The NEXTGEN TV Web Platform

Run3TV Advertising Implementation Guidelines: SIMID Creatives

Version: 0.4 (Pre-release)

Date: May 31st 2024

Doc: R3TV-IG-0512

Framework version: 2.3

This document is a pre-release version, intended to gather early feedback.
Please be aware that the information presented here is not final and may be revised in subsequent releases.

Revision history

Version	Date	Framework version	Update
v0.2 (Pre-release)	February 23rd 2024	Version 2.3	First Pre-release
v0.4 (Pre-release)	**May 31st 2024**	**Version 2.3**	Correction to .onSIMIDEvent() definition Correction to .simid.sendMessage() definition Document renamed to “Run3TV Advertising Implementation Guidelines: SIMID Creatives”

Copyright notice

Contents

1. Glossary

Glossary of unfamiliar words and acronyms.

Term	Definition
AMP	Application Media Player (ATSC 3.0)
RMP	Receiver Media Player (ATSC 3.0)

2. References

ID	Publisher	Document
A/344	ATSC	ATSC Standard: "ATSC 3.0 Interactive Content" *(The applicable version(s) of this document are dependent upon the SIMID Creative's markets' receiver base)*
SIMID	IAB	Secure Interactive Media Interface Definition (SIMID) v1.1.0 https://interactiveadvertisingbureau.github.io/SIMID/simid-1.1.0.html
Google IMA	Google	https://developers.google.com/interactive-media-ads/docs/sdks/html5/client-side/architecture
R3TV-IG-0203	Run3TV	Run3TV Implementation Guidelines: Run3TV SDK APIs
R3TV-IG-0212	Run3TV	Run3TV Implementation Guidelines: I-Frame Sub-Application integration
R3TV-IG-0221	Run3TV	Run3TV Implementation Guidelines: Q-Bar Application

3. Introduction

VOD content can be configured to include addressable pre-roll advertisements within the Run3TV Framework. These pre-roll advertisements can include interactive elements known as **SIMID Creative**[1] applications. These applications are launched alongside the advertisement's video and terminated either when the advertisement ends or when the viewer stops interacting with them, whichever is later.

SIMID Creatives are associated with adverts using VAST signaling that includes a `<InteractiveCreativeFile>` element. For more information on signaling SIMID Creatives, see Associating SIMID Creatives To VOD Content.

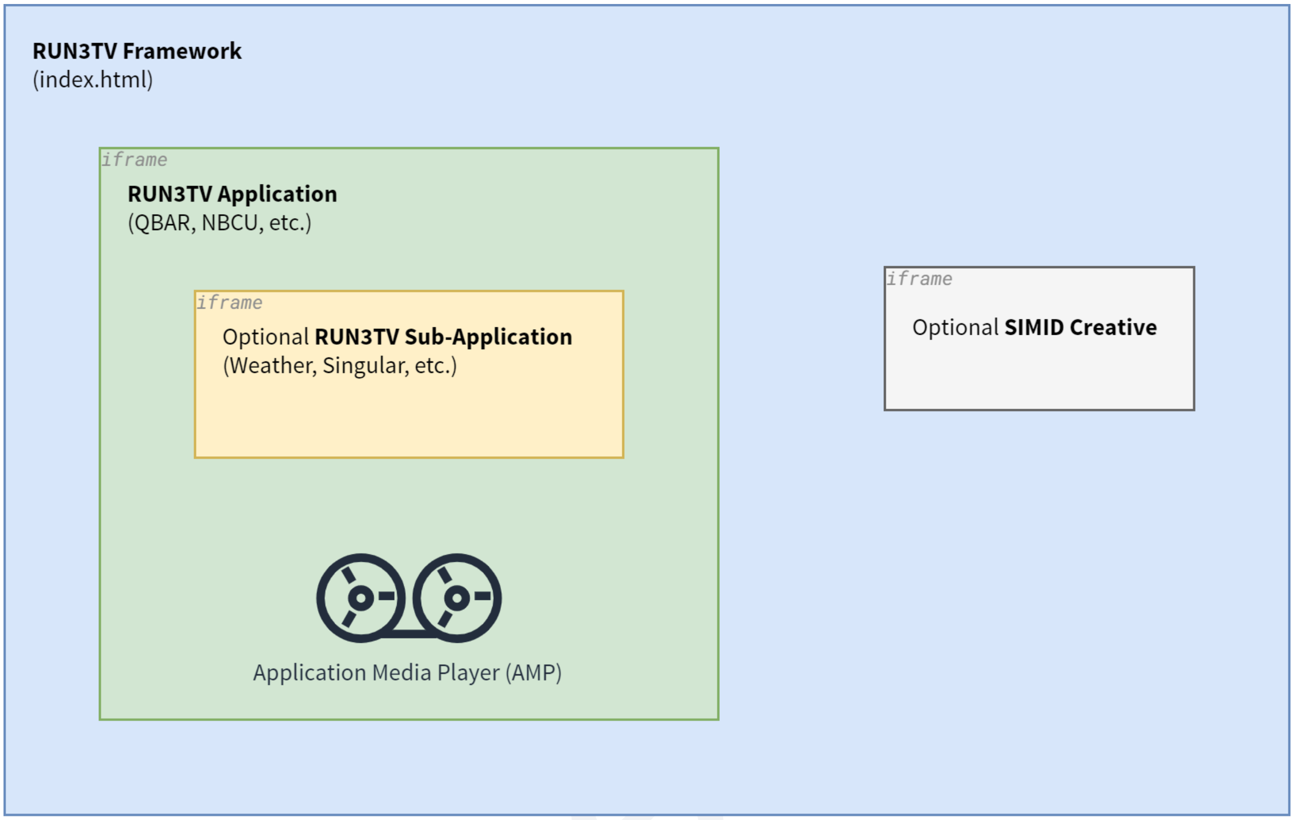
The Framework launches SIMID Creatives into a separate iframe from the Run3TV Application and manages their application lifecycle.

Developers implementing SIMID Creatives can access SIMID events, lifecycle announcements and certain keypresses by using the `a3fa-framework.ad-overlay.min.js` API.

Note	Run3TV Sub-Applications share a similar interface and application lifecycle to SIMID Creatives. For more details see [R3TV-IG-0212].

[1] Secure Interactive Media Interface Definition

This hierarchy of windows within the Framework is summarized in the diagram below.



