# HybridGallery.Common.VDX.Components.AdEx

**Template/Demo URLs:** [Click Here](https://wiki.exponential.com/display/VDXConnect/SingleFramework.VDXConnect.Templates.AdEx)  
**New HybridGallery with extended functionality:**

## Functionality:

Component allows to show set of image files or videos within the defined container as a gallery.  
With optional points/thumbs, side arrows and swipe navigations.  
Video ids are taken from the widget setup.

## Muse setup:

Designer should drop a placeholder on a page, set size and position and add a unique title id. This placeholder will be used as a gallery.

Configuration properties details

## ****Latest property updates:****

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Property** | **Type** | **Default** | **Available in**  **Non-Muse Flow** | **Screenshot of the Property** | **Description** |
| placeholderId | String | — |  |  | Placeholder title id |
| slides [     "path\_to\_image"     "video\_id"     object {         imageUrl         videoId         videoBoundsId         videoStyle         contentId         contentHtml         loop         overlay {             contentId             contentHtml             style         }     } ] | Url-String String  Url-String String String Style-String String Html-String Boolean | Number  String Html-String Style-String | — |  |  | array["image-url | videoid", "image-url | videoid", ... ]  |  array[object, object, ... ]     Image Url, for ex. "images/poster\_1.jpg"     Video Id, for ex. "video2"     Extended slide description         Image Url, for ex. "images/poster\_1.jpg"         Video Id, for ex. "video2"         Video placeholder title id within contentId (place a rectangle or container into the slide block to fit video in that area)         Resize and position with provided css-style (for ex. videoStyle: "width:198px; height:351px; left:545px; top:104px; ")         Content title id         Content html code to be rendered, contentId is ignored         false | true | 1 .. N — true means infinite video loop, number leads to specific count of video replays         Here might be set up an element ro overlay whole slide, like the CTA or notification             Content title id, to be rendered if contentHtml is not providen             Html code to be rendered, contentId is ignored             Inline css to be applied to the overlay block |
| thumbs | Template-String | Array | none |  |  | EMPTY STRING: no thumbs are created. STRING: base file name template; for example "local\_folder/thumb\_{xx}.jpg", where {xx} is a thumb number; if thumb number less than template suggested size, then leading zeros adding. ARRAY: urls to each thumb. |
| overlay {     contentId     contentHtml     style     enabled: []     disabled: [] } | String Html-String Style-String Index-Array Index-Array | none |  |  | Here might be set up an element ro overlay whole gallery, like the CTA or notification     Content title id, to be rendered if contentHtml is not providen     Html code to be rendered, contentId is ignored     Inline css to be applied to the overlay block     If provided, then overlay would be shown on these slides only (slide index numeration starts from 1)     If provided, then overlay would be hidden on these slides, even if it enabled (slide index numeration starts from 1) |
| slideStyle | Style-String | none |  |  | **EXPERT** Custom CSS styling for the slide. |
| imageFit | String |  |  |  | cover | contain | unset | auto | initial For the details see [MDN](https://developer.mozilla.org/en-US/docs/Web/CSS/background-size) |
| videoFit | String |  |  |  | cover | contain | none | fill | scale-down For the details see [MDN](https://developer.mozilla.org/en-US/docs/Web/CSS/object-fit) If not providen, then imageFit property is used |
| ~~controlsPlacement~~ | ~~String~~ | ~~"br"~~ |  |  | ~~tl | tr | bl | br~~ ~~which means top-left | top-right | bottom-left | bottom-right correspondigly~~ |
| dotNavigation | Boolean | true |  |  | Enable/disable dots gallery navigation. |
| arrowNavigation | Boolean | true |  |  | Enable/disable left/right arrows navigation. |
| swipeNavigation | Boolean | String | true |  |  | Enable/disable horizontal swipe navigation (true | false | "auto"). |
| transitionStyle | String | "chain" |  |  | chain | push | overlay | crossfade | fade |
| transitionDuration | Seconds-Number | 1 |  |  | Time (sec) — transition time value to complete transition from slide to slide. |
| loopSlides | Boolean | true |  |  | Enable/disable seamless slide change on border slide numbers. |
| autoPlay | Seconds-Number | 0 |  |  | Time (sec) — allows to automatically change slides if no interaction happened. |
| resumeAutoPlay | Seconds-Number | 0 |  |  | Time (sec) — allows to resume autoplay from the last interaction if "**autoPlay**" property is not 0. |
| videoFreeze | Seconds-Number | none |  |  | Time (sec) — allows to control how long last frame of the video would stay before switch to next slide If not providen then autoPlay value is used |
| dotsPosition | String | "bottom" |  |  | top | bottom |
| dotsOverflow | String | none |  |  | fade | scale — [**Details**](http://prntscr.com/tl5saq) |
| dotsStyle | Style-String | none |  |  | **EXPERT** Custom CSS styling for the dots container (for ex. for the better positioning). |
| dotsGap | Style-Size-String | "10px" |  |  | CSS width value to set the gap size between dots. |
| dotStyle {     normal     hover     active } | Object {Style-String} | none |  |  | **EXPERT** Custom CSS styling for the specific dot states (each is optional).     normal state — (for ex.: "border:0px solid #FFF; background-color:rgba(255, 255, 255, 0.1); ")     mouseover state — (for ex.: "border-size:2px; background-color:rgba(255, 255, 255, 0.25); ")     active state — (for ex.: "border-size:1px; background-color:#9999FF; ") |
| dotOutlineNormal | Style-Border-String | none |  |  | CSS border value. If provided - dot normal state would have this style; For ex. "1px solid #FFFFFF". For details see [MDN](https://developer.mozilla.org/en-US/docs/Web/CSS/border). |
| dotOutlineActive | Style-Border-String | none |  |  | CSS border value. If provided - dot active state would have this style; For ex. "2px solid #FFFFFF". For details see [MDN](https://developer.mozilla.org/en-US/docs/Web/CSS/border). |
| dotSize | Style-Size-String | none |  |  | CSS size like value. For ex. "10px" - width and height at same time, "10px 20px" - width and height separately. |
| dotShape | String | "circle" |  |  | rectangle | circle |
| dotThumbFit | String | "contain" |  |  | cover | contain | unset | auto | initial For the details see [MDN](https://developer.mozilla.org/en-US/docs/Web/CSS/background-size) |
| arrowsPosition | String | "center" |  |  | center | top | bottom |
| arrowsMargin | Style-Margin-String | none |  |  | CSS margin like value. For ex. "10px", "0px 10px", "0px 0px 100px", "0px 30px 0px 0px"; any valid css-margin value for arrows container. |
| arrowStyle {     normal     hover     disabled }, | Object {Style-String} | none |  |  | **EXPERT** Custom CSS styling for the specific arrow states (each is optional).     normal state — (for ex.: "opacity:0.8 ")     mouseover state — (for ex.: "opacity:1 ")     disabled state (first and last slide in case of loopSlides: false) — (for ex.: "opacity:0.1 ") |
| arrowSize | Style-Size-String | "44px 80px" |  |  | CSS size like value. For ex. "50px" - width and height at same time, "30px 50px" - width and height separately |
| arrowColor | Style-Color-String | "#FFFFFF" |  |  | CSS color like value. For ex. "#FFFFFF" |
| arrowBackgroundColor | Style-Color-String | none |  |  | CSS color like value. For ex. "rgba(255, 255, 255, 0.25)" |
| arrowImageLeft | Url-String | none |  |  | URL path to left (previous) arrow image. If provided, custom image for the left arrow will be used; If not provided, but right (next) image is provided - mirrored right (next) image would be used. |
| arrowImageRight | Url-String | none |  |  | URL path to right (next) arrow image. If provided, custom image for the right (next) arrow will be used; If not provided, but left (previous) image is provided - mirrored left (previous) image would be used. |
| arrowImageFit | String | "intial" |  |  | cover | contain | unset | auto | initial For the details see [MDN](https://developer.mozilla.org/en-US/docs/Web/CSS/background-size) |
| arrowsCustomize {     container     left     right } | Object {Style-String} | none |  |  | **EXPERT** Custom CSS styling for the specific arrow elements (each is optional).     container style (to define position and size of the arrows container element) — (for ex.: "left:10px; top:10px; width:200px; height:100px;")     left/previous style (to define position and size of the left/previous arrow element) — (for ex.: "width:80px; height:100%;")     right/next style (to define position and size of the right/next arrow element) — (for ex.: "width:80px; height:100%;") |
| ~~progressStyle {~~ ~~time~~ ~~full~~ ~~}~~ | ~~Object {Style-Color-String}~~ | ~~none~~ |  |  | ~~Custom color styling for the specific progress bar elements (each is optional).~~ ~~Playback progress bar coloring, for ex.: "#0099FF"~~ ~~Background layer coloring, for ex.: "rgba(255, 0, 0, 0.5)"~~ |
| controls {     enabled     host     position     style     color } | Object     Boolean     String     String     Style-String     Style-Color |  |  |  | Playback controls (mute & play/pause, each is optional)     true | false (true by default)     video | slide (to place controls within the video bounds, or within the slide bounds)     tl | tr | bl | br (which means top-left | top-right | bottom-left | bottom-right relatively)     additional styles (to fine-tune position, or add some filters)     controls color (#FFFFFF by default) |
| progress {     enabled     host     timeColor     fullColor } | Object     Boolean     String     Style-Color-String     Style-Color-String |  |  |  | Progress bar settings (each is optional)     true | false (true by default)     video | slide (to place controls within the video bounds, or within the slide bounds)     Playback progress bar coloring, for ex.: "#0099FF"     Background layer coloring, for ex.: "rgba(255, 0, 0, 0.5)" |
| theme { **\*update in 3.10.1**     name     size     timeSize     timeWidth     bgColor     playPauseColor     style } | Object     String     Number     Number     Number     Style-Color-String     Style-Color-String     Style-String | "circular" theme | Yes |  | Progress theme customization     Default theme selection should be **circular.**     **name: default | circular** (default theme means line bar at the bottom, circular theme means progress bar in circular shape)     **size**: Set the size of background circle.     **timeSize**: Set the size of circular progress bar.     **timeWidth**: Set the thickness of the circular progress bar.     **bgColor**: Control playpause color in circular theme apart of mute/unmute button     **playPauseColor**: Control playpause color in circular theme apart of mute/unmute button     **style**: Positioning styles like top | bottom | left | right | transform applicable. |
| preventRender | Boolean | false |  |  | Prevent automatic rendering at the ad-unit initialization If value is false — the "render" function should be called anytime later |
| restrict {     execution     size } | Object {String} | none |  |  | If provided, component would be rendered or not based on inputs execution — "inframe", "instream", "display", "mobile"; or multiple at once "instream, display" and so on size — "phone", "tablet"; or multiple at once "phone, tablet" and so on |

## Component Callbacks:

* **onSlideClick(Expo, event)** (available event parameters: slideNo, videoId; fires on video click event)
* **onSlideChange(Expo, event)** (available event parameters: slideFrom, slideTo, videoIdFrom, vieoIdTo, auto; fires on slide change event)
* **onDotClick(Expo, event)** (available event parameters: slideFrom, slideTo; fires on bottom navigation click event)
* **onLeftArrowClick(Expo)** (fires on left arrow click event)
* **onRightArrowClick(Expo)** (fires on right arrow click event)

## Font setup:

Refer to the font-embedding documentation: [Fonts.Common.VDX.Components.AdEx](file:////display/VDXComponents/Fonts.Common.VDX.Components.AdEx)

## Limitations:

1. Dots navigation position either controlled by **dotsPosition:** "**top**"/"**bottom**", either by **dotsStyle**.
2. Options **dotsStyle**, **dotStyle** and **arrowStyle** should be used with care, as they might change visual appearance dramatically, and these options might override other single-styling options

## Notes:

1. **The placeholder should be a container ('State button' or 'Rectangle').**
2. The place holder id will only have component name without the initial **VDXC\_ in designer-config, while in muse designer need to define the placehoder id with VDXC\_component name**
3. Keep **transition time** under 1 second for smoothest animation look.
4. **Swipe navigation** is preferable to use in **mobile units** only
5. if **onSlideClick** event is provided, slides has cursor: pointer.
6. Currently for image url we can't use urls with keyword "**video**" in it, it will cause issues
7. To function properly on second show, hybrid gallery component has to be called with either **reset()**, either **showSlide(x)**
8. To handle issues between **overlays** and **progress bar** or **control buttons**, you can add z-index style to overlay elements as shown here <https://snipboard.io/2NCP8J.jpg>
9. **restrict** usage: if component has to be restricted to mobile phone only, this setting should be provided:
   * restrict: {
   * execution: "mobile",
   * size: "phone",
   * },

## Advanced use:

Get gallery component instance:

* var **hybridGalleryInstance** = Expo.designerAPI.hybridGallery.get("placeholderId");

## Component methods:

* **hybridGalleryInstance.update(config)** (here could be provided parameters to override existing config and init new gallery view)
* **hybridGalleryInstance.reset()** (reset component to initial state)
* **hybridGalleryInstance.release()** (reset component to initial state and release video)
* **hybridGalleryInstance.showSlide(slideNo)** (show particular slide, numeration starts from 1)
* **hybridGalleryInstance.showVideo(videoId)** (show slide where **videoId** is the content video)
* **hybridGalleryInstance.getCurrentSlide()** (returns current slide number)
* **hybridGalleryInstance.getCurrentVideo()** (returns current videoId or null if no video in the slide)
* **hybridGalleryInstance.getStructure()** (set of gallery root DOM elements, like slidesEl, dotsEl, leftArrowEl, rightArrowEl)
* **hybridGalleryInstance.playPause()** (pauses or resumes playback - might be used in **onSlideClick** event)
* **hybridGalleryInstance.render**() - render the component with default config
* **hybridGalleryInstance.play**() - resumes component's video and autoplay
* **hybridGalleryInstance.resume**() - resumes component's video and autoplay
* **hybridGalleryInstance.pause**() - pauses component's video and autoplay
* **hybridGalleryInstance.stop**() - stops component's video and autoplay

## Additional global methods:

Expo.designerAPI.components.get("placeholderId"); // to get component by id  
Expo.designerAPI.components.reset(); // to reset all components

Code Snippet

* **Use this line of code on Tab to Initialize the HybridGallery**  
  Expo.designerAPI.components.get("hybridGallery\_placeholder1").render();
* **Use this line of code on Tab to reset the HybridGallery**  
  Expo.designerAPI.components.get("hybridGallery\_placeholder1").reset();

**Sample Code** Expand source

/\* Start HybridGallery Code from Here \*/

        {

            name: "hybridGallery", // Do not change this value

            props: {

                placeholderId: "hybridGallery\_placeholder1",

                slides: [ // array["image-url | videoid", "image-url | videoid", ...]

                    "assets/slide\_01.jpg",

                    "video2",

                    "assets/slide\_02.jpg",

                    "video3",

                    "assets/slide\_03.jpg",

                ],

                thumbs: "assets/thumb\_{xx}.jpg", // url-template | array["url1", "url2", ...]

                imageFit: "cover", // cover | contain | auto | XXpx | XXpx YYpx | XX% | XX% YY% | inherit | initial

videoFit: "cover", // cover | contain | none | fill | scale-down

controlsPlacement: "br", // tl | tr | bl | br

                dotNavigation: true, // true | false // enable/disable dots/thumbs navigation

                arrowNavigation: true, // true | false // enable/disable side arrows navigation

                swipeNavigation: "auto", // "auto" | true | false // enable/disable mouse/touch swipe navigation

                transitionStyle: "chain", // chain | push | overlay | crossfade | fade

                transitionDuration: 1, // transition time (sec)

                loopSlides: true,

                autoPlay: 3, // time (sec) to start autoplay

                resumeAutoPlay: 3, // time (sec) to resume autoplay after stop

                dotsPosition: "bottom", // top | bottom

                dotsOverflow: "fade", // fade | scale

dotsStyle: "", // css values

                dotStyle: {

                    normal: "", // css values

                    hover: "", // css values

                    active: "", // css values

                },

                dotOutlineNormal: "1px solid #FFFFFF", // css border value

                dotOutlineActive: "2px solid #FFFFFF", // css border value

                dotSize: "10px", // {size} | {width, height}

                dotShape: "circle", // rectangle | circle

                dotThumbFit: "contain", // cover | contain | auto | XXpx | XXpx YYpx | XX% | XX% YY% | inherit | initial

                arrowsPosition: "center", // center | top | bottom

                arrowsMargin: "0px", // css margin value

                arrowStyle: {

                    normal: "", // css values

                    hover: "", // css values

                    disabled: "", // css values

                },

                arrowSize: "44px 80px", // {size} | {width, height}

                arrowColor: "#FFFFFF", // css color

                arrowBackgroundColor: "rgba(0, 0, 0, 0.25)", // css color

                arrowImageLeft: "", // url

                arrowImageRight: "assets/gallery\_arrowRight.png", // url

                arrowImageFit: "contain", // cover | contain | auto | XXpx | XXpx YYpx | XX% | XX% YY% | inherit | initial

                arrowsCustomize: {

                    container: "", // css values

                    left: "", // css values

                    right: "", // css values

                },

          theme: {

             name: "circular",

             progressBgSize: 60,

             progressSize: 50,

             progressWidth: 6

             style: "left:40px; bottom:60px;",

          }

           },

            events: {

                onSlideClick: function(Expo, event) {

                },

                onSlideChange: function(Expo, event) {

                },

                onDotClick: function(Expo, event) {

                },

                onLeftArrowClick: function(Expo) {

                },

                onRightArrowClick: function(Expo) {

                },

            },

      },

/\* End HybridGallery Code \*/

## Component Callbacks:

* **onSlideClick(Expo, event)** (available event parameters: slideNo, videoId; fires on video click event)
* **onSlideChange(Expo, event)** (available event parameters: slideFrom, slideTo, videoIdFrom, vieoIdTo, auto; fires on slide change event)
* **onDotClick(Expo, event)** (available event parameters: slideFrom, slideTo; fires on bottom navigation click event)
* **onLeftArrowClick(Expo)** (fires on left arrow click event)
* **onRightArrowClick(Expo)** (fires on right arrow click event)

## Supported interaction Events and Detail

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **User Action** | **Event Name** | **Default Source All in bound pixel with have a suffix TabTracker id which is defined by user** | **Default Sub Source** | **Parameter in insight pixel** | **Can i change source value in non muse BEFORE Advance Pixel Management Release?** | **Can i change source value in non muse AFTER Advance Pixel Management Release?** | **Can i change source value in Muse BEFORE Advance Pixel Management Release?** | **Can i change source value in Muse AFTER Advance Pixel Management Release?** | **Initiator(function/callback initiating the pixel)** |
| click on left Arrow | inbound | leftArrow | {   componentName: "COMPONENT NAME",   tabName: "HOST TAB LABEL", // **Offers**   tabSequence: "tab{HOST\_TAB\_ID}", // **tab1, tab2, ...**   subSource: "slide{ID}", // **slide1, slide2, ...** } | event=intLive and fr:{source) | NO | NO | YES | YES | onLeftArrowClick |
| click on right Arrow | inbound | rightArrow | {   componentName: "COMPONENT NAME",   tabName: "HOST TAB LABEL", // **Offers**   tabSequence: "tab{HOST\_TAB\_ID}", // **tab1, tab2, ...**   subSource: "slide{ID}", // **slide1, slide2, ...** } | event=intLive and fr:{source) | NO | NO | YES | YES | onRightArrowClick |
| click on dot | inbound | dotSelect | {   componentName: "COMPONENT NAME",   tabName: "HOST TAB LABEL", // **Offers**   tabSequence: "tab{HOST\_TAB\_ID}", // **tab1, tab2, ...**   subSource: "slide{ID}", // **slide1, slide2, ...** } | event=intLive and fr:{source) | NO | NO | YES | YES | onDotClick |
| on click on slide | outbound | tracker id (as defined by user) | {   componentName: "COMPONENT NAME",   tabName: "HOST TAB LABEL", // **Offers**   tabSequence: "tab{HOST\_TAB\_ID}", // **tab1, tab2, ...**   subSource: "slide{ID}", // **slide1, slide2, ...** } | event=clickLive and fr:{source) | NO | YES | YES | YES | onSlideClick |
| on slide change | inbound | slideChange | {   componentName: "COMPONENT NAME",   tabName: "HOST TAB LABEL", // **Offers**   tabSequence: "tab{HOST\_TAB\_ID}", // **tab1, tab2, ...**   subSource: "slide{ID}", // **slide1, slide2, ...** } | event=intLive and fr:{source) | NO | NO | YES | YES | onSlideChange |
| Video events | all supported video event | video{ID} | {   componentName: "COMPONENT NAME",   tabName: "HOST TAB LABEL", // **Offers**   tabSequence: "tab{HOST\_TAB\_ID}", // **tab1, tab2, ...**   subSource: "video{ID}", // **video1, video2, ...** } | all supported video event and fr:{videoID} | NO | NO | NO | NO | Create/Update video player instance |
| play/pause button clicked | pauseLive resumeLive | pauseLive resumeLive | {   componentName: "COMPONENT NAME",   tabName: "HOST TAB LABEL", // **Offers**   tabSequence: "tab{HOST\_TAB\_ID}", // t**ab1, tab2, ...**   subSource: "slide{ID}", // **slide1, slide2, ...** } | event=intLive and fr:{source) | NO | NO | NO | YES | onPlayPauseClick |
| Mute/Unmute button Clicked | muteLive unmuteLive | muteLive unmuteLive | {   componentName: "COMPONENT NAME",   tabName: "HOST TAB LABEL", // **Offers**   tabSequence: "tab{HOST\_TAB\_ID}", // **tab1, tab2, ...**   subSource: "slide{ID}", // **slide1, slide2, ...** } | event=intLive and fr:{source) | NO | NO | NO | YES | onMuteUnmuteClick |

Open Items and Point need to discuss for finalize :

* Currently we are identifying hybrid gallery video using tabName and VideoName inside the reporting... suppose we build a adUnit which has second tabName is "Offers" like this adunit [here](https://creative.vdx.tv/Chevrolet/#GPIJ5AEndt.Q/1) (for example). Suppose we placed hybrid gallery having multiple video inside this tab in that case source and subsoure should be like this "video1" and "Offers".... in video interaction detail report it should appear like video1\_offers... in later release, whenever we can have option to change the video name from VDX studio and Muse both places together .... we can trackerId provided inside the VDX studio and Muse firePixel API.
* Currently there is ambiguity during test cases and reviewing the JSON outcome what is the expectation of Default source and subsoure value to review in JSON and what is the actual parameter expecting in pixel firing?
* Need to update specs saying callback instead of Component Events .. review the all specs if change are required across system.
* Need Explicit clarity on which event CSM has option to ask change on trackerId or which events there is no option to change TrackerID. it should be fine that all events CSM don't have option to change the trackerID just like leftArrow or rightArrow but that should be explicit clear.
* Suppose we are building this [Example](https://creative.vdx.tv/Neds_Lottery/#sn7GaA/10) using RichMedia Gallery in that case what we are expecting as default value of source and subSource or what is the expectation on trackerID...
* We need This table across all Component to bring more clarity around test cases ...
* please note that , Cases where we don't have option to provide trackerId... outcome JSON doesn't have trackerId ...in that case we need clear picture what need to pick for widget input...for example leftArrow , right arrow , hybrid gallery video .... whenever we are providing tracker Id  which is equivalent to key (inside outcome json in vdxstudio). we will use key only .... but when tracker id is not available please clear what is the value of key ..
* CTA on all Component always has trackerId option irrespective of Muse and non Muse ...is this correct statement...

# Fonts.Common.VDX.Components.AdEx

# Font embedding setup for use with any component

Technically there are several options to import fonts

## Local fonts:

1. Place required web-fonts to the local "**fonts**" folder <https://i.imgur.com/Zgk1tml.jpg>
2. Put font-family property to the **menuDesign -> items -> normal** property
3. Add code in the **designer-config.js** in the init section as follows (use emeding rules from [MDN](https://developer.mozilla.org/en-US/docs/Web/CSS/@font-face))

**designer-config.js -> init()**

init: function(Expo) {

var fontStyle = document.head.appendChild( document.querySelector("style") );

fontStyle.innerText = "\

@font-face {font-family:Bogle; font-weight:normal; src:url(fonts/Bogle-Regular.woff) format('woff'), url(fonts/Bogle-Regular.ttf) format('truetype');}\

@font-face {font-family:Bogle; font-weight:bold; src:url(fonts/Bogle-Bold.woff) format('woff'), url(fonts/Bogle-Bold.ttf) format('truetype');}\

";

},

## Google Fonts:

1. Find required font on google fonts (for ex. [Open Sans](https://fonts.google.com/specimen/Open+Sans))
2. Put font-family property <https://i.imgur.com/tEL8VLf.jpg> to the **menuDesign -> items -> normal** property
3. Copy @import embedding option <https://i.imgur.com/buHsLI4.png>
4. Add code in the **designer-config.js** in the init section as follows: <https://i.imgur.com/QlrE7rG.png>

**designer-config.js -> init()**

init: function(Expo) {

var fontStyle = document.head.appendChild( document.querySelector("style") );

fontStyle.innerText = "@import url('https://fonts.googleapis.com/css2?family=Open+Sans&display=swap');";

},

## Muse — Typekit, self-hosted:

1. Create text field outside of main canvas, and apply required font to it <https://i.imgur.com/3hr0MzB.jpeg>
2. Publish mainunit and open index.html in browser; then find this element using web-inspector
3. Copy **font-family** property <https://i.imgur.com/e6laklG.jpeg> to the **menuDesign -> items -> normal** property

## Watch video recording how to apply above techniques to the ad-units:

.

.

.

Temporary (to be removed soon):

Legacy

1. Setup font by Muse:
   1. Create text field outside of main canvas, and apply required font to it <https://i.imgur.com/3hr0MzB.jpeg>
   2. Publish mainunit and open index.html in browser; then find this element using web-inspector, and copy **font-family** property <https://i.imgur.com/e6laklG.jpeg> to the **menuDesign -> items -> normal** property
2. Setup font by code:
   1. Find required font on google fonts or similar service (for ex. [Nunito Sans](https://fonts.google.com/specimen/Nunito+Sans))
   2. Copy provided embedding option <https://i.imgur.com/JPBZANc.jpg> (this not work properly anymore)
   3. Create Muse **Html Block** and paste embeding option there <https://i.imgur.com/dLi4BsI.jpg> (this not work properly anymore)
   4. Copy font-family property <https://i.imgur.com/tEL8VLf.jpg> to the **menuDesign -> items -> normal** property
3. Setup local fonts with code:
   1. Place required web-fonts to the local "**assets**" folder <https://i.imgur.com/Zgk1tml.jpg>
   2. Create Muse **Html Block** and enter embeding there <https://i.imgur.com/RxK9g4U.jpg> (use emeding rules from [MDN](https://developer.mozilla.org/en-US/docs/Web/CSS/@font-face))
   3. Copy entered font-family property <https://i.imgur.com/272upTt.jpg> to the **menuDesign -> items -> normal** property

# MobileExpandable.DesignerAPI.AdEx

|  |  |
| --- | --- |
| **Code Snippet** | **Usage** |
| Expo.designerAPI.firePixel(PixelID,[{"eventName":"clickLive", "multi":false}]);  Expo.designerAPI.firePixel(PixelID,[{"eventName":"intLive", "multi":true}]); | to trigger click and interaction pixel  "true" means multiple times  "false" means single time |
| anchorCLick(tabName); | this will take user to that particular card with smooth scrolling effect. |
| Expo.designerAPI.openUrl([C](http://www.lexus.com/)TAURL, CTAName); | to trigger CTA URL |
| mainUnitCloseCallback: function(callback, defaultVideoId) { /\* sample Code start this.designerAPI.switchPlayerInTab(defaultVideoId, "videoContainer"); this.trigger(tabName, false); callback(); // dont remove this from framework  /\* sample Code End } | called when the mainunit is closed to reset the mainunit tabs and to perform any function if required. |
|  | define following code snippet in tooltip (as shown in image) to get element's click/tap event.  EXPOEVENT\_tabVideo  where tabVideo is the name of callback function. |
| Expo.designerAPI.pause(); | to pause video |
| Expo.designerAPI.resume(); | to resume video |
| videoEndedEvent: function(videoId, eventObj) { if (videoId == 'video1') { console.log("end of video1 is called from designer API is called"); } else if (videoId == 'video2') { console.log("end of video2 is called from designer API is called"); } else if (videoId == 'video3') { console.log("end of video3 is called from designer API is called"); } } | called when video playing in mainunit ends. |
| Expo.designerAPI.fireDynamicPixel(PixelID,[{"eventName":"EventName", "data":{"key1":sampleData1,"key2":sampleData2,"key3":sampleData3}}]); | to fire Dynamic pixel. |
| { name: buttonName, type: 'click', callback: function(Expo, event) {  } | callback provided to designer for buttons  Here Expo gives the access to component framework api's |
| onOrientationChange : function(obj){ if(obj==="landscape"){  } | triggers when orientation change happens. |

# InFrame.DesignerAPI.AdEx

|  |  |
| --- | --- |
| **Code Snippet** | **Usage** |
| Expo.designerAPI.pause(); | to pause video |
| Expo.designerAPI.resume(); | to resume video |
| Expo.designerAPI.firePixel(pixelID,[{"eventName":"intLive", "multi":true}]);  Expo.designerAPI.firePixel(pixelID,[{"eventName":"clickLive", "multi":false}]); | to fire clickLive and IntLive pixel  "true" means multiple times  "false" means single time |
| Expo.designerAPI.openUrl(CTAURL, CTAName); | to open CTA URL |
|  | define following code snippet in tooltip (as shown in image) to get element's click/tap event.  EXPOEVENT\_tabVideo  where tabVideo is the name of callback function. |
| mainUnitCloseCallback: function(callback, defaultVideoId) {  /\* sample Code start this.setTempValue("IsComingFromMainUnit", true); this.designerAPI.switchPlayerInTab(defaultVideoId, "videoContainer"); this.trigger(tabName, false); callback(); // dont remove this from framework arrowPlaylist.resetCarousel(); // dont remove this from carousel component  /\* sample Code End } | called when the mainunit is closed to reset the mainunit tabs and to perform any function if required. |
| videoEndedEvent: function(videoId, eventObj) { if (videoId == 'video1') { console.log("end of video1 is called from designer API is called"); } else if (videoId == 'video2') { console.log("end of video2 is called from designer API is called"); } else if (videoId == 'video3') { console.log("end of video3 is called from designer API is called"); } } | called when video playing in mainunit ends. |
| Expo.designerAPI.fireDynamicPixel(PixelID,[{"eventName":"EventName", "data":{"key1":sampleData1,"key2":sampleData2,"key3":sampleData3}}]); | to fire Dynamic pixel. |
| { name: buttonName, type: 'click', callback: function(Expo, event) {  } | callback provided to designer for buttons  Here Expo gives the access to component framework api's |

# DisplayExpandable.DesignerAPI.AdEx

|  |  |
| --- | --- |
| **Code Snippet** | **Usage** |
| Expo.designerAPI.openUrl([C](http://www.lexus.com/)TAURL, CTAName); | to trigger CTA URL |
| Expo.designerAPI.firePixel(PixelID,[{"eventName":"clickLive", "multi":false}]);  Expo.designerAPI.firePixel(PixelID,[{"eventName":"intLive", "multi":true}]); | to trigger click and interaction pixel  "true" means multiple times  "false" means single time |
|  |  |
|  | define following code snippet in tooltip (as shown in image) to get element's click/tap event.  EXPOEVENT\_tabVideo  where tabVideo is the name of callback function. |
| Expo.designerAPI.pause(); | to pause video |
| Expo.designerAPI.resume(); | to resume video |
| mainUnitCloseCallback: function(callback, defaultVideoId) { /\* sample Code start this.designerAPI.switchPlayerInTab(defaultVideoId, "videoContainer"); this.trigger(tabName, false); callback(); // dont remove this  /\* sample Code End } | called when the mainunit is closed to reset the mainunit tabs and to perform any function if required. |
| videoEndedEvent: function(videoId, eventObj) { if (videoId == 'video1') { console.log("end of video1 is called from designer API is called"); } else if (videoId == 'video2') { console.log("end of video2 is called from designer API is called"); } else if (videoId == 'video3') { console.log("end of video3 is called from designer API is called"); } } | called when video playing in mainunit ends. |
| Expo.designerAPI.switchPlayerInTab(videoId, videoHolder); | to switch to different video. |
| Expo.designerAPI.fireDynamicPixel(PixelID,[{"eventName":"EventName", "data":{"key1":sampleData1,"key2":sampleData2,"key3":sampleData3}}]); | to fire Dynamic pixel. |
| { name: buttonName, type: 'click', callback: function(Expo, event) {  } | callback provided to designer for buttons  Here Expo gives the access to component framework api's |

# ColorPicker.Common.VDX.Components.AdEx

## Description:

Color Picker component based on HybridGallery functionality

## Functionality:

The component allows showing a set of image files by switching via dots. Each dot can be positioned independently at all sizes of the parent tab.  
Image switching used **'fade'** transition animation

## Muse setup:

**ColorPicker component not supported for the muse flow**

## Configuration:

Configuration properties details...

