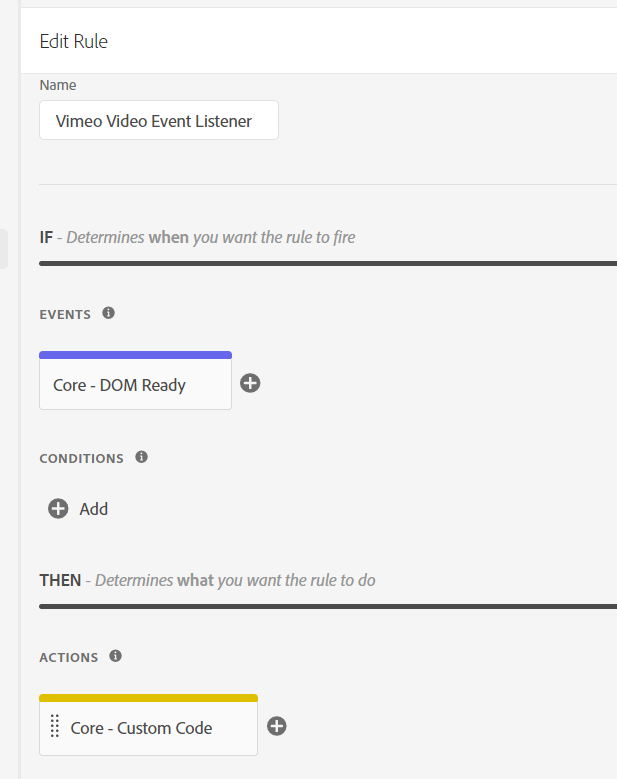
**VIMEO VIDEO TRACKING IN ADOBE ANALYTICS**

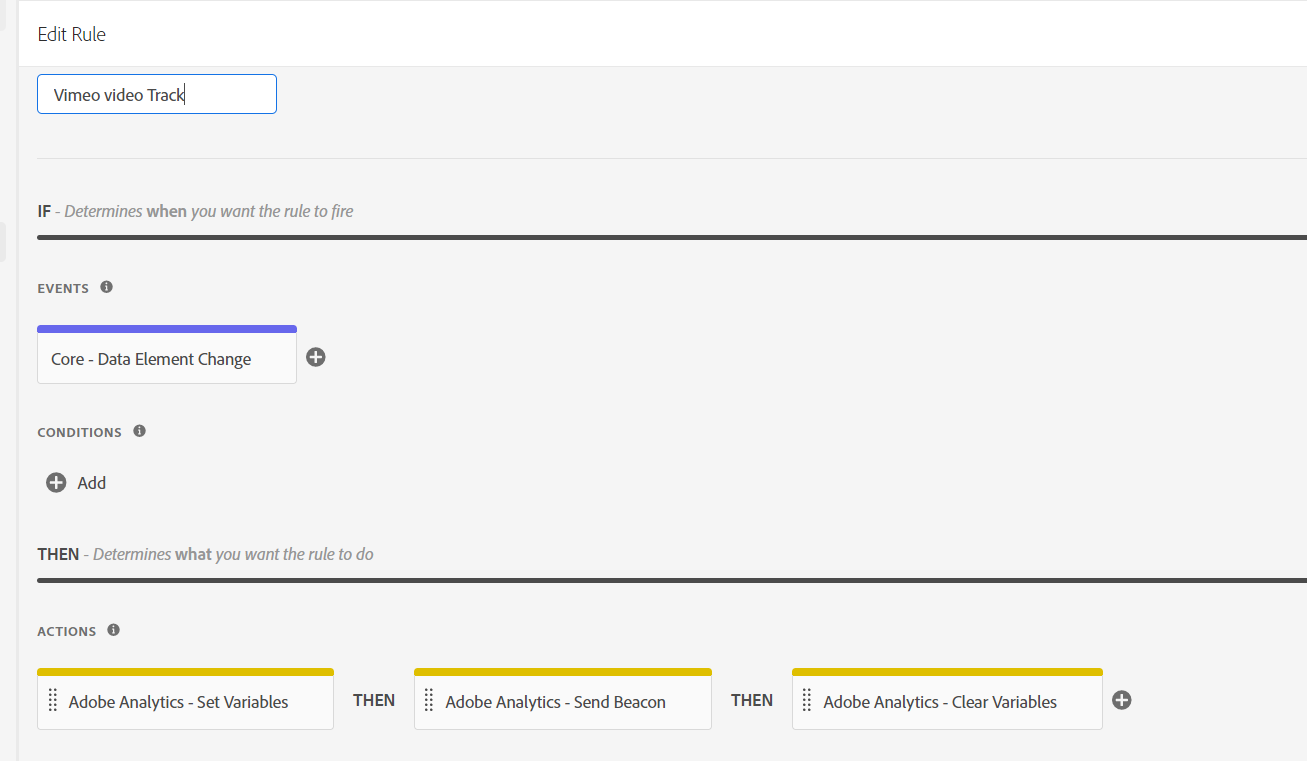
**Step 1 : Create a rule for Event Listener for Vimeo Video.**



