nomensa
humanising technology

Nomensa - Accessible Media Player

Technical manual

Contents

Technical documentation ____ 3
- Core required files 3

Using the Youtube AMP ____ 4
- Additional required files 4
- Initialising the YouTube AMP 4
- Loading the YouTube AMP 5

Using the JW Player AMP ____ 8
- Additional required files 8
- Initialising the JW Player AMP 9
- Loading the JW Player AMP 10

Using the Vimeo AMP ____ 13
- Additional required files 13
- Initialising the Vimeo AMP 14
- Loading the Vimeo AMP 14

Using captions ____ 17
- Example code: Caption formatting 17

Appendix A ____ 19
- Required properties 19
- Optional properties 20

T +44 (0)117 929 7333 **E** info@nomensa.com
F +44 (0)117 929 7543 **W** www.nomensa.com

nomensa
humanising technology

Overview

The Accessible Media Player (AMP) is a flexible multimedia solution for websites and intranets. It can be configured to support a choice of formats, and customised to match your website designs.

T +44 (0)117 929 7333
F +44 (0)117 929 7543
E info@nomensa.com
W www.nomensa.com

nomensa
humanising technology

Technical documentation

The AMP consists of an accessible HTML/CSS wrapper, around a media player. The AMP is configured to work with the YouTube player by default, but it can also be configured to work with the Vimeo player or the JW Player (for MP3/MP4/FLV support).

Core required files

The following external files and libraries are required in order to use the AMP.

JavaScript files

- jQuery JavaScript library at version 1.3.x (or later);
- jQuery UI JavaScript library version 1.7.x (or later);
- core/javascript/jquery.player.min.js;
- swfObject JavaScript flash loader 2.2.

To ensure compatibility with the AMP we recommend using a pairing of the two libraries that are known to work reliably with each other. At the time of writing jQuery recommend using jQuery version 1.3.X with jQuery UI version 1.7.3. If a more recent version of jQuery is being used you should use jQuery UI version 1.8.16. If you are unsure up to date information can be found at http://www.jqueryui.com.

CSS files

- core/css/player-core.min.css;
- custom/css/player-theme.min.css.

T +44 (0)117 929 7333 **E** info@nomensa.com
F +44 (0)117 929 7543 **W** www.nomensa.com

nomensa
humanising technology

Using the Youtube AMP

Additional required files

The AMP is configured to work with the YouTube player by default. No additional files are required.

Example code: Including the required files

The following code will include all the required files. You'll need to make sure the local files are in the appropriate directories for the code to work.

```
<link type="text/css" href="core/css/player-
core.min.css" rel="stylesheet" >

<link type="text/css" href="custom/css/player-
theme.min.css" rel="stylesheet" >

<!-- Pull in the JQuery libraries from Google,
although local versions could be used instead
-->

<script type="text/javascript"
src="http://www.google.com/jsapi"></script>

<script type="text/javascript">

google.load("jquery", "1.3.2");

google.load("jqueryui", "1.7.2");

</script>

<script type="text/javascript"
src="core/javascript/swfobject.js"></script>

<script type="text/javascript"
src="core/javascript/jquery.player.min.js"></s
cript>
```

Initialising the YouTube AMP

In order to create a new instance of the AMP, a JavaScript object must be passed into the method call containing at

T +44 (0)117 929 7333 **E** info@nomensa.com
F +44 (0)117 929 7543 **W** www.nomensa.com

least two properties. The two properties that must be provided are:

- id : The id to assign to the media player. Note that this MUST be unique;
- media : The name of the media file. This is usually the last 11 characters from the YouTube videos URL. For example, for the URL http://www.youtube.com/watch?v=kXiGXcq4pqY the 'media' argument to supply would be "kXiGXcq4pqY".

Example code: Initialising the YouTube AMP

The following code will generate an instance of the YouTube AMP:

```
$(selector).player({
                    id:'yt_player',
                    media:'kXiGXcq4pqY'
                    });
```

A number of other configuration settings can be applied to the AMP. A full list of all configuration properties can be found in Appendix A.

Loading the YouTube AMP

The AMP initialization process can also be automated across your entire website. By applying HTML classes to links for each different video provider you can target these links and instantiate them in a more automated fashion. For example, if we ensure that we give all youtube links a class of 'youtube' we can find and instantiate them automatically.

Example code: Loading the YouTube AMP

HTML

```
<h3>Youtube Player with captions</h3>

<p>

    <a
href="http://www.youtube.com/watch?v=kXiGXcq4p
qY">Order of content by Emily Coward</a>
```

T +44 (0)117 929 7333 E info@nomensa.com
F +44 (0)117 929 7543 W www.nomensa.com

```
    <a class="captions"
href="captions/captions-order-of-content.xml"
style="display:none;">Captions</a>

</p>


<h3>Youtube Player without captions</h3>

<p>

    <a
href="http://www.youtube.com/watch?v=SUNmLuNdi
L8">Cats</a>

</p>
```