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Property** | **Type** | **Default** | **Available in**  **Non-Muse Flow** | **Screenshot of the Property** | **Description** |
| placeholderId | String | — |  |  | Placeholder title id |
| mode | String | colorPicker |  |  | Fixed value that is switching on colorPicker functionality, **Not allow to change this value** |
| preventRender | Boolean | false |  |  | Prevent automatic rendering at the ad-unit initialization If value is false — the "render" function should be called anytime later |
| slides [     "image-url"     object {         "image-url"         contentId     } ] | Url-String | — |  |  | array["image-url", ...] | array[object, object, ... ] **image-url** - for ex. "images/poster\_1.jpg"  Extended slide description: **image-url** - for ex. "images/poster\_1.jpg" **contentId** - Content title id |
| dotsOverflow | String | none |  |  | fade | scale — [**Details**](http://prntscr.com/tl5saq) |
| dotSize | Style-Size-String | none |  |  | CSS size like value. For ex. "10px" - width and height at same time, "10px 20px" - width and height separately. |
| dotShape | String | "circle" |  |  | Possible values - rectangle | circle |
| colorPicker | Object | — |  |  | Object with specified only for Color Picker component parameters |
| autoArrange | Boolean |  |  |  | if it is true, config will provide position and direction for dots container, and gap, size, and styling for all dots (each dot will have state styling configuration only, bgImage, color etc…).  If it will be false, the container will take the full size of the parent tab container and each dot will have its own configuration for position, stylings, and states. |
| dots [         object {             thumb             thumbSize             position             stateStyles         }     ] | Array     Object         String | null         String         Object         Object {Style-String} |  |  |  | Array of objects with configuration for each dot element **thumb:**      null - no thumbs are created.     String: file name path; for example "local\_folder/thumb\_1.jpg" **thumbSize:** cover | contain | unset | auto | initial- For the details see [MDN](https://developer.mozilla.org/en-US/docs/Web/CSS/background-size) **position:** Object with x, y position from top-left corner of parent tab container **stateStyles:** Custom CSS styling for the specific dot states (each is optional).         normal state — (for ex.: "border:0px solid #FFF; background-color:rgba(255, 255, 255, 0.1); ")         mouseover state — (for ex.: "border-size:2px; background-color:rgba(255, 255, 255, 0.25); ")         active state — (for ex.: "border-size:1px; background-color:#9999FF; ") |
| transitionDuration | Seconds-Number |  |  |  | Time (sec) — transition time value to complete transition from slide to slide. |
| loopSlides | Boolean | true |  |  | Enable/disable seamless slide change on border slide numbers. |
| autoPlay | Seconds-Number | 0 |  |  | Time (sec) — allows to automatically change slides if no interaction happened. |
| resumeAutoPlay | Seconds-Number | 0 |  |  | Time (sec) — allows to resume autoplay from the last interaction if "**autoPlay**" property is not 0. |
| controls {     enabled     host     position     style     color } | Object     Boolean     String     String     Style-String     Style-Color |  |  |  | Playback controls (mute & play/pause, each is optional)     true | false (true by default)     video | slide (to place controls within the video bounds, or within the slide bounds)     tl | tr | bl | br (which means top-left | top-right | bottom-left | bottom-right relatively)     additional styles (to fine-tune position, or add some filters)     controls color (#FFFFFF by default) |
| theme {      name     size     timeSize     timeWidth     bgColor     playPauseColor     style } | Object     String     Number     Number     Number     Style-Color-String     Style-Color-String     Style-String | "circular" theme | Yes |  | Progress theme customization     **name: default | circular** (default theme means line bar at the bottom, circular theme means progress bar in circular shape)     **size**: Set the size of background circle.     **timeSize**: Set the size of circular progress bar.     **timeWidth**: Set the thickness of the circular progress bar.     **bgColor**: Control play-pause color in circular theme apart from mute/unmute button     **playPauseColor**: Control playpause color in circular theme apart from mute/unmute button     **style**: Positioning styles like top | bottom | left | right | transform applicable. |
| progress {     enabled     host     timeColor     fullColor } | Object     Boolean     String     Style-Color-String     Style-Color-String |  |  |  | Progress bar settings (each is optional)     true | false (true by default)     video | slide (to place controls within the video bounds, or within the slide bounds)     Playback progress bar coloring, for ex.: "#0099FF"     Background layer coloring, for ex.: "rgba(255, 0, 0, 0.5)" |

JSON config example

Config updated for advanced pixels configuration

{

"name": "hybridGallery",

"containerTabName": "video-tab-cRzjjE7lNa",

"containerTabSequence": 2,

"props": {

"type": "container",

"placeholderId": "colorPicker-galley-Ri0ohx9gFK",

"mode": "colorPicker",

"preventRender": true,

"slides": [

{

"contentId": "colorPicker-galley-slide-Ri0ohx9gFK"

},

{

"contentId": "zICDUcQIek-colorPicker-galley-slide-Ri0ohx9gFK"

}

],

"dotsOverflow": "none",

"dotSize": "40px 40px",

"dotShape": "circle",

"colorPicker": {

"autoArrange": false,

"dots": [

{

"thumb": null,

"thumbSize": "contain",

"position": {

"x": "85px",

"y": "287px"

},

"stateStyles": {

"normal": "box-sizing: border-box; border: 1px solid rgba(0, 0, 0, 1); background-color: rgba(0, 0, 0, 1);",

"hover": "box-sizing: border-box; border: 1px solid rgba(0, 0, 0, 1); background-color: rgba(0, 0, 0, 1);",

"active": "box-sizing: border-box; border: 1px solid rgba(0, 0, 0, 1); background-color: rgba(0, 0, 0, 1);"

}

},

{

"thumb": null,

"thumbSize": "contain",

"position": {

"x": "131px",

"y": "285px"

},

"stateStyles": {

"normal": "box-sizing: border-box; border: 1px solid rgba(0, 0, 0, 1); background-color: rgba(0, 0, 0, 1);",

"hover": "box-sizing: border-box; border: 1px solid rgba(0, 0, 0, 1); background-color: rgba(0, 0, 0, 1);",

"active": "box-sizing: border-box; border: 1px solid rgba(0, 0, 0, 1); background-color: rgba(0, 0, 0, 1);"

}

}

]

},

"transitionDuration": 0.1,

"loopSlides": true,

"autoPlay": 5,

"resumeAutoPlay": 0,

"controls": {

"position": "br",

"color": "#FFFFFF"

},

"theme": {

"name": "circular",

"progressBgSize": 80,

"progressSize": 70,

"progressWidth": 6,

"playPauseColor": "#FFFFFF",

"style": "left:50%; bottom:50px; transform:translate(-50%, -50%);"

},

"progress": {

"timeColor": "#ffffff",

"fullColor": "#4362E370"

}

},

"callbacks": {

"onSlideClick": "function(Expo, event) { }",

"onSlideChange": "function(Expo, event) {

var vdxStudioInfo = {

name: ["IMP\_{TAB\_NAME}.SlideChange"],

tabName: "tabName",

tabSequence: "tabNumber",

eventName: "intLive"

};

var vdxFrameworkPixelParams = {

componentName: "hotSpot",

eventName: "intLive",

multi: true,

tabName: vdxStudioInfo.tabName,

tabSequence: vdxStudioInfo.tabSequence,

subSource: "slide" + event.slideTo,

};

if (!event.auto && event.index >= 0) {

if (vdxStudioInfo.name.length === 1) event.index = 0;

Expo.designerAPI.firePixel(vdxStudioInfo.name[event.index], vdxFrameworkPixelParams);

}

}",

"onDotClick": "function(Expo, event) {

var vdxStudioInfo = {

name: ["IMP\_{TAB\_NAME}.DotSelectHotspotSelect"],

tabName: "tabName",

tabSequence: "tabNumber",

eventName: "intLive",

};

var vdxFrameworkPixelParams = {

componentName: "hotSpot",

eventName: "intLive",

multi: true,

tabName: vdxStudioInfo.tabName,

tabSequence: vdxStudioInfo.tabSequence,

subSource: "slide" + event.slideTo,

};

if (vdxStudioInfo.name.length === 1) event.index = 0;

Expo.designerAPI.firePixel(vdxStudioInfo.name[event.index], vdxFrameworkPixelParams);

}",

"onPlayPauseClick": "function(Expo, event) {

var vdxStudioInfo = {

name: ["IMP\_{TAB\_NAME}.AnimatedPause", "IMP\_{TAB\_NAME}.AnimatedResume"],

tabName: "tabName",

tabSequence: "tabNumber",

eventName: "intLive",

};

var vdxFrameworkPixelParams = {

componentName: "hotSpot",

eventName: "intLive",

multi: true,

tabName: vdxStudioInfo.tabName,

tabSequence: vdxStudioInfo.tabSequence,

subSource: "slide" + event.slide,

};

if (event.index >= 0) {

if (vdxStudioInfo.name.length === 1) event.index = 0;

Expo.designerAPI.firePixel(vdxStudioInfo.name[event.index], vdxFrameworkPixelParams);

}

}"

  }

}

## Component Callbacks:

* **onSlideChange(Expo, event)** (available event parameters: slideFrom, slideTo, videoIdFrom, vieoIdTo, auto; fires on slide change event)
* **onDotClick(Expo, event)** (available event parameters: slideFrom, slideTo; fires on bottom navigation click event)
* **onPlayPauseClick(Expo, event)** (available event parameters: ; fires on play pause toggle button click event)

## Limitations:

## Notes:

1. **The placeholder should be a container ('State button' or 'Rectangle').**
2. The placeholder id will only have component name without the initial **VDXC\_ in designer-config, while in muse designer needs to define the placeholder id with VDXC\_component name**
3. Keep **transition time** under 1 second for the smoothest animation look.
4. Currently, for image url, we can't use urls with the keyword "**video**" in them, it will cause issues

## Advanced use:

* var **colorPickerInstance** = Expo.designerAPI.hybridGallery.get("placeholderId");

## Component methods:

* **colorPickerInstance.update(config)** (here could be provided parameters to override existing config and init new gallery view)
* **colorPickerInstance.reset()** (reset component to the initial state)
* **colorPickerInstance.showSlide(slideNo)** (show particular slide, numeration starts from 1)
* **colorPickerInstance.getCurrentSlide()** (returns current slide number)
* **colorPickerInstance.getStructure()** (set of gallery root DOM elements, like slidesEl, dotsEl, leftArrowEl, rightArrowEl)
* **colorPickerInstance.playPause()** (pauses or resumes playback - might be used in **onSlideClick** event)
* **colorPickerInstance.render**() - render the component with the default config

## Additional global methods:

* Expo.designerAPI.components.get("placeholderId"); // to get component by id
* Expo.designerAPI.components.reset(); // to reset all components

## Supported interaction Events and Detail

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **User Action** | **Event Name** | **Default Source All in bound pixel with have a suffix TabTracker id which is defined by user** | **Default Sub Source** | **Parameter in insight pixel** | **Can i change source value in non muse BEFORE Advance Pixel Management Release?** | **Can i change source value in non muse AFTER Advance Pixel Management Release?** | **Can i change source value in Muse BEFORE Advance Pixel Management Release?** | **Can i change source value in Muse AFTER Advance Pixel Management Release?** | **Initiator(function/callback initiating the pixel)** |
| click on dot | inbound | dotSelect | {   componentName: "colorPicker",   tabName: "HOST TAB LABEL", // **Offers**   tabSequence: "tab{HOST\_TAB\_ID}", // **tab1, tab2, ...**   subSource: "slide{ID}", // **slide1, slide2, ...** } | event=intLive and fr:{source) | NO | NO | YES | YES | onDotClick |
| on slide change | inbound | slideChange | {   componentName: "colorPicker",   tabName: "HOST TAB LABEL", // **Offers**   tabSequence: "tab{HOST\_TAB\_ID}", // **tab1, tab2, ...**   subSource: "slide{ID}", // **slide1, slide2, ...** } | event=intLive and fr:{source) | NO | NO | YES | YES | onSlideChange |
| play/pause button clicked | pauseLive resumeLive | pauseLive resumeLive | {   componentName: "colorPicker",   tabName: "HOST TAB LABEL", // **Offers**   tabSequence: "tab{HOST\_TAB\_ID}", // t**ab1, tab2, ...**   subSource: "slide{ID}", // **slide1, slide2, ...** } | event=intLive and fr:{source) | NO | NO | NO | YES | onPlayPauseClick |

# View360.Common.VDX.Components.AdEx

* **Template/Demo URLs:** [ClickHere](https://wiki.exponential.com/display/RMDOC/View360.Templates.AdEx)
* Config Details

# Threesixty View Component

## Functionality:

Component allows to show a 360 view of an object ( a car ) by draging or clicking left/right arrows. There can be any amount of components in a unit.

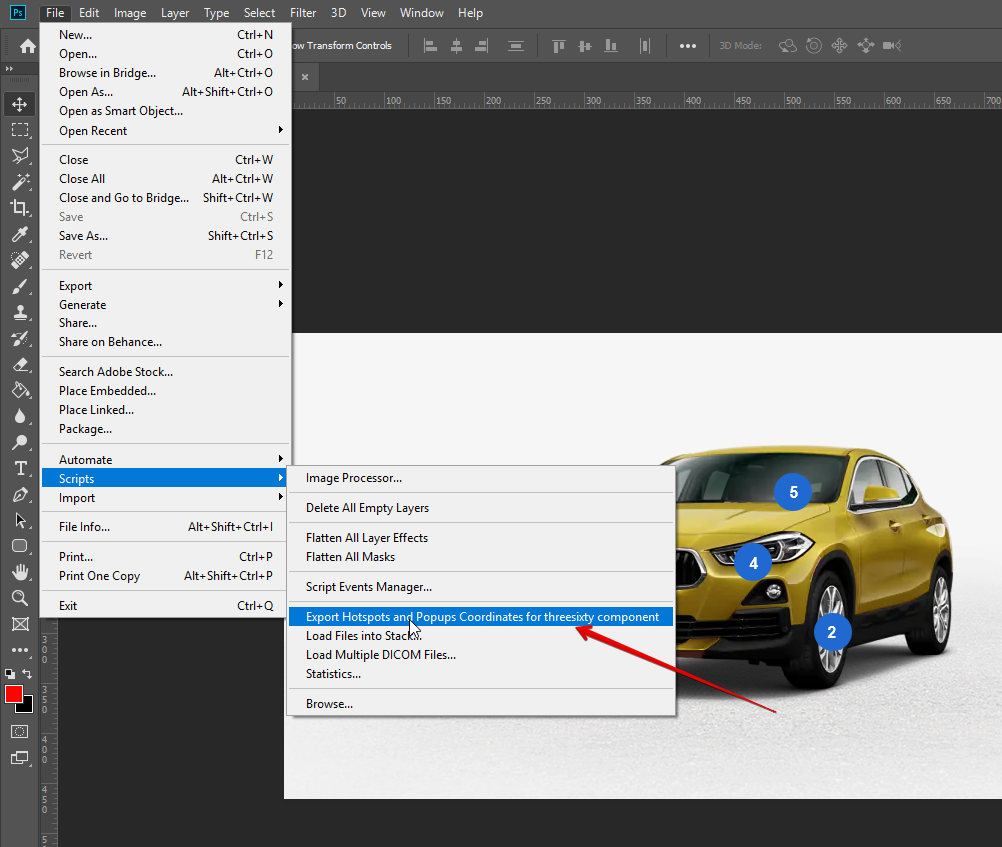
## Muse setup:

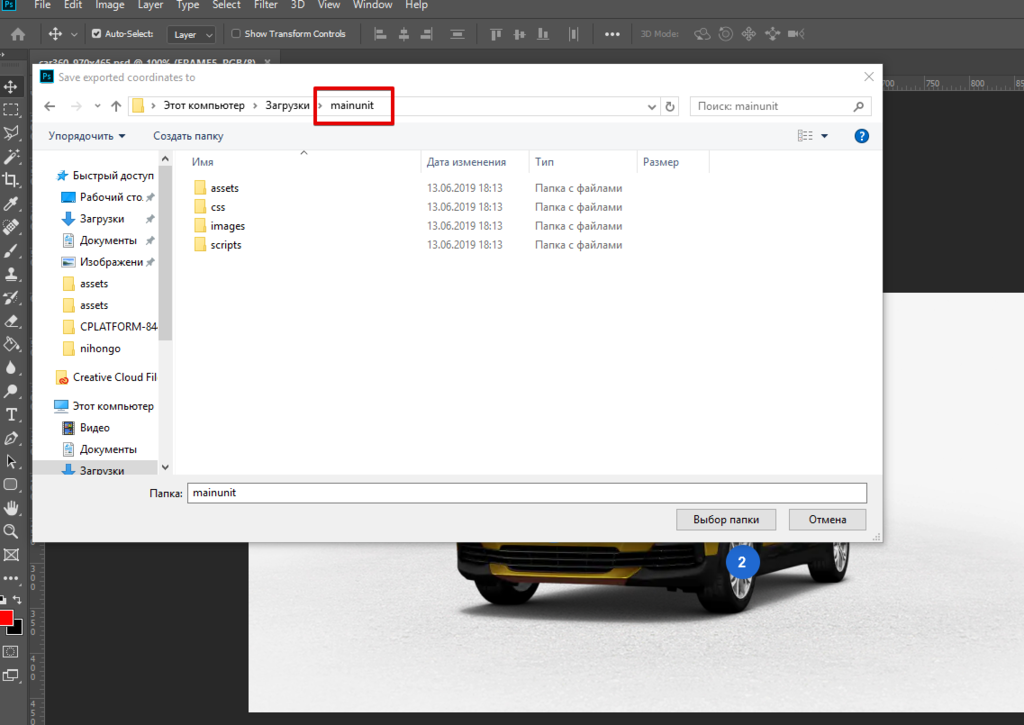
Designer should drop a placeholder on a page, set size and position and add a unique title id. This placeholder will be used as a gallery.

## User defined parameters:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Property** | **Type** | **Default** | **Screenshot of property** | **Description** |
| placeholderId | string |  |  | placeholder title |
| imagesPath | string | "images/" |  | path to frame images, including left/right arrows |
| frames | string[] | string |  |  | array of images names, or filename template, plus **totalSlides** to be defined; for ex. "image{xx}.jpg", where {xx} is a slide number. |
| totalSlides | number |  |  | total slides to be shown, if **frames** is a string |
| speed | number | 15 |  | speed of item rotation |
| width | number | auto-calculated |  | width of the placeholder has been choice during hotspots setting **not necessary** -  needed only for responsive component |
| height | number | auto-calculated |  | height of the placeholder has been choice during hotspots setting **not necessary** -  needed only for responsive component |
| navigationButtons[] {     id     direction   } | array {   string   "left" | "right" } |  |  | navigation buttons |
| bgCursor | string-css-pointer | "pointer" |  | defines background pointer style |
| arrowCursor | string-css-pointer | "pointer" |  | defines navigation arrows pointer style |
| popupsBackdropColor | string-css-color |  |  | popup background color |
| popupOpenByRollover | boolean | false |  | open corresponding pop-ups by hovering hotspots |
| bgImageFill **\*new in 3.10.1** | string-css-bg-size | "contain" |  | slides image fitting |
| preloaderColor | string-css-color | "rgb(0, 174, 239)" |  | color of preloader line / preloader background, if image is provided |
| preloaderContainerHeight | string-css-length | "3px" |  | height of the preloader possible values: XXpx | XX% |
| preloaderImage | string-url |  |  | image like jpg|gif|svg rendered as background of preloader |
| preloaderImagePosition | string-css-bg-position | "center" |  | position of the preload image inside the preload container,  possible values: top | bottom | center | left | right |
| preloaderImageSize | string-css-bg-size | "auto" |  | "background-size" for image inside the preloader container, possible values: auto | contain | cover |
| preventRender | boolean | false |  | Prevent automatic rendering at the ad-unit initialization If value is false — the "render" function should be called anytime later |
| restrict {     execution     size } | object {String} |  |  | If provided, component would be rendered or not based on inputs execution — "inframe", "instream", "display", "mobile"; or multiple at once "instream, display" and so on size — "phone", "tablet"; or multiple at once "phone, tablet" and so on |

or you can export it directly from Photoshop using Export Hotspots and Popups Coordinates for threesixty component script (doesn't export any images, only json):

* 1. Place [this file](https://exponential.box.com/s/p2v3lno02bwfeftlvbynvgm7ofykbera) in
     1. Mac: '~/Applications/Adobe Photoshop CS#/Presets/Scripts/'  <http://prntscr.com/ryazmo>
     2. Win: 'C:\Program Files\Adobe\Adobe Photoshop CS#\Presets\Scripts\'
  2. Restart Photoshop
  3. File -> Scripts -> Export Hotspots and Popups Coordinates for threesixty component (Or you can just select it here File -> Scripts -> Browse)  
     

4. Choose the "mainunit" folder in the dialog form.   


5. The threesixty view settings will be put in the place marked below (after the last component and before this text /\*\* next components code will start from here start with "," \*\*/)  (comma will be added automatically). Be sure that you delete the old threesixty settings before run the script again (it doesn't remove it automatically)  


6. You can use [this template.](https://jira.exponential.com/secure/attachment/147288/threesiztyViewHotspotsExportTemplate.psd)  Here is the PSD [example](https://jira.exponential.com/secure/attachment/147289/car360_970x465.psd).  
  
7. To hide a hotspot or popup from a frame you can delete the layer with his name (hotspot5 for example) or change name to "hotspot5-hidden"

## Component Callbacks:

* 1. "**onImageClicked**" (fires on an image click)
  2. "**onLeftArrowClick**" (fires on left arrow click event)
  3. "**onRightArrowClick**" (fires on right arrow click event)

## Methods:

* 1. Expo.designerAPI.threesixtyViewWithHotspots  - get all copies of the component,
  2. Expo.designerAPI.threesixtyViewWithHotspots.get('placeholderId') - find a unique copy of the component
  3. **reset**(animationOn) - set the slider to the default position, animationOn can be true or false, default - false.
  4. **init**(options) - reset default config, options is a object of componenta parameters
  5. **render**() - render a dom of the component with default config
  6. **update**(options) - destroy the dom and rerender it with a new options

## Examples:

* 1. Expo.designerAPI.threesixtyViewWithHotspots.get('threesixtyViewWithHotspots\_1').reset(true); - reset only one copy of the component with a placeholder 'threesixtyViewWithHotspots\_1'
  2. Expo.designerAPI.threesixtyViewWithHotspots.reset(true); - reset all copies of the component.
* Code Snippet
  1. **Use this line of code on Tab to Initialise the Three Sixty View With Hotspots (you need to set "preventRender" to true to use it)**  
     Expo.designerAPI.components.get("threesixtyViewWithHotspots\_1").render();
  2. **Use this line of code on Tab to reset the Three Sixty View With Hotspots**Expo.designerAPI.components.get("threesixtyViewWithHotspots\_1").reset();
* {
* name: "threesixtyViewWithHotspots",
* props: {
* preventRender: true,
* placeholderId: "threesixtyViewWithHotspots\_1",
* imagesPath: "images/",
* frames: [
* "img1.jpg",
* "img2.jpg",
* "img3.jpg",
* "img4.jpg",
* "img5.jpg",
* "img6.jpg",
* "img7.jpg",
* "img8.jpg",
* "img9.jpg",
* "img10.jpg"
* ],
* width: 970,
* height: 466,
* speed: 15,
* navigationButtons: [
* {
* id: "threesixtyViewWithHotspots\_1\_leftArrow",
* direction: "left",
* events: [
* {
* onArrowClicked: function(Expo, obj) {Expo.designerAPI.firePixel("onLeftArrowClick\_1", [{ "eventName": "intLive", "multi": true }]);}
* }
* ]
* },
* {
* id: "threesixtyViewWithHotspots\_1\_rightArrow",
* direction: "right",
* events: [
* {
* onArrowClicked: function(Expo, obj) {Expo.designerAPI.firePixel("onRightArrowClick\_1", [{ "eventName": "intLive", "multi": true }]);}
* }
* ]
* }
* ]
* },
* events: {
* onImageClicked: function(Expo, obj) { Expo.designerAPI.firePixel("onImageClicked\_1", [{"eventName":"intLive", "multi":true}]); }
* }

}

## Supported interaction Events and Detail

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **User Action** | **Event Name** | **Default Source**  **All in bound pixel with have a suffix TabTracker id which is defined by user** | **Default Sub Source** | **Parameter in insight pixel** | **Can i change source value in non muse BEFORE Advance Pixel Management Release?** | **Can i change source value in non muse AFTER Advance Pixel Management Release? (if component is present in VDX STUDIO)** | **Can i change source value in Muse BEFORE Advance Pixel Management Release?** | **Can i change source value in Muse AFTER Advance Pixel Management Release?** | **Initiator(function/callback initiating the pixel)** |
| click on left Arrow | inbound | leftArrow | {  componentName: "COMPONENT NAME"  tabName: TAB Label like **Offers**  tabSequence: TAB Number like tab{Number} for example tab1, tab2 where Number is integer value  subSource: hotSpotNumber like hotSpot{Number} for example hotSpot1, hotSpot2 where Number is integer value  } | event=intLive and fr:{source) | NO | NO | YES | YES | onLeftArrowClick |
| click on right Arrow | inbound | rightArrow | {  componentName: "COMPONENT NAME"  tabName: TAB Label like **Offers**  tabSequence: TAB Number like tab{Number  } for example tab1, tab2 where Number is integer value  subSource: hotSpotNumber like hotSpot{Number} for example hotSpot1, hotSpot2 where Number is integer value  } | event=intLive and fr:{source) | NO | NO | YES | YES | onRightArrowClick |
| click on image | inbound | imageClicked | {  componentName: "COMPONENT NAME"  tabName: TAB Label like **Offers**  tabSequence: TAB Number like tab{Number  } for example tab1, tab2 where Number is integer value  subSource: hotSpotNumber like hotSpot{Number} for example hotSpot1, hotSpot2 where Number is integer value  } | event=intLive and fr:{source) | NO | NO | YES | YES | onImageClicked |

# TabComponent.Common.VDX.Components.AdEx

**Template/Demo URLs:** [ClickHere](https://wiki.exponential.com/display/RMDOC/TabComponent.Templates.AdEx)

## Functionality:

Component allows to show set of creative blocks or custom html code as a tabs gallery with navigation.

## Muse setup:

Designer should drop a view placeholder on a page, set size and position and add a unique title id. This placeholder will be used as a tabs-gallery view.

Next, designer should drop a menu placeholder on a page, set size and position and add a unique title id. This placeholder will be used as a tabs-gallery menu.

Also designer have to design each tab as a separate creative block.