**Step 2 : Add the following HTML code in custom code and save the rule.**

|  |
| --- |
| **<script>**  **//declare name of array in which you want data. “VideoData” is used here.**  **var videoData = (typeof(videoData) !== "undefined" && videoData instanceof Array) ? videoData : [];**  **var videoLabels=[];**  **var lastP=[];**  **//we declare variables that will hold information about the video being played**  **var \_playerTitle = {};**  **try{**  **init();**  **}**  **catch(err){**  **videoData.push({**  **'event': 'video.error',**  **'errorMessage': e.message,**  **'tag': 'CP - UA - Vimeo Video Listener'**  **})**  **}**  **function init(){**  **try{**  **var player=document.getElementsByTagName("iframe");**  **for (i = 0; i < player.length; ++i) {**  **var url=player[i].getAttribute("src");**    **if(/player\.vimeo\.com\/video/.test(url)){ // vimeo iframe found**  **if(!player[i].hasAttribute("id")){ // id attribute missing**  **player[i].setAttribute("id","vimeo\_id\_"+i); // add id attribute**  **}**  **var urlUpdated=false;**  **if(!/api=/.test(url)){ // check to see if api parameter is in src attribute**  **url=updateUrl(url,"api",1);**  **urlUpdated=true;**  **}**    **if(!/player\_id=/.test(url)){ // check if player\_id is in src attribute**  **url=updateUrl(url,"player\_id",player[i].getAttribute("id"));**  **urlUpdated=true;**  **}**  **if(urlUpdated){ // repopulate src attribute with added parameters**  **player[i].setAttribute("src",url)**  **}**  **videoLabels[player[i].getAttribute("id")]=player[i].getAttribute("src"); // id to label dictionary**  **}**  **}**    **// Listen for messages from the player**  **if (window.addEventListener){**  **window.addEventListener('message', onMessageReceived, false);**  **}**  **else {**  **window.attachEvent('onmessage', onMessageReceived, false);**  **}**  **}**  **catch(err){**  **}**  **}**    **function updateUrl(url,param,value){**  **try{**  **return url+((/\?/.test(url)) ? "&" : "?")+param+"="+value;**  **}**  **catch(err){**  **}**  **}**    **// Handle messages received from the player**  **function onMessageReceived(e) {**  **try{**  **var data = e.data;**    **if(typeof data === "string"){**  **data = JSON.parse(data);**  **}**    **switch (data.event) {**  **case 'ready':**  **onReady(data);**  **break;**  **case 'play':**  **onPlay(data);**  **break;**  **case 'pause':**  **onPause(data);**  **break;**  **case 'timeupdate':**  **onPlayProgress(data);**  **break;**  **}**  **}**  **catch(err){**  **}**  **}**    **// Helper function for sending a message to the player**  **function post(action, value) {**  **try{**  **var data = {**  **method: action**  **};**    **if (value) {**  **data.value = value;**  **}**    **var message = JSON.stringify(data);**  **var player = document.getElementsByTagName("iframe");**  **var url;**  **var prot;**    **for (i = 0; i < player.length; ++i) {**  **url=player[i].getAttribute("src");**    **if(/player\.vimeo\.com\/video/.test(url)){**  **// Check if protocol exists**  **prot = player[i].getAttribute('src').split('?')[0].split('//')[0];**    **// If protocol doesn't exist, then need to append to "url"**  **if (!prot){**  **url="https:" + player[i].getAttribute("src").split('?')[0];**  **}**  **player[i].contentWindow.postMessage(data, url);**  **}**  **}**  **}**  **catch(err){**  **}**  **}**    **function getLabel(id){**  **try{**  **return videoLabels[id].split('?')[0].split('/').pop();**  **}**  **catch(err){**  **}**  **}**    **//our function that will use the Vimeo video Embed API to retrieve additional information about the video**  **function getVimeoInfo(url, callback) {**    **var script = document.createElement('script');**  **script.type = 'text/javascript';**  **script.src = url;**    **document.getElementsByTagName('body')[0].appendChild(script);**  **}**    **//the callback function which takes the data received from the Vimeo oEmbed API and places it into the corresponding objectes**  **function vimeoCallback(e){**  **//console.log(e);**  **\_playerTitle[e.video\_id] = e.title;**  **}**    **function onReady(data) {**  **try{**  **//execute our function which queries the Vimeo oEmbed API once the embedded videos are "ready"**  **getVimeoInfo("https://www.vimeo.com/api/oembed.json?url=https://vimeo.com/"+getLabel(data.player\_id)+"&callback=vimeoCallback", vimeoCallback);**    **post('addEventListener', 'play');**  **post('addEventListener', 'pause');**  **post('addEventListener', 'finish');**  **post('addEventListener', 'playProgress');**  **}**  **catch(err){**  **}**  **}**    **function onPlay(data){**  **try{**  **videoData.push({**  **vimeo\_videoStatus: "Play",**  **vimeo\_videoTitle: \_playerTitle[getLabel(data.player\_id)].charAt(0).toUpperCase() + \_playerTitle[getLabel(data.player\_id)].slice(1)**  **});**  **}**    **catch(err){**    **}**  **}**    **function onPause(data){**  **try{**  **//console.log(data.data.seconds);**  **if((data.data.duration - data.data.seconds)>1){**  **videoData.push({**  **vimeo\_videoStatus: "Pause",**  **vimeo\_videoTitle: \_playerTitle[getLabel(data.player\_id)].charAt(0).toUpperCase() + \_playerTitle[getLabel(data.player\_id)].slice(1)**  **});**  **}**  **}**  **catch(err){**  **}**  **}**    **// Track progress: 25%, 50%, 75%, 100%**  **function onPlayProgress(data) {**  **try{**  **var t = (Math.floor(data.data.seconds / data.data.duration \* 4) / 4).toFixed(2);**  **//console.log((Math.floor(data.data.seconds / data.data.duration \* 4) / 4).toFixed(2));**  **if (!lastP[data.player\_id] || t > lastP[data.player\_id] ||t < lastP[data.player\_id]) {**  **lastP[data.player\_id]=t;**  **if (parseFloat(t) !== 0){**    **videoData.push({**  **vimeo\_videoStatus: t\*100+ "% Complete",**  **vimeo\_videoTitle: \_playerTitle[getLabel(data.player\_id)].charAt(0).toUpperCase() + \_playerTitle[getLabel(data.player\_id)].slice(1)**  **})**  **}**  **}**  **}**  **catch(err){**  **}**  **}**    **</script>** |
|  |
|  |