The descriptions throughout this document use the Q-Bar application as an example Run3TV Application that runs alongside SIMID Creatives, however the Framework can present SIMID Creatives alongside any application.

3.1. Example SIMID Creatives In The Starter Kit

The Run3TV Starter Kit provides the following example SIMID Creative:

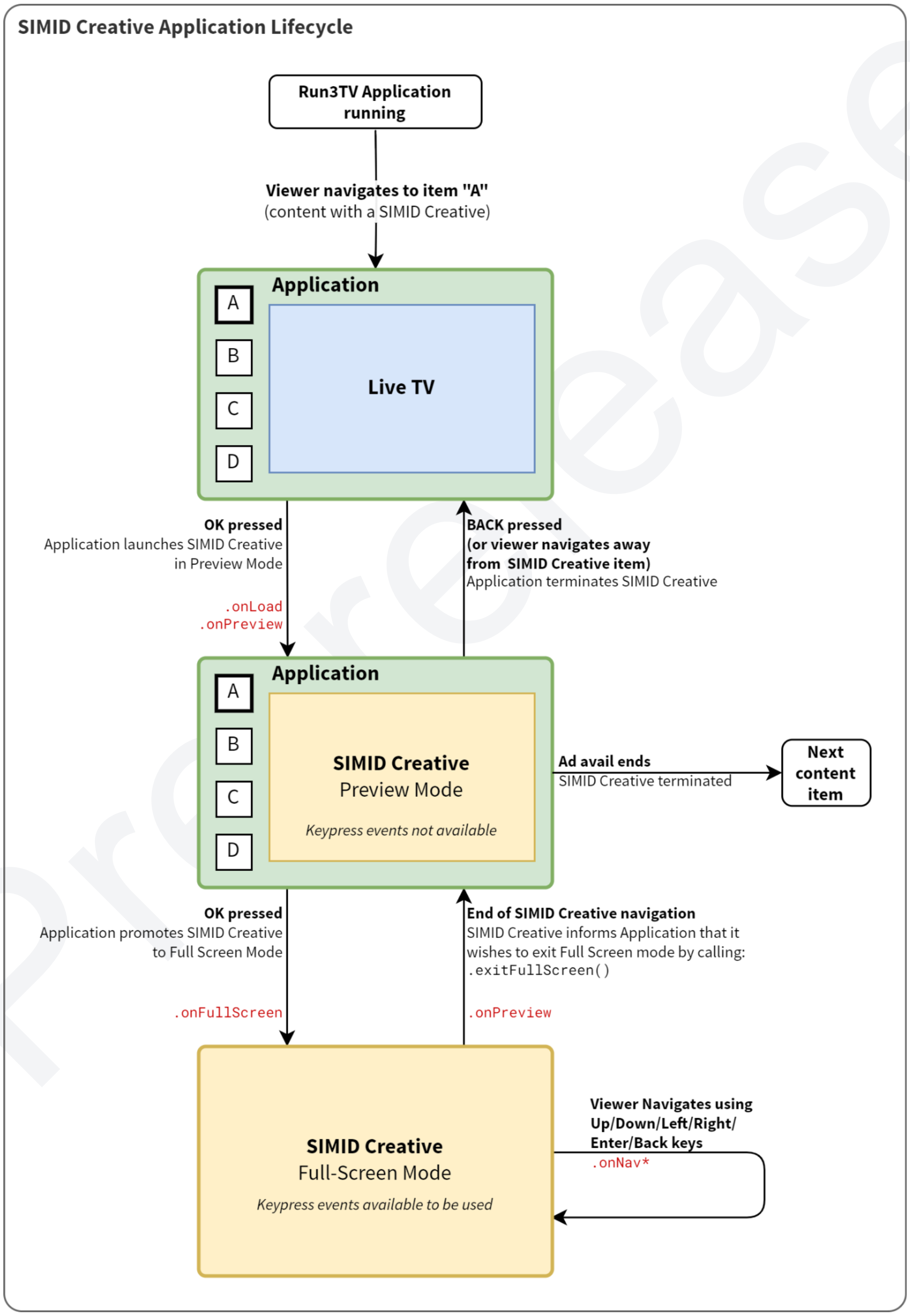
Starter Kit Location	Description
`/london/apps/simid-creative`	An example SIMID Creative demonstrating the application lifecycle. The VAST response that references this Creative is hosted

This is a complete, worked example of a SIMID Creative, and are described in more detail in Annex A - Example SIMID Creative.

4. SIMID Creatives Application Lifecycle

This section describes the application lifecycle of Run3TV SIMID Creatives.

The diagram below provides a high level overview of this lifecycle:


SIMID Creative Application Lifecycle
Run3TV Application running
Viewer navigates to item "A"
(content with a SIMID Creative)
Application
A
B
C
D
Live TV
OK pressed
Application launches SIMID Creative in Preview Mode
.onLoad
.onPreview
BACK pressed
(or viewer navigates away from SIMID Creative item)
Application terminates SIMID Creative
Application
A
B
C
D
SIMID Creative
Preview Mode
Keypress events not available
Ad avail ends
SIMID Creative terminated
Next content item
OK pressed
Application promotes SIMID Creative to Full Screen Mode
.onFullScreen
End of SIMID Creative navigation
SIMID Creative informs Application that it wishes to exit Full Screen mode by calling:
.exitFullScreen()
.onPreview
SIMID Creative
Full-Screen Mode
Keypress events available to be used
Viewer Navigates using Up/Down/Left/Right/Enter/Back keys
.onNav*

SIMID Creatives can be presented either of two modes:

- **Preview Mode** - Presentation is scaled down (with the same 16:9 aspect ratio as full screen content) to fit within the frame of the Run3TV Application. No key events are passed on to the SIMID Creative
- **Full Screen Mode** - Presentation is scaled to fill the screen. SIMID Creatives receive events for certain keys from the Run3TV Application.

The Framework and SIMID Creatives inform each other of lifecycle changes and SIMID events via the `a3fa-framework.ad-overlay.min.js` interface.

In addition, when the SIMID Creative is in Full Screen Mode, the same interface informs the SIMID Creative of certain keypresses.

4.1. SIMID Creative Modes And a3fa-framework.ad-overlay.min.js Events And Functions

The table below summarizes which `a3fa-framework.ad-overlay.min.js` functions and events are applicable for each SIMID Creative Mode.

Run3TV provides a local version of `a3fa-framework.ad-overlay.min.js` for development. This must **not** be used in the production environment.