## User defined config:

|  |  |  |  |
| --- | --- | --- | --- |
| **Property** | **Type** | **Default** | **Description** |
| isRoot | Boolean | false | Should be true in case if it's root component to be used as main unit navigation |
| handleOrientation | Boolean | false | Should be true in case if same tabs-component placehlder is used in different breakpoints |
| placeholderId | String | — | Placeholder title id |
| menuPlaceholderId | String | — | In case of vdxmobile theme this would be a trigger element title id; in other cases — menu placeholder title id |
| arrange | String | "slides" | slides | stack  slides is for tab organize stack is for single web page organize |
| tabs: [ {     type     title     menuImage     menuStyle     placeholderId     onClick }, ... ] | Array     String     String     Url-String     Css-String     String     Function | — | Each array item is a config for each of the tab     tab | landing — tab is for content, landing is for opening landing page     Tab title to be shown in menu, if menuImage is not provided     If provided would be rendered as menu title     Tab-specific menu styling options     If type is tab, this placeholderId element would be shown as a content block     Callback function on click on menu item, or on scroll event in case of stack arrange |
| menuDesign: {     direction     orientation     theme     align     overflow     style     logoStyle     triggerStyle     triggerImg     items: {         normal         hover         active     } } | Object     String     String     String     String     String     Css-String     Css-String     Css-String     Url-String     Object         JS-Css         JS-Css         JS-Css | "horizontal" "cw0" "default" "center" "visible" none none none none none none none none | Menu styling options     horizontal | vertical     [*cw0*](https://i.imgur.com/45iOJlI.png) | [*cw90*](https://i.imgur.com/a6VljHe.png) | [*ccw90*](https://i.imgur.com/BgtbMla.png) | cw180 | ccw180     default | vdxconnect | vdxmobile (vdxmobile generates side menu, other generates buttons)     {left|right|top|bottom|center} | {left|center|right, top|center|bottom} — single or double value     [*visible*](https://i.imgur.com/AvHVG9m.png) | [*hidden*](https://i.imgur.com/ZcbjZYq.png) | [resize](https://i.imgur.com/QgMnSLA.png) (doesn't affect vdxmobile theme)     Stylize of menu items container block     In case of vdxmobile theme this would stylize hamburger-side-menu logo element     In case of vdxmobile theme this option would stylize the trigger     In case of vdxmobile theme this option would render image inside of the trigger     Menu items stylize for different states (each is optional). See [MDN: Block](https://developer.mozilla.org/en-US/docs/Learn/CSS/Building_blocks/Backgrounds_and_borders), [MDN: Text](https://developer.mozilla.org/en-US/docs/Learn/CSS/Styling_text/Fundamentals)            normal state — (for ex.: {font:"Helvetica, Arial, sans-serif", color: "#EEE", background:"#666"})         mouseover state — (for ex.: {"background": "#777"})         active state — (for ex.: {color: "#333", background:"#EEE"}) |
| dotNavigation | Boolean | false | **EXPERT** Enable/disable dots gallery navigation (see additional dots properties in Standard Image Gallery wiki documentation). |
| arrowNavigation | Boolean | false | **EXPERT** Enable/disable left/right arrows navigation (see additional arrows properties in Standard Image Gallery wiki documentation). |
| swipeNavigation | Boolean | String | false | **EXPERT** Enable/disable horizontal swipe navigation (true | false | "auto"). |
| transitionStyle | String | "fade" | chain | push | overlay | crossfade | fade |
| transitionDuration | Seconds-Number | 1 | Time (sec) — transition time value to complete transition from slide to slide. |
| preventRender | Boolean | false | Prevent automatic rendering at the ad-unit initialization If value is false — the "render" function should be called anytime later |
| restrict: {     execution,     size, } | Object {String} | none | If provided, component would be rendered or not based on inputs execution — "inframe", "instream", "display", "mobile"; or multiple at once "instream, display" and so on size — "phone", "tablet"; or multiple at once "phone, tablet" and so on |

## Component Callbacks:

* **onTabChange(Expo, event)** (event parameters: type, title, tabFrom, tabTo, auto; fires on tab change event)
* **onMenuItemClick(Expo, event)** (event parameters: type, title, tabFrom, tabTo; fires on menu item click event)
* **onHamburgerLogoClick(Expo)** (vdx mobile side menu logo click callback)
* **onHamburgerOpened(Expo, event)** (vdx mobile side menu shown callback // event.isUserInitiated)
* **onHamburgerClosed(Expo, event)** (vdx mobile side menu closed callback // event.isUserInitiated)

## Font setup:

Refer to the font-embedding documentation: [Fonts.Common.VDX.Components.AdEx](file:////display/VDXComponents/Fonts.Common.VDX.Components.AdEx)

## Special techniques:

1. To make a **screenshot** of the mainunit for the **Mobile Instream** follow the [instruction](https://exponential.box.com/s/r568vbxql9k9ls97i2ranaseohay2xrx)

## Limitations:

1. Make sure in designer-config **tabs** component came before it's nested component
2. If Tabs Component used as main navigation, **Video Tab** has to be placed in it's final position, as shown <https://i.imgur.com/BdzmcQ7.jpg> (otherwise video would fly away and disappear during transition from teaser to mainunit stages)
3. Menu items of type "**landing**" should came as last menu elements, and no tabs should follow after landing buttons
4. In Muse for the placeholder block "**State Button**" should be used. While "Rectangle" block is also works fine, sometimes it renders as a block of shorten height than designer defined it.
5. **Internet Explorer** **doesn't** support vdxconnect menu items **shadow** outline

## Advanced use:

Get gallery component instance:  
var **tabsInstance** = Expo.designerAPI.tabs.get("placeholderId");

## Component methods:

* **tabsInstance.render()** (used to initially render the component)
* **tabsInstance.update(config)** (here could be provided parameters to override existing config and init new gallery view)
* **tabsInstance.reset()** (reset component to initial state)
* **tabsInstance.showTab(tabNo)** (show particular tab, numeration starts from 1)
* **tabsInstance.getCurrentTab()** (returns current tab number)
* **tabsInstance.getStructure()** (set of tabs root DOM elements, like tabsEl, tabsArray, menuEl, menuArray)

Code Snippet

* **Use this line of code on Tab to reset the Tab Component on initial state**  
  this.designerAPI.tabs.get("tabs\_placeholder").reset();
* **Use this line of code on Tab to hide Video Component (if video overlays tabs component, but not placed inside of the slide)**  
  this.designerAPI.playerHide();
* **Use this line of code on Tab to show hidden Video Component (if video overlays tabs component, but not placed inside of the slide)**  
  this.designerAPI.playerShow();

**Sample Config** Expand source

/\* Start TabsComponent Code from Here \*/

{

name: "tabs", // Do not change this value

props: {

            preventRender: false,

placeholderId: "tabs\_placeholder1",

menuPlaceholderId: "tabs\_menu1",

tabs: [

{

type: "tab", // "tab", "landing"

title: "VIDEO",

placeholderId: "tabs\_content1\_1",

onClick: function(Expo, event) {

Expo.designerAPI.firePixel("Video\_Tab", [{ "eventName": "intLive", "multi": true }]);

},

},

{

type: "tab", // "tab", "landing"

title: "ABOUT RX",

placeholderId: "tabs\_content1\_2",

onClick: function(Expo, event) {

Expo.designerAPI.firePixel("AboutRX\_Tab", [{ "eventName": "intLive", "multi": true }]);

},

},

{

type: "tab", // "tab", "landing"

title: "FIND DEALER",

placeholderId: "tabs\_content1\_3",

onClick: function(Expo, event) {

Expo.designerAPI.firePixel("FindDealer\_Tab", [{ "eventName": "intLive", "multi": true }]);

},

},

],

menuDesign: {

direction: "horizontal", //"horizontal", "vertical"

orientation: "cw0",

theme: "vdxconnect", // "default", "vdxconnect"

align: "center bottom", // "left", "right", "top", "bottom", "center" // "{horizontal} {vertical}"

overflow: "resize", // "visible", "hidden", "resize"

items: {

normal: {

"font": "11px brandon-grotesque, Helvetica, Arial, sans-serif",

"font-weight": "700",

"color": "#FFF",

"background": "#252525",

},

hover: {

"text-decoration": "underline",

},

active: {

"background": "#FFF",

"color": "#000",

},

},

},

transitionStyle: "fade", // "chain", "push", "overlay", "fade", "crossfade"

transitionDuration: 0.2,

},

events: {

onTabChange: function(Expo, event) {

switch(event.tabFrom) {

case 1:

Expo.designerAPI.pause(); // pause videoplayer when exiting 1st tab

break;

}

switch(event.tabTo) {

case 1:

Expo.designerAPI.resume(); // resume videoplayer when entering 1st tab

break;

case 2:

Expo.designerAPI.components.get("tabs\_content1\_2").reset();

break;

case 3:

if(!Expo.getTempValue("activeMapTab")){

Expo.setTempValue("activeMapTab", true);

if(Expo.designerAPI.getDynamicData().rawData || Expo.designerAPI.getDynamicData().rawData === null) {

var mapDefault = document.querySelector('div[data-expo-event="EXPOEVENT\_mapDefault"]');

mapDefault.style.zIndex = -1;

}

map.generateMap({mapID: "mapA", dynamicData: Expo.designerAPI.getDynamicData().rawData, localData: "json/data.json", defaultImageURL: "images/map\_default.jpg", noDataImage: "images/map\_default.jpg"});

};

break;

}

},

onMenuItemClick: function(Expo, event) {

}

}

},

{

name: "tabs", // Do not change this value

props: {

            preventRender: true,

placeholderId: "tabs\_content1\_2",

menuPlaceholderId: "tabs\_menu2",

tabs: [

{

type: "tab", // "tab", "landing"

title: "GALLERY",

placeholderId: "tabs\_content2\_1",

onClick: function(Expo, event) {

Expo.designerAPI.firePixel("AboutRX\_GallerySubtab", [{ "eventName": "intLive", "multi": true }]);

},

},

{

type: "tab", // "tab", "landing"

title: "COMPARE",

placeholderId: "tabs\_content2\_2",

onClick: function(Expo, event) {

Expo.designerAPI.firePixel("AboutRX\_CompareSubtab", [{ "eventName": "intLive", "multi": true }]);

},

},

{

type: "tab", // "tab", "landing"

title: "SPECS",

placeholderId: "tabs\_content2\_3",

onClick: function(Expo, event) {

Expo.designerAPI.firePixel("AboutRX\_SpecsSubtab", [{ "eventName": "intLive", "multi": true }]);

},

},

],

menuDesign: {

direction: "horizontal", //"horizontal", "vertical"

orientation: "cw0",

theme: "default", // "default", "vdxconnect"

align: "center", // "left", "right", "top", "bottom", "center" // "{horizontal} {vertical}"

overflow: "resize", // "visible", "hidden", "resize"

items: {

normal: {

"font": "13px brandon-grotesque, Helvetica, Arial, sans-serif",

"font-weight": "700",

"color": "#000",

"height": "40px",

// "padding": "0 20px",

"letter-spacing": "2px",

"background": "#FFF",

},

hover: {

"text-decoration": "underline",

},

active: {

"background": "#000",

"color": "#FFF",

},

},

},

transitionStyle: "push", // "chain", "push", "overlay", "fade", "crossfade"

transitionDuration: 0.2,

},

events: {

onTabChange: function(Expo, event) {

switch(event.tabTo) {

case 1:

Expo.designerAPI.components.get("tabs\_content2\_1").reset();

break;

case 2:

Expo.designerAPI.components.get("tabs\_content2\_2").reset(true);

break;

case 3:

Expo.designerAPI.components.get("tabs\_content2\_3").reset();

break;

}

},

onMenuItemClick: function(Expo, event) {

}

}

}

/\* End TabsComponent Code till Here \*/

**VDX Connect Mobile Instream solution** Expand source

// add style option into menuDesign section, this would scale menu to x2 size

menuDesign: {

style: "transform:scale(2); transform-origin:bottom right",

}

**Tab's Default Configuration for VDX MobileExpandable** Expand source

// tabs component, primary tabs navigation

{

name: "tabs", // Do not change this value

props: {

isRoot: true, // true in case if it's root tabs component

placeholderId: "tabs\_placeholder1",

menuPlaceholderId: "tabs\_menu1",

arrange: "stack", // slides | stack

tabs: [

{

type: "tab", // tab | landing

title: "Video",

menuImage: "", // url

menuStyle: "", // css values

placeholderId: "tabs\_content1\_1",

onClick: function(Expo, event) {

// console.info(event.tabFrom, event.tabTo);

},

},

// ...

],

menuDesign: {

theme: "vdxmobile", // default | vdxconnect | vdxmobile

style: "background-color: #000000", // css values // container-block style

logoStyle: "background-image: url(assets/menu-logo.png)", // css values // vdxmobile logo styling

items: {

normal: { // object{css-property-1: css-value-1, css-property-2: css-value-2, ...}

"font": "Helvetica, Arial, sans-serif",

"font-weight": "bold",

"color": "#EEE",

"background-color": "#666",

"text-transform": "uppercase",

},

hover: { // object{css-property-1: css-value-1, css-property-2: css-value-2, ...}

"background-color": "#777",

},

active: { // object{css-property-1: css-value-1, css-property-2: css-value-2, ...}

"color": "#333",

"background-color": "#EEE",

},

},

},

transitionStyle: "fade", // chain | push | overlay | crossfade | fade

transitionDuration: .333, // transition time (sec)

},

events: {

onTabChange: function(Expo, event) {

// console.info(event.type, event.title, event.tabFrom, event.tabTo, event.auto);

},

onMenuItemClick: function(Expo, event) {

// console.info(event.type, event.title, event.tabFrom, event.tabTo);

},

onHamburgerLogoClick: function(Expo) {

// vdx mobile side menu logo click callback

},

onHamburgerOpened: function(Expo, event) {

// vdx mobile side menu shown callback // event.isUserInitiated

},

onHamburgerClosed: function(Expo, event) {

// vdx mobile side menu closed callback // event.isUserInitiated

},

   },

},

## Supported interaction Events and Detail

| **User Action** | **Event Name** | **Default Source** | **Default Sub Source** | **Parameter in insight pixel** | **Can i change source value in non muse BEFORE Advance Pixel Management Release?** | **Can i change source value in non muse AFTER Advance Pixel Management Release?** | **Can i change source value in Muse BEFORE Advance Pixel Management Release?** | **Can i change source value in Muse AFTER Advance Pixel Management Release?** | **Initiator(function/callback initiating the pixel)** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| user-initiated tab change | inbound | tab tracker id (as defined by user) | {   componentName: "tabs",  tabSequence: "tab{TARGET\_TAB\_ID}", // **tab1, tab2, ...**   subSource: "tab{TARGET\_TAB\_ID}", // **tab1, tab2, ...** } | event=intLive and fr:{source) | NO | NO | YES | YES | onTabChange |

# SyncGallery.Common.VDX.Components.AdEx

**Template/Demo URLs:** [Click Here](https://wiki.exponential.com/display/RMDOC/SyncGallery.Templates.AdEx)

## Functionality:

* + Pin points over the video are optional. It might be used if video has products inside of the video.
  + Products click pixels defined as
    - Product{N}Thumb and Product{N}Pin, where {N} is product no
  + clickLive pixels for product pin and thumb will be defined in designer-config.js file with custom4 value as shown below:
    - **Expo.designerAPI.fireDynamicPixel("Product{N}Card", [{"eventName": "clickLive", "data": {"itemID": "p{N}", "sku": {product\_sku}, "src": "card"}, "isBillable": true}]);**
    - **Expo.designerAPI.fireDynamicPixel("Product{N}Pin", [{"eventName": "clickLive", "data": {"itemID": "p{N}", "sku": {product\_sku}, "src": "pin"}, "isBillable": true}]);**
    - **Sample:**
  + Landing page for pins are same as corresponding thumbs landing page.
  + **IMPORTANT: Auto scroll is preventing when user interacts with the product list navigation, until video is replayed, or user srolls back to the last product himself**

## Muse setup:

* + Designer should drop a placeholder on a page, near the video, set size and position and add a unique title id. This placeholder will be used as a container for the thumbs.
  + ~~If SyncGallery planned to be used on secondary tab, it has to be provided with video container unique titale id. This placeholder will be used as a container for the video.~~

## User defined config:

|  |  |  |  |
| --- | --- | --- | --- |
| **Property** | **Type** | **Default** | **Description** |
| placeholderId | String | — | Placeholder title id. |
| ~~videoPlaceholderId~~ | String | none | Video placeholder title id. If provided, video would be switched into this container. |
| ~~videoId~~ | String | "video1" | Video Id, for ex. "video2" |
| ~~startAdState~~ | String | "vpm" | teaser | vpm | maiunit Ad-unit state where pins are starting show. |
| direction | String | "vertical" | horizontal | vertical Product cards arranging horizontal or vertical. |
| dataGeneral {     tagsTimeShift     introDuration     outtroDuration } | Object {Number}     Seconds-Number     Seconds-Number     Seconds-Number | none | Video-specific properties:     tagsTimeShift if it needs to show/hide tags earlier/later all at once     introDuration beginning part of the video which is meant to be no tags were shown before this time     outtroDuration ending part of the video which is meant all tags were shown before this time |
| dataTags [     {        timeIn        timeOut        pin {x, y}     }, ... ] | Array [Object]      Seconds-Number     Seconds-Number     Position-Object | none | Array of tags      if provided, tag would be show at this point of video time     if provided, tag would be removed from the video at this point of video time     if provided pin would be shown over the video at this video-space position  if both timeIn/timOut are not providen then show time would calculated automatically beetween [introDuration .. outtroDuration] |
| dataProvider {     rawData     jsonUrl     card } | Object     Object { properties }     Url-String     Object { properties } | none | Inventory for loading or handling token data to render product cards.     Raw json-like data to be processed     Global or local path to json with raw data to be processed     Each property might be provided as one of option:         string path, for ex. "products[{x}].imageURL" (where {x} is a product index)         object {path: path\_to\_value, post: function(val){}}, }, for ex. { path: "products[{x}].deal", post: function(val){return "<span" + (val.match("¢") ? " class='cent'" : "") + ">" + val + "</span>"} } (where {x} is a product index) |
| fonts | Array [ String ] | none | Array of fonts as css notation to use local/remote fonts. See [MDN](https://developer.mozilla.org/en-US/docs/Web/CSS/@font-face)  For ex.: [ 'font-family: Bogle; font-weight: normal; src: url("assets/Bogle-Regular.woff") format("woff"), url("assets/Bogle-Regular.ttf") format("truetype");' ], |
| css | Css-String | none | Inline css to be applied to the gallery to stylize dynamic html content. |
| exceptionHtml | Html-String | "Data Error" | Html code to be shown in case of loading or parsing issues.     Add data-event="ClickCallbackId" attribute to track click events (where ClickCallbackId is unique event id). |
| cardHtml | Html-String | — | Html code to render card within slide.     Add data-event="ClickCallbackId" attribute to track click events (where ClickCallbackId is unique event id).     Add {{property}} in html code to substitute it with dynamic data. |
| cardSize | Style-Size-String | "100% 33.3%" | "{size}" | "{width} {height}" size in px/% of container |
| cardsSeparator | Style-Border-String | Style-String | none | Separator between cards. Two different notations available: css border-like | inline-style, for ex.:     "1px solid #999999"     "width: 3px; height: 50%; background: #FF0000" |
| pinImage | Url-String | — | URL path to pin image to overlay video and/or card. |
| pinImageFit | String | "contain" | cover | contain | unset | auto | initial For the details see [MDN](https://developer.mozilla.org/en-US/docs/Web/CSS/background-size) |
| pinVideoSize | Style-Size-String | "64px" | CSS size like value. For ex. "50px" - width and height at same time, "30px 50px" - width and height separately |
| pinCardSize | Style-Size-String | "64px" | CSS size like value. For ex. "50px" - width and height at same time, "30px 50px" - width and height separately |
| arrowSize | Style-Size-String | "20px" | One value for width or height (based on direction). |
| arrowImagePrev | Url-String | none | URL path to previous arrow image. If provided, custom image for the previous arrow will be used; If not provided, but right (next) image is provided - mirrored next image would be used. |
| arrowImageNext | Url-String | none | URL path to next arrow image. If provided, custom image for the next arrow will be used; If not provided, but previous image is provided - mirrored previous image would be used. |
| arrowImageFit | String | none | cover | contain | unset | auto | initial For the details see [MDN](https://developer.mozilla.org/en-US/docs/Web/CSS/background-size) |
| arrowsSeparator | Style-Border-String | Style-String | none | Separator between arrows and cards. Two different notations available: css border-like | inline-style, for ex.:     "1px solid #999999"     "width: 3px; height: 50%; background: #FF0000" |
| preventRender | Boolean | false | Prevent automatic rendering at the ad-unit initialization If value is false — the "render" function should be called anytime later |
| restrict {     execution     size } | Object {String} | none | If provided, component would be rendered or not based on inputs execution — "inframe", "instream", "display", "mobile"; or multiple at once "instream, display" and so on size — "phone", "tablet"; or multiple at once "phone, tablet" and so on |

## Component Callbacks:

* **onCardClick(Expo, event)** (available event parameters: id, tag; fires on card click event)
* **onPinClick(Expo, event)** (available event parameters: id, tag; fires on video-pin click event)
* **onLeftArrowClick(Expo)** (fires on left arrow click event)
* **onRightArrowClick(Expo)** (fires on right arrow click event)
* **onDataError(Expo)** (fires if data provider not able to load data, or loads non valid data)
* **onErrorClick(Expo)** (fires on click on error html)

## Font setup:

Refer to the font-embedding documentation: [Fonts.Common.VDX.Components.AdEx](file:////display/VDXComponents/Fonts.Common.VDX.Components.AdEx)

## Limitations:

1. Works for single video in primary tab only.
2. ~~MacOS Safari doesn't support smooth scroll, so new thumbs would be switched permanently~~

## Notes:

1. **The placeholder should be a container ('State button' or 'Rectangle').**
2. The placeholder id will only have component name without the initial **VDXC\_ in designer-config, while in muse designer need to define the placeholder id with VDXC\_component name**
3. **restrict** usage: if component has to be restricted to mobile phone only, this setting should be provided:
   * restrict: {
   * execution: "mobile",
   * size: "phone",
   * },

## Advanced use:

Get gallery component instance:

* var **syncGalleryInstance** = Expo.designerAPI.syncGallery.get("placeholderId");

## Component methods:

* **syncGalleryInstance.update(config)** (here could be provided parameters to override base config and init new product gallery view)
* **syncGalleryInstance.reset()** (reset component to initial state)
* **syncGalleryInstance.release()** (to hide video overlays such as pins)
* **syncGalleryInstance.getStructure()** (set of gallery root DOM elements, like slidesEl, slideArray, dotsEl, leftArrowEl, rightArrowEl)

## Additional global methods:

Expo.designerAPI.components.get("placeholderId"); // to get component by id  
Expo.designerAPI.components.reset(); // to reset all components

Code Snippet

* **Use this line of code on Tab to Initialize the SyncGallery**  
  Expo.designerAPI.components.get("syncGallery\_placeholder1").render();
* **Use this line of code on Tab to reset the SyncGallery**  
  Expo.designerAPI.components.get("syncGallery\_placeholder1").reset();
* **Use this line of code on Tab to Initialize the SyncGallery, or reset if it's already initialized**  
  Expo.designerAPI.components.get("syncGallery\_placeholder1").render(true);

**Sample Code** Expand source

/\* Start SyncGallery Code from Here \*/

{

name: "syncGallery", // Do not change this value

props: {

placeholderId: "syncGallery\_placeholder1",

videoPlaceholderId: "syncGallery\_video1", // optional

videoId: "video2",

dataGeneral: {

tagsTimeShift: 0, // time in sec // if it needs to show/hide tags earlier/later all at once

introDuration: 1, // time in sec // beginning part of the video which is meant to be no tags were shown before this time

outtroDuration: 2, // time in sec // ending part of the video which is meant all tags were shown before this time

},

dataTags: [ // array of tags

{

timeIn: 3.6, // time in sec // if provided, tag would be show at this point of video time; if nothing provided show time would calculated automatically beetween [introDuration .. outtroDuration]

timeOut: 5.06, // time in sec //if provided, tag would be removed from the video at this point of video time; if not then hide time would calculated automatically beetween [introDuration .. outtroDuration]

pin: {x: 671, y: 363}, // {px, px} // if provided pin would be shown over the video at this position (coordinates are relative to video natural width & height)

// ...

},

// ...

],

dataProvider: {

rawData: {}, // same data object as in json

jsonUrl: 'assets/products.json', // url // path to json

card: { // properties to be used in card template

imageURL: 'products[{x}].thumb',

landingPageUrl: 'products[{x}].url',

},

},

direction: "vertical", // horizontal | vertical

fonts: [ // array of fonts as css notation

'font-family: Bogle; font-weight: normal; src: url("assets/Bogle-Regular.woff") format("woff"), url("assets/Bogle-Regular.ttf") format("truetype");',

'font-family: Bogle; font-weight: bold; src: url("assets/Bogle-Bold.woff") format("woff"), url("assets/Bogle-Bold.ttf") format("truetype");',

],

css: '.wrapper {position:relative; width:100%; height:100%; font-family:Bogle,sans-serif; background:#fff center/contain;}', // inline-css

exceptionHtml: 'Data error', // html with {{properties}} from dataProvider.card

cardHtml: '<div class="wrapper" style="background-image:url({{imageURL}})"></div>', // html with {{properties}}

cardSize: '100% 112px', // {size} | {width, height}

cardsSeparator: '1px solid #999999', // css border value

pinImage: 'assets/pin.svg', // url

pinImageFit: 'contain', // cover | contain | auto | unset | initial    pinVideoSize: '64px', // {css-size} // one value for width & height

pinCardSize: '42px', // {css-size} // one value for width & height

arrowSize: '20px', // {css-size} // one value for width or height (based on direction)

arrowImagePrev: 'assets/nav\_arrow\_prev.svg', // url

arrowImageNext: 'assets/nav\_arrow\_next.svg', // url

arrowImageFit: 'contain', // cover | contain | auto | unset | initial   },

events: {

onDataError: function(Expo) {

},

onErrorClick: function(Expo) {

},

onCardClick: function(Expo, event) {

// event.id holds tag index (starting from 1)

// event.tag is an object{..} and it holds tag data

},

onPinClick: function(Expo, event) {

// event.id holds tag index (starting from 1)

// event.tag is an object{..} and it holds tag data

},

onPrevArrowClick: function(Expo, event) {

// event.partial = [ .. ] // partial visible cards

// event.full = [ .. ] // fully visible cards

// event.full[0] would return index of first visible card

},

onNextArrowClick: function(Expo, event) {

// event.partial = [ .. ] // partial visible cards

// event.full = [ .. ] // fully visible cards

// event.full[0] would return index of first visible card

},

      },

},

/\* End SyncGallery Code \*/

## Supported interaction Events and Detail

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **User Action** | **Event Name** | **Default Source**  **All in bound pixel with have a suffix TabTracker id which is defined by user** | **Default Sub Source** | **Parameter in insight pixel** | **Can i change source value in non muse BEFORE Advance Pixel Management Release?** | **Can i change source value in non muse AFTER Advance Pixel Management Release? (if component is present in VDX STUDIO)** | **Can i change source value in Muse BEFORE Advance Pixel Management Release?** | **Can i change source value in Muse AFTER Advance Pixel Management Release?** | **Initiator(function/callback initiating the pixel)** |
| click on left Arrow | inbound | leftArrow | {  componentName: "COMPONENT NAME"  tabName: TAB Label like **Offers**  tabSequence: TAB Number like tab{Number  } for example tab1, tab2 where Number is integer value  subSource: slideNumber like slide{Number} for example slide1, slide2 where Number is integer value  } | event=intLive and fr:{source) | NO | NO | YES | YES | onLeftArrowClick |
| click on right Arrow | inbound | rightArrow | {  componentName: "COMPONENT NAME"  tabName: TAB Label like **Offers**  tabSequence: TAB Number like tab{Number  } for example tab1, tab2 where Number is integer value  subSource: slideNumber like slide{Number} for example slide1, slide2 where Number is integer value  } | event=intLive and fr:{source) | NO | NO | YES | YES | onRightArrowClick |
| click on video-pin | outbound | tracker id (as defined by user) | {  componentName: "COMPONENT NAME"  tabName: TAB Label like **Offers**  tabSequence: TAB Number like tab{Number  } for example tab1, tab2 where Number is integer value  sku: sku id present in the data provided  subSource: slideNumber like slide{Number} for example slide1, slide2 where Number is integer value  } | event=clickLive and fr:{source) | NO | YES | YES | YES | onPinClick |
| on click on card click | outbound | tracker id (as defined by user) | {  componentName: "COMPONENT NAME"  tabName: TAB Label like **Offers**  tabSequence: TAB Number like tab{Number  } for example tab1, tab2 where Number is integer value  sku: sku id present in the data provided  subSource: itemID like item{Number} for example item1, item2 where Number is integer value  } | event=clickLive and fr:{source) | NO | YES | YES | YES | onCardClick |
| if data provider not able to load data, or loads non valid data | inbound | dataError | {  componentName: "COMPONENT NAME"  tabName: TAB Label like **Offers**  tabSequence: TAB Number like tab{Number  } for example tab1, tab2 where Number is integer value  subSource: slideNumber like slide{Number} for example slide1, slide2 where Number is integer value  } | event=intLive and fr:{source) | NO | NO | YES | YES | onDataError |
| on click on error html | inbound | error | {  componentName: "COMPONENT NAME"  tabName: TAB Label like **Offers**  tabSequence: TAB Number like tab{Number  } for example tab1, tab2 where Number is integer value  subSource: slideNumber like slide{Number} for example slide1, slide2 where Number is integer value  } | event=intLive and fr:{source) | NO | NO | YES | YES | onErrorClick |

# StandardImageGallery.Common.VDX.Components.AdEx

**Template/Demo URLs:** [ClickHere](https://wiki.exponential.com/display/RMDOC/StandardImageGallery.Templates.AdEx)

## Functionality:

Component allows to show set of image files within the defined container as a gallery.  
With optional points/thumbs, side arrows and swipe navigations.

## Muse setup:

Designer should drop a placeholder on a page, set size and position and add a unique title id. This placeholder will be used as a gallery.