**Step 3 : Create Rule for Tracking . I have used Data element which holds Status which is populated from above rule for event purpose.**



**Event: Data Element Change**

**Data element: Custom code**

|  |
| --- |
| **try{**  **var status = videoData[videoData.length-1].vimeo\_videoStatus;**    **return status ;**    **}**  **catch(e){}** |

**Action : Set Variables : Custom Editor**

|  |
| --- |
| **try{**    **var status = \_satellite.getVar("Vimeo Video status"); //DataElement name**  **var name = videoData[0].vimeo\_videoTitle;**  **s.linkTrackVars = "events,eVar6"; // evar6 for title ;**  **s.linkTrackEvents = "event10,event11,event13,event14,event15,event19";**  **s.eVar6 = name;**  **if(status == "Play"){**  **s.events = "event10"; //write ur own code here.**  **}**  **if(status == "Pause"){**  **s.events = "event19"; //write ur own code here.**  **}**  **if(status == "25% Complete"){**  **s.events = "event13"; //write ur own code here.**  **}**  **if(status == "50% Complete"){**  **s.events = "event14"; //write ur own code here.**  **}**  **if(status == "75% Complete"){**  **s.events = "event15"; //write ur own code here.**  **}**  **if(status == "100% Complete"){**  **s.events = "event11"; //write ur own code here.**  **}**  **}**  **catch(e){}** |