⚠ Please contact your Run3TV representative to gain access to the hosted production version of this API.

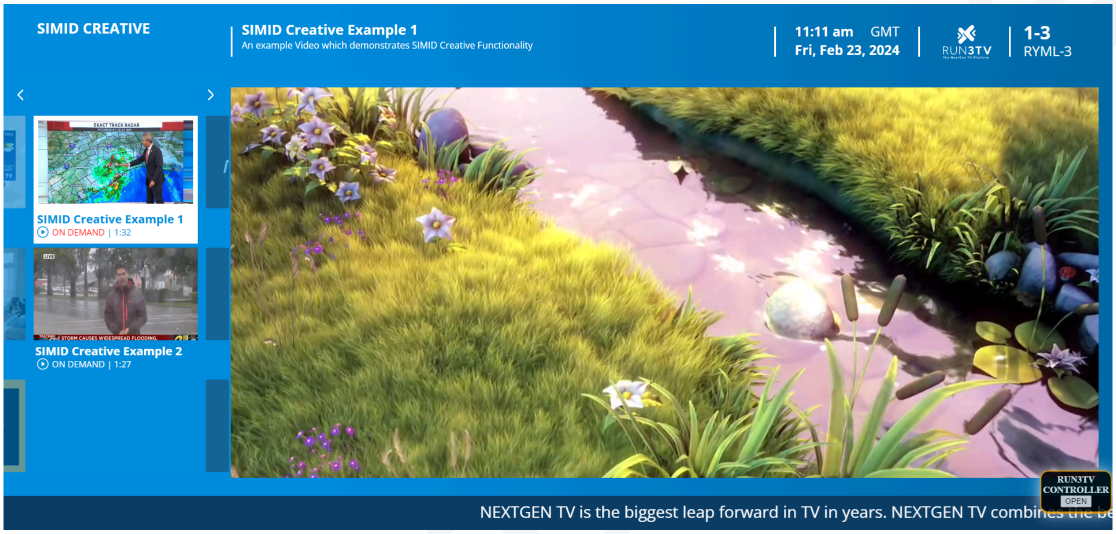
				Applicable in	
	`a3fa-framework.ad-overlay.min.js` **Events & Functions**	**Description**	**Defined by**	**Preview Mode**	**Full Screen Mode**
Lifecycle	`.onload()`	Fired when the SIMID Creative is first loaded	[R3TV-IG-0212]	✔	—
	`.onPreview()`	Fired when the Run3TV Application moves the SIMID Creative to Preview Mode	[R3TV-IG-0212]	✔ *(on first start)*	✔
	`.onFullScreen()`	Fired when the Run3TV Application moves the SIMID Creative to Full Screen Mode	[R3TV-IG-0212]	✔	—
	`.exitFullScreen()`	The SIMID Creative must call this function to inform the Run3TV Application to exit Full Screen Mode	[R3TV-IG-0212]	—	✔
	`.isFullScreen()`	Returns `true` if theSIMID Creative is in Full Screen Mode, otherwise returns `false`	[R3TV-IG-0212]	✔	✔
	`.getPreviewWindowScale()`	Returns the position and scale factor of the Preview Window	[R3TV-IG-0212]	✔	✔
	`.simid.sendMessage()`	The SIMID Creative can use this function to send SIMID messages. For more details, see SIMID Events And Messages	This document	✔	✔
	`.onSIMIDEvent()`	Fired when a new SIMID event is received. For more details, see SIMID Events And Messages	This document	✔	✔
	`.onIMAEvent()`	Fired when a new Interactive Media Ads (IMA) event is received. For more details see IMA Events	This document	✔	✔

				Applicable in	
	`a3fa-framework.ad-overlay.min.js` **Events & Functions**	**Description**	**Defined by**	**Preview Mode**	**Full Screen Mode**
Navigation	`.onNavNext()`	Fired when the viewer presses **RIGHT**	[R3TV-IG-0212]	—	✔
	`.onNavBack()`	Fired when the viewer presses **LEFT**	[R3TV-IG-0212]	—	✔
	`.onNavUp()`	Fired when the viewer presses **UP**	[R3TV-IG-0212]	—	✔
	`.onNavDown()`	Fired when the viewer presses **DOWN**	[R3TV-IG-0212]	—	✔
	`.onNavEnter()`	Fired when the viewer presses **OK** or **SELECT**	[R3TV-IG-0212]	—	✔
	`.onNavExit()`	Fired when the viewer presses **BACK**	[R3TV-IG-0212]	—	✔

4.2. Prior To Launch

Viewers can navigate within the Run3TV Application to highlight VOD content. When highlighted (but not selected), the item thumbnails are presented — the underlying content does not automatically start to render.

An example of a VOD item that has been highlighted but not selected is shown below:



SIMID CREATIVE

SIMID Creative Example 1
An example Video which demonstrates SIMID Creative Functionality

11:11 am GMT
Fri, Feb 23, 2024

1-3
RYML-3

SIMID Creative Example 1
ON DEMAND | 1:32

SIMID Creative Example 2
ON DEMAND | 1:27

NEXTGEN TV is the biggest leap forward in TV in years. NEXTGEN TV combines the be

VOD assets (and their associated pre-roll ads) are not started until the viewer selects them from the on-screen menu by pressing **OK** or **SELECT**.

4.3. Launching And Preview Mode

When a viewer selects a content rail item containing VOD content, the Framework uses its configuration to generate a playlist to present. If the content rail configuration contains a reference to a VAST server, it will query the server for ads suitable to present in advance of the content. For information on this configuration, see Associating SIMID Creatives To VOD Content.

If the VAST Ad configuration returned contains a valid reference to a SIMID Creative, the Run3TV Application will request that the Framework launches this SIMID Creative alongside the ad video.

4.3.1. Run3TV Application ⇄ Framework Communication

This section is informative for SIMID Creative developers.

On the request of the Run3TV Application, the Framework launches the SIMID Creative and informs the Run3TV Application that it has been started by calling `fmw.onPreview()`, as defined in [R3TV-IG-0203].

4.3.2. Framework ⇄ SIMID Creative Communication

SIMID Creatives are first launched into **Preview Mode**. This mode places the contents of the Creative within the existing frame of the Run3TV Application's scaled video window.

Therefore, the SIMID Creative will receive two events from the `a3fa-framework.ad-overlay.min.js` interface when first launching:

- `.onLoad()`
- `.onPreview()`

Note All example SIMID Creatives provided in the Starter Kit follow the application lifecycle detailed here. See run3tv-SIMID-Creative-basic for a minimal example.