## User defined config:

|  |  |  |  |
| --- | --- | --- | --- |
| **Property** | **Type** | **Default** | **Description** |
| placeholderId | String | — | Placeholder title id |
| slides | Template-String | Array | — | EMPTY STRING: empty slides are created, suitable for manual control. STRING: base file name template; for example "local\_folder/image\_{xx}.jpg", where {xx} is a slide number; if slide number less than template suggested size, then leading zeros adding. ARRAY: urls to each slide. |
| thumbs | Template-String | Array | none | EMPTY STRING: no thumbs are created. STRING: base file name template; for example "local\_folder/thumb\_{xx}.jpg", where {xx} is a thumb number; if thumb number less than template suggested size, then leading zeros adding. ARRAY: urls to each thumb. |
| totalSlides | Number |  | If slides set up as template string then this property is total slides count to be shown from 1st to this value. If value is 0 then component is not initializing. |
| imageFit | String |  | cover | contain | unset | auto | initial For the details see [MDN](https://developer.mozilla.org/en-US/docs/Web/CSS/background-size) |
| slideStyle | Style-String | none | **EXPERT** Custom CSS styling for the slide. |
| dotNavigation | Boolean | true | Enable/disable dots gallery navigation. |
| arrowNavigation | Boolean | true | Enable/disable left/right arrows navigation. |
| swipeNavigation | Boolean | String | true | Enable/disable horizontal swipe navigation (true | false | "auto"). |
| transitionStyle | String | "chain" | chain | push | overlay | crossfade | fade |
| transitionDuration | Seconds-Number | 1 | Time (sec) — transition time value to complete transition from slide to slide. |
| loopSlides | Boolean | true | Enable/disable seamless slide change on border slide numbers. |
| autoPlay | Seconds-Number | 0 | Time (sec) — allows to automatically change slides if no interaction happened. |
| resumeAutoPlay | Seconds-Number | 0 | Time (sec) — allows to resume autoplay from the last interaction if "**autoPlay**" property is not 0. |
| dotsPosition | String | "bottom" | top | bottom |
| dotsOverflow | String | none | fade | scale — [**Details**](http://prntscr.com/tl5saq) |
| dotsStyle | Style-String | none | **EXPERT** Custom CSS styling for the dots container (for ex. for the better positioning). |
| dotsGap | Style-Size-String | "10px" | CSS width value to set the gap size between dots. |
| dotStyle {     normal     hover     active } | Object {Style-String} | none | **EXPERT** Custom CSS styling for the specific dot states (each is optional).     normal state — (for ex.: "border:0px solid #FFF; background-color:rgba(255, 255, 255, 0.1); ")     mouseover state — (for ex.: "border-size:2px; background-color:rgba(255, 255, 255, 0.25); ")     active state — (for ex.: "border-size:1px; background-color:#9999FF; ") |
| dotOutlineNormal | Style-Border-String | none | CSS border value. If provided - dot normal state would have this style; For ex. "1px solid #FFFFFF". For details see [MDN](https://developer.mozilla.org/en-US/docs/Web/CSS/border). |
| dotOutlineActive | Style-Border-String | none | CSS border value. If provided - dot active state would have this style; For ex. "2px solid #FFFFFF". For details see [MDN](https://developer.mozilla.org/en-US/docs/Web/CSS/border). |
| dotSize | Style-Size-String | none | CSS size like value. For ex. "10px" - width and height at same time, "10px 20px" - width and height separately. |
| dotShape | String | "circle" | rectangle | circle |
| dotThumbFit | String | "contain" | cover | contain | unset | auto | initial For the details see [MDN](https://developer.mozilla.org/en-US/docs/Web/CSS/background-size) |
| arrowsPosition | String | "center" | center | top | bottom |
| arrowsMargin | Style-Margin-String | none | CSS margin like value. For ex. "10px", "0px 10px", "0px 0px 100px", "0px 30px 0px 0px"; any valid css-margin value for arrows container. |
| arrowStyle {     normal     hover     disabled } | Object {Style-String} | none | **EXPERT** Custom CSS styling for the specific arrow states (each is optional).     normal state — (for ex.: "opacity:0.8 ")     mouseover state — (for ex.: "opacity:1 ")     disabled state (first and last slide in case of loopSlides: false) — (for ex.: "opacity:0.1 ") |
| arrowSize | Style-Size-String | "44px 80px" | CSS size like value. For ex. "50px" - width and height at same time, "30px 50px" - width and height separately |
| arrowColor | Style-Color-String | "#FFFFFF" | CSS color like value. For ex. "#FFFFFF" |
| arrowBackgroundColor | Style-Color-String | none | CSS color like value. For ex. "rgba(255, 255, 255, 0.25)" |
| arrowImageLeft | Url-String | none | URL path to left (previous) arrow image. If provided, custom image for the left arrow will be used; If not provided, but right (next) image is provided - mirrored right (next) image would be used. |
| arrowImageRight | Url-String | none | URL path to right (next) arrow image. If provided, custom image for the right (next) arrow will be used; If not provided, but left (previous) image is provided - mirrored left (previous) image would be used. |
| arrowImageFit | String | "intial" | cover | contain | unset | auto | initial For the details see [MDN](https://developer.mozilla.org/en-US/docs/Web/CSS/background-size) |
| arrowsCustomize {     container     left     right } | Object {Style-String} | none | **EXPERT** Custom CSS styling for the specific arrow elements (each is optional).     container style (to define position and size of the arrows container element) — (for ex.: "left:10px; top:10px; width:200px; height:100px;")     left/previous style (to define position and size of the left/previous arrow element) — (for ex.: "width:80px; height:100%;")     right/next style (to define position and size of the right/next arrow element) — (for ex.: "width:80px; height:100%;") |
| controls { **\*new in 3.10.1**     enabled     position     style     color } | Object     Boolean     String     Style-String     Style-Color | false   "br"   ""   "#FFF" | Playback controls (mute & play/pause, each is optional)     true | false (true by default)     tl | tr | bl | br (which means top-left | top-right | bottom-left | bottom-right relatively)     additional styles (to fine-tune position, or add some filters)     controls color (#FFFFFF by default) |
| progress { **\*new in 3.10.1**     enabled     timeColor     fullColor } | Object     Boolean     Style-Color-String     Style-Color-String | false   ""   "" | Progress bar settings (each is optional)     true | false (true by default)     Playback progress bar coloring, for ex.: "#0099FF"     Background layer coloring, for ex.: "rgba(255, 0, 0, 0.5)" |
| theme { **\*new in 3.10.1**     name     size     timeSize     timeWidth     bgColor     playPauseColor     style } | Object     String     Number     Number     Number     Style-Color-String     Style-Color-String     Style-String | "bar"   ""   ""   ""   ""   ""   "" | Progress theme customisation     Default theme selection should be **circular.**     **name: default | circular** (default theme means line bar at the bottom, circular theme means progress bar in circular shape)     **size**: Set the size of background circle.     **timeSize**: Set the size of circular progress bar.     **timeWidth**: Set the thickness of the circular progress bar.     **bgColor**: Control playpause color in circular theme apart of mute/unmute button     **playPauseColor**: Control playpause color in circular theme apart of mute/unmute button     **style**: Positioning styles like top | bottom | left | right | transform applicable. |
|  |  |  |  |

## Component Callbacks:

* **onSlideClick(Expo, event)** (available event parameters: slide; fires on slide click event)
* **onSlideChanged(Expo, event)** (available event parameters: slideFrom, slideTo, auto; fires on slide change event)
* **onDotClick(Expo, event)** (available event parameters: slideFrom, slideTo; fires on bottom navigation click event)
* **onLeftArrowClick(Expo)** (fires on left arrow click event)
* **onRightArrowClick(Expo)** (fires on right arrow click event)

## Limitations:

1. **Dots or thumbs** navigation is only supported in Engage it is not supported for VDX connect.
2. Thumbs and dots navigation in Engage adunits can be used keeping the thumbs width less or equal to the container size.
3. Dots navigation position either controlled by **dotsPosition:** "**top**"/"**bottom**", either by **dotsStyle**.
4. Options **dotsStyle**, **dotStyle** and **arrowStyle** should be used with care, as they might change visual appearance dramatically, and these options might override other single-styling options.

## Notes:

1. **All properties having "RGB" value can also be use with "#" values.**
2. **The placeholder should be a container ('State button' or 'Rectangle').**
3. The place holder id will only have component name without the initial **VDXC\_ in designer-config, while in muse designer need to define the placehoder id with VDXC\_component name.**
4. Keep **transition time** under 1 second for smoothest animation look.
5. **Swipe navigation** is preferable to use in **mobile units** only. Use "auto" to handling mobile/desktop behavior automatically.
6. If **onSlideClick** event is provided, slides has cursor: pointer. Empty **onSlideClick** callback might cause confusing.

## Advanced use:

Get gallery component instance:  
var **galleryInstance** = Expo.designerAPI.imageGallery.get("placeholderId");

## Component methods:

* **galleryInstance.update(config)** (here could be provided parameters to override existing config and init new gallery view)
* **galleryInstance.reset()** (reset component to initial state)
* **galleryInstance.showSlide(slideNo)** (show particular slide, numeration start from 1)
* **galleryInstance.play()** (start autoplay, if it's available)
* **galleryInstance.pause()** (pause autoplay, if it's available)
* **galleryInstance.getCurrentSlide()** (returns current slide number)
* **galleryInstance.getStructure()** (set of gallery root DOM elements, like slidesEl, dotsEl, leftArrowEl, rightArrowEl)
* **galleryInstance.render**() - render the component with default config

## Additional global methods:

Expo.designerAPI.components.get("placeholderId"); // to get component by id  
Expo.designerAPI.components.reset(); // to reset all components

Code Snippet

* **Use this line of code on Tab to Initialise the Standard ImageGallery**  
  Expo.designerAPI.imageGallery.get("imageGallery\_placeholder1").render();
* **Use this line of code on Tab to reset the Standard ImageGallery**  
  Expo.designerAPI.imageGallery.get("imageGallery\_placeholder1").reset();

**Sample Code** Expand source

/\* Start StandardImageGallery Code from Here \*/

{

name: "imageGallery", // Do not change this value

props: {

            preventRender: true,

placeholderId: "imageGallery\_placeholder1",

slides: "images/img{x}.png",

// thumbs: "images/t{x}.jpg",

totalSlides: 4,

imageSize: "cover", // "unset", "cover", "contain"

slideStyle: "background-size:unset; background-position:center center; background-color:#000;",

dotNavigation: false,

arrowNavigation: true,

swipeNavigation: false,

transitionStyle: "chain", /\* "chain", "push", "overlay", "fade" \*/

transitionDuration: 1,

loopSlides: true,

autoPlay: 0,

resumeAutoPlay: 0,

dotStyle: {

"normal": "border:2px solid #FFF; background:rgba(255, 255, 255, 0.1); margin:5px; padding:5px; width:11px;",

"hover": "border:2px solid #29ADE4; background:rgba(255, 255, 255, 0.25);",

"active": "border:2px solid #29ADE4;",

},

dotSize: "11px",

dotShape: "rectangle", /\* "circle", "rectangle" \*/

dotsPosition: "bottom", /\* "top", "bottom" \*/

dotThumbSize: "cover", /\* "original", "cover", "contain" \*/

arrowSize: "60px 60px",

arrowColor: "#29ADE4",

arrowBackgroundColor: "rgba(255, 255, 255, 0.25)",

arrowsMargin: "5px", /\* "top left/right bottom" or "uniform"\*/

arrowImageRight: "images/arrow\_next.png",

arrowImageLeft: "images/arrow\_pre.png",

preventRender: true,

arrowStyle: {

"normal": "background-color:rgba(255, 255, 255, 0.5); opacity:0.9; ",

"hover": "background-color:rgba(0, 0, 0, 0.25); opacity:1; ",

"disabled": "background-color:rgba(255, 255, 255, 0.1); opacity:0.1; ",

},

},

events: {

onSlideClick: function(Expo, event) {

Expo.designerAPI.fireDynamicPixel("gallery\_slide\_click\_" + event.slide, [{"eventName":"clickLive", "multi":true}]);

switch(event.slide) {

case 1:

Expo.designerAPI.openUrl("http://www.clicktag1.com", "gallery\_slide\_click\_" + event.slide);

break;

case 2:

Expo.designerAPI.openUrl("http://www.clicktag2.com", "gallery\_slide\_click\_" + event.slide);

break;

case 3:

Expo.designerAPI.openUrl("http://www.clicktag3.com", "gallery\_slide\_click\_" + event.slide);

break;

case 4:

Expo.designerAPI.openUrl("http://www.clicktag4.com", "gallery\_slide\_click\_" + event.slide);

break;

};

Expo.designerAPI.pause();

},

onSlideChange: function(Expo, event) {

if (!event.auto) Expo.designerAPI.fireDynamicPixel("gallery1\_slide\_change\_" + event.slideTo, [{ "eventName": "intLive", "multi": true }]);

},

onDotClick: function(Expo) {

Expo.designerAPI.firePixel("gallery1\_dot\_click", [{"eventName":"intLive", "multi":true}]);

},

onLeftArrowClick: function(Expo) {

Expo.designerAPI.firePixel("gallery1\_left\_arrow\_click", [{"eventName":"intLive", "multi":true}]);

},

onRightArrowClick: function(Expo) {

Expo.designerAPI.firePixel("gallery1\_right\_arrow\_click", [{"eventName":"intLive", "multi":true}]);

}

}

},

/\* End StandardImageGallery Code \*/

## Supported interaction Events and Detail

| **User Action** | **Event Name** | **Default Source**  **All in bound pixel with have a suffix TabTracker id which is defined by user** | **Default Sub Source** | **Parameter in insight pixel** | **Can i change source value in non muse BEFORE Advance Pixel Management Release?** | **Can i change source value in non muse AFTER Advance Pixel Management Release?** | **Can i change source value in Muse BEFORE Advance Pixel Management Release?** | **Can i change source value in Muse AFTER Advance Pixel Management Release?** | **Initiator(function/callback initiating the pixel)** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| click on Left Arrow | inbound | leftArrow | {   componentName: "COMPONENT NAME",   tabName: "HOST TAB LABEL", // **Offers**   tabSequence: "tab{HOST\_TAB\_ID}", // **tab1, tab2, ...**   subSource: "slide{ID}", // **slide1, slide2, ...** } | event=intLive and fr:{source) | NO | NO | YES | YES | onLeftArrowClick |
| click on Right Arrow | inbound | rightArrow | {   componentName: "COMPONENT NAME",   tabName: "HOST TAB LABEL", // **Offers**   tabSequence: "tab{HOST\_TAB\_ID}", // **tab1, tab2, ...**   subSource: "slide{ID}", // **slide1, slide2, ...** } | event=intLive and fr:{source) | NO | NO | YES | YES | onRightArrowClick |
| click on dot | inbound | dotSelect | {   componentName: "COMPONENT NAME",   tabName: "HOST TAB LABEL", // **Offers**   tabSequence: "tab{HOST\_TAB\_ID}", // **tab1, tab2, ...**   subSource: "slide{ID}", // **slide1, slide2, ...** } | event=intLive and fr:{source) | NO | NO | YES | YES | onDotClick |
| on click on slide | outbound | tracker id (as defined by user) | {   componentName: "COMPONENT NAME",   tabName: "HOST TAB LABEL", // **Offers**   tabSequence: "tab{HOST\_TAB\_ID}", // **tab1, tab2, ...**   subSource: "slide{ID}", // **slide1, slide2, ...** } | event=clickLive and fr:{source) | NO | YES | YES | YES | onSlideClick |
| on slide change | inbound | slideChange | {   componentName: "COMPONENT NAME",   tabName: "HOST TAB LABEL", // **Offers**   tabSequence: "tab{HOST\_TAB\_ID}", // **tab1, tab2, ...**   subSource: "slide{ID}", // **slide1, slide2, ...** } | event=intLive and fr:{source) | NO | NO | YES | YES | onSlideChange |

# Scrollbar.Common.VDX.Components.AdEx

* **Template/Demo URLs:** [ClickHere](https://wiki.exponential.com/display/RMDOC/Scrollbar.Templates.AdEx)
* Config Details

## Functionality:

Component allows to show design block or raw html source within the defined container with scrollbar overlay (native or custom).  
The component has a set of properties that allow you to customize the appearance and functioning of the components.

## Muse setup:

Designer should drop a placeholder (should be block element, not a textfield or image) on a page, set size and position and add a unique title id (which will be used as a selector). This placeholder will be used as a scroll area.

Also designer have to design a content block, which will be shown within the scroll area. This block is also have to be titled with unique id.

## User defined parameters:

* + - **restrict**: Object (if provided, component would be rendered or not based on inputs)
      * **execution**: String ("inframe", "instream", "display", "mobile"; or multiple at once "instream, display")
      * **size**: String ("phone", "tablet"; or multiple at once "phone, tablet")
    - **placeholderId**: String (placeholder title)
    - **contentId**: String (content title)
    - **contentHtml**: String (raw text html source, if provided contentId is ignored)
    - **direction**: String (available options: "horizontal", "vertical", "both")
    - **customScroll**: Boolean (enable custom js scrollbars instead of native)
    - **customScrollAlwaysShow**: Boolean (show always custom scrollbar / show by user mouse-over interaction)
    - **customScrollTheme**: String (custom scrollbar design scheme, options: "default", "dark", "light", "blue")
    - **preventRender:** Boolean (prevent calling of "render" function at the initialization, the "render" function can be called anytime by a designer, default: true)

## Component Callbacks:

* + - **onScrolled(Expo)** - general event (fires on each scroll event, neither wheel, neither click&drag, tap&drag)
    - **onScrollStart(Expo)** - service event (fires on custom scrollbar scroll start event, after mouse click or touch start and move)
    - **onScrollEnd(Expo)** - service event (fires on custom scrollbar scroll end event, after mouse release or touch end)

## Limitations:

* + - IE (IE11 in particular) has some limited code features support, that cause:
      * Scrollbar is flickering and looks jerky.
      * Also scrollbar is not resetting to the initial state when reset is called.

## Notes:

* + **The placeholder should be a container ('State button' or 'Rectangle').**
    - The placeholder id will only have component name without the initial **VDXC\_ in designer-config, while in muse designer need to define the placehoder id with VDXC\_component\_name.**
    - **restrict** usage: if component has to be restricted to mobile phone only, this setting should be provided:
    - restrict: {
      * execution: "mobile",
      * size: "phone",
      * },

## Advanced use:

Get scrollbar component instance:  
var **scrollbarInstance** = Expo.designerAPI.scrollbar.get("placeholderId");

## Component methods:

* + **scrollbarInstance.update(config)** (here could be provided parameters to override existing config and init new scrollbar view)
    - **scrollbarInstance.reset()** (reset component to initial state)
    - **scrollbarInstance.updateCustomScroll()** (if custom scrollbar was used this method allows to redraw bars)
    - **scrollbarInstance.render**() - render a dom of the component with default config

## Additional global methods:

Expo.designerAPI.components.get("placeholderId"); // to get component by id  
Expo.designerAPI.components.reset(); // to reset all components

* Code Snippet
  + **Use this line of code on Tab to reset the Scrollbar**  
    Expo.designerAPI.scrollbar.get("scrollbar\_placeholder").reset();
* /\* Start Scrollbar Code from Here \*/
* {
* name: "scrollbar", // Do not change this value
* props: {
* preventRender: true,
* placeholderId: "scrollbar\_placeholder",
* contentId: "scrollbar\_content",
* direction: "vertical", /\* "vertical", "horizontal", "both" \*/
* customScroll: true,
* customScrollAlwaysShow: false,
* customScrollTheme: "blue" /\* "default", "dark", "light", "blue" \*/
* },
* events: {
* onScrolled: function(Expo) {
* Expo.designerAPI.firePixel("content\_1\_scrolled", [{"eventName":"intLive", "multi":true}]);
* },
* onScrollStart: function(Expo) {
* Expo.designerAPI.firePixel("content\_1\_scrollStart", [{"eventName":"intLive", "multi":true}]);
* },
* onScrollEnd: function(Expo) {
* Expo.designerAPI.firePixel("content\_1\_scrollEnd", [{"eventName":"intLive", "multi":true}]);
* }
* }
* }

/\* End Scrollbar Code \*/

## Font setup:

Refer to the font-embedding documentation: [Fonts.Common.VDX.Components.AdEx](file:////display/VDXComponents/Fonts.Common.VDX.Components.AdEx)

## Component Callbacks:

* + **onScrolled(Expo)** - general event (fires on each scroll event, neither wheel, neither click&drag, tap&drag)
  + **onScrollStart(Expo)** - service event (fires on custom scrollbar scroll start event, after mouse click or touch start and move)
  + **onScrollEnd(Expo)** - service event (fires on custom scrollbar scroll end event, after mouse release or touch end)

## Supported interaction Events and Detail

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **User Action** | **Event Name** | **Default Source**  **All in bound pixel with have a suffix TabTracker id which is defined by user** | **Default Sub Source** | **Parameter in insight pixel** | **Can i change source value in non muse BEFORE Advance Pixel Management Release?** | **Can i change source value in non muse AFTER Advance Pixel Management Release?**  **(if component is present in VDX STUDIO)** | **Can i change source value in Muse BEFORE Advance Pixel Management Release?** | **Can i change source value in Muse AFTER Advance Pixel Management Release?** | **Initiator(function/callback initiating the pixel)** |
| on scrolled | inbound | scrolled | {  componentName: "COMPONENT NAME"  tabName: TAB Label like **Offers**  tabSequence: TAB Number like tab{Number  } for example tab1, tab2 where Number is integer value  subSource: srcolled  } | event=intLive and fr:{source) | NO | NO | YES | YES | onScrolled |
| on start of scroll | inbound | scrollStart | {  componentName: "COMPONENT NAME"  tabName: TAB Label like **Offers**  tabSequence: TAB Number like tab{Number  } for example tab1, tab2 where Number is integer value  subSource: scrollStart  } | event=intLive and fr:{source) | NO | NO | YES | YES | onScrollStart |
| on end of scroll | inbound | scrollEnd | {  componentName: "COMPONENT NAME"  tabName: TAB Label like **Offers**  tabSequence: TAB Number like tab{Number  } for example tab1, tab2 where Number is integer value  subSource: scrollEnd  } | event=intLive and fr:{source) | NO | NO | YES | YES | onScrollEnd |

# ProductGallery.Common.VDX.Components.AdEx

**Template/Demo URLs:** [Click Here](https://wiki.exponential.com/display/VDXConnect/SingleFramework.VDXConnect.Templates.AdEx#tab-Components+Templates)

## Functionality:

Component allows to build and organize set of product cards as a product gallery.  
With optional bottom, side arrows and swipe navigations.  
This could be used with static and dynamic content.

## Muse setup:

Designer should drop a placeholder on a page, set size and position and add a unique title id. This placeholder will be used as a product gallery.

## User defined config:

|  |  |  |  |
| --- | --- | --- | --- |
| **Property** | **Type** | **Default** | **Description** |
| placeholderId | String | — | Placeholder title id |
| dataGeneral | Object { properties } | none | Common data properties to be rendered in galleryHtml, slideHtml and cardHtml without processing |
| dataProducts | Array [ Object ] | none | Per-product data properties to be rendered in cardHtml without processing |
| dataProvider {     rawData     jsonUrl     maxProducts     general     card } | Object     Object { properties }     Url-String     Number     Object { properties }     Object { properties } | none | Inventory for loading or handling token data to render product cards     Raw json-like data to be processed     Global or local path to json with raw data to be processed     If provided this will limit maximum products count     Each property might be provided as one of option:         string path, for ex. "products[{x}].imageURL" (where {x} is a product index)         object {path: path\_to\_value, post: function(val){}}, }, for ex. { path: "products[{x}].deal", post: function(val){return "<span" + (val.match("¢") ? " class='cent'" : "") + ">" + val + "</span>"} } (where {x} is a product index) |
| cssUrl | Url-String | none | Path to local css-file with product gallery styles |
| css | Css-String | none | Inline css to be applied to the gallery to stylize dynamic html content. |
| exceptionHtml | Html-String | "<h1>No data or error</h1>" | Html code to be shown in case of loading or parsing issues.     Add data-event="ClickCallbackId" attribute to track click events (where ClickCallbackId is unique event id). |
| galleryHtml | Html-String | none | **EXPERT** Html code to be rendered as gallery container. |
| slideHtml | Html-String | none | **EXPERT** Html code to be rendered as slide container. |
| cardHtml | Html-String | — | Html code to render card within slide.     Add data-event="ClickCallbackId" attribute to track click events (where ClickCallbackId is unique event id).     Add {{property}} in html code to substitute it with dynamic data. |
| ~~slideStyle~~ | Style-String | none | **EXPERT** Custom CSS styling for the slide. |
| cardsGrid | String | "1x1" | {colums}x{rows} |
| cardsGap | Style-Size-String | "10px" | "{size}" | "{horizontal} {vertical}" size in px/% of gaps between cards |
| cardSize | Style-Size-String | none | "auto" | "{size}" | "{width} {height}" size in px/% of container |
| cardsAlign | String | none | "{horizontal&vertical}" | "{horizontal} {vertical}" top | bottom | left | right | space-between | space-around | space-evenly | stretch | center |
| slidePadding | Style-Padding-String | none | "{size}" | "{horizontal} {vertical}" | other slide inner padding, so cards wouldn't stick to slide border |
| preloaderImage | Url-String | none | If provided then custom preloader would be shown while data is loading |
| dotNavigation | Boolean | true | Enable/disable dots gallery navigation. |
| arrowNavigation | Boolean | true | Enable/disable left/right arrows navigation. |
| swipeNavigation | Boolean | String | true | Enable/disable horizontal swipe navigation (true | false | "auto"). |
| transitionStyle | String | "chain" | chain | push | overlay | crossfade | fade |
| transitionDuration | Seconds-Number | 1 | Time (sec) — transition time value to complete transition from slide to slide. |
| loopSlides | Boolean | true | Enable/disable seamless slide change on border slide numbers. |
| autoPlay | Seconds-Number | 0 | Time (sec) — allows to automatically change slides if no interaction happened. |
| resumeAutoPlay | Seconds-Number | 0 | Time (sec) — allows to resume autoplay from the last interaction if "**autoPlay**" property is not 0. |
| dotsPosition | String | "bottom" | top | bottom |
| dotsOverflow | String | none | fade | scale — [**Details**](http://prntscr.com/tl5saq) |
| dotsStyle | Style-String | none | **EXPERT** Custom CSS styling for the dots container (for ex. for the better positioning). |
| dotsGap | Style-Size-String | "10px" | CSS width value to set the gap size between dots. |
| dotStyle {     normal     hover     active } | Object {Style-String} | none | **EXPERT** Custom CSS styling for the specific dot states (each is optional).     normal state — (for ex.: "border:0px solid #FFF; background-color:rgba(255, 255, 255, 0.1); ")     mouseover state — (for ex.: "border-size:2px; background-color:rgba(255, 255, 255, 0.25); ")     active state — (for ex.: "border-size:1px; background-color:#9999FF; ") |
| dotOutlineNormal | Style-Border-String | none | CSS border value. If provided - dot normal state would have this style; For ex. "1px solid #FFFFFF". For details see [MDN](https://developer.mozilla.org/en-US/docs/Web/CSS/border). |
| dotOutlineActive | Style-Border-String | none | CSS border value. If provided - dot active state would have this style; For ex. "2px solid #FFFFFF". For details see [MDN](https://developer.mozilla.org/en-US/docs/Web/CSS/border). |
| dotSize | Style-Size-String | none | CSS size like value. For ex. "10px" - width and height at same time, "10px 20px" - width and height separately. |
| dotShape | String | "circle" | rectangle | circle |
| dotThumbFit | String | "contain" | cover | contain | unset | auto | initial For the details see [MDN](https://developer.mozilla.org/en-US/docs/Web/CSS/background-size) |
| arrowsPosition | String | "center" | center | top | bottom |
| arrowsMargin | Style-Margin-String | none | CSS margin like value. For ex. "10px", "0px 10px", "0px 0px 100px", "0px 30px 0px 0px"; any valid css-margin value for arrows container. |
| arrowStyle {     normal     hover     disabled } | Object {Style-String} | none | **EXPERT** Custom CSS styling for the specific arrow states (each is optional).     normal state — (for ex.: "opacity:0.8 ")     mouseover state — (for ex.: "opacity:1 ")     disabled state (first and last slide in case of loopSlides: false) — (for ex.: "opacity:0.1 ") |
| arrowSize | Style-Size-String | "44px 80px" | CSS size like value. For ex. "50px" - width and height at same time, "30px 50px" - width and height separately |
| arrowColor | Style-Color-String | "#FFFFFF" | CSS color like value. For ex. "#FFFFFF" |
| arrowBackgroundColor | Style-Color-String | none | CSS color like value. For ex. "rgba(255, 255, 255, 0.25)" |
| arrowImageLeft | Url-String | none | URL path to left (previous) arrow image. If provided, custom image for the left arrow will be used; If not provided, but right (next) image is provided - mirrored right (next) image would be used. |
| arrowImageRight | Url-String | none | URL path to right (next) arrow image. If provided, custom image for the right (next) arrow will be used; If not provided, but left (previous) image is provided - mirrored left (previous) image would be used. |
| arrowImageFit | String | "intial" | cover | contain | unset | auto | initial For the details see [MDN](https://developer.mozilla.org/en-US/docs/Web/CSS/background-size) |
| arrowsCustomize {     container     left     right } | Object {Style-String} | none | **EXPERT** Custom CSS styling for the specific arrow elements (each is optional).     container style (to define position and size of the arrows container element) — (for ex.: "left:10px; top:10px; width:200px; height:100px;")     left/previous style (to define position and size of the left/previous arrow element) — (for ex.: "width:80px; height:100%;")     right/next style (to define position and size of the right/next arrow element) — (for ex.: "width:80px; height:100%;") |
| controls { **\*new in 3.10.1**     enabled     position     style     color } | Object     Boolean     String     Style-String     Style-Color | false   "br"   ""   "#FFF" | Playback controls (mute & play/pause, each is optional)     true | false (true by default)     tl | tr | bl | br (which means top-left | top-right | bottom-left | bottom-right relatively)     additional styles (to fine-tune position, or add some filters)     controls color (#FFFFFF by default) |
| progress { **\*new in 3.10.1**     enabled     timeColor     fullColor } | Object     Boolean     Style-Color-String     Style-Color-String | false   ""   "" | Progress bar settings (each is optional)     true | false (true by default)     Playback progress bar coloring, for ex.: "#0099FF"     Background layer coloring, for ex.: "rgba(255, 0, 0, 0.5)" |
| theme { **\*new in 3.10.1**     name     size     timeSize     timeWidth     bgColor     playPauseColor     style } | Object     String     Number     Number     Number     Style-Color-String     Style-Color-String     Style-String | "bar"   ""   ""   ""   ""   ""   "" | Progress theme customization     Default theme selection should be **circular.**    **name: default | circular** (default theme means line bar at the bottom, circular theme means progress bar in circular shape)     **size**: Set the size of background circle.     **timeSize**: Set the size of circular progress bar.     **timeWidth**: Set the thickness of the circular progress bar.     **bgColor**: Control playpause color in circular theme apart of mute/unmute button     **playPauseColor**: Control playpause color in circular theme apart of mute/unmute button     **style**: Positioning styles like top | bottom | left | right | transform applicable. |
| preventRender | Boolean | false | Prevent automatic rendering at the ad-unit initialization If value is false — the "render" function should be called anytime later |
| restrict {     execution     size } | Object {String} | none | If provided, component would be rendered or not based on inputs execution — "inframe", "instream", "display", "mobile"; or multiple at once "instream, display" and so on size — "phone", "tablet"; or multiple at once "phone, tablet" and so on |

## Component Callbacks:

* **onSlideChanged(Expo, event)** (available event parameters: slideFrom, slideTo, auto; fires on slide change event)
* **onDotClick(Expo, event)** (available event parameters: slideFrom, slideTo; fires on bottom navigation click event)
* **onLeftArrowClick(Expo)** (fires on left arrow click event)
* **onRightArrowClick(Expo)** (fires on right arrow click event)
* **onDataError(Expo)** (fires if data provider not able to load data, or loads non valid data)
* **userDefinedCallbacks(Expo, event)** (available event parameters: index, product; fires on click on template-defined elements)

## Font setup:

Refer to the font-embedding documentation: [Fonts.Common.VDX.Components.AdEx](file:////display/VDXComponents/Fonts.Common.VDX.Components.AdEx)

## Limitations:

1. **Dots or thumbs** navigation is only supported in Engage it is not supported for VDX connect.
2. Thumbs and dots navigation in Engage adunits can be used keeping the thumbs width less or equal to the container size.