JavaScript

```
$(document).ready(function(){

      var $yt_links =
$("a[href*='http://www.youtube.com/watch']");
      // Selects all links that have an href
value pointing at youtube

      $.each($yt_links, function(i) {     //
Iterate through the links creating a new media
player for each

            var $holder = $(‘<span/>’);   //
Create a container for the media player


      $(this).parent().replaceWith($holder);
      // Replace the parent element of the
link (a <p> tag) with the holder

            var $caption_link =
$(this).siblings('.captions');      // Get a
reference to any sibling captions links that
exist

            // If the captions link exists
extract the href, otherwise set the captions
file variable to null

            var captions_file =
($caption_link.length > 0) ?
$caption_link.attr('href') : null;

var link = $(this).attr('href').split("=")[1];
      // Extract the video id from the YouTube
url in the link

$holder.player({  // Instantiate the player
```

T +44 (0)117 929 7333 E info@nomensa.com
F +44 (0)117 929 7543 W www.nomensa.com

```
      id : 'yt_player_' + i,  // ensure that
the index of the current element is appended
to the base id

      media : link,     // The media property
is assigned the id of the YouTube video that
was extracted from the link

      // A reference to the captions file is
also provided.  Note that this will be null if
no captions are found

      captions : captions_file

});

      });

});
```

T +44 (0)117 929 7333 **E** info@nomensa.com
F +44 (0)117 929 7543 **W** www.nomensa.com

nomensa
humanising technology

Using the JW Player AMP

Additional required files

In addition to the core required files, the following additional files are required in order to use the JW Player AMP. Please note that a JW Player license is not included with the AMP, so you may need to obtain one before using this configuration. The JW Player can be purchased and downloaded from http://www.longtailvideo.com/.

JavaScript files

- jw-config.js (included with the AMP);
- jwplayer.js (provided within the JW Player distribution).

Flash files

- core/javascript/player.swf (provided within the JW Player distribution).

Example code: Including the required files

The following code will include all the required files. You'll need to make sure the local files are in the appropriate directories for the code to work.

```
<head>

<link type="text/css" href="core/css/player-
core.min.css" rel="stylesheet" >

<link type="text/css" href="custom/css/player-
theme.min.css" rel="stylesheet" >

<!-- Pull in the JQuery libraries from Google,
although local versions could be used instead
-->

<script type="text/javascript"
src="http://www.google.com/jsapi"></script>

<script type="text/javascript">

google.load("jquery", "1.3.2");
```

T +44 (0)117 929 7333 E info@nomensa.com
F +44 (0)117 929 7543 W www.nomensa.com

```
google.load("jqueryui", "1.7.2");

</script>

<script type="text/javascript"
src="custom/javascript/config/jwplayer-
5/jwplayer.js"></script>

<script type="text/javascript" src="
custom/javascript/config/jwplayer-5/jw-
player.config.js "></script>

<script type="text/javascript"
src="core/javascript/swfobject.js"></script>

<script type="text/javascript"
src="core/javascript/jquery.player.min.js"></s
cript>

</head>
```

Initialising the JW Player AMP

Once the required files are loaded, initialising a JW Player AMP is straight forward.

Example code: Initialising the JW Player AMP

The following code will generate an instance of the JW Player AMP:

```
$(selector).player({
      id:'jw_player',
      media:'example/video/test.mp4',
      url: 'core/javascript/player.swf',
      swfCallback : jwPlayerReady
    },
    jwconfig);
```

The property 'URL' should provide a path (either relative or full) to the JW Player .swf file. Please ensure you have purchased a license for the JW Player for use on your website.

In addition a reference to a JavaScript callback function should be provided. The JavaScript callback function can be found in the JW Player configuration file provided with the AMP.

T +44 (0)117 929 7333 **E** info@nomensa.com
F +44 (0)117 929 7543 **W** www.nomensa.com

The jwconfig variable is a JavaScript object that provides custom functionality for the JW player. This will have been provided within the jw-player.config.js file. Provided the jw-player.config.js file is included within the head of the document you should not need to do anything aside from passing the jwconfig variable into the player() method as the second argument.

Loading the JW Player AMP

The AMP initialization process can also be automated across your entire website. For JW Player instances the process is almost identical to the one for creating AMP instances for YouTube videos.

The main practical difference is that the player() method takes a second argument. This is the jwplayer config object that provides functionality specific to the jwplayer. This config should have been included in the archive that you were sent when you bought the AMP (it can be found at /custom/javascript/config/jwplayer-5/jw-player.config.min.js). It is important to ensure that this configuration file is loaded before you call the player() method.

The other difference is that you need to pass another parameter into the settings object (the first parameter to the player() method. This property should be named 'url' and should be a path to the jwplayer.swf file that you have downloaded or licenced.