When in Preview Mode, the SIMID Creative is presented in the same aspect ratio (16:9) as when in Full Screen Mode.

For details on how to determine the presented size of the SIMID Creative, see Window size, position and scaling in Preview Mode.

Whilst in Preview Mode, SIMID Creatives will **not** receive navigation key events from the `a3fa-framework.ad-overlay.min.js` interface. Therefore, SIMID Creatives in Preview Mode shall not present navigation buttons (other than a call to action) or on-screen media controls.

4.4. Full Screen Mode

When the viewer clicks **OK** whilst the SIMID Creative is in Preview Mode, the Framework promotes it to Full Screen Mode.

The Framework informs the SIMID Creative of the promotion to Full Screen Mode by firing the `.onFullScreen()` event from the `a3fa-framework.ad-overlay.min.js` interface.

Whilst in Full Screen Mode, the following key events will be passed to the SIMID Creative via the `a3fa-framework.ad-overlay.min.js` interface:

Key	**`a3fa-framework.ad-overlay.min.js` event**
Right	`.onNavNext()`
Left	`.onNavBack()`
Up	`.onNavUp()`
Down	`.onNavDown()`
OK or **SELECT**	`.onNavEnter()`
BACK	`.onNavExit()` *SIMID Creatives must always capture this event and call `.exitFullScreen()` as appropriate*

Note SIMID Creatives do not receive focus, so cannot capture key events directly.

The Run3TV Application will inform the SIMID Creative that it has been reverted from Full Screen Mode to Preview Mode by sending the `.onPreview()` event from `a3fa-framework.ad-overlay.min.js`.

Note The example SIMID-Creative (available in the Starter Kit) provides a worked example of how to make use of navigation events provided by `a3fa-framework.ad-overlay.min.js`.

The SIMID Creative is responsible for exiting Full Screen mode. To do so, it **must** call `.exitFullScreen()` when the viewer wishes to return back to the Run3TV Application. This must occur when the viewer presses the **BACK** key to complete navigation within the SIMID Creative. In other words, all Creatives must always register for `.onNavExit()` events and call `.exitFullScreen()` as appropriate.

Depending on the intended user experience of the SIMID Creative, it may be appropriate to call `.exitFullScreen()` at other times.

The SIMID Creative must remove any navigation controls and on-screen media playback controls when it returns to Preview Mode.

Please refer to SIMID Creative Development for more details on viewer navigation within SIMID Creatives.

4.5. Termination

The Framework will terminate the SIMID Creative in the following situations:

- When the viewer chooses to navigate away from the advertisement,
- When the SIMID Creative is in Preview Mode and its advertisement ends playback,
- When the SIMID Creative leaves Full Screen Mode after its advertisement has already ended playback
- When the SIMID Creative sends any of the following SIMID Messages:
    - `SIMID:Creative:requestStop`
    - `SIMID:Creative:fatalError`

5. SIMID Events And Messages

SIMID Events and Messages, as defined in §4 of [SIMID] provide a means for standardized communication between the Framework and the SIMID Creative. This interface enables the SIMID Creative to (amongst other abilities) control Ad video playback and monitor the state of that playback.

SIMID Creatives can use the `a3fa-framework.ad-overlay.min.js` interface to:

- Send SIMID messages using the `.simid.sendMessage()` function
- Register to receive SIMID events using the `.onSIMIDEvent()` event.

The `a3fa-framework.ad-overlay.min.js` interface is defined in detail in Interface API For SIMID Creatives.

5.1. Receiving (Event) Messages

The `a3fa-framework.ad-overlay.min.js` interface will inform the SIMID Creative of all SIMID messages via the `.onSIMIDEvent()` event.

5.2. Sending SIMID Messages

The SIMID Creative can send SIMID messages to the Player as defined by §4.4 of [SIMID] and profiled by the table below.

SIMID Message	**Permitted to be sent by the SIMID Creative**	**Comments**
`SIMID:Creative:clickThru`	✔	
`SIMID:Creative:fatalError`	✔	
`SIMID:Creative:getMediaState`	✔	
`SIMID:Creative:log`	✔	
`SIMID:Creative:reportTracking`	✔	
`SIMID:Creative:requestChangeAdDuration`	✔	
`SIMID:Creative:requestChangeVolume`	✔	
`SIMID:Creative:requestFullscreen`	—	Not applicable. The Framework will promote the SIMID Creative to Full Screen Mode
`SIMID:Creative:requestExitFullscreen`	—	SIMID Creatives shall use `.exitFullScreen()` to request a return to Preview Mode
`SIMID:Creative:requestNavigation`	—	Not applicable. The Framework will grant access to navigation key events whilst the SIMID Creative is in Full Screen Mode
`SIMID:Creative:requestPause`	✔	
`SIMID:Creative:requestPlay`	✔	
`SIMID:Creative:requestResize`	—	Not applicable. The Framework will resize the SIMID Creative when moving between Preview Mode and Full Screen Mode
`SIMID:Creative:requestSkip`	✔	
`SIMID:Creative:requestStop`	✔	

6. IMA Events

IMA (Interactive Media Ads) events, as defined in [Google IMA] provide a means for standardized communication between the Framework and the Ad Server.

SIMID Creatives can use the `a3fa-framework.ad-overlay.min.js` interface to receive IMA events.

The following IMA events will be passed on to the SIMID Creative:

- `AD_BREAK_READY`
- `AD_BUFFERING`
- `AD_METADATA`
- `AD_PROGRESS`
- `ALL_ADS_COMPLETED`
- `CLICK`
- `CONTENT_PAUSE_REQUESTED`
- `CONTENT_RESUME_REQUESTED`
- `DURATION_CHANGE`
- `FIRST_QUARTILE`
- `IMPRESSION`
- `INTERACTION`
- `LINEAR_CHANGED`
- `LOADED`
- `MIDPOINT`
- `PAUSED`
- `RESUMED`
- `SKIPPABLE_STATE_CHANGED`
- `SKIPPED`
- `THIRD_QUARTILE`
- `USER_CLOSE`
- `VIDEO_CLICKED`
- `VIDEO_ICON_CLICKED`
- `VOLUME_CHANGED`
- `VOLUME_MUTED`

The truncated example below prints all IMA events that arrive to the console:

IMA event example code snippet

```
<!DOCTYPE html>
<html>
  <head>
    <title>Overlay</title>
    <meta charset="utf-8" />
    <script src="a3fa-framework.ad-overlay.min.js"></script>
  </head>
  <body>
    <script>
      // ...
      overlay.onIMAEvent(function (name) {
        console.log("overlay onIMAEvent()", name);
      });
      // ...
    </script>
  </body>
</html>
```

The `a3fa-framework.ad-overlay.min.js` interface is defined in detail in Interface API For SIMID Creatives.