## Notes:

1. **The placeholder should be a container ('State button' or 'Rectangle').**
2. The placeholder id will only have component name without the initial **VDXC\_ in designer-config, while in muse designer need to define the placeholder id with VDXC\_component name**
3. Keep **transition time** under 1 second for smoothest animation look.
4. **Swipe navigation** is preferable to use in **mobile units** only
5. **restrict** usage: if component has to be restricted to mobile phone only, this setting should be provided:
   * restrict: {
   * execution: "mobile",
   * size: "phone",
   * },

## Advanced use:

Get product gallery component instance:  
var **productGalleryInstance** = Expo.designerAPI.productGallery.get(placeholderId);

Component methods:

* **productGalleryInstance.update(config)** (here could be provided parameters to override base config and init new product gallery view)
* **productGalleryInstance.reset()** (reset component to initial state)
* **productGalleryInstance.showSlide(slideNo)** (show particular slide, numeration start from 1)
* **productGalleryInstance.play()** (start autoplay, if it's available)
* **productGalleryInstance.pause()** (pause autoplay, if it's available)
* **productGalleryInstance.getCurrentSlide()** (returns current slide number)
* **productGalleryInstance.getStructure()** (set of gallery root DOM elements, like slidesEl, slideArray, dotsEl, leftArrowEl, rightArrowEl)

## Additional global methods:

Expo.designerAPI.components.get("placeholderId"); // to get component by id  
Expo.designerAPI.components.reset(); // to reset all components

Code Snippet

* **Use this line of code on Tab to Initialize the ProductGallery**  
  Expo.designerAPI.components.get("productGallery\_placeholder1").render();
* **Use this line of code on Tab to reset the ProductGallery**  
  Expo.designerAPI.components.get("productGallery\_placeholder1").reset();
* **Use this line of code on Tab to Initialize the ProductGallery, or reset if it's already rendered**  
  Expo.designerAPI.components.get("productGallery\_placeholder1").render(true);

**Sample Code** Expand source

/\* Start ProductGallery Code from Here \*/

        {

name: "productGallery", // Do not change this value

props: {

placeholderId: "productGallery\_placeholder1",

dataGeneral: { // object{propertyX: "valueX", ...} // optional // general data to be rendered in galleryHtml and slidesHtml

},

dataProducts: [ // array[object{propertyX: "valueX", ...}, ...] // optional // array of products to render

],

dataProvider: { // if dataProducts is not provided, then dataProvider is used to fetch data dynamically

rawData: {}, // object | array // optioanl // raw data to be processed

jsonUrl: "assets/products.json", // url to fetch raw data, if rawData is not provided

card: { // object{propertyX: string | object{path: path\_to\_value, post: function(val){}}, ...} // template processor

thumb: "products[{x}].imageURL",

price: {

path: "products[{x}].deal",

post: function(val) {

return "<span" + (val.match("¢") ? " class='cent'" : "") + ">" + val + "</span>";

},

},

title: "products[{x}].title",

description: "products[{x}].description",

url: "products[{x}].buyOnlineLinkURL",

},

},

css: "", // inline css to be used with template and processor

exceptionHtml: "<div>Data error</div>", // valid html code

// galleryHtml: "<div></div>", // valid html code

// slideHtml: "<div></div>", // valid html code

cardHtml: "\

<div class='product-card' data-event='onCtaClick'>\

<div class='product-thumb'><img src='{{thumb}}'/></div>\

<div class='product-price' data-overflow='fit-w;6;clip'>{{price}}</div>\

<div class='walmart-title' data-overflow='fit;6;ellipsis'>{{title}}<br>{{description}}</div>\

<div class='walmart-cta'>Shop the weekly ad</div>\

</div>\

", // valid html code

cardsGrid: "1x1", // {colums}x{rows}

cardsGap: "10px", // "{size}" | "{horizontal} {vertical}" // size in px of gaps between cards

cardSize: "100% 100%", // "auto" | "{size}" | "{width} {height}" // size in px or % of container

cardsAlign: "center", // "{horizontal&vertical}" | "{horizontal} {vertical}" // top | bottom | left | right | space-between | space-around | space-evenly | stretch | center

dotNavigation: true, // true | false // enable/disable dots/thumbs navigation

arrowNavigation: true, // true | false // enable/disable side arrows navigation

swipeNavigation: "auto", // "auto" | true | false // enable/disable mouse/touch swipe navigation

transitionStyle: "chain", // chain | push | overlay | crossfade | fade

transitionDuration: 1, // transition time (sec)

loopSlides: true,

autoPlay: 3, // time (sec) to start autoplay

resumeAutoPlay: 3, // time (sec) to resume autoplay after stop

dotsPosition: "bottom", // top | bottom

dotsOverflow: "fade", // fade | scale

dotsStyle: "", // css values

dotStyle: {

normal: "", // css values

hover: "", // css values

active: "", // css values

},

dotOutlineNormal: "1px solid #FFFFFF", // css border value

dotOutlineActive: "2px solid #FFFFFF", // css border value

dotSize: "10px", // {size} | {width, height}

dotShape: "circle", // rectangle | circle

dotThumbFit: "contain", // cover | contain | auto | unset | initial

arrowsPosition: "center", // center | top | bottom

arrowsMargin: "0px", // css margin value

arrowStyle: {

normal: "", // css values

hover: "", // css values

disabled: "", // css values

},

arrowSize: "44px 80px", // {size} | {width, height}

arrowColor: "#FFFFFF", // css color

arrowBackgroundColor: "rgba(0, 0, 0, 0.25)", // css color

arrowImageLeft: "", // url

arrowImageRight: "assets/gallery\_arrowRight.png", // url

arrowImageFit: "contain", // cover | contain | auto | unset | initial

arrowsCustomize: {

container: "", // css values

left: "", // css values

right: "", // css values

},

},

events: {

onDataError: function(Expo) {

},

onErrorClick: function(Expo) {

// callback function name is defined in exceptionHtml

// there might be any name, any amount of callbacks

},

onCtaClick: function(Expo, event) {

// callback function name is defined in cardHtml

// there might be any name, any amount of callbacks

// event.id holds product index (starting from 1)

// event.product is an object{..} and it holds product data

},

onSlideChange: function(Expo, event) {

},

onDotClick: function(Expo, event) {

},

onLeftArrowClick: function(Expo) {

},

onRightArrowClick: function(Expo) {

},

        },

  },

/\* End ProductGallery Code \*/

## Supported interaction Events and Detail

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **User Action** | **Event Name** | **Default Source**  **All in bound pixel with have a suffix TabTracker id which is defined by user** | **Default Sub Source** | **Parameter in insight pixel** | **Can i change source value in non muse BEFORE Advance Pixel Management Release?** | **Can i change source value in non muse AFTER Advance Pixel Management Release? (if component is present in VDX STUDIO)** | **Can i change source value in Muse BEFORE Advance Pixel Management Release?** | **Can i change source value in Muse AFTER Advance Pixel Management Release?** | **Initiator(function/callback initiating the pixel)** |
| click on left Arrow | inbound | leftArrow | {  componentName: "COMPONENT NAME"  tabName: TAB Label like **Offers**  tabSequence: TAB Number like tab{Number  } for example tab1, tab2 where Number is integer value  subSource: slideNumber like slide{Number} for example slide1, slide2 where Number is integer value  } | event=intLive and fr:{source) | NO | NO | YES | YES | onLeftArrowClick |
| click on right Arrow | inbound | rightArrow | {  componentName: "COMPONENT NAME"  tabName: TAB Label like **Offers**  tabSequence: TAB Number like tab{Number  } for example tab1, tab2 where Number is integer value  subSource: slideNumber like slide{Number} for example slide1, slide2 where Number is integer value  } | event=intLive and fr:{source) | NO | NO | YES | YES | onRightArrowClick |
| on slide change | inbound | slideChanged | {  componentName: "COMPONENT NAME"  tabName: TAB Label like **Offers**  tabSequence: TAB Number like tab{Number  } for example tab1, tab2 where Number is integer value  subSource: slideNumber like slide{Number} for example slide1, slide2 where Number is integer value  } | event=intLive and fr:{source) | NO | NO | YES | YES | onSlideChanged |
| on bottom navigation click | inbound | dotClicked | {  componentName: "COMPONENT NAME"  tabName: TAB Label like **Offers**  tabSequence: TAB Number like tab{Number  } for example tab1, tab2 where Number is integer value  subSource: slideNumber like slide{Number} for example slide1, slide2 where Number is integer value  } | event=intLive and fr:{source) | NO | NO | YES | YES | onDotClick |
| if data provider not able to load data, or loads non valid data | inbound | dataError | {  componentName: "COMPONENT NAME"  tabName: TAB Label like **Offers**  tabSequence: TAB Number like tab{Number  } for example tab1, tab2 where Number is integer value  subSource: slideNumber like slide{Number} for example slide1, slide2 where Number is integer value  } | event=intLive and fr:{source) | NO | NO | YES | YES | onDataError |

# Playlist.Common.VDX.Components.AdEx

|  |  |
| --- | --- |
| **Arrow Playlist Code snippet** | **usage** |
| arrowPlaylist.showHideCarousel("show"); | to show carousel/arrow playlist |
| arrowPlaylist.showHideCarousel("hide"); | to hide carousel/arrow playlist |
| videoCarouselEndEvent: function(Expo) { Expo.trigger(tabName, false); } | playlist end event.  define your tab name (highlighted via red color)  Here Expo gives the access to component framework api's |
| totalVideosCarousel: 3, | to define number of videos |
| autoPlay: true, | to play video in continuation.  keep it true always. |
| arrowOpacity: .3, | to show inactive arrows.  do not change this. |
| hideArrowTime: 3000, | to get inactive timer.  do not change this. |
| skipResetOnPlaylistEnd: false, | By default when playlist ends current video resets to "video1" and pauses.  If this property is set to true, then playlist would freeze on last frame of the last video. |
| videoCarouselNext: function(Expo) {  } | to track playlist "next arrow" click/tap  Here Expo gives the access to component framework api's |
| videoCarouselPrev: function(Expo) {  } | to track playlist "back arrow" click/tap  Here Expo gives the access to component framework api's |
| Expo.designerAPI.showReplayIcon(true); | to show replay icon (in case of single tab means video tab only) |

## Supported interaction Events and Detail

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **User Action** | **Event Name** | **Default Source**  **All in bound pixel with have a suffix TabTracker id which is defined by user** | **Default Sub Source** | **Parameter in insight pixel** | **Can i change source value in non muse BEFORE Advance Pixel Management Release?** | **Can i change source value in non muse AFTER Advance Pixel Management Release?** | **Can i change source value in Muse BEFORE Advance Pixel Management Release?** | **Can i change source value in Muse AFTER Advance Pixel Management Release?** | **Initiator(function/callback initiating the pixel)** |
| click on Left Arrow | inbound | leftArrow | {   componentName: "COMPONENT NAME",   tabName: "HOST TAB LABEL", // **Video**   tabSequence: "tab1", // **tab1**   subSource: "slide{ID}", // **slide1, slide2, ...** } | event=intLive and fr:{source) | NO | NO | YES | YES | videoCarouselPrev |
| click on Right Arrow | inbound | rightArrow | {   componentName: "COMPONENT NAME",   tabName: "HOST TAB LABEL", // **Video**   tabSequence: "tab1", // **tab1**   subSource: "slide{ID}", // **slide1, slide2, ...** } | event=intLive and fr:{source) | NO | NO | YES | YES | videoCarouselNext |

|  |  |
| --- | --- |
| **Thumb Playlist Code snippet** | **usage** |
| totalVideos: 3, | to define number of videos |
| onVideoStart: function(Expo, obj) {  } | triggers on playlist video start event  Here Expo gives the access to component framework api's  obj.videoID gives the id of the video started |
| onVideoEnd: function(Expo, obj) {  } | triggers on playlist video end event  Here Expo gives the access to component framework api's  obj.videoID gives the id of the video ended |
| onPlaylistEnd: function(Expo, obj) {  } | triggers on playlist end event and can be used to perform any action on playlist end. |

## Supported interaction Events and Detail

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **User Action** | **Event Name** | **Default Source** | **Default Sub Source** | **Parameter in insight pixel** | **Can i change source value in non muse BEFORE Advance Pixel Management Release?** | **Can i change source value in non muse AFTER Advance Pixel Management Release?** | **Can i change source value in Muse BEFORE Advance Pixel Management Release?** | **Can i change source value in Muse AFTER Advance Pixel Management Release?** | **Initiator(function/callback initiating the pixel)** |
|  |  |  |  |  |  |  |  |  |  |

# Map.Common.VDX.Components.AdEx

**Template/Demo URLs:** [Click Here](https://wiki.exponential.com/display/VDXConnect/SingleFramework.VDXConnect.Templates.AdEx)

## Updated for components-release - [HTML5 Components 3.1-1 Release](https://wiki.exponential.com/display/VDXComponents/HTML5+Components+3.1-1+Release)

## Updates:

* + Added animation functionality for the component.

## Functionality:

Component showing client's offices with address, contact phone, and CTA at popup block. Allow animation automation for show/hide and switch between stores markers on the map.

## Muse setup:

Make the container with "VDXC\_mapComponent\_map" as its hyperlink name, for placing map, it's height and width should be as per the requirement.

Make the container with "EXPOEVENT\_mapDefault" as its hyperlink name, for placing default/noData image. the placed image should be of the same size.

Designer need to update data for local store information at **json/data.json** file.

Configuration properties details

## ****Auto-mode config updates:****

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Property** | | **Description** | **Values** | **Details** | **Available in**  **Non-Muse Flow** | **Screenshot of property** | | **Creative Description** |
| maps.mapA.common.theme | | popup theme | string [ default | bubble ] |  | Bubble theme set by default  Default theme available without auto play |  | | We have two themes available with this release i.e., default and bubble.  **Default:** The same old theme without auto play and with [old popup](https://prnt.sc/JDaDUbFVwx0k) design.  **Bubble:** New theme with auto play and with [new popup](https://prnt.sc/gsgETaZsq29z) design. Location pin sequence is added in the data.json. |
| maps.mapA.common.dropSound | | sound to be played when pin is dropped to map | string-url | optional | Yes |  | | Drop sound will play when pins drop on the map initially. |
| maps.mapA.common.popSound | | sound to be played when pin is animated during auto-mode | string-url | optional | Yes |  | | Pop sound plays when the next location pin appears through auto play. |
| autoPlay | | time to start auto-play after initialization | number-time-ms |  | Yes | A black rectangular object with a black rectangle  AI-generated content may be incorrect. | | Time to start auto-play after initialization.  To achieve the “***Default***” theme we need to set the property value as 0. |
| autoPlayPauseOnHover | | indicates if auto-play has to be paused on hover or tap on popup | boolean | default: true | No |  | |  |
| autoPlayHoverResume | | time to resume auto-play after hover pausing | number-time-ms |  | Yes |  | | This is the time after which the auto play will resume if the user mouse out from pin/popup. |
| autoPlayResume | | Resumes the auto play if user interacts with map | number-time-ms |  | Yes |  | | In case we don’t want to resume the auto play, we can set the property as 0. |
| autoPlayScenario | | script for the actors (marker, popup, cut)  normally it should be (with some number variations):  autoPlayScenario: [     [0, 'marker', 'pop'],     [500, 'popup', 'show'],     [4000, 'popup', 'hide'], ] | list of three-item arrays [startTimeMs, actorName, actionName] |  | Yes | A screenshot of a game  AI-generated content may be incorrect. | | With Auto Play Scenario properties, we can set the auto play animation behavior with 3 step sequence.  Example:  [0, 'marker', 'pop'] – Step 1  [500, 'popup', 'show'] – Step 2  [4000, 'popup', 'hide'] – Step 3  In these 3 steps we have 3 items: Time (StartTime), Target (Actor Name) and Action  StartTime – This will be key parameter which defines when the Actor will be in action.  Actor Name – Which object going to be in Action.  Action – What Actor to be done.  Let’s understand how animation will work as per above mentioned Auto Play Scenario.  [0, 'marker', 'pop'] – As soon as the auto play starts, at 0 milliseconds, the marker(pin) will pop.  [500, 'popup', 'show'] – after 0.5 seconds the popup will be visible.  [4000, 'popup', 'hide'] – After 4 seconds, it will hide the popup and move to next marker and continues the same Auto Play Scenario again…  We recommend only to change the Time property as per the Autoplay properties you want. Rest are the fixed parameters. |
| **Code snippet** | **usage** | | | | | |
| popupWidth | Will define the max width of the popup. if the content is less than the specified width than the popup adjust according to content and if content is more than the specified width than line break take place. No horizontal scrollbar. | | | | | |
| popupHeight | Will define the max height of the popup. if the content is less than the specified height than the popup adjust according to content and if content is more than the specified height than vertical scrollbar appears. | | | | | |
| popUpCTAClick: function(Expo, event) { } | Will perform the function defined on click of the CTA deined inside the popup. | | | | | |
| "icon": "imagePath/imageName" | Will define the image used as a the pin. | | | | | |
| "pinEvents": { rollover: function(Expo, event) { }, rollout: function(Expo, event) { }, click: function(Expo, event) { } } | Roll-over function will work on roll-over of the pin.  Roll-out function will work on roll-out of the pin.  click function will work on click of the pin. | | | | | |
| event.layer.closePopup(); | Will close the popup | | | | | |
| event.layer.openPopup(); | Will open the popup | | | | | |
| defaultURL | if cta is not present in the dynamic data than this URL will open on popup CTA button | | | | | |
| pixelID | this pixel id will attached to the CTA of the CTA button in popup | | | | | |

**Default Properties can be overridden by using them in designer-config**

|  |  |
| --- | --- |
| **Code Snippet For default properties not exposed to designer-config** | **Usage** |
| zoomLevel | Will change the zoom level of the map. |
| maxZoom | it is used to set the maximum level of zoom user can do (will get supported in release 2.2) |
| minZoom | it is used to set the minimum level of zoom user can do (will get supported in release 2.2) |
| zoomControl | Will show/hide the zoom increase/decrease button |
| attributionControl | Will show/hide the mapbox copyright. |
| showCloseBtn | Will show/hide the close button on the popup. |
| openPopByDefault | Will show popup for the first marker initially (this property is ignored in auto-mode with defined popup action in scenario). |
| iconSize | Will manage the pin size (width x height) |
| center | Will make the one of the pin position to center if true. |
| msgString | html properties of the message in the popup can be managed through it. |
| marginFromTop | it will provide the margin to the popup window from top of the map container. |
| twoFingersEnabled | if the value is true then user can scroll the adunit from the map section with one finger and if he uses the two fingers he can pan and zoom the adunit. The default value will be true. The feature is enabled for mobile adunits |
| iconAnchor | icon anchor determines the position of the tip pointing the target location. |
| popupAnchor | Helps in positioning the popup from the pin. |
| tooltip | this property will show/hide the tooltip of the popup. default value is "visible" "visible,hidden" |

## Limitations:

* Location Map is not supported on Primary Tab.

Knowns**:**

* Tracking on popup click is not implemented.

## Notes:

* The placeholder should be a container('button')

Code Snippet

* **Initialise the Location Map on tab click**
  + Place the below line of code in the init function in the designer-config.js

Expo.setTempValue("activeMapTab", false);

* + Place the below line of code in the tab function in the designer-config.js on click of which map component needs to be initialize.

if(!Expo.getTempValue("activeMapTab")){

Expo.setTempValue("activeMapTab", true);

if(Expo.designerAPI.getDynamicData().rawData || Expo.designerAPI.getDynamicData().rawData === null) {

var mapDefault = document.querySelector('div[data-expo-event="EXPOEVENT\_mapDefault"]');

mapDefault.style.zIndex = -1;

};

map.generateMap({mapID: "mapA", dynamicData: Expo.designerAPI.getDynamicData().rawData, localData: "json/data.json", defaultImageURL: "images/map\_default.jpg", noDataImage: "images/no\_data.jpg"});

};

* + Place the below line of code in the designer-config.js for adding the CTA on the default/noData image.

{

name: "mapDefault",

type: 'click',

callback: function(Expo, event) {

Expo.designerAPI.firePixel("mapDefault",[{"eventName":"clickLive"}]);

Expo.designerAPI.openUrl("place the default/noData URL here", "mapDefault");

Expo.designerAPI.pause();

}

}

* **Additional code to  Initialise the Location Map on second tab and IntLive pixel is not required.**
  + Place the below line of code in the init function in the designer-config.js

Expo.setTempValue("isComingFromPlayList", false);

* + Place the below line of code in the videoCarouselEndEvent function in the designer-config.js
  + Expo.setTempValue("isComingFromPlayList", true);

Expo.trigger('tabAdFormats');

* + Place the below line of code in the tab function in the designer-config.js on click of which map component needs to be initialise.
  + if(!Expo.getTempValue("isComingFromPlayList")){
  + Expo.designerAPI.firePixel("tabMap",[{"eventName":"intLive", "multi":true}]);
  + }else{
  + Expo.setTempValue("isComingFromPlayList", false);

};

/\* Start LocationMap Code from Here \*/

{

name: "map", // Do not change this value

props: {

maps: {

mapA: {

mapID: "mapDefault",

pinType: ["stores"],

pinValue: ["pin0"],

twoFingersEnabled: true,

common: {

"zoomLevel": 12,

"maxZoom": 19, (will get supported in release 2.2)

"minZoom": 2, (will get supported in release 2.2)

"zoomControl": false,

"attributionControl": false,

"theme": "bubble",

"dropSound": "assets/dropSound.mp3",

"popSound": "assets/dropSound.mp3",

},

pin0: [{

"pop": {

"popupWidth": 250,

"popupHeight": 180,

"marginFromTop": .004,

"iconAnchor": [12, 35],

"minPopupWidth": 100,

"popupAnchor": [0, -35], // ideal value [0, -height of iconSize]

"tooltip": "visible", //visible,hidden

"showCloseBtn": true,

"openPopByDefault": true,

"msgString": '<div id="contentPopup" align="center" style="width:' + '100%' + '; height:' + '100%' + ';">' + 'tokenContentMap.label' + " </br> " + 'tokenContentMap.address' + ", " + 'tokenContentMap.city' + " " + ",</br>" + " " + 'tokenContentMap.state' + " " + 'tokenContentMap.zip' + " " + "</br>" + " " + 'tokenContentMap.phone' + " " + '</br><img id=popUpCount style="margin-top: 5px; cursor: pointer;"' + "s" + 'rc="images/popupCtaBtn.png" alt="" height="21" width="70" name="pin0" onclick="window.parent.map.popUpCTA(this)"></div>',

"popupCTA": {

"defaultURL": "place the default URL here",

"pixelID": "popUpCTAClick",

popUpCTAClick: function (Expo, event) {

Expo.designerAPI.firePixel("popUpCTAClick",[{"eventName":"clickLive"}]);

Expo.designerAPI.pause();

}

}

},

"iconSize": "24x35",

"icon": "images/mapPin1.png",

"center": true,

"pinEvents": {

rollover: function(Expo, mapObject, event) {},

rollout: function(Expo, mapObject, event) {},

click: function(Expo, mapObject, event) {}

}

}]

}

},

               autoPlay: 2000,

               autoPlayPauseOnHover: true,

               autoPlayHoverResume: 2000,

               autoPlayResume: 5000,

autoPlayScenario: [

[0, 'marker', 'pop'],

[500, 'popup', 'show'],

[4000, 'popup', 'hide'],

],

          }

}

/\* End LocationMap Code \*/

## Component methods:

* **map.render(true)** render/reset component view
* **map.update(config)** here could be provided parameters to override existing config
* **map.reset()** reset component to initial state
* **map.generateMap(data)** generate map after (map instance has to be initialized and rendered already)
* **map.resume**() - resumes component's popup switching autoplay
* **map.pause**() - pauses component's popup switching autoplay
* **map.stop**() - stops component's popup switching autoplay

## Supported interaction Events and Detail

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **User Action** | **Event Name** | **Default Source**  **All in bound pixel with have a suffix TabTracker id which is defined by user** | **Default Sub Source** | **Parameter in insight pixel** | **Can i change source value in non muse BEFORE Advance Pixel Management Release?** | **Can i change source value in non muse AFTER Advance Pixel Management Release?** | **Can i change source value in Muse BEFORE Advance Pixel Management Release?** | **Can i change source value in Muse AFTER Advance Pixel Management Release?** | **Initiator(function/callback initiating the pixel)** |
| click on default image | outbound | tracker id (as defined by user) | {   componentName: "COMPONENT NAME",   tabName: "HOST TAB LABEL", // **Offers**   tabSequence: "tab{HOST\_TAB\_ID}", // **tab1, tab2, ...**   subSource: "map", } | event=clickLive and fr:{source) | ? | ? | YES | YES | onDefaultClick |
| click on popup CTA | outbound | tracker id (as defined by user) | {   componentName: "COMPONENT NAME",   tabName: "HOST TAB LABEL", // **Offers**   tabSequence: "tab{HOST\_TAB\_ID}", // **tab1, tab2, ...**   subSource: "{pinGroupID}\_{pinID}", // **pin0\_1, pin0\_2, ...** } | event=clickLive and fr:{source) | ? | ? | YES | YES | onPopupCtaClick |

# InteractiveView360.Common.VDX.Components.AdEx

**Template/Demo URLs:** [ClickHere](https://wiki.exponential.com/display/RMDOC/InteractiveView360.Templates.AdEx)

Config Details

# Threesixty View With Hotspots Component

## Functionality:

Component allows to show a 360 view of an object ( a car ) by draging or clicking left/right arrows and hotspots over it. User can see popups by clicking to hotspots. There can be any amount of components in a unit.

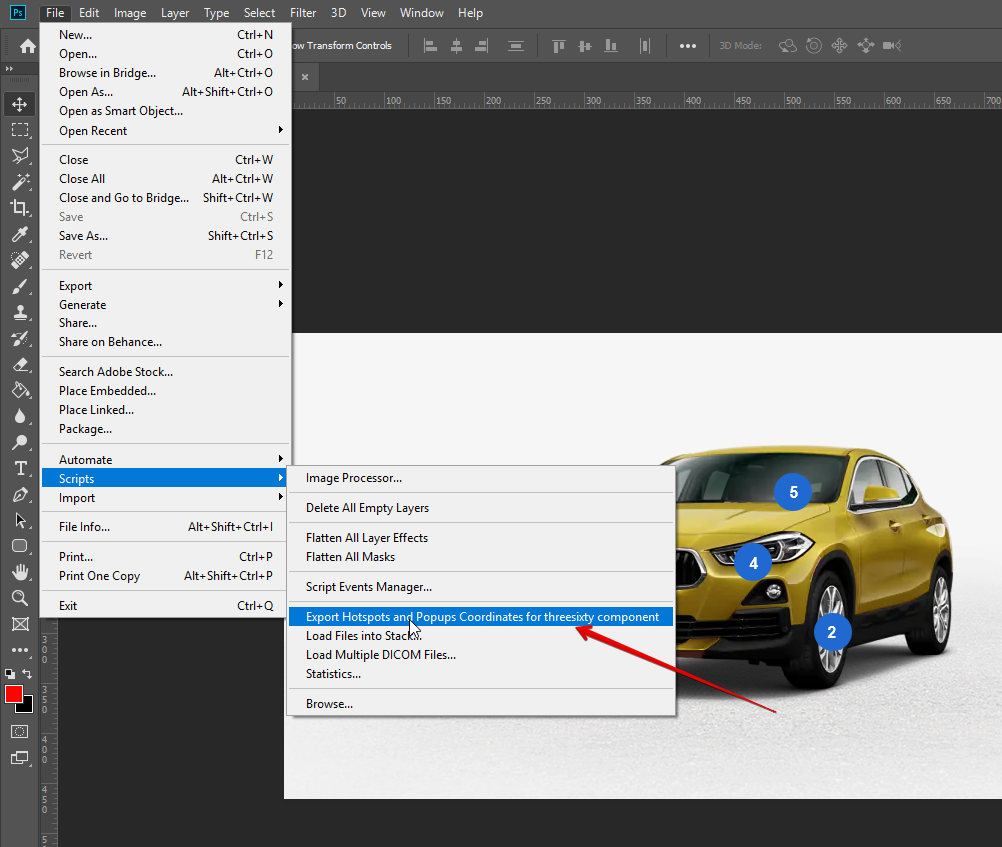
## Muse setup:

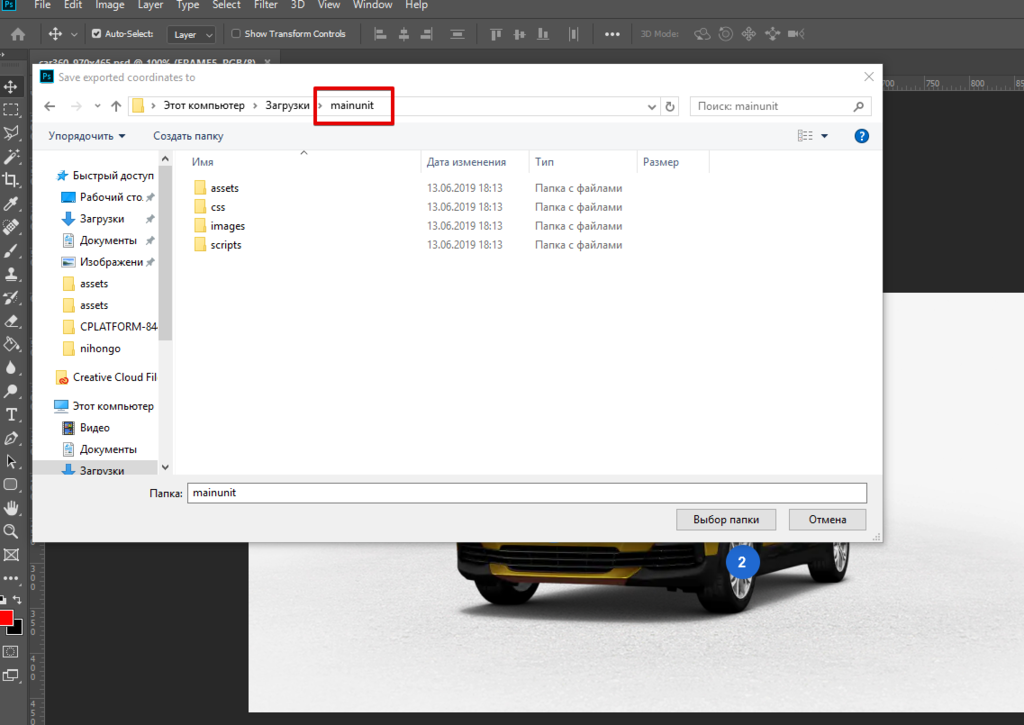
Designer should drop a placeholder on a page, set size and position and add a unique title id. This placeholder will be used as a gallery. Also the designer should create hotspots and popups in the unit.