HTML

```
<h3>JW Player with captions</h3>

<p>

    <a class=”jwlink”
href="resources/videos/video1.mp4">Test video
1</a>

    <a class="captions"
href="resources/captions/captions1.xml"
style="display:none;">Captions</a>

</p>


<h3>JW Player without captions</h3>
```

T +44 (0)117 929 7333 E info@nomensa.com
F +44 (0)117 929 7543 W www.nomensa.com

```
      <p>

    <a class=”jwlink”
href="resources/videos/video2.mp4">Test video
2</a>

</p>
```

JavaScript

```
$(document).ready(function(){

      $.each($("a.jwlink"), function(i) { //
Iterate through the links creating a new media
player for each

            var $holder = $(‘<span/>’);   //
Create a container for the media player


      $(this).parent().replaceWith($holder);
      // Replace the parent element of the
link (a <p> tag) with the holder

            var $caption_link =
$(this).siblings('.captions');      // Get a
reference to any sibling captions links that
exist

            // If the captions link exists
extract the href, otherwise set the captions
file variable to null

            var captions_file =
($caption_link.length > 0) ?
$caption_link.attr('href') : null;

var link = $(this).attr('href');    // Extract
the path to the video from the link

$holder.player({  // Instantiate the player

      id : ‘jw_player_’ + i,  // ensure that
the index of the current element is appended
to the base id

      media : link,     // The media property
is assigned the id of the YouTube video that
was extracted from the link

      url:’path/to/jwplayer.swf’,   // Provide
a path to the jwplayer swf file.

      swfCallback : jwPlayerReady,  // Provide
a reference to the callback function for when
the swf has loaded

captions : captions_file // A reference to the
captions file is also provided.

}, jwconfig);

      });
```

T +44 (0)117 929 7333 **E** info@nomensa.com
F +44 (0)117 929 7543 **W** www.nomensa.com

```
});
```

T +44 (0)117 929 7333
F +44 (0)117 929 7543
E info@nomensa.com
W www.nomensa.com

nomensa
humanising technology

Using the Vimeo AMP

Additional required files

In addition to the core required files, the following additional files are required in order to use the Vimeo AMP.

JavaScript files

- vimeoconfig.js ((included with the AMP).

Example code: Including the required files

The following code will include all the required files. You'll need to make sure the local files are in the appropriate directories for the code to work.

<head>

```
<link type="text/css" href="core/css/player-
core.min.css" rel="stylesheet" >

<link type="text/css" href="custom/css/player-
theme.min.css" rel="stylesheet" >

<!-- Pull in the JQuery libraries from Google,
although local versions could be used instead
-->

<script type="text/javascript"
src="http://www.google.com/jsapi"></script>

<script type="text/javascript">

google.load("jquery", "1.3.2");

google.load("jqueryui", "1.7.2");

</script>

<script type="text/javascript"
src="custom/javascript/config/vimeo/vimeo.conf
ig.min.js"></script>

<script type="text/javascript"
src="core/javascript/swfobject.js"></script>

<script type="text/javascript"
src="core/javascript/jquery.player.min.js"></s
cript>
```

T +44 (0)117 929 7333 **E** info@nomensa.com
F +44 (0)117 929 7543 **W** www.nomensa.com

```
</head>
```

Initialising the Vimeo AMP

Once the required files are loaded, creating a Vimeo AMP is straight forward.

Example code: Initialising the Vimeo AMP

The following code will generate an instance of the Vimeo AMP:

```
$(selector).player({

    id:'vimeo_player',

    url :
'http://vimeo.com/moogaloop.swf?clip_id=',

    media:’25453241’

  },

  vimeoconfig);
```

When creating a Vimeo AMP the property ‘URL’ should provide a full path to the Vimeo moogaloop (shown above).

The vimeoconfig variable is a JavaScript object that provides custom functionality for the Vimeo AMP. This can be found in the vimeo.config.min.js file. Provided the vimeo.config.min.js file is included within the head of the document you should not need to do anything aside from passing the vimeoconfig variable into the player() method as the second argument.

Loading the Vimeo AMP

The AMP initialization process can also be automated across your entire website. For Vimeo Player instances the process is almost identical to the one for creating AMP instances for YouTube videos.

The main practical difference is that the player() method takes a second argument. This is the vimeo config object that provides functionality specific to the Vimeo player. This config should have been included in the archive that you

were sent when you bought the AMP (it can be found at /custom/javascript/config/vimeo/vimeo.config.min.js). It is important to ensure that this configuration file is loaded before you call the player() method.

Example code: Loading the Vimeo AMP

HTML

```
<h3>Vimeo Player with captions</h3>

<p>

    <a class=”vimeolink”
href="http://vimeo.com/3985019">Vimeo Test
1</a>

    <a class="captions"
href="resources/captions/captions1.xml"
style="display:none;">Captions</a>

</p>
```

JavaScript

```
$(document).ready(function(){

      $.each($("a.vimeolink"), function(i) {
      // Iterate through the links creating a
new media player for each

            var $holder = $(‘<span/>’);   //
Create a container for the media player


      $(this).parent().replaceWith($holder);
      // Replace the parent element of the
link (a <p> tag) with the holder

            var $caption_link =
$(this).siblings('.captions');      // Get a
reference to any sibling captions links that
exist

            // If the captions link exists
extract the href, otherwise set the captions
file variable to null

            var captions_file =
($caption_link.length > 0) ?
$caption_link.attr('href') : null;

var urlBits = $(this).attr('href').split(“/”);
      // Split the URL