7. Associating Ads Containing SIMID Creatives To VOD Content

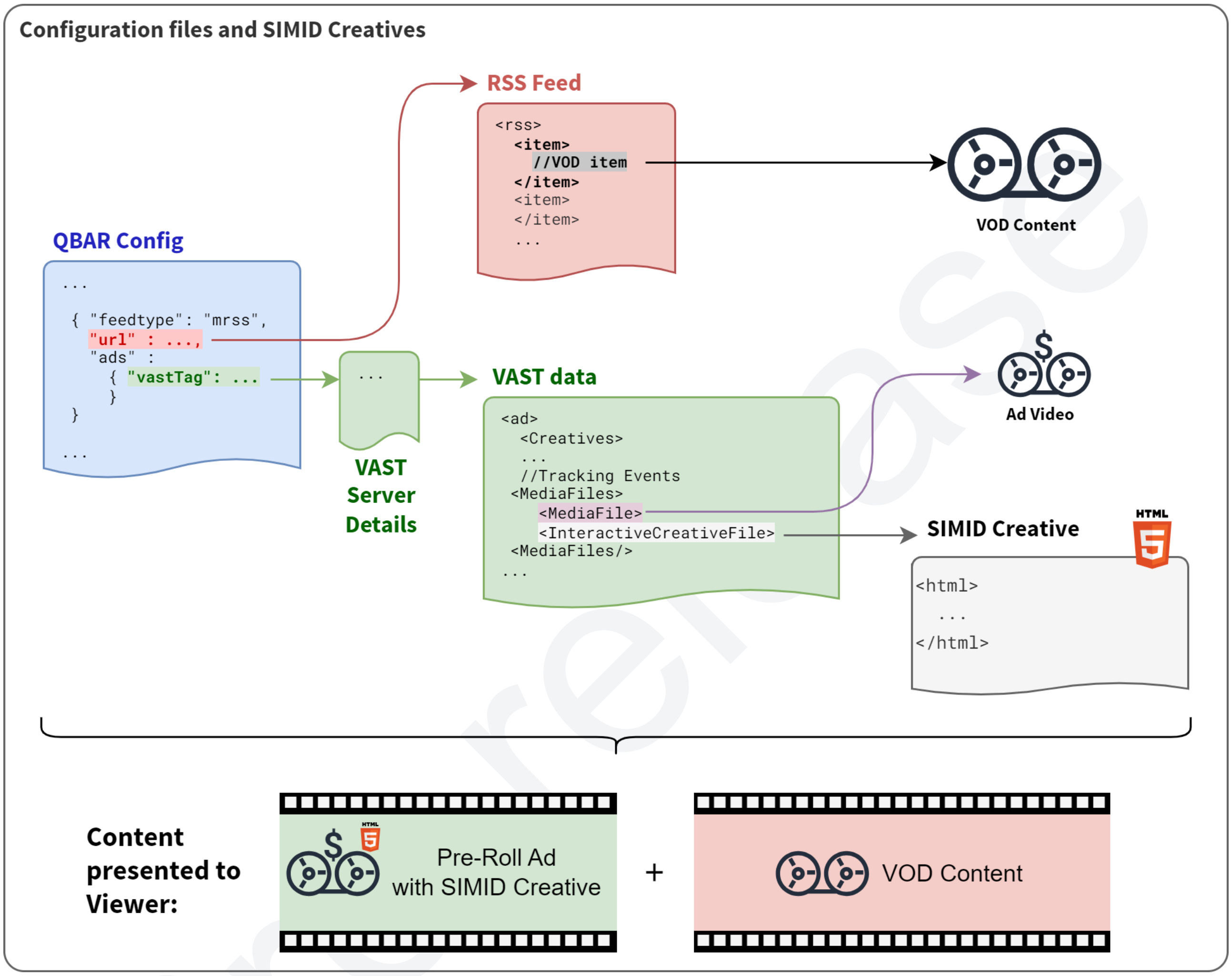
This section describes at a high level how to associate a SIMID Creative to a pre-roll advertisement.

Note Addressable ad insertion is inherently a complex process. This document does not describe the complete process of addressing individual adverts to individual receivers.

To present a SIMID Creative over a pre-roll advertisement, the following configuration and infrastructure must be in place:

- The Content Rail of the application must be configured to insert pre-roll advertisements.
- A VAST server must be in place, configured to include the correct <InteractiveCreativeFile> element for the advertisement

The diagram below summarizes this configuration, along with the configuration required to present the VOD content itself, using the Q-Bar as an example.



7.1. Q-Bar Configuration

The Q-Bar allows enabling and disabling of pre-roll advertisements for VOD content at a Content Rail level.

Any Content Rail that includes a well-configured `ads` element in the Q-Bar's `config.json` file will request pre-roll advertisements from the configured VAST server. An example of this configuration is shown below:

Q-Bar configuration example (Two content rails: The first with Pre-roll ads, the second without)

```
1    ...
2    {
3      "feeds": [
4        {
5          "feedType": "mrss",
6          "title": "VOD with Pre-Roll Advertisements",
7          "url": "https://example.com/VOD_rail1_withPreRollAds.rss",
8          "ads": {
9            "vastTag": "https://example.com/VAST/linear.xml?platform=run3tv"
10         }
11       },
12       {
13         "feedType": "mrss",
14         "title": "VOD without Pre-Roll Advertisements",
15         "url": "https://example.com/VOD_rail2_noAds.rss"
16       }
17     ]
18   }
19   ...
```

In this example, two content rails are defined:

- The first (starting at **line 4**) includes an `ads` field containing a `vastTag` field. When viewers select VOD content from this rail, the Q-Bar application will request suitable advertisements via the details provided by the VAST server.
- The second (starting at **line 12**) does not contain an ads field. When viewers select VOD content from this rail, the Q-Bar application plays back the VOD asset immediately, without requesting advertisements.

The URL in the `vastTag` field should include a means to inform the VAST server that SIMID Creatives will be launched within the Run3TV environment. In the example above, this is achieved by including the `platform=run3tv` query string.