## User defined parameters:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Property** | **Type** | **Default** | **Screenshot of property** | **Description** |
| placeholderId | string |  |  | placeholder title |
| imagesPath | string | "images/" |  | path to frame images, including left/right arrows |
| frames | string[] | string |  |  | array of images names, or filename template, plus **totalSlides** to be defined; for ex. "image{xx}.jpg", where {xx} is a slide number. |
| totalSlides | number |  |  | total slides to be shown, if **frames** is a string |
| speed | number | 15 |  | speed of item rotation |
| width | number | auto-calculated |  | width of the placeholder has been choice during hotspots setting **not necessary**-  needed only for responsive component |
| height | number | auto-calculated |  | height of the placeholder has been choice during hotspots setting **not necessary**-  needed only for responsive component |
| navigationButtons[] {     id     direction } | array {   string   "left" | "right" } |  |  | navigation buttons |
| hotSpots {     id     frames     popUpId     popUpCloseId     popUpFrames } | array {   string   array<{x,y}|"hidden">   string   string   array<{x,y}|"hidden"> } |  |  | list of hotspot and related parameters     **id** - hotspot placeholder id     **frames** - array of hotspot coordinates in each frame , if it is 'hidden' doesn't shown     **popUpId** - popup placeholder id     **popUpCloseId** - popup close button id     **popUpFrames** - array of popup coordinates in each frame , if it is 'hidden' doesn't shows |
| popupAppearDuration | number | 500 |  | duration of popup appearing animation |
| popupAppearEasing | string | "easeOutCubic" |  | easing of popup appearing animation, linear | easeInQuad | easeOutQuad | easeInOutQuad | easeInCubic | easeOutCubic | easeInOutCubic | easeInQuart | easeOutQuart | easeInOutQuart | easeInQuint | easeOutQuint | easeInOutQuint | easeInSine | easeOutSine | easeInOutSine | easeInExpo | easeOutExpo | easeInOutExpo | easeInCirc | easeOutCirc | easeInOutCirc | easeInBack | easeOutBack | easeInOutBack | easeInElastic | easeOutElastic | easeInOutElastic |
| popupCloseDuration | number | 500 |  | duration of popup closing animation |
| popupCloseEasing | string | "easeInCubic" |  | easing of popup closing animation, linear | easeInQuad | easeOutQuad | easeInOutQuad | easeInCubic | easeOutCubic | easeInOutCubic | easeInQuart | easeOutQuart | easeInOutQuart | easeInQuint | easeOutQuint | easeInOutQuint | easeInSine | easeOutSine | easeInOutSine | easeInExpo | easeOutExpo | easeInOutExpo | easeInCirc | easeOutCirc | easeInOutCirc | easeInBack | easeOutBack | easeInOutBack | easeInElastic | easeOutElastic | easeInOutElastic |
| clickBgToHidePopups | boolean | true |  | click on background to hide opened pop-up |
| bgCursor | string-css-pointer | "pointer" |  | defines background pointer style |
| arrowCursor | string-css-pointer | "pointer" |  | defines navigation arrows pointer style |
| popupsBackdropColor | string-css-color |  |  | popup background color |
| popupOpenByRollover | boolean | false |  | open corresponding pop-ups by hovering hotspots |
| bgImageFill **\*new in 3.10.1** | string-css-bg-size | "contain" |  | slides image fitting |
| popupTheme { **\*new in 3.10.1**     name     background     rounding     tipSize     tipCurved   } | object {     "default" | "bubble"     string-css-background     number     number     boolean   } | "default" |  | re-defines popup theme (for bubble theme)   **name**: "bubble",   **background**: "#FFF",   **rounding**: 10, // [10] | [10, 20] | [10, 20, 30, 40] | "10" | "10 20" | "10 20 30 40"   **tipSize**: 40,   **tipCurved**: false, |
| autoPlayTheme { **\*new in 3.10.1**     name     position     size     timeSize     timeWidth     timeColor     bgColor   } | object {     "none" | "circular"     "[tbc][lrc]" | string-inline-style     number     number     number     string-css-color     string-css-color   } |  |  | Progress theme customization     Default theme selection should be **circular.**    **name:** circular theme means progress bar in circular shape     **position**: defines circular placement / position     **size**: sets the size of background circle     **timeSize**: set the size of circular progress bar     **timeWidth**: Set the thickness of the circular progress bar     **timeColor**: Control play/pause color     **bgColor**: Control play/pause color |
| dropSound **\*new in 3.10.1** | string-url |  |  | Drop sound will play when pins drop on the screen initially. |
| popSound **\*new in 3.10.1** | string-url |  |  | Pop sound plays when the next location pin appears through auto play. |
| autoPlay **\*new in 3.10.1** | number-time-ms |  | A black rectangular object with a black rectangle  AI-generated content may be incorrect. | Time to start auto-play after initialization. 0 = no autoplay  To achieve the “***Default***” theme we need to set the property value as 0. |
| autoPlayPauseOnHover **\*new in 3.10.1** | boolean |  |  | indicates if auto-play has to be paused on hover or tap on popup |
| autoPlayHoverResume **\*new in 3.10.1** | number-time-ms |  |  | This is the time after which the auto play will resume if the user mouse out from pin/popup. |
| autoPlayResume **\*new in 3.10.1** | number-time-ms |  |  | In case we don’t want to resume the auto play, we can set the property as 0. |
| autoPlayScenario **\*new in 3.10.1** | list of three-item arrays [startTimeMs, actorName, actionName] |  | A screenshot of a game  AI-generated content may be incorrect. | With Auto Play Scenario properties, we can set the auto play animation behavior with 3 step sequence.  Example:  [0, 'marker', 'pop'] – Step 1  [500, 'popup', 'show'] – Step 2  [4000, 'popup', 'hide'] – Step 3  In these 3 steps we have 3 items: Time (StartTime), Target (Actor Name) and Action  StartTime – This will be key parameter which defines when the Actor will be in action.  Actor Name – Which object going to be in Action.  Action – What Actor to be done.  Let’s understand how animation will work as per above mentioned Auto Play Scenario.  [0, 'marker', 'pop'] – As soon as the auto play starts, at 0 milliseconds, the marker(pin) will pop.  [500, 'popup', 'show'] – after 0.5 seconds the popup will be visible.  [4000, 'popup', 'hide'] – After 4 seconds, it will hide the popup and move to next marker and continues the same Auto Play Scenario again…  We recommend only to change the Time property as per the Autoplay properties you want. Rest are the fixed parameters. |
| preloaderColor | string-css-color | "rgb(0, 174, 239)" |  | color of preloader line / preloader background, if image is provided |
| preloaderContainerHeight | string-css-length | "3px" |  | height of the preloader possible values: XXpx | XX% |
| preloaderImage | string-url |  |  | image like jpg|gif|svg rendered as background of preloader |
| preloaderImagePosition | string-css-bg-position | "center" |  | position of the preload image inside the preload container, possible values: top | bottom | center | left | right |
| preloaderImageSize | string-css-bg-size | "auto" |  | "background-size" for image inside the preloader container, possible values: auto | contain | cover |
| preventRender | boolean | false |  | Prevent automatic rendering at the ad-unit initialization If value is false — the "render" function should be called anytime later |
| restrict {     execution     size } | object {String} |  |  | If provided, component would be rendered or not based on inputs execution — "inframe", "instream", "display", "mobile"; or multiple at once "instream, display" and so on size — "phone", "tablet"; or multiple at once "phone, tablet" and so on |

or you can export it directly from Photoshop using Export Hotspots and Popups Coordinates for threesixty component script:

1. Place [this file](https://exponential.box.com/s/p2v3lno02bwfeftlvbynvgm7ofykbera) in
   1. Mac: '~/Applications/Adobe Photoshop CS#/Presets/Scripts/'  <http://prntscr.com/ryazmo>
   2. Win: 'C:\Program Files\Adobe\Adobe Photoshop CS#\Presets\Scripts\'
2. Restart Photoshop
3. File -> Scripts -> Export Hotspots and Popups Coordinates for threesixty component (Or you can just select it here File -> Scripts -> Browse)  
   

4. Choose the "mainunit" folder in the dialog form.   


5. The threesixty view settings will be put in the place marked below (after the last component and before this text /\*\* next components code will start from here start with "," \*\*/)  (comma will be added automatically). Be sure that you delete the old threesixty settings before run the script again (it doesn't remove it automatically)  


6. You can use [this template.](https://jira.exponential.com/secure/attachment/147288/threesiztyViewHotspotsExportTemplate.psd)  Here is the PSD [example](https://jira.exponential.com/secure/attachment/147289/car360_970x465.psd).  
  
7. To hide a hotspot or popup from a frame you can delete the layer with his name (hotspot5 for example) or change name to "hotspot5-hidden"

## Component Callbacks:

* "**onImageClicked**" (fires on an image click)
* "**onLeftArrowClick**" (fires on left arrow click event)
* "**onRightArrowClick**" (fires on right arrow click event)

For each hotspot/popup:

* "**onHotspotClicked**" (fires on a hotspot click)
* "**onPopupCloseClicked**" (fires on popuup close button click)

## Methods:

* Expo.designerAPI.threesixtyViewWithHotspots  - get all copies of the component,
* Expo.designerAPI.threesixtyViewWithHotspots.get('placeholderId') - find a unique copy of the component
* **reset**(animationOn) - set the slider to the default position, animationOn can be true or false, default - false.
* **init**(options) - reset default config, options is a object of componenta parameters
* **render**() - render a dom of the component with default config
* **update**(options) - destroy the dom and rerender it with a new options

## Examples:

* Expo.designerAPI.threesixtyViewWithHotspots.get('threesixtyViewWithHotspots\_1').reset(true); - reset only one copy of the component with a placeholder 'threesixtyViewWithHotspots\_1'
* Expo.designerAPI.threesixtyViewWithHotspots.reset(true); - reset all copies of the component.

Code Snippet

* **Use this line of code on Tab to Initialise the Three Sixty View With Hotspots (you need to set "preventRender" to true to use it)**  
  Expo.designerAPI.components.get("threesixtyViewWithHotspots\_1").render();
* **Use this line of code on Tab to reset the Three Sixty View With Hotspots**Expo.designerAPI.components.get("threesixtyViewWithHotspots\_1").reset();

{

name: "threesixtyViewWithHotspots",

props: {

            preventRender: true,

placeholderId: "threesixtyViewWithHotspots\_1",

imagesPath: "images/",

frames: [

"img1.jpg",

"img2.jpg",

"img3.jpg",

"img4.jpg",

"img5.jpg",

"img6.jpg",

"img7.jpg",

"img8.jpg",

"img9.jpg",

"img10.jpg"

],

width: 970,

height: 466,

speed: 15,

hotSpots: [

{

id: "threesixtyViewWithHotspots\_1\_hotspot\_1",

frames: [

{ x: 320, y: 240 },

{ x: 165, y: 240 },

"hidden",

"hidden",

"hidden",

"hidden",

"hidden",

{ x: 770, y: 240 },

{ x: 610, y: 240 },

{ x: 467, y: 240 }

],

popUpId: "threesixtyViewWithHotspots\_1\_hotspot\_1\_Popup",

popUpCloseId: "threesixtyViewWithHotspots\_1\_hotspot\_1\_PopupClose",

popUpFrames: [

{ x: 140, y: 40 },

{ x: 185, y: 40 },

"hidden",

"hidden",

"hidden",

"hidden",

"hidden",

{ x: 590, y: 40 },

{ x: 630, y: 40 },

{ x: 285, y: 40 }

],

events: {

onHotspotClicked: function(Expo, obj) { Expo.designerAPI.firePixel("onHotspotClicked\_1\_1", [{"eventName":"intLive", "multi":true}]); },

onPopupCloseClicked: function(Expo, obj) { Expo.designerAPI.firePixel("onPopupCloseClicked\_1\_1", [{"eventName":"intLive", "multi":true}]); }

}

},

{

id: "threesixtyViewWithHotspots\_1\_hotspot\_2",

frames: [

{ x: 530, y: 280 },

{ x: 380, y: 280 },

{ x: 262, y: 280 },

{ x: 280, y: 270 },

"hidden",

{ x: 660, y: 270 },

{ x: 665, y: 280 },

{ x: 560, y: 280 },

{ x: 400, y: 280 },

"hidden"

],

events: {

onHotspotClicked: function(Expo, obj) { Expo.designerAPI.firePixel("onHotspotClicked\_1\_2", [{"eventName":"intLive", "multi":true}]); },

onPopupCloseClicked: function(Expo, obj) { Expo.designerAPI.firePixel("onPopupCloseClicked\_1\_2", [{"eventName":"intLive", "multi":true}]); }

},

popUpId: "threesixtyViewWithHotspots\_1\_hotspot\_2\_Popup",

popUpCloseId: "threesixtyViewWithHotspots\_1\_hotspot\_2\_PopupClose",

popUpFrames: [

{ x: 550, y: 40 },

{ x: 400, y: 40 },

{ x: 280, y: 40 },

{ x: 300, y: 40 },

"hidden",

{ x: 480, y: 40 },

{ x: 485, y: 40 },

{ x: 380, y: 40 },

{ x: 420, y: 40 },

"hidden"

]

},

{

id: "threesixtyViewWithHotspots\_1\_hotspot\_3",

frames: [

"hidden",

"hidden",

{ x: 770, y: 260 },

{ x: 690, y: 280 },

{ x: 465, y: 285 },

{ x: 249, y: 280 },

{ x: 165, y: 260 },

"hidden",

"hidden",

"hidden"

],

events: {

onHotspotClicked: function(Expo, obj) { Expo.designerAPI.firePixel("onHotspotClicked\_1\_3", [{"eventName":"intLive", "multi":true}]); },

onPopupCloseClicked: function(Expo, obj) { Expo.designerAPI.firePixel("onPopupCloseClicked\_1\_3", [{"eventName":"intLive", "multi":true}]); }

},

popUpId: "threesixtyViewWithHotspots\_1\_hotspot\_3\_Popup",

popUpCloseId: "threesixtyViewWithHotspots\_1\_hotspot\_3\_PopupClose",

popUpFrames: [

"hidden",

"hidden",

{ x: 590, y: 40 },

{ x: 510, y: 40 },

{ x: 485, y: 40 },

{ x: 270, y: 40 },

{ x: 185, y: 40 },

"hidden",

"hidden",

"hidden"

]

},

{

id: "threesixtyViewWithHotspots\_1\_hotspot\_4",

frames: [

{ x: 450, y: 210 },

{ x: 275, y: 210 },

{ x: 200, y: 210 },

"hidden",

"hidden",

"hidden",

"hidden",

{ x: 665, y: 210 },

{ x: 480, y: 210 },

{ x: 594, y: 210 }

],

events: {

onHotspotClicked: function(Expo, obj) { Expo.designerAPI.firePixel("onHotspotClicked\_1\_4", [{"eventName":"intLive", "multi":true}]); },

onPopupCloseClicked: function(Expo, obj) { Expo.designerAPI.firePixel("onPopupCloseClicked\_1\_4", [{"eventName":"intLive", "multi":true}]); }

},

popUpId: "threesixtyViewWithHotspots\_1\_hotspot\_4\_Popup",

popUpCloseId: "threesixtyViewWithHotspots\_1\_hotspot\_4\_PopupClose",

popUpFrames: [

{ x: 270, y: 40 },

{ x: 295, y: 40 },

{ x: 220, y: 40 },

"hidden",

"hidden",

"hidden",

"hidden",

{ x: 480, y: 40 },

{ x: 300, y: 40 },

{ x: 610, y: 40 }

]

},

{

id: "threesixtyViewWithHotspots\_1\_hotspot\_5",

frames: [

{ x: 490, y: 140 },

{ x: 405, y: 140 },

{ x: 375, y: 140 },

{ x: 365, y: 150 },

"hidden",

"hidden",

{ x: 565, y: 140 },

{ x: 555, y: 140 },

{ x: 570, y: 140 },

{ x: 530, y: 140 }

],

events: {

onHotspotClicked: function(Expo, obj) { Expo.designerAPI.firePixel("onHotspotClicked\_1\_5", [{"eventName":"intLive", "multi":true}]); },

onPopupCloseClicked: function(Expo, obj) { Expo.designerAPI.firePixel("onPopupCloseClicked\_1\_5", [{"eventName":"intLive", "multi":true}]); }

},

popUpId: "threesixtyViewWithHotspots\_1\_hotspot\_5\_Popup",

popUpCloseId: "threesixtyViewWithHotspots\_1\_hotspot\_5\_PopupClose",

popUpFrames: [

{ x: 510, y: 40 },

{ x: 425, y: 40 },

{ x: 395, y: 40 },

{ x: 385, y: 40 },

"hidden",

"hidden",

{ x: 385, y: 40 },

{ x: 375, y: 40 },

{ x: 390, y: 40 },

{ x: 350, y: 40 }

]

}

],

navigationButtons: [

{

id: "threesixtyViewWithHotspots\_1\_leftArrow",

direction: "left",

events: [

{

onArrowClicked: function(Expo, obj) {Expo.designerAPI.firePixel("onLeftArrowClick\_1", [{ "eventName": "intLive", "multi": true }]);}

}

]

},

{

id: "threesixtyViewWithHotspots\_1\_rightArrow",

direction: "right",

events: [

{

onArrowClicked: function(Expo, obj) {Expo.designerAPI.firePixel("onRightArrowClick\_1", [{ "eventName": "intLive", "multi": true }]);}

}

]

}

]

},

events: {

onImageClicked: function(Expo, obj) { Expo.designerAPI.firePixel("onImageClicked\_1", [{"eventName":"intLive", "multi":true}]); }

}

}

## Component Callbacks:

* "**onImageClicked**" (fires on an image click)
* "**onLeftArrowClick**" (fires on left arrow click event)
* "**onRightArrowClick**" (fires on right arrow click event)

For each hotspot/popup:

* "**onHotspotClicked**" (fires on a hotspot click)
* "**onPopupCloseClicked**" (fires on popuup close button click)

## Terms & Restrictions:

* Play-Pause-Progress element appears only if auto-play is enabled (autoPlay > 0) and auto-play-theme is defined as circular (autoPlayTheme.name: "circular")
* Play-Pause-Progress element automatically hides if there's less than two popups in the frame (one or zero)

## Supported interaction Events and Detail

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **User Action** | **Default Source**  **All in bound pixel with have a suffix TabTracker id which is defined by user** | **Event Name** | **Default Sub Source** | **Parameter in insight pixel** | **Can i change source value in non muse BEFORE Advance Pixel Management Release?** | **Can i change source value in non muse AFTER Advance Pixel Management Release?** | **Can i change source value in Muse BEFORE Advance Pixel Management Release?**  **(if component is present in VDX STUDIO)** | **Can i change source value in Muse AFTER Advance Pixel Management Release?** | **Initiator(function/callback initiating the pixel)** |
| click on left Arrow | leftArrow | inbound | {  componentName: "COMPONENT NAME"  tabName: TAB Label like **Offers**  tabSequence: TAB Number like tab{Number  } for example tab1, tab2 where Number is integer value  subSource: slideNumber like slide{Number} for example slide1, slide2 where Number is integer value  } | event=intLive and fr:{source) | NO | NO | YES | YES | onLeftArrowClick |
| click on right Arrow | rightArrow | inbound | {  componentName: "COMPONENT NAME"  tabName: TAB Label like **Offers**  tabSequence: TAB Number like tab{Number  } for example tab1, tab2 where Number is integer value  subSource: slideNumber like slide{Number} for example slide1, slide2 where Number is integer value  } | event=intLive and fr:{source) | NO | NO | YES | YES | onRightArrowClick |
| click on image | imageClicked | inbound | {  componentName: "COMPONENT NAME"  tabName: TAB Label like **Offers**  tabSequence: TAB Number like tab{Number  } for example tab1, tab2 where Number is integer value  subSource: hotspotNumber like hotspot{Number} for example hotspot1, hotspot2 where Number is integer value  } | event=intLive and fr:{source) | NO | NO | YES | YES | onImageClicked |
| on click on hotspot | hotspotClicked | inbound | {  componentName: "COMPONENT NAME"  tabName: TAB Label like **Offers**  tabSequence: TAB Number like tab{Number  } for example tab1, tab2 where Number is integer value  subSource: hotspotNumber like hotspot{Number} for example hotspot1, hotspot2 where Number is integer value  } | event=intLive and fr:{source) | NO | NO | YES | YES | onHotspotClicked |
| on click on popup close button | popupCloseClicked | inbound | {  componentName: "COMPONENT NAME"  tabName: TAB Label like **Offers**  tabSequence: TAB Number like tab{Number  } for example tab1, tab2 where Number is integer value  subSource: hotspotNumber like hotspot{Number} for example hotspot1, hotspot2 where Number is integer value  } | event=intLive and fr:{source) | NO | NO | YES | YES | onPopupCloseClicked |

# HotSpot.Common.VDX.Components.AdEx

* **Template/Demo URLs:** [ClickHere](https://wiki.exponential.com/display/RMDOC/Hotspot.Templates.AdEx)
* Config Details

# Hotspot Component

## Functionality:

Component allows to show an image with hotspots over it. User can see popups by clicking to hotspots. There can be any amount of components in a unit.

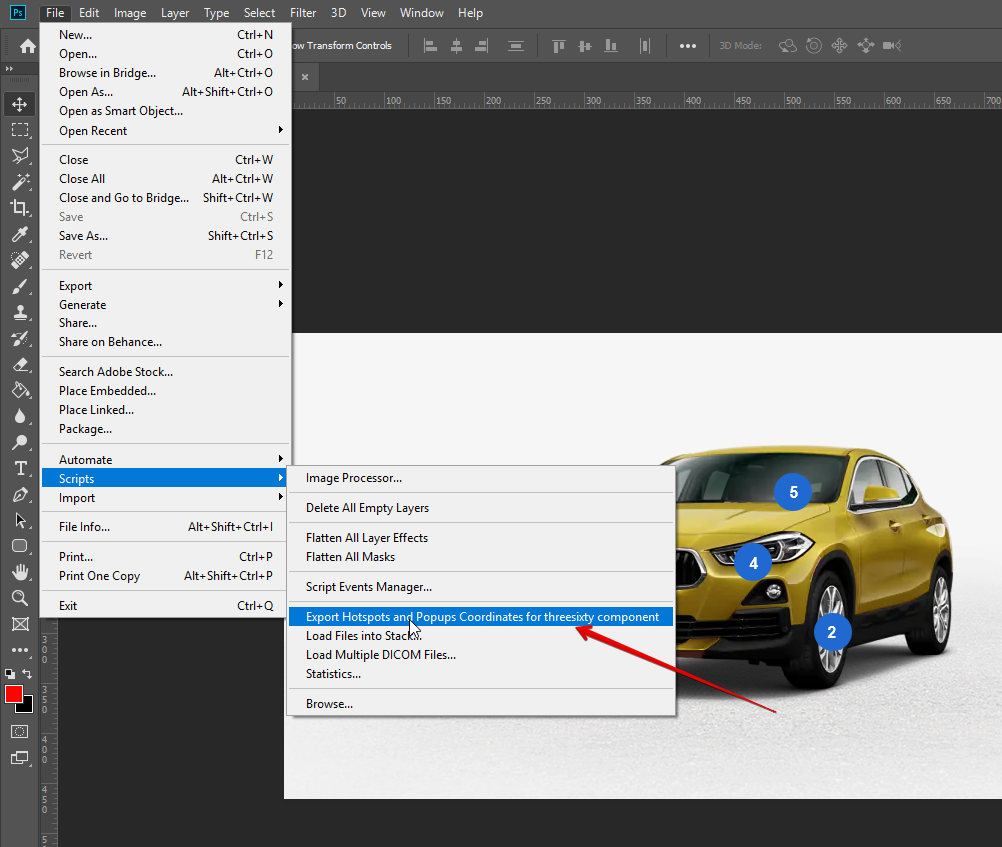
## Muse setup:

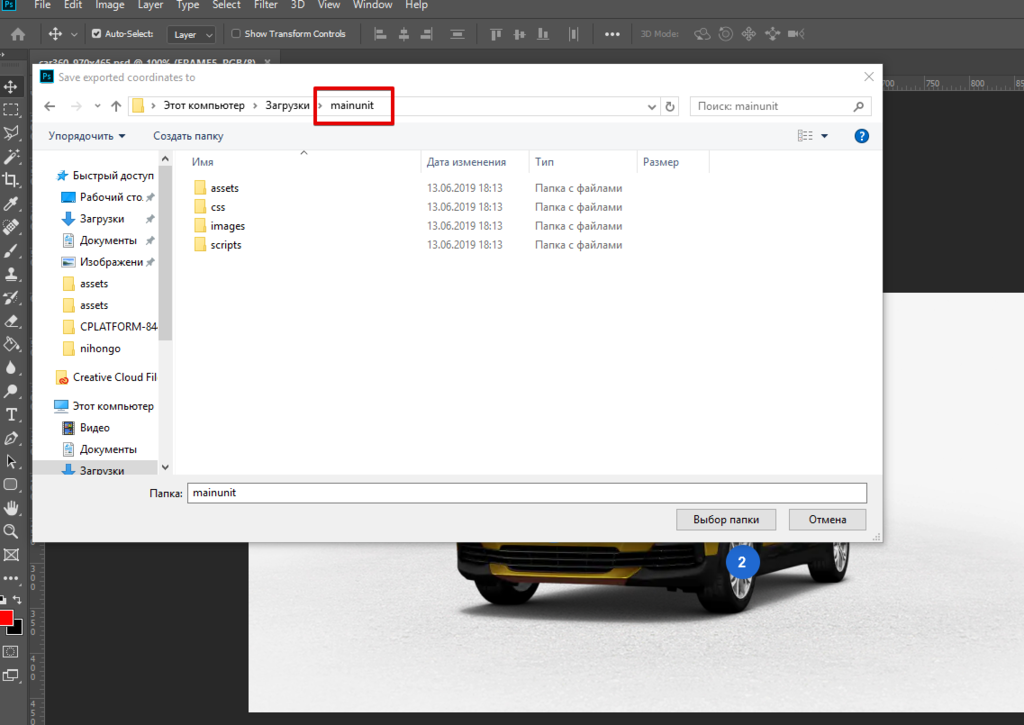
Designer should drop a placeholder on a page, set size and position and add a unique title id. This placeholder will be used as a container for the image. Also Designer should create hotspots and popups in the unit.

## User defined parameters:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Property** | **Type** | **Default** | **Screenshot of property** | **Description** |
| placeholderId | string |  |  | placeholder title |
| imagesPath | string | "images/" |  | path to frame images, including left/right arrows |
| frames | string[] | string |  |  | array of images names, or filename template, plus **totalSlides** to be defined; for ex. "image{xx}.jpg", where {xx} is a slide number.  **It should have one frame only** |
| totalSlides | number |  |  | total slides to be shown, if **frames** is a string  **If provided should be equal to 1** |
| speed | number | 15 |  | speed of item rotation |
| width | number | auto-calculated |  | width of the placeholder has been choice during hotspots setting **not necessary**-  needed only for responsive component |
| height | number | auto-calculated |  | height of the placeholder has been choice during hotspots setting **not necessary**-  needed only for responsive component |
| hotSpots {     id     frames     popUpId     popUpCloseId     popUpFrames } | array {   string   array<{x,y}|"hidden">   string   string   array<{x,y}|"hidden"> } |  |  | list of hotspot and related parameters     **id** - hotspot placeholder id     **frames** - array of hotspot coordinates in each frame , if it is 'hidden' doesn't shown     **popUpId** - popup placeholder id     **popUpCloseId** - popup close button id     **popUpFrames** - array of popup coordinates in each frame , if it is 'hidden' doesn't shows  **It should have one frame only** |
| popupAppearDuration | number | 500 |  | duration of popup appearing animation |
| popupAppearEasing | string | "easeOutCubic" |  | easing of popup appearing animation, linear | easeInQuad | easeOutQuad | easeInOutQuad | easeInCubic | easeOutCubic | easeInOutCubic | easeInQuart | easeOutQuart | easeInOutQuart | easeInQuint | easeOutQuint | easeInOutQuint | easeInSine | easeOutSine | easeInOutSine | easeInExpo | easeOutExpo | easeInOutExpo | easeInCirc | easeOutCirc | easeInOutCirc | easeInBack | easeOutBack | easeInOutBack | easeInElastic | easeOutElastic | easeInOutElastic |
| popupCloseDuration | number | 500 |  | duration of popup closing animation |
| popupCloseEasing | string | "easeInCubic" |  | easing of popup closing animation, linear | easeInQuad | easeOutQuad | easeInOutQuad | easeInCubic | easeOutCubic | easeInOutCubic | easeInQuart | easeOutQuart | easeInOutQuart | easeInQuint | easeOutQuint | easeInOutQuint | easeInSine | easeOutSine | easeInOutSine | easeInExpo | easeOutExpo | easeInOutExpo | easeInCirc | easeOutCirc | easeInOutCirc | easeInBack | easeOutBack | easeInOutBack | easeInElastic | easeOutElastic | easeInOutElastic |
| clickBgToHidePopups | boolean | true |  | click on background to hide opened pop-up |
| bgCursor | string-css-pointer | "pointer" |  | defines background pointer style |
| arrowCursor | string-css-pointer | "pointer" |  | defines navigation arrows pointer style |
| popupsBackdropColor | string-css-color |  |  | popup background color |
| popupOpenByRollover | boolean | false |  | open corresponding pop-ups by hovering hotspots |
| bgImageFill **\*new in 3.10.1** | string-css-bg-size | "contain" |  | slides image fitting |
| popupTheme { **\*new in 3.10.1**     name     background     rounding     tipSize     tipCurved   } | object {     "default" | "bubble"     string-css-background     number     number | number | string     boolean   } | "default" |  | re-defines popup theme (for bubble theme)   **name**: "bubble",   **background**: "#FFF",   **rounding**: 10, // [10] | [10, 20] | [10, 20, 30, 40] | "10" | "10 20" | "10 20 30 40"   **tipSize**: 40,   **tipCurved**: false, |
| autoPlayTheme { **\*new in 3.10.1**     name     position     size     timeSize     timeWidth     timeColor     bgColor   } | object {     "none" | "circular"     "[tbc][lrc]" | string-inline-style     number     number     number     string-css-color     string-css-color   } |  |  | Progress theme customization     Default theme selection should be **circular.**    **name:** circular theme means progress bar in circular shape     **position**: defines circular placement / position     **size**: sets the size of background circle     **timeSize**: set the size of circular progress bar     **timeWidth**: Set the thickness of the circular progress bar     **timeColor**: Control play/pause color     **bgColor**: Control play/pause color |
| dropSound **\*new in 3.10.1** | string-url |  |  | Drop sound will play when pins drop on the screen initially. |
| popSound **\*new in 3.10.1** | string-url |  |  | Pop sound plays when the next location pin appears through auto play. |
| autoPlay **\*new in 3.10.1** | number-time-ms |  |  | Time to start auto-play after initialization. 0 = no autoplay  To achieve the “***Default***” theme we need to set the property value as 0. |
| autoPlayPauseOnHover **\*new in 3.10.1** | boolean |  |  | indicates if auto-play has to be paused on hover or tap on popup |
| autoPlayHoverResume **\*new in 3.10.1** | number-time-ms |  |  | This is the time after which the auto play will resume if the user mouse out from pin/popup. |
| autoPlayResume **\*new in 3.10.1** | number-time-ms |  |  | In case we don’t want to resume the auto play, we can set the property as 0. |
| autoPlayScenario **\*new in 3.10.1** | list of three-item arrays [startTimeMs, actorName, actionName] |  |  | With Auto Play Scenario properties, we can set the auto play animation behavior with 3 step sequence.  Example:  [0, 'marker', 'pop'] – Step 1  [500, 'popup', 'show'] – Step 2  [4000, 'popup', 'hide'] – Step 3  In these 3 steps we have 3 items: Time (StartTime), Target (Actor Name) and Action  StartTime – This will be key parameter which defines when the Actor will be in action.  Actor Name – Which object going to be in Action.  Action – What Actor to be done.  Let’s understand how animation will work as per above mentioned Auto Play Scenario.  [0, 'marker', 'pop'] – As soon as the auto play starts, at 0 milliseconds, the marker(pin) will pop.  [500, 'popup', 'show'] – after 0.5 seconds the popup will be visible.  [4000, 'popup', 'hide'] – After 4 seconds, it will hide the popup and move to next marker and continues the same Auto Play Scenario again…  We recommend only to change the Time property as per the Autoplay properties you want. Rest are the fixed parameters. |
| preloaderColor | string-css-color | "rgb(0, 174, 239)" |  | color of preloader line / preloader background, if image is provided |
| preloaderContainerHeight | string-css-length | "3px" |  | height of the preloader possible values: XXpx | XX% |
| preloaderImage | string-url |  |  | image like jpg|gif|svg rendered as background of preloader |
| preloaderImagePosition | string-css-bg-position | "center" |  | position of the preload image inside the preload container, possible values: top | bottom | center | left | right |
| preloaderImageSize | string-css-bg-size | "auto" |  | "background-size" for image inside the preloader container, possible values: auto | contain | cover |
| preventRender | boolean | false |  | Prevent automatic rendering at the ad-unit initialization If value is false — the "render" function should be called anytime later |
| restrict {     execution     size } | object {String} |  |  | If provided, component would be rendered or not based on inputs execution — "inframe", "instream", "display", "mobile"; or multiple at once "instream, display" and so on size — "phone", "tablet"; or multiple at once "phone, tablet" and so on |

or you can export it directly from Photoshop using Export Hotspots and Popups Coordinates for threesixty component script (doesn't export any images, only json):

* 1. Place [this file](https://exponential.box.com/s/p2v3lno02bwfeftlvbynvgm7ofykbera) in
     1. Mac: '~/Applications/Adobe Photoshop CS#/Presets/Scripts/'  <http://prntscr.com/ryazmo>
     2. Win: 'C:\Program Files\Adobe\Adobe Photoshop CS#\Presets\Scripts\'
  2. Restart Photoshop
  3. File -> Scripts -> Export Hotspots and Popups Coordinates for threesixty component (Or you can just select it here File -> Scripts -> Browse)  
     

4. Choose the "mainunit" folder in the dialog form.   


5. The threesixty view settings will be put in the place marked below (after the last component and before this text /\*\* next components code will start from here start with "," \*\*/)  (comma will be added automatically). Be sure that you delete the old threesixty settings before run the script again (it doesn't remove it automatically)  


6. You can use [this template.](https://jira.exponential.com/secure/attachment/147288/threesiztyViewHotspotsExportTemplate.psd)  Here is the PSD [example](https://jira.exponential.com/secure/attachment/147289/car360_970x465.psd).  
  
7. To hide a hotspot or popup from a frame you can delete the layer with his name (hotspot5 for example) or change name to "hotspot5-hidden"

## Component Callbacks:

* 1. "**onImageClicked**" (fires on an image click)

For each hotspot/popup:

* 1. "**onHotspotClicked**" (fires on a hotspot click)
  2. "**onPopupCloseClicked**" (fires on popuup close button click)

## Methods:

* 1. Expo.designerAPI.threesixtyViewWithHotspots  - get all copies of the component,
  2. Expo.designerAPI.threesixtyViewWithHotspots.get('placeholderId') - find a unique copy of the component
  3. **reset**(animationOn) - set the slider to the default position, animationOn can be true or false, default - false.
  4. **init**(options) - reset default config, options is a object of componenta parameters
  5. **render**() - render a dom of the component with default config
  6. **update**(options) - destroy the dom and rerender it with a new options

## Examples:

* 1. Expo.designerAPI.threesixtyViewWithHotspots.get('threesixtyViewWithHotspots\_1').reset(true); - reset only one copy of the component with a placeholder 'threesixtyViewWithHotspots\_1'
  2. Expo.designerAPI.threesixtyViewWithHotspots.reset(true);- reset all copies of the component.
* Code Snippet
  1. **Use this line of code on Tab to Initialise the Three Sixty View With Hotspots (you need to set "preventRender" to true to use it)**  
     Expo.designerAPI.components.get("threesixtyViewWithHotspots\_1").render();
  2. **Use this line of code on Tab to reset the Three Sixty View With Hotspots**Expo.designerAPI.components.get("threesixtyViewWithHotspots\_1").reset();
* {
* name: "threesixtyViewWithHotspots",
* props: {
* preventRender: true,
* placeholderId: "threesixtyViewWithHotspots\_1",
* imagesPath: "images/",
* frames: [
* "bg\_intetrior.jpg"
* ],
* width: 970,
* height: 466,
* hotSpots: [
* {
* id: "threesixtyViewWithHotspots\_1\_hotspot\_1",
* frames: [
* { x: 140, y: 160 }
* ],
* popUpId: "threesixtyViewWithHotspots\_1\_hotspot\_1\_Popup",
* popUpCloseId: "threesixtyViewWithHotspots\_1\_hotspot\_1\_PopupClose",
* popUpFrames: [
* { x: 157, y: 150 }
* ],
* events: {
* onHotspotClicked: function(Expo, obj) { Expo.designerAPI.firePixel("onHotspotClicked\_1\_1", [{"eventName":"intLive", "multi":true}]); },
* onPopupCloseClicked: function(Expo, obj) { Expo.designerAPI.firePixel("onPopupCloseClicked\_1\_1", [{"eventName":"intLive", "multi":true}]); }
* }
* },
* {
* id: "threesixtyViewWithHotspots\_1\_hotspot\_2",
* frames: [
* { x: 460, y: 360 }
* ],
* events: {
* onHotspotClicked: function(Expo, obj) { Expo.designerAPI.firePixel("onHotspotClicked\_1\_2", [{"eventName":"intLive", "multi":true}]); },
* onPopupCloseClicked: function(Expo, obj) { Expo.designerAPI.firePixel("onPopupCloseClicked\_1\_2", [{"eventName":"intLive", "multi":true}]); }
* },
* popUpId: "threesixtyViewWithHotspots\_1\_hotspot\_2\_Popup",
* popUpCloseId: "threesixtyViewWithHotspots\_1\_hotspot\_2\_PopupClose",
* popUpFrames: [
* { x: 280, y: 170 }
* ]
* },
* {
* id: "threesixtyViewWithHotspots\_1\_hotspot\_3",
* frames: [
* { x: 670, y: 180 }
* ],
* events: {
* onHotspotClicked: function(Expo, obj) { Expo.designerAPI.firePixel("onHotspotClicked\_1\_3", [{"eventName":"intLive", "multi":true}]); },
* onPopupCloseClicked: function(Expo, obj) { Expo.designerAPI.firePixel("onPopupCloseClicked\_1\_3", [{"eventName":"intLive", "multi":true}]); }
* },
* popUpId: "threesixtyViewWithHotspots\_1\_hotspot\_3\_Popup",
* popUpCloseId: "threesixtyViewWithHotspots\_1\_hotspot\_3\_PopupClose",
* popUpFrames: [
* { x: 477, y: 206 }
* ]
* },
* {
* id: "threesixtyViewWithHotspots\_1\_hotspot\_4",
* frames: [
* { x: 760, y: 210 }
* ],
* events: {
* onHotspotClicked: function(Expo, obj) { Expo.designerAPI.firePixel("onHotspotClicked\_1\_4", [{"eventName":"intLive", "multi":true}]); },
* onPopupCloseClicked: function(Expo, obj) { Expo.designerAPI.firePixel("onPopupCloseClicked\_1\_4", [{"eventName":"intLive", "multi":true}]); }
* },
* popUpId: "threesixtyViewWithHotspots\_1\_hotspot\_4\_Popup",
* popUpCloseId: "threesixtyViewWithHotspots\_1\_hotspot\_4\_PopupClose",
* popUpFrames: [
* { x: 580, y: 240 }
* ]
* },
* {
* id: "threesixtyViewWithHotspots\_1\_hotspot\_5",
* frames: [
* { x: 850, y: 110 }
* ],
* events: {
* onHotspotClicked: function(Expo, obj) { Expo.designerAPI.firePixel("onHotspotClicked\_1\_5", [{"eventName":"intLive", "multi":true}]); },
* onPopupCloseClicked: function(Expo, obj) { Expo.designerAPI.firePixel("onPopupCloseClicked\_1\_5", [{"eventName":"intLive", "multi":true}]); }
* },
* popUpId: "threesixtyViewWithHotspots\_1\_hotspot\_5\_Popup",
* popUpCloseId: "threesixtyViewWithHotspots\_1\_hotspot\_5\_PopupClose",
* popUpFrames: [
* { x: 670, y: 80 }
* ]
* },
* {
* id: "threesixtyViewWithHotspots\_1\_hotspot\_6",
* frames: [
* { x: 510, y: 50 }
* ],
* events: {
* onHotspotClicked: function(Expo, obj) { Expo.designerAPI.firePixel("onHotspotClicked\_1\_6", [{"eventName":"intLive", "multi":true}]); },
* onPopupCloseClicked: function(Expo, obj) { Expo.designerAPI.firePixel("onPopupCloseClicked\_1\_6", [{"eventName":"intLive", "multi":true}]); }
* },
* popUpId: "threesixtyViewWithHotspots\_1\_hotspot\_6\_Popup",
* popUpCloseId: "threesixtyViewWithHotspots\_1\_hotspot\_6\_PopupClose",
* popUpFrames: [
* { x: 530, y: 40 }
* ]
* }
* ]
* },
* events: {
* onImageClicked: function(Expo, obj) { Expo.designerAPI.firePixel("onImageClicked\_1", [{"eventName":"intLive", "multi":true}]); }
* }

}

## Component Callbacks:

* 1. "**onImageClicked**" (fires on an image click)

For each hotspot/popup:

* 1. "**onHotspotClicked**" (fires on a hotspot click)
  2. "**onPopupCloseClicked**" (fires on popuup close button click)

## Supported interaction Events and Detail

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **User Action** | **Default Source**  **All in bound pixel with have a suffix TabTracker id which is defined by user** | **Event Name** | **Default Sub Source** | **Parameter in insight pixel** | **Can i change source value in non muse BEFORE Advance Pixel Management Release?** | **Can i change source value in non muse AFTER Advance Pixel Management Release?**  **(if component is present in VDX STUDIO)** | **Can i change source value in Muse BEFORE Advance Pixel Management Release?** | **Can i change source value in Muse AFTER Advance Pixel Management Release?** | **Initiator(function/callback initiating the pixel)** |
| click on image | imageClicked | inbound | {  componentName: "COMPONENT NAME"  tabName: TAB Label like **Offers**  tabSequence: TAB Number like tab{Number  } for example tab1, tab2 where Number is integer value  subSource: hotspotNumber like hotspot{Number} for example hotspot1, hotspot2 where Number is integer value  } | event=intLive and fr:{source) | NO | NO | YES | YES | onImageClicked |
| click on hotspot | hotspotClicked | inbound | {  componentName: "COMPONENT NAME"  tabName: TAB Label like **Offers**  tabSequence: TAB Number like tab{Number  } for example tab1, tab2 where Number is integer value  subSource: hotspotNumber like hotspot{Number} for example hotspot1, hotspot2 where Number is integer value  } | event=intLive and fr:{source) | NO | NO | YES | YES | onHotspotClicked |
| click on popuup close button | popupCloseClicked | inbound | {  componentName: "COMPONENT NAME"  tabName: TAB Label like **Offers**  tabSequence: TAB Number like tab{Number  } for example tab1, tab2 where Number is integer value  subSource: hotspotNumber like hotspot{Number} for example hotspot1, hotspot2 where Number is integer value  } | event=intLive and fr:{source) | NO | NO | YES | YES | onPopupCloseClicked |

# ComparisonSlider.Common.VDX.Components.AdEx

**Template/Demo URLs:** [Click Here](https://wiki.exponential.com/display/VDXConnect/SingleFramework.VDXConnect.Templates.AdEx)

## Updated for components-release - [HTML5 Components 3.11.0-5 Release](file:////display/VDXComponents/HTML5+Components+3.11.0-5)

## Functionality:

Component allows showing a comparison of 2 elements (image-image, video-image, video-video) by sliding or auto moving a border to show/hide the top element. There can be any amount of components in a unit.

## Muse setup:

Designer should drop a placeholder on a page, set size and position and add a unique title id.   
Also designer has to upload 2 files (image-image, video-video) to the **mainunit/assets** folder.

Configuration properties details

## ****Latest property updates:****

| **Property** | **Type** | **Default** | **Available in**  **Non-Muse Flow** | **Screenshot of property** | **Description** |
| --- | --- | --- | --- | --- | --- |
| isVertical | Boolean | false | No |  | experimental if true then separator would switch comparison into vertical mode) |
| leftVideo {      videoId     isMuted     hasAudio     loop } | Object     String     Boolean     Boolean     Boolean | Image Comparison by default | Yes (partial) | A screenshot of a video  AI-generated content may be incorrect. | Parameters required for Left video: **videoId**: Define the video name which we want to show on the left side in video comparison.     **isMuted**: If set to true then sound would be initially muted.     **hasAudio**: If set to false then video would play muted permanently.     **loop**: If set to true then video would play in loop |
| rightVideo {      videoId     isMuted     hasAudio     loop } | Object     String     Boolean     Boolean     Boolean | Image Comparison by default | Yes | A screenshot of a video  AI-generated content may be incorrect.  A screenshot of a video slide  AI-generated content may be incorrect. | Parameters required for Right video:     **videoId**: Define the video name which we want to show on the right side in video comparison.     **isMuted**: If set to true then sound would be initially muted.     **hasAudio**: If set to false then video would play muted permanently.     **loop**: If set to true then video would play in loop |
| leftOverlay { **\*new in 3.10.2**     contentId     contentHtml     style } | Object     String     String     Css-Style |  |  |  | Defines left overlay:     **contentId**: [optional] id of the document block to overlay content.     **contentHtml**: [optional] html markup to overlay content.     **style**: additional inline-style for the overlay |
| rightOverlay { **\*new in 3.10.2**     contentId     contentHtml     style } | Object     String     String     Css-Style |  |  |  | Defines right overlay:     **contentId**: [optional] id of the document block to overlay content.     **contentHtml**: [optional] html markup to overlay content.     **style**: additional inline-style for the overlay |
|  |  | Left video will play with the sound and right video will be muted. |  | A black rectangular object with white text  AI-generated content may be incorrect. | This enables the sound icons for both videos in comparison slider and we can toggle the sound by clicking on the icon.  Once we enable this property, we can access all 4 possible states by clicking on the sound icons:   * Default – On initialisation | Property values: leftVideo.isMuted = false, rightVideo.isMuted = false, showMuteIcon = true. * Both Muted - Property values: leftVideo.isMuted = true, rightVideo.isMuted = true, showMuteIcon = false. * Left Video Muted - Property values: leftVideo.isMuted = true, rightVideo.isMuted = false, showMuteIcon = true. * Right Video Muted -  Property values: leftVideo.isMuted = false, rightVideo.isMuted = true, showMuteIcon = true. |
| videoFit | String | cover | Yes | A black rectangular object with white text  AI-generated content may be incorrect. | “Object-fit" for videos inside the container. |
| playback | String | sync | No |  | Do not change the value. |
| syncGap | Number-time-s | 0.2 | No |  | Defines maximum playback progress difference between main and secondary videos for "sync" mode. Not supported yet ın VDX Studıo. |
| syncEndDiff | Number-time-s | 1.0 | No |  | Defines maximum duration difference between main and secondary videos for "sync" mode. Planned to be below 1.0 all the time. Not supported yet ın VDX Studıo. |
| slideIconWidth | Number |  | Yes | A black rectangular object with white text  AI-generated content may be incorrect. | Set the size of slider icon through this property. |
| autoPlayEnable | Boolean | true | No |  | If set to false then auto play mode wouldn't start no event if autoPlay value is not 0 |
| autoPlay | Number-time-ms |  | Yes | A screen shot of a video game  AI-generated content may be incorrect. | Time to start auto-play after initialisation |
| autoPlayResume | Number-time-ms |  | Yes | A screenshot of a video game  AI-generated content may be incorrect. | The time after which the auto play will resume once the user interacts with the component. |
| autoPlayRepeat | Number | 0 | No |  | Amount of automatic swipe repetitions, 0 = infinite times |
| autoPlayTick | Number-time-ms |  | Yes | A screenshot of a video game  AI-generated content may be incorrect. | Travel duration in which the slider(with hand cursor) will travel from left-to-right and right-to-back in case of auto play. |
| autoPlayLeftRightPx | num Number | [0, 0] | Yes | A screenshot of a black rectangular object with white text  AI-generated content may be incorrect. | Defines left and right offsets of both sides in AutoPlay mode.   If we set it as [10, 20] then during AutoPlay mode slider line will stop 10px away from the left side when going to the left and would stop 20px away from the right side when going to the right.  In this case, designer must know that maximum values for both sides can't be more than half of the comparison slider container. This parameter is optional and can be removed from the configuration, if not parameter at the configuration, default value will be set as [0, 0]. |
| autoPlayHintHtml | String-html | Optional in Muse flow | No |  | We can customize the [hand cursor](https://prnt.sc/9RRwwo8aWHFt) icon for “autoPlayTick” animation through css. |
| autoPlayHintImage | String-url | - | No |  | Alternative to autoPlayHintHtml property, image to be shown above side-icon during auto-play |
| autoPlayHandCursor | Boolean | true | No |  | Show hand hint image over slider icon to invite to interact with component |

## User defined parameters:

* "**placeholderId**" (String - placeholder title)
* **"isVertical"** (Boolean - experimental if true then separator would switch comparison into vertical mode)
* "**leftImageName**" (String - the name of the left image)
* "**rightImageName**" (String - the name of the right image)
* **"slideIcon"** (String - image name of an icon that will be showed to let user know how to use it)
* "**slideIconPosition**" (String - position of a slide icon, uses css property "background-position", default: "center", possible values - canter|top|bottom|0px 30% )
* "**animationOn**" (Boolean - enable/disable animation, default: true)
* "**startingPosition**" (Number - starting position of the slider in percents, default: 50, possible values - from 0 to 100)
* "**clickToMoveOn**" (Boolean - enable/disable sliding by click to an image, default: true)
* "**sliderWidth**" (Number - width of the slider line)
* "**sliderColor**" (String - color of the slider line, default: #000)
* "**sliderStyle**" (String, style of the slider line, default: solid, possible: none|dotted|dashed|solid|double)
* "**backgroundImageSize**" (String, css style - "background-size" for images inside the container, possible values - auto|contain|cover, default - cover)
* "**preloaderColor**" (String, - css style - 'background-color' of the preloader, default: rgb(0, 174, 239), possible values - rgb(x, x, x)|rgba(x, x, x, y)|#ffffff
* "**preloaderContainerHeight**" (String, - height of the preloader, default: 3px, possible values - Xpx|X% ),
* "**preloaderImage**" (String, - image/gif/svg for background of preloader, if set - height becames 100% and transparent background, you can set any height and background colors by setting preloaderHeight and preloaderColor, default: ""),
* "**preloaderImagePosition**" - (String, the position of the preload image inside the preload container, default: "center", possible values - top|bottom|center|left|right|top left|top center|top right|center left|center right|Xpx Ypx|X% Y% )
* "**preloaderImageSize**" (String, css style - "background-size" for images inside the container, possible values - auto|contain|cover, default - auto)
* "**preventRender**" (Boolean - can be used for any component, prevent calling of "render" function at the initialization, the "render" function can be called anytime by a designer, default: true)
* **"leftVideo"** (Object - contain data about leftVideo element)
  + **"videoId"** (String - string with videoId number)
  + **"isMuted"** (Boolean - controlled if the video is muted from component config, if it's muted from configuration user can't control mute/unmute state for this video, default: true)
  + **"hasAudio"** (Boolean - optional indicates if video has soundtrack or not)
  + **"loop"** (Boolean - if true then video would be playing in loop)
* **"rightVideo"** (Object - contain data about leftVideo element)
  + **"videoId"** (String - string with videoId number)
  + **"isMuted"** (Boolean - controlled if the video is muted from component config, if it's muted from configuration user can't control mute/unmute state for this video, default: true)
  + **"hasAudio"** (Boolean - optional indicates if video has soundtrack or not)
  + **"loop"** (Boolean - if true then video would be playing in loop)
* **"showMuteIcon"** (Boolean - show/hide control element with mute/unmute functionality)
* **"videoFit"** (String - css style - "object-fit" for videosinside the container, possible values - cover | contain | none, default - cover)
* **"playback"** (String - configures videoPllayback functionality, possible values - primary | sync | async, default - sync)
* **"syncGap"** (Number - time in second, means how much time gap can be before videos will be synchronized at playback "sync" mode, default - 0.2)
* **"slideIconWidth"** (Number - sets size of slide icon, default - 50)
* **"autoPlayEnable"** (Boolean - if false then auto play mode wouldn't start no event if autoPlay value is not 0)
* **"autoPlay"** (Number-time-ms - time to start auto-play after initialization, default - )
* **"autoPlayResume"** (Number-time-ms - time to resume auto-play after comparison click/tap interaction, default - 0)
* **"autoPlayTick"** (Number-time-ms - time between animation ticks, when sliding left-to-right and back, default - 0)
* **"autoPlayRepeat"** (Number - amount of automatic swipe repetitions, 0 = infinite times, default - 0)
* **"autoPlayLeftRightPx"** ([number, number] - left and right offsets for auto-play positions in px, default - [ 0, 0 ])
* **"autoPlayHintHtml"** (String-html - html overlay to be shown above side-icon during auto-play, default - optional)
* **"autoPlayHintImage"** (String-url - alternative to autoPlayHintHtml property, image to be shown above side-icon during auto-play, default - optional)
* **"autoPlayHandCursor"** (Boolean - show hand hint image over slider icon to invite to interact with component, default true)

## Video parameters description:

* The replay button will be shown when all videos will be finished
* The video controls section is not showing (progress bar, play/pause buttons)
* The comparison slider can show video and image at the same config (image/image, image/video, video/video)
* The comparison slider has its own mute/unmute button with 3 states support:
  1. enable left sound - disable right sound
  2. disable left sound - disable right sound
  3. disable left sound - enable right sound
* playback parameter description (component default value - 'sync')
  1. "primary" - left video will start, right video will be paused
  2. "sync" - video will be playing synchronous when video positions gap between 2 videos will be more than **syncGap** parameter value, right video position will be overwritten by left video position value. For the first phase, **syncGap** value will be set by component (default value - 0.2) second.
  3. "async" - videos will be played independently
* for VDXStudio flow value for 'showMuteIcon' parameter will have a dependency on the Video sound dropdown ([xd prototype from Harry](https://xd.adobe.com/view/cd984ff7-e7d9-4ffd-86be-c53c417eabdb-98cb/screen/dbd421e5-08c0-4d38-9b34-7e0da4bacc45?fullscreen)), Video sound dropdown values description:
  1. Default - User can control both of the videos mute/unmute states, left video start plaing with sound. Property values: leftVideo.isMuted = false, rightVideo.isMuted = false, showMuteIcon = true.
  2. Both Muted - Videos will be played without sound and user can't control mute/unmute states. Property values: leftVideo.isMuted = true, rightVideo.isMuted = true, showMuteIcon = false.
  3. Left Muted - Left video is muted, user can control mute/unmute state for right video.  Property values: leftVideo.isMuted = true, rightVideo.isMuted = false, showMuteIcon = true.
  4. Right Muted - Right video is muted, user can control mute/unmute state for left video.  Property values: leftVideo.isMuted = false, rightVideo.isMuted = true, showMuteIcon = true.

## Other parameters description and explanation:

* **"autoPlayLeftRightPx"** - defines left and right offsets of component sides for auto-mode. So if you set it as **[10, 20]** then during auto-mode separator line will stop 10px away from the left side when going to the left. And would stop 20px away from the right side when going to the right. In this case, designer needs to know that maximum values for both sides can't be more than half of the comparison slider container. This parameter is optional and can be removed from the configuration, if not parameter at the configuration, default value will be set as **[0, 0]**.

**Configuration parameters for video support (this parameter is added  to support video functionality, all other parameter are still in use):**

## Methods:

Expo.designerAPI.comparisonSlide   - get all copies of the component,  
Expo.designerAPI.comparisonSlide.get('placeholderId') - find a unique copy of the component

**reset**(animationOn) - set the slider to the default position, animationOn can be true or false, default - false.  
**init**(config) - reset default config, options is a object of componenta parameters  
**render**() - render a dom of the component with default config  
**update**(config) - destroy the dom and rerender it with a new options

**play**() - resumes playback of the component (video playback and autoplay)  
**resume**() - resumes playback of the component (video playback and autoplay)  
**pause**() - pauses playback of the component (video playback and autoplay)  
**stop**() - stop component (video playback and autoplay)

## Examples:

Expo.designerAPI.comparisonSlide.get('VDX\_comparisonSlide\_1').reset(); - reset only one copy of the component with a placeholder 'VDX\_comparisonSlide\_1'  
Expo.designerAPI.comparisonSlide.reset(); - reset all copies of the component.

## Component callbacks:

* "**onSliderClicked**" (fires when slider is clicked)
* "**onLeftImageClicked**" (fires when the left image is clicked)
* "**onRightImageClicked**" (fires when the right image is clicked)

## Designer-config code example:

{  
        name: "comparisonSlide",  
        props: {

            preventRender: true,  
            placeholderId:"comparisonSlide\_1",  
            leftImageName:"images/before1.jpg",  
            rightImageName:'images/after1.jpg',  
            slideIcon:'images/drag\_item.png',  
              
            startingPosition:30,  
            sliderWidth: 1,  
            sliderColor: 'rgba(0,0,0,.3)',  
        },  
        events: {  
            onSliderClicked: function(Expo, obj) {  
                Expo.designerAPI.firePixel("onSliderClicked\_1",[{"eventName":"intLive", "multi":true}]);  
            },  
            onLeftImageClicked: function(Expo, obj) {  
                Expo.designerAPI.firePixel("onLeftImageClicked\_1",[{"eventName":"intLive", "multi":true}]);  
            },  
            onRightImageClicked: function(Expo, obj) {  
                Expo.designerAPI.firePixel("onRightImageClicked\_1",[{"eventName":"intLive", "multi":true}]);  
            }  
        }  
    },              
    {  
        name: "comparisonSlide",  
        props: {  
            placeholderId:"comparisonSlide\_2",  
            preventRender: true,  
            leftVideo: {

                videoId: "video2", // videoId  
                isMuted: true,     // if video is muted, it can't be controlled mute/unmute by user   
            },  
            rightVideo: {  
                videoId: "video3",  
                isMuted: false,  
            },  
            showMuteIcon: true, // show/hide controls with mute/unmute functionality  
            videoFit: "cover",  // cover | contain | none  
            playback: "sync",   // "primary" | "sync" | "async"  
            syncGap: 0.2, // time gap between videos in seconds

            slideIcon:'',  
            animationOn:false,  
            startingPosition:70,  
            clickToMoveOn:false,  
            sliderWidth: 8,  
            sliderColor: 'rgba(0,0,0,.3)',  
            sliderStyle: 'dotted',  
        },  
        events: {  
            onSliderClicked: function(Expo, obj) {  
                Expo.designerAPI.firePixel("onSliderClicked\_2",[{"eventName":"intLive", "multi":true}]);  
            },  
            onLeftImageClicked: function(Expo, obj) {  
                Expo.designerAPI.firePixel("onLeftImageClicked\_2",[{"eventName":"intLive", "multi":true}]);  
            },  
            onRightImageClicked: function(Expo, obj) {  
                Expo.designerAPI.firePixel("onRightImageClicked\_2",[{"eventName":"intLive", "multi":true}]);  
            }  
        }  
    }

## Limitations:

* Both the videos sound cannot be unmuted at the same time.
* For the first release Phase, we allow videos to compare only with the same length of the video.
* For the first release Phase, we do not allow video-image compare configurations at both (Muse, nonMuse) flows.
* Supported video durations between min 15 seconds to 5 minutes maximum.

