# 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...