The sample Q-Bar application provided within the Starter Kit includes a content rail configured to present pre-roll advertisements.

For more information about the Q-Bar application and its configuration, please see [R3TV-IG-0221].

7.2. RSS Configuration

The RSS format used by the Framework to define content rails is detailed in [R3TV-IG-0231]. This format permits each item within a content rail to be one of a set of Content Types.

The Framework will insert pre-roll advertisements in advance of content items as shown below (presuming the content rail is configured to do so. For more information see Q-Bar Configuration):

RSS Content Type	Pre-roll ads can be inserted?
Live TV Service	—
Live Radio Service	—
Current Live TV Program	—
Next program on Live TV	—
Current Live Radio Program	—
Next Radio content live	—
Live Streaming Service	—
Live Radio	—
Video on Demand	✔
On-Demand Audio	✔
Imagery	✔
Sub-Applications	—
Application Links	—
Animated Content	—
Advertisement	—
Promotional Video	—
Alerts	—

7.3. VAST Configuration

To associate a SIMID Creative with an advertisement, the VAST response for that advertisement must include an `<InteractiveCreativeFile>` element. An example VAST response is shown below.

VAST-defined advertisement with SIMID Creative

```
1    <VAST xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
2          xsi:noNamespaceSchemaLocation="vast.xsd" version="4.2">
3      <Ad id="1234567">
4        <InLine>
5          <AdSystem>AdSystem</AdSystem>
6          <AdTitle>Linear SIMID Example</AdTitle>
7          <Description>SIMID Linear Video Ad</Description>
8          <Error>https://example.com/ads/err</Error>
9          <Impression>https://example.com/ads/imp</Impression>
10         <Creatives>
11           <Creative sequence="1">
12             <Linear>
13               <Duration>00:00:30</Duration>
14               <TrackingEvents>
15                 <Tracking event="creativeView">https://example.com/a</Tracking>
16                 <Tracking event="start">https://example.com/b</Tracking>
17                 <Tracking event="midpoint">https://example.com/d</Tracking>
18                 <Tracking event="complete">https://example.com/f</Tracking>
19               </TrackingEvents>
20               <VideoClicks>
21                 <ClickTracking id="adSystem">
22                   <![CDATA[https://example.com/ads/ct]]>
23                 </ClickTracking>
24               </VideoClicks>
25               <MediaFiles>
26                 <MediaFile delivery="progressive"
27                   type="video/mp4">https://example.com/video.mp4</MediaFile>
28                 <InteractiveCreativeFile type="text/html"
29   apiFramework="SIMID">https://example.com/creative.html?platform=run3tv</InteractiveCreativeFile>
30               </MediaFiles>
31             </Linear>
32           </Creative>
33         </Creatives>
34       </InLine>
35     </Ad>
36   </VAST>
```

In this example, the `<MediaFiles>` element (starting at line 25) includes an `<InteractiveCreativeFile>` element (lines 28 & 29).

The `<InteractiveCreativeFile>` element informs the Framework of the location of the SIMID Creative.

If the SIMID Creative is to be run in both Run3TV and other environments, the URL provided in the `<InteractiveCreativeFile>` element should include a means to inform the Creative of the environment that it is to be run in. In the example above, this is achieved with the inclusion of the `platform=run3tv` query string.

7.4. Profiling of VAST <InteractiveCreativeFile>

Any number of `<InteractiveCreativeFile>` elements can be provided within the `<MediaFiles>` element.

The Framework will select and attempt to launch the SIMID Creative URI defined within the first `<InteractiveCreativeFile>` element that meets all of the following criteria:

`<InteractiveCreativeFile>` Field	**Permitted Value**
`type`	`text/html`
`apiFramework`	*`SIMID`*

If this SIMID Creative fails to launch, the Framework will not attempt to load any Creatives defined within later `<InteractiveCreativeFile>` elements. The ad video will still play.

The content of `<InteractiveCreativeFile>` must be the URI to the SIMID Creative `.html` file. This URI can optionally be wrapped as CDATA.

8. SIMID Creative Development

This section provides a summary of the guidelines (and guidance) for SIMID Creative development.

8.1. Window size, position and scaling in Preview Mode

Whilst SIMID Creatives are always presented in a 16:9 aspect ratio, they may not always fill the screen. Depending on the graphical design of the SIMID Creative, it may be appropriate to adjust how the GUI is presented depending on the window size.

The window size can be determined using the following function from the `a3fa-framework.ad-overlay.min.js` interface:

```
.getPreviewWindowScale()
```

This function will always return the size, position and scaling factor of the Preview Mode window, even when the SIMID Creative is in Full Screen Mode.

8.2. Z-Order

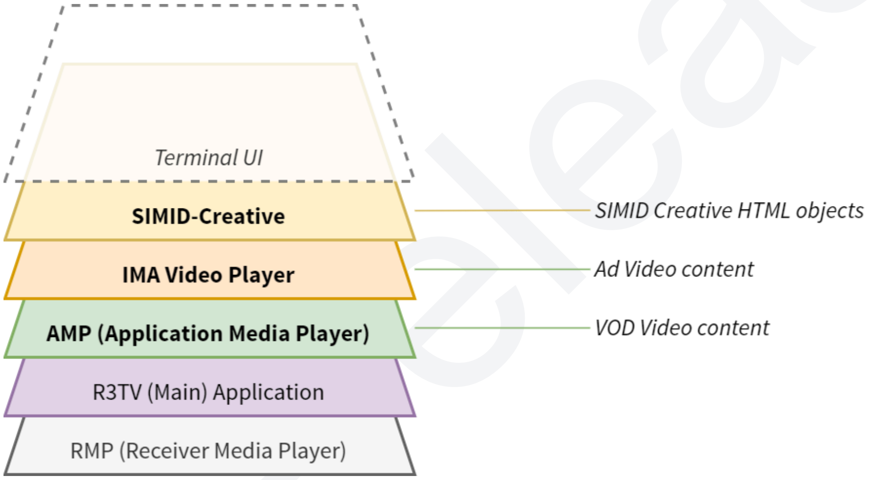
On-screen assets created by SIMID Creatives are placed directly on top of advertisement video.

Advertisement video is rendered by the IMA Video Player.

When switching between modes, the IMA Video Player will continue to play the audio/video.

The Framework will stop the IMA Video Player when the end of the advert is reached, or when the SIMID Creative sends a `SIMID:Creative:requestStop` SIMID message.