## Notes:

* The placeholder should be a container('button')

Code Snippet

* **Use this line of code on Tab to Initialise the Comparison Slider**  
  Expo.designerAPI.comparisonSlide.get("comparisonSlide").render();
* **Use this line of code on Tab to reset the Comparison Slider**  
  Expo.designerAPI.comparisonSlide.get("comparisonSlide").reset();

/\* Start ComparisonSlider Code from Here \*/

{

name: "comparisonSlide",

props: {

            preventRender: true,

placeholderId:"comparisonSlide1",

leftImageName:"images/before.jpg",

rightImageName:'images/after.jpg',

slideIcon:'images/drag\_icon.png',

startingPosition:50,

sliderWidth: 3,

sliderColor: 'rgba(0,0,0,.3)',

sliderStyle: 'dotted',

preloaderImage: 'images/clock\_preloader.gif',

preloaderImageSize: "auto",

preventRender: true

},

events: {

onSliderClicked: function(Expo, obj) {

Expo.designerAPI.firePixel("ComparisonSlide1\_onSliderClicked",[{"eventName":"intLive", "multi":true}]);

},

onLeftImageClicked: function(Expo, obj) {

Expo.designerAPI.firePixel("ComparisonSlide1\_onLeftImageClicked",[{"eventName":"intLive", "multi":true}]);

},

onRightImageClicked: function(Expo, obj) {

Expo.designerAPI.firePixel("ComparisonSlide1\_onRightImageClicked",[{"eventName":"intLive", "multi":true}]);

}

}

},

{

name: "comparisonSlide",

props: {

preventRender: true,

placeholderId: "comparisonSlide2",

// leftImageName: "images/comparison\_before.jpg",

// rightImageName: "images/comparison\_after.jpg",

leftVideo: {

videoId: "video2",

isMuted: false,

},

rightVideo: {

videoId: "video3",

isMuted: false,

},

showMuteIcon: true,

videoFit: "cover",

playback: "sync",

// syncGap: 0.2, // for current version is used default value from bundle

slideIcon: "images/comparison\_drag\_icon.png",

slideIconPosition: "center",

animationOn: true,

startingPosition: 50,

clickToMoveOn: true,

sliderWidth: 3,

sliderColor: "rgba(0, 0, 0, .3)",

sliderStyle: "dotted",

backgroundImageSize: "cover",

preloaderColor: "rgba(0, 179, 239, .9)",

autoPlay: 1000,

autoPlayResume: 3000,

autoPlayTick: 2000,

autoPlayLeftRightPx: [30, 50],

autoPlayHintHtml: "<img src=\"images/comparison\_handCursor.png\" style=\"width:40px; height:50px; transform:translate(-13px, 6px) rotate(-25deg) scale(.8)\">",

},

events: {

onSliderClicked: function(Expo, obj) {

Expo.designerAPI.firePixel("ComparisonSlide2\_onSliderClicked", [{"eventName":"intLive", "multi":true}]);

},

onLeftImageClicked: function(Expo, obj) {

Expo.designerAPI.firePixel("ComparisonSlide2\_onLeftImageClicked", [{"eventName":"intLive", "multi":true}]);

},

onRightImageClicked: function(Expo, obj) {

Expo.designerAPI.firePixel("ComparisonSlide2\_onRightImageClicked", [{"eventName":"intLive", "multi":true}]);

}

}

},

/\* End ComparisonSlider Code \*/

## Component Callbacks:

* "**onSliderClick**" (fires when slider is clicked)
* "**onLeftImageClick**" (fires when the left image is clicked)
* "**onRightImageClick**" (fires when the right image is clicked)

## Supported interaction Events and Detail

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **User Action** | **Event Name** | **Default Source**  **All in bound pixel with have a suffix TabTracker id which is defined by user** | **Default Sub Source** | **Parameter in insight pixel** | **Can i change source value in non muse BEFORE Advance Pixel Management Release?** | **Can i change source value in non muse AFTER Advance Pixel Management Release?** | **Can i change source value in Muse BEFORE Advance Pixel Management Release?** | **Can i change source value in Muse AFTER Advance Pixel Management Release?** | **Initiator(function/callback initiating the pixel)** |
| click on left Image | inbound | leftImage | {   componentName: "COMPONENT NAME",   tabName: "HOST TAB LABEL", // **Offers**   tabSequence: "tab{HOST\_TAB\_ID}", // **tab1, tab2, ...**   subSource: "leftImage", } | event=intLive and fr:{source) | NO | NO | YES | YES | onLeftImageClick |
| click on right Image | inbound | rightImage | {   componentName: "COMPONENT NAME",   tabName: "HOST TAB LABEL", // **Offers**   tabSequence: "tab{HOST\_TAB\_ID}", // **tab1, tab2, ...**   subSource: "rightImage", } | event=intLive and fr:{source) | NO | NO | YES | YES | onRightImageClick |
| click on slider | inbound | slider | {   componentName: "COMPONENT NAME",   tabName: "HOST TAB LABEL", // **Offers**   tabSequence: "tab{HOST\_TAB\_ID}", // **tab1, tab2, ...**   subSource: "slider", } | event=intLive and fr:{source) | NO | NO | YES | YES | onSliderClick |
| Video events | all supported video event | video{ID} | {   componentName: "COMPONENT NAME",   tabName: "HOST TAB LABEL", // **Offers**   tabSequence: "tab{HOST\_TAB\_ID}", // **tab1, tab2, ...**   subSource: "video{ID}", // **video1, video2, ...** } | all supported video event and fr:{videoID} | NO | NO | NO | NO | Create Video player Instance |
| replay button clicked | replayLive | replayLive | {   componentName: "COMPONENT NAME",   tabName: "HOST TAB LABEL", // **Offers**   tabSequence: "tab{HOST\_TAB\_ID}", // **tab1, tab2, ...**   subSource: "replayButton", } | event=intLive and fr:{source) | NO | NO | NO | YES | onReplayClick |
| Mute/Unmute button Clicked | muteLive unmuteLive | muteLive unmuteLive | {   componentName: "COMPONENT NAME",   tabName: "HOST TAB LABEL", // **Offers**   tabSequence: "tab{HOST\_TAB\_ID}", // **tab1, tab2, ...**   subSource: "muteButton" | "unmuteButton", } | event=intLive and fr:{videoID) | NO | NO | NO | YES | onMuteUnmuteClick |

# Creative Features and Exceptions

## VDX Ad Features List

### Image Gallery (Standard, Dot, Thumbnail)

* **VDX Standard:** No
* **VDX Advanced:** Yes
* **VDX Custom:** Yes
* **Demo:**
  + **Standard:** <https://creative.vdx.tv/#DJh0ggleXwSw0/3>
  + **Dot:** <https://creative.vdx.tv/#UJ.SEARl2euQ0/9>
  + **Thumbnail:** <https://creative.vdx.tv/#8OXOrw52AoEA1/2>
* **Notes:** Not Applicable
* **Descriptions:**
  + Standard Gallery: Interactive image gallery with side arrow navigation for seamless browsing.
  + Dot Gallery: Compact image gallery with dot-based navigation for quick selection.
  + Thumbnail Gallery: Visual image gallery with thumbnail previews for enhanced user control.
* **Keywords:** image gallery, navigation, interactive, standard, dot, thumbnail, visual
* **Use Case:** Ideal for e-commerce ads to display multiple product images efficiently.

### Color Selector

* **VDX Standard:** No
* **VDX Advanced:** Yes
* **VDX Custom:** Yes
* **Demo:** <https://creative.vdx.tv/#VeJENA/21>
* **AutoPlay Demo:** <https://creative.vdx.tv/#EuD-sgQYSRwA/0>
* **Notes:** Not Applicable
* **Descriptions:** Dynamic color selection tool allowing users to view products in various colors, optimized for customization.
* **Keywords:** color selector, product customization, interactive, automotive, fashion
* **Use Case:** Used in car ads to let users explore different paint options.

### Multiple Video Support

* **VDX Standard:** Yes
* **VDX Advanced:** Yes
* **VDX Custom:** Yes
* **Demo:** <https://creative.vdx.tv/#GbIqxg5GanhA1/3>
* **Notes:** Not Applicable
* **Descriptions:** Feature enabling multiple videos in a single tab to showcase products or services comprehensively.
* **Keywords:** video, multiple, product showcase, multimedia, tabbed content
* **Use Case:** Perfect for tech ads displaying various product features via video clips.

### Trigger Tab

* **VDX Standard:** No
* **VDX Advanced:** Yes
* **VDX Custom:** Yes
* **Demo:** <https://creative.vdx.tv/#oWbL4QgyYRMA/8>
* **Notes:** Not Applicable
* **Descriptions:** Interactive tab trigger activated by button clicks or taps for engaging user navigation.
* **Keywords:** trigger tab, interactive, navigation, user engagement, button
* **Use Case:** Useful in educational ads to reveal additional content on demand.

### Location Map

* **VDX Standard:** No
* **VDX Advanced:** Yes
* **VDX Custom:** Yes
* **Demo:** <https://creative.vdx.tv/#butvjwQYSRwA0/9>
* **AutoPlay Demo:** <https://creative.vdx.tv/#HxowtQaWFcPg/10>
* **Notes:** <https://wiki.exponential.com/display/PRODBLOG/2019/08/19/VDX+Component+Framework+%28v2.1%29+Release>
* **Descriptions:** Dynamic map displaying locations by zip code with auto-transitioning pins and popups for localized marketing.
* **Keywords:** location map, dynamic, zip code, geolocation, pins, popups
* **Use Case:** Great for retail ads to show nearby store locations.

### Scroll

* **VDX Standard:** No
* **VDX Advanced:** Yes
* **VDX Custom:** Yes
* **Demo:** <https://creative.vdx.tv/#podj1w2c6.Aw1/4>
* **Notes:** <https://wiki.exponential.com/display/PRODBLOG/2019/08/19/VDX+Component+Framework+%28v2.1%29+Release>
* **Descriptions:** Scrollable content display for presenting extensive information in a compact ad space.
* **Keywords:** scroll, content display, compact, large content, user-friendly
* **Use Case:** Effective in travel ads to list itinerary details.

### Comparison Slider

* **VDX Standard:** No
* **VDX Advanced:** Yes
* **VDX Custom:** Yes
* **Demo:** <https://creative.vdx.tv/#.LC-zw/0>
* **AutoPlay Demo:** <https://creative.vdx.tv/#mia-IAaWFcPg/8>
* **Notes:** <https://wiki.exponential.com/display/PRODBLOG/2019/08/19/VDX+Component+Framework+%28v2.1%29+Release>
* **Descriptions:** Interactive slider comparing two images or videos with optional auto-transition for product comparisons.
* **Keywords:** comparison slider, interactive, image comparison, video comparison, auto-transition
* **Use Case:** Used in tech ads to compare product versions side-by-side.

### View 360

* **VDX Standard:** No
* **VDX Advanced:** Yes
* **VDX Custom:** Yes
* **Demo:** <https://creative.vdx.tv/#8ug49AaWFcPg/10>
* **Notes:** <https://wiki.exponential.com/display/productarchive/Component+v2.2+-+View+360%2C+Interactive+View+360+and+Hotspots>
* **Descriptions:** Rotatable 360-degree image viewer for immersive product exploration.
* **Keywords:** 360 view, rotatable, immersive, product exploration
* **Use Case:** Ideal for real estate ads to showcase property interiors.

### Interactive View 360

* **VDX Standard:** No
* **VDX Advanced:** Yes
* **VDX Custom:** Yes
* **Demo:** <https://creative.vdx.tv/#DpFrDgPCNbUQ0/3>
* **AutoPlay Demo:** <https://creative.vdx.tv/#jEcDYAaWFcPg/10>
* **Notes:** <https://wiki.exponential.com/display/productarchive/Component+v2.2+-+View+360%2C+Interactive+View+360+and+Hotspots>
* **Descriptions:** Enhanced 360-degree viewer with interactive feature callouts for detailed product insights.
* **Keywords:** interactive 360 view, callouts, immersive, product details
* **Use Case:** Useful in automotive ads to highlight car features interactively.

### Hotspots

* **VDX Standard:** No
* **VDX Advanced:** Yes
* **VDX Custom:** Yes
* **Demo:** <https://creative.vdx.tv/#sOQ-oA9izNPw0/2>
* **AutoPlay Demo:** <https://creative.vdx.tv/#PEHtlwaWFcPg/0>
* **Notes:** <https://wiki.exponential.com/display/productarchive/Component+v2.2+-+View+360%2C+Interactive+View+360+and+Hotspots>
* **Descriptions:** Interactive hotspots with detailed popups for engaging content exploration.
* **Keywords:** hotspots, interactive, popups, engagement, content
* **Use Case:** Perfect for lifestyle ads to highlight product benefits.

### Tab Component

* **VDX Standard:** No
* **VDX Advanced:** Yes
* **VDX Custom:** Yes
* **Demo:** <https://creative.vdx.tv/#1P8rAA5a2vOA1/1>
* **Notes:** <https://wiki.exponential.com/pages/viewpage.action?pageId=131341352>
* **Descriptions:** Tabbed interface focusing on different elements within an ad unit for organized content delivery.
* **Keywords:** tab component, navigation, organized, content delivery
* **Use Case:** Great for financial ads to separate service offerings.

### Hybrid Gallery

* **VDX Standard:** No
* **VDX Advanced:** Yes
* **VDX Custom:** Yes
* **Demo:** <https://creative.vdx.tv/#/J5frCgRTdKZA>
* **AutoPlay Demo:** <https://creative.vdx.tv/#bJfsWggyYRMA/0>
* **Notes:** <https://wiki.exponential.com/display/VP/Component+v2.4+-+Hybrid+Gallery>
* **Descriptions:** Video gallery in a secondary tab (max 10 videos, 5 min each, 100 MB total) with auto-transition option.
* **Keywords:** hybrid gallery, video, secondary tab, auto-transition, multimedia
* **Use Case:** Used in entertainment ads to showcase movie trailers.

### Sync Gallery

* **VDX Standard:** No
* **VDX Advanced:** Yes
* **VDX Custom:** Yes
* **Demo:**
  + **Demo1 (with pins):** <https://creative.vdx.tv/#7Ab1Zw/2>
  + **Demo2 (without pins):** <https://creative.vdx.tv/#wJqgJA/4>
* **Notes:** Not Applicable
* **Descriptions:** Image gallery synchronized with video timeline for seamless multimedia integration.
* **Keywords:** sync gallery, video timeline, images, multimedia, synchronized
* **Use Case:** Ideal for fashion ads to sync runway footage with still images.

### Product Gallery

* **VDX Standard:** No
* **VDX Advanced:** Yes
* **VDX Custom:** Yes
* **Demo:** <https://creative.vdx.tv/#sLHqsQ/2>
* **Notes:** Developer support needed
* **Descriptions:** Dynamic gallery powered by JSON/API with product details (title, description, pricing) for real-time updates.
* **Keywords:** product gallery, dynamic, JSON, API, real-time
* **Use Case:** Perfect for retail ads with frequently updated product catalogs.

### Bokeh Mode

* **VDX Standard:** Yes
* **VDX Advanced:** Yes
* **VDX Custom:** Yes
* **Demo:** <https://creative.vdx.tv/#SZnlhg/10>
* **Notes:** Not Applicable
* **Descriptions:** Visual effect blurring the background with a video overlay for a cinematic focus.
* **Keywords:** bokeh mode, visual effect, video overlay, cinematic
* **Use Case:** Used in luxury ads to emphasize premium products.

### Long Form Video on Primary Tab

* **VDX Standard:** Yes
* **VDX Advanced:** Yes
* **VDX Custom:** Yes
* **Demo:** <https://creative.vdx.tv/#xNrwXw/28>
* **Notes:** Not Applicable
* **Descriptions:** Extended video playback (>30 sec) with "Click to continue" prompt after 30 sec (max 5 min recommended).
* **Keywords:** long form video, extended playback, primary tab, interactive
* **Use Case:** Great for storytelling ads like brand documentaries.

## Custom Features - Basic Guidelines

### Pre-Sale (Mock) Campaign via Muse

* **US:** Design team + Oliver approval
* **Outside US:** Design team + Oliver approval
* **VDX Standard:** No
* **VDX Advanced:** Yes
* **VDX Custom:** Yes
* **Notes:** Discuss feasibility with CSM; use Muse if approved by Oliver.
* **Timeline:** Custom, based on requirement
* **Descriptions:** Pre-sale mock campaign creation using Muse for rapid prototyping.
* **Keywords:** pre-sale, mock campaign, Muse, prototyping, custom
* **Use Case:** Ideal for pitching creative concepts to clients.

### Nested Gallery/Features

* **US:** CSM
* **Outside US:** SF Approval Request
* **VDX Standard:** No
* **VDX Advanced:** Yes
* **VDX Custom:** Yes
* **Notes:** Combines multiple ad features (e.g., hotspots on gallery).
* **Timeline:** 5–10 business days (tentative)
* **Descriptions:** Nested gallery integrating multiple features for complex ad experiences.
* **Keywords:** nested gallery, combined features, hotspots, complex
* **Use Case:** Used in luxury ads to layer interactive elements.

### GIF Animation & Other Animations

* **US:** SF Approval Request
* **Outside US:** SF Approval Request
* **VDX Standard:** No
* **VDX Advanced:** Yes
* **VDX Custom:** Yes
* **Notes:** Feasibility test required; not available on VDX Standard ET. <https://wiki.exponential.com/display/PRDTMGT/Animation+requests>
* **Timeline:** Based on feasibility
* **Descriptions:** Custom GIF and animation support for visually engaging ads.
* **Keywords:** GIF, animation, custom, visual, engagement
* **Use Case:** Perfect for social media ads requiring eye-catching motion.

### "Video" Creation (Video Solution)

* **US:** Ask CSM
* **Outside US:** CSM
* **VDX Standard:** Yes (Regional)
* **VDX Advanced:** Yes
* **VDX Custom:** Yes
* **Notes:** Custom SLA; SF approval for mock video solutions.
* **Timeline:** +5 days per video
* **Descriptions:** Custom video creation service for tailored ad content.
* **Keywords:** video creation, custom, mock, SLA
* **Use Case:** Useful for campaigns needing bespoke video assets.

### Query String Search

* **US:** Ask CSM
* **Outside US:** SF Approval Request
* **VDX Standard:** No
* **VDX Advanced:** No
* **VDX Custom:** Yes
* **Notes:** Feasibility test required.
* **Timeline:** Custom, AdEx support may be needed
* **Descriptions:** Search functionality using query strings for dynamic ad content.
* **Keywords:** query string, search, dynamic, custom
* **Use Case:** Ideal for searchable product listings in ads.

### Dynamic Location/CityName on Video Tab

* **US:** No Approval
* **Outside US:** No Approval
* **VDX Standard:** No
* **VDX Advanced:** No
* **VDX Custom:** Yes
* **Notes:** Supported only with Muse flow.
* **Timeline:** AdEx support for integration
* **Descriptions:** Dynamic location or city name display on video tab for personalized ads.
* **Keywords:** dynamic location, city name, video tab, personalization
* **Use Case:** Great for local service ads targeting specific regions.

### Dynamic Functionality (including API)

* **US:** CSM
* **Outside US:** CSM ($100K USD+)
* **VDX Standard:** No
* **VDX Advanced:** No
* **VDX Custom:** Yes
* **Notes:** Feasibility check via AdEx JIRA ticket; dual timelines for creative and AdEx setup.
* **Timeline:** Custom, AdEx support required
* **Descriptions:** API-driven dynamic functionality for real-time ad customization.
* **Keywords:** dynamic, API, real-time, custom, functionality
* **Use Case:** Used in e-commerce for live inventory updates.

### Dynamic Listing

* **US:** CSM
* **Outside US:** CSM ($100K USD+)
* **VDX Standard:** No
* **VDX Advanced:** No
* **VDX Custom:** Yes
* **Notes:** Dynamic content display; feasibility check required.
* **Timeline:** Custom, AdEx support required
* **Descriptions:** Dynamic listing feature for displaying real-time content in ads.
* **Keywords:** dynamic listing, real-time, content display, custom
* **Use Case:** Perfect for real estate ads with live property listings.

### Dynamic Hotspot

* **US:** CSM
* **Outside US:** CSM ($100K USD+)
* **VDX Standard:** No
* **VDX Advanced:** No
* **VDX Custom:** Yes
* **Notes:** Interactive hotspot with dynamic content; feasibility check required.
* **Timeline:** Custom, AdEx support required
* **Descriptions:** Dynamic hotspots delivering interactive, real-time content in ads.
* **Keywords:** dynamic hotspot, interactive, real-time, content
* **Use Case:** Useful in travel ads for live tour information.

### Click to Call

* **US:** SF Approval Request
* **Outside US:** SF Approval Request
* **VDX Standard:** No
* **VDX Advanced:** No
* **VDX Custom:** Yes
* **Notes:** Muse flow feasibility test required; not supported in Canvas flow. <https://jira.exponential.com/browse/CREQ-352>
* **Timeline:** Custom, AdEx support if needed
* **Descriptions:** Click-to-call feature for direct user contact within ads.
* **Keywords:** click to call, direct contact, interactive, Muse
* **Use Case:** Ideal for service-based ads like plumbing or insurance.

## Pharma Guidelines

### Pre-Qualifier (Teaser)

* **US:** Pharma Clients Only
* **Outside US:** Pharma Clients Only
* **VDX Standard:** Yes
* **VDX Advanced:** Yes
* **VDX Custom:** Yes
* **Notes:** Sales support material required. <https://creative.vdx.tv/#LxyrOAaWFcPg> (Demo)
* **Timeline:** As per SLA
* **Descriptions:** Teaser pre-qualifier for pharma ads to engage users before main content.
* **Keywords:** pre-qualifier, teaser, pharma, engagement
* **Use Case:** Used in drug ads to introduce benefits before details.

### ISI (Teaser)

* **US:** Pharma Clients Only
* **Outside US:** Pharma Clients Only
* **VDX Standard:** Yes
* **VDX Advanced:** Yes
* **VDX Custom:** Yes
* **Notes:** Sales support material required. <https://creative.vdx.tv/#.NH3ewaWFcPg> (Demo)
* **Timeline:** As per SLA
* **Descriptions:** Important Safety Information teaser for pharma ads to inform users briefly.
* **Keywords:** ISI, teaser, pharma, safety information
* **Use Case:** Great for quick safety disclaimers in pharma campaigns.

### ISI (Main Unit)

* **US:** Pharma Clients Only
* **Outside US:** Pharma Clients Only
* **VDX Standard:** Yes
* **VDX Advanced:** Yes
* **VDX Custom:** Yes
* **Notes:** Sales support material required. <https://creative.vdx.tv/#4X3OaAaWFcPg> (Demo)
* **Timeline:** As per SLA
* **Descriptions:** Full Important Safety Information display in the main ad unit for compliance.
* **Keywords:** ISI, main unit, pharma, safety, compliance
* **Use Case:** Essential for detailed drug safety info in ads.

### ISI (Teaser + Main Unit)

* **US:** Pharma Clients Only
* **Outside US:** Pharma Clients Only
* **VDX Standard:** Yes
* **VDX Advanced:** Yes
* **VDX Custom:** Yes
* **Notes:** Sales support material required. <https://creative.vdx.tv/#MaX2CQaWFcPg/0> (Demo)
* **Timeline:** As per SLA
* **Descriptions:** Combined teaser and main unit ISI for comprehensive pharma safety info.
* **Keywords:** ISI, teaser, main unit, pharma, comprehensive
* **Use Case:** Used in pharma ads for a two-stage safety approach.

### Pre-Qualifier (Teaser) + ISI (Main Unit)

* **US:** Pharma Clients Only
* **Outside US:** SF Approval Request
* **VDX Standard:** Yes
* **VDX Advanced:** Yes
* **VDX Custom:** Yes
* **Notes:** Sales support material required. <https://creative.vdx.tv/#ZKTviAaWFcPg> (Demo)
* **Timeline:** As per SLA
* **Descriptions:** Pre-qualifier teaser paired with full ISI in main unit for pharma ads.
* **Keywords:** pre-qualifier, ISI, teaser, main unit, pharma
* **Use Case:** Ideal for engaging pharma users with safety details.

### Pre-Qualifiers & ISI (Teaser) + ISI (Main Unit)

* **US:** Pharma Clients Only
* **Outside US:** Pharma Clients Only
* **VDX Standard:** Yes
* **VDX Advanced:** Yes
* **VDX Custom:** Yes
* **Notes:** Sales support material required. <https://creative.vdx.tv/#11K6bQaWFcPg/0> (Demo)
* **Timeline:** As per SLA
* **Descriptions:** Comprehensive pre-qualifiers with teaser and main unit ISI for pharma compliance.
* **Keywords:** pre-qualifiers, ISI, teaser, main unit, pharma, compliance
* **Use Case:** Perfect for complex pharma ads requiring full disclosure.

## VDX Products – Exceptions

### Animation in Header (Inframe 300x250)

* **US:** Not Supported
* **Outside US:** Not Supported
* **VDX Standard:** No
* **VDX Advanced:** No
* **VDX Custom:** Yes
* **Notes:** Not recommended.
* **Timeline:** Custom timeline
* **Descriptions:** Animation in header for 300x250 inframe units, available only in custom setups.
* **Keywords:** animation, header, inframe, custom, exception
* **Use Case:** Used in niche campaigns needing header animations.

### Remove Logo in Header from Specific Tab

* **US:** CSM
* **Outside US:** CSM
* **VDX Standard:** Yes
* **VDX Advanced:** Yes
* **VDX Custom:** Yes
* **Notes:**
* **Timeline:** No additional timeline
* **Descriptions:** Option to remove logo from specific tabs for branding flexibility.
* **Keywords:** remove logo, header, tab, branding, flexibility
* **Use Case:** Useful for co-branded ads requiring selective logo display.

### Clicktag on Background/Whole Unit (Video Tab)

* **US:** SF Approval Request
* **Outside US:** SF Approval Request
* **VDX Standard:** Yes
* **VDX Advanced:** Yes
* **VDX Custom:** Yes
* **Notes:** Not recommended due to false clicks.
* **Timeline:** No additional timeline
* **Descriptions:** Clickable background or whole unit on video tab for broader interactivity.
* **Keywords:** clicktag, background, video tab, interactivity, exception
* **Use Case:** Used in high-engagement campaigns despite click risks.

### Separate Teaser Video

* **US:** SF Approval Request
* **Outside US:** SF Approval Request
* **VDX Standard:** No
* **VDX Advanced:** No
* **VDX Custom:** Yes
* **Notes:** Muse flow only.
* **Timeline:** +1 day + AdEx support
* **Descriptions:** Standalone teaser video separate from main content for custom campaigns.
* **Keywords:** teaser video, separate, Muse, custom, exception
* **Use Case:** Ideal for phased campaign rollouts.

### Remove Sizes from Showcase

* **US:** CSM
* **Outside US:** CSM
* **VDX Standard:** Yes
* **VDX Advanced:** Yes
* **VDX Custom:** Yes
* **Notes:** Sizes hidden in showcase but delivered live.
* **Timeline:** No additional timeline
* **Descriptions:** Option to hide size options in showcase while maintaining delivery.
* **Keywords:** remove sizes, showcase, hidden, delivery, exception
* **Use Case:** Used in simplified ad previews for clients.

## VDX Studio - Available Components/Features

### Standard Image Gallery

* **Limitation:** No separate CTA per frame.
* **Descriptions:** Basic image gallery for simple visual displays in studio projects.
* **Keywords:** standard image gallery, visual, simple, studio
* **Use Case:** Suitable for quick studio mockups.

### Location Map

* **Limitation:** None
* **Descriptions:** Fully functional location map for studio-based ad designs.
* **Keywords:** location map, studio, geolocation, functional
* **Use Case:** Used in studio prototypes for local ads.

### Hybrid Gallery

* **Limitation:**
  + No nested content.
  + No separate CTA per frame (feasible via RM Gallery).
* **Descriptions:** Video-inclusive gallery for studio projects with limited nesting.
* **Keywords:** hybrid gallery, video, studio, limited nesting
* **Use Case:** Ideal for studio video showcases.

### Comparison Slider

* **Limitation:**
  + No image-to-video comparison.
  + No nested content.
  + No separate CTA per frame.
* **Descriptions:** Basic comparison slider for studio designs with static content.
* **Keywords:** comparison slider, studio, static, basic
* **Use Case:** Used in studio A/B testing visuals.

### Long-Form

* **Limitation:** None
* **Descriptions:** Extended content support for long-form studio ad units.
* **Keywords:** long-form, studio, extended, content
* **Use Case:** Perfect for detailed studio narratives.

### Bokeh

* **Limitation:** None
* **Descriptions:** Bokeh effect for studio ads to enhance visual appeal.
* **Keywords:** bokeh, studio, visual effect, appeal
* **Use Case:** Used in studio luxury ad prototypes.

### Pharma

* **Limitation:**
  + ISI (Main Unit) + Pre-Qualifiers (Teaser)
  + ISI + Pre-Qualifiers (Teaser)
  + ISI (Main Unit only)
* **Descriptions:** Pharma-compliant features for studio ads with safety info options.
* **Keywords:** pharma, studio, ISI, pre-qualifiers, compliance
* **Use Case:** Ideal for studio pharma campaign drafts.

### Rich Media Gallery

* **Limitation:** No nested content.
* **Descriptions:** Rich media gallery for studio projects with multimedia support.
* **Keywords:** rich media, gallery, studio, multimedia
* **Use Case:** Used in studio interactive ad designs.

### Color Picker

* **Limitation:**
  + No nested content.
  + No arrows.
  + No transition controls.
* **Descriptions:** Simple color picker for studio ads with basic customization.
* **Keywords:** color picker, studio, customization, simple
* **Use Case:** Suitable for studio product color previews.

### Hotspot

* **Limitation:**
  + No nested content.
  + Single fade effect only.
  + No video in popup.
* **Descriptions:** Basic hotspot feature for studio ads with simple interactivity.
* **Keywords:** hotspot, studio, interactivity, basic, fade
* **Use Case:** Used in studio prototypes for feature highlights.