPPING

console.log("\*-\* Player is on skipping mode " + event.data);

s.Media.stop(video\_name, event.target.getCurrentTime()); //this will cause the monitor to have media.event='STOP'

} else if (event.data === 0) {

//\*-\* Completed

console.log("\*-\* Player has been completed " + event.data);

s.Media.stop(video\_name, event.target.getCurrentTime());

s.Media.close(video\_name);

}

}else{

console.log('\*\*\* s is undefined \*\*\*');

}

// DTM tracking

}

1. FancyBox was used to display the YouTube Video iframe.

2.1> On the ‘beforeShow‘ callback of the Fancy Box YouTube player object ‘\_YTPlayer’ was created.

//Code

YTPlayer = new YT.Player('videoOverlay\_player', {

width: "100%",

height:"100%",

videoId: videoId,

playerVars: { 'autoplay': 1,

listType:'playlist',

list: playListId,

},

events : {

'onReady' : onPlayerReady,

'onStateChange': onPlayerStateChange

}

});

//Code

2.2> On the ‘beforeClose’ callback of the FancyBox the YouTube player Object was destroyed. Instead, we first paused the player and then destroyed it. The Code looked as below after this change:

beforeClose: function(){

try{

\_YTPlayer.pauseVideo();

}catch(err){

console.log(err);

}

},

afterClose: function(){

\_YTPlayer.destroy();

\_isOpen = false;

$(\_this).trigger(VideoOverlay.COMPLETE);

// auto close only first time it is played

\_autoPlay = false;

// show home

Global.navManager.showHome();

}