The Z-order of the various layers and media players is summarized below. The SIMID Creative cannot modify this order.



Ad video will only be visible if the SIMID Creative provides “open” space to allow the video to cut through the Creative’s own Z-Order layer.

It is the responsibility of the SIMID Creative to present any required video player controls.

8.3. Ad Video Control

The SIMID Creative can programmatically control ad playback. This is achieved by sending the appropriate SIMID message using the `.simid.sendMessage()` function from the `a3fa-framework.ad-overlay.min.js` interface.

For example:

SIMID Message	Action
`SIMID:Creative:requestPause`	Requests the IMA Video Player to pause media playback
`SIMID:Creative:requestPlay`	Requests the IMA Video Player to resume media playback
`SIMID:Creative:requestStop`	Requests the IMA Video Player to stop media playback
`SIMID:Creative:requestSkip`	Requests the IMA Video Player skip the current ad

See SIMID Events And Messages for a complete list of permitted SIMID messages.

8.4. Navigation

SIMID Creatives cannot capture keypresses directly. Instead, the navigation event API provided by the `a3fa-framework.ad-overlay.min.js` interface shall be used to detect keypresses.

SIMID Creatives must be designed to make use of these navigation events and update their GUI as appropriate so that the viewer can easily navigate within the Creative.

Browser based focusable navigation is not supported within SIMID Creatives.

The SIMID Creative user interface must be designed in such a way that the viewer can exit Full Screen Mode (and return control to the Run3TV Application) by pressing the **BACK** key. Other keys may also be used to enable the viewer to exit the Creative, so long as their function is made clear.

9. SIMID Creative Development Environment

This section will follow in a future version of this document.

10. Interface API For SIMID Creatives

This section provides a description of the `a3fa-framework.ad-overlay.min.js` interface API.

This API shares some functions and events with the Sub-App interface `subApp.run3tv.min.js`. To avoid duplication, these are not defined in the current document; please refer to [R3TV-IG-0212] for details on the following events and functions:

- `.onload()`
- `.onPreview()`
- `.onFullScreen()`
- `.onNavNext()`
- `.onNavBack()`
- `.onNavUp()`
- `.onNavDown()`
- `.onNavEnter()`
- `.onNavExit()`
- `.exitFullScreen()`
- `.isFullScreen()`
- `.getPreviewWindowScale()`

10.1. .simid.sendMessage()

The`.simid.sendMessage()` function is called by the SIMID Creative when it needs to send SIMID messages to the player.

Syntax

```
overlay.simid.sendMessage(simidMessage,
                       callbackFunction(data object, messageEvent <object>), //Optional
                       args{opt object});
```

```
overlay.simid.sendMessage(simidMessage,
                          callbackFunction(data, message), //Optional
                          args                             //Optional
                          );
```

Parameters

`simidMessage <String>`

A SIMID message, as defined by [SIMID] and profiled by this document.

For more details, see SIMID Events And Messages

`callbackFn` ***`Optional`***

A function to execute containing the SIMID response message. If this callback function is not provided, the SIMID response message will instead be fired via `.onSIMIDEvent()`.

This function is called with the following parameters:

> **`data <Object>`**
>
> The SIMID message
>
> **`messageEvent <MessageEvent>`**
>
> A MessageEvent object, as described in `.onSIMIDEvent()`

`args <Object>` ***`Optional`***

The `args` data for the SIMID message. If the `simidMessage` type requires that `args` be provided, this object must be supplied

Return value

None

Basic usage

The example below requests the player to pause the ad.

```
overlay.simid.sendMessage("SIMID:Creative:requestPlay",
                          function(data, messageEvent) {console.log(data)});

// {"type":"resolve","sessionId":"f294d67c-0a62-4435-8b8f-f1434fd6bdf7","  ...
```

10.2. .onSIMIDEvent()

The `.onSIMIDEvent()` event is fired for every new SIMID event. For details, see SIMID Events And Messages.

Syntax

```
overlay.onSIMIDEvent(callbackFunction(message));
```

Parameters

`callbackFn`

A function to execute containing the response. The function is called with a single parameter:

> **`message <MessageEvent>`**
>
> A MessageEvent object. Fields of note are summarized below:
>
> **`data <String>`**
>
> A response object containing the following fields (as defined in detail by [SIMID]:
>
> **`type <String>`**
>
> The SIMID event message name
>
> **`sessionId <String>`**
>
> The SIMID session ID
>
> **`args <Optional Object>`**
>
> The SIMID message arguments
>
> **`messageId <Number>`**
>
> The SIMID messageId
>
> **`timestamp <Number>`**
>
> The timestamp (Unix epoch) of the SIMID message

Return value

None

10.3. .onIMAEvent()

The `.onIMAEvent()` event is fired for every new IMA event.

Syntax

```
overlay.onIMAEvent(callbackFunction(imaEvent));
```

Parameters

`callbackFn`

A function to execute containing the response. The function is called with a single parameter:

> **`imaEvent <String>`**
>
> The IMA Event String, as defined by [Google IMA] and profiled by this document.
>
> For more details see IMA Events

None

Basic usage

The example prints all IMA events to the console.

```
overlay.onIMAEvent(function (imaEvent) {
  console.log("IMA Event: ", imaEvent);
});

// IMA Event: AD_PROGRESS
// IMA Event: USER_CLOSE
```

Annex A - Example SIMID Creative

This annex describes the example SIMID Creative available within the Starter Kit.

This SIMID Creative is available in the `public/simid-creative` directory and can be explored on screen via the Q-Bar's "SIMID CREATIVE" content rail.

This content rail contains two VOD entries. Both entries will present an ad with the same example application (`SIMID-Creative`) followed by unique VOD content.

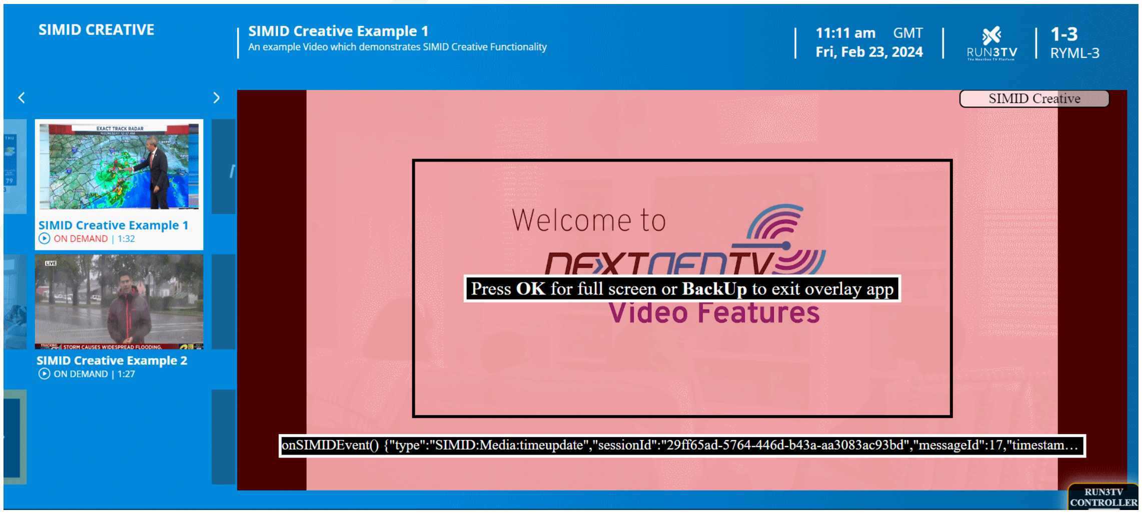
For access to the Starter Kit, please contact your Run3TV representative.

SIMID-Creative Example

This SIMID Creative is an example of the near-minimum amount of code required to create a SIMID Creative.

When accessed via the Q-Bar's "SIMID CREATIVE" content rail, the Framework uses an example VAST response hosted at https://dai-tester-app.a3fa.yottacloud.tv/vast/local-linear-overlay.xml to define the location of this application.

The screenshots below show this SIMID Creative in Preview Mode and Full Screen Mode:



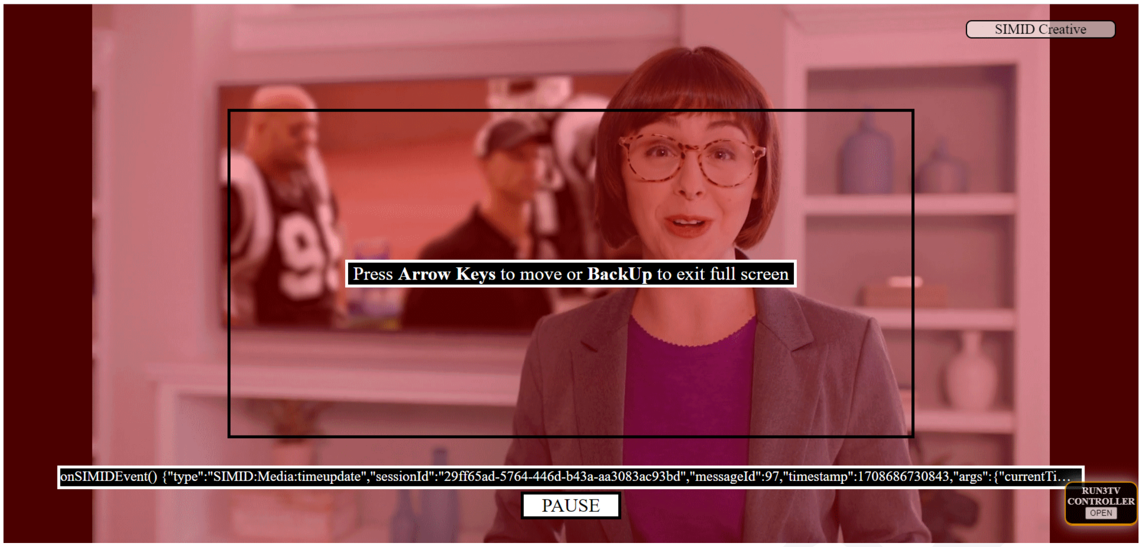
SIMID Creative Example 1

An example Video which demonstrates SIMID Creative Functionality

Press **OK** for full screen or **BackUp** to exit overlay app

```
onSIMIDEvent() {"type":"SIMID:Media:timeupdate","sessionId":"29ff65ad-5764-446d-b43a-aa3083ac93bd","messageId":17,"timestam…
```

Preview Mode


SIMID Creative
Press **Arrow Keys** to move or **BackUp** to exit full screen
PAUSE
RUN3TV
CONTROLLER
OPEN

```
onSIMIDEvent() {"type":"SIMID:Media:timeupdate","sessionId":"29ff65ad-5764-446d-b43a-aa3083ac93bd","messageId":97,"timestamp":1708686730843,"args":{"currentTi…
```

Full Screen Mode

This SIMID Creative captures events from `a3fa-framework.ad-overlay.min.js`, as summarized below:

Action	Causes Event	Description
The Framework informs the SIMID Creative that it has been launched	⇒ `.onLoad()`	• Sets the message in the text box depending on whether the SIMID Creative has been launched in Preview Mode or Full Screen Mode • Sets the button visibility depending on whether the SIMID Creative has been launched in Preview Mode or Full Screen Mode
The SIMID Creative enters Preview Mode	⇒ `.onPreview()`	• Updates the content of the text box on screen • Hides the button
The SIMID Creative enters Full Screen Mode	⇒ `.onFullScreen()`	• Updates the contents of text box on screen • Shows the button
The viewer presses **RIGHT**	⇒ `.onNavNext()`	• The text box is shifted 10 pixels in the direction of the arrow key chosen
The viewer presses **LEFT**	⇒ `.onNavBack()`	
The viewer presses **UP**	⇒ `.onNavUp()`	
The viewer presses **DOWN**	⇒ `.onNavDown()`	
The viewer presses **OK** / **SELECT**	⇒ `.onNavDown()`	• Changes the color of the text • Sends a SIMID message to request that the play state be changed (based on the current button text)

Action	Causes Event	Description
The viewer presses **BACK**	⇒ `.onNavExit()`	● Calls `.exitFullScreen()`, informing the Run3TV Application that this SIMID Creative wishes to exit Full Screen Mode. ● The Application will return the SIMID Creative to Preview Mode
An IMA event is presented to the SIMID Creative	⇒ `.onIMAEvent()`	● The IMA event is written to the console
A SIMID event is presented to the SIMID Creative	⇒ `.onSIMIDEvent()`	● The SIMID event message is written to the console ● If the message is to inform the Creative that the playback state has changed (`SIMID:Media:play` or `SIMID:Media:pause`), the button text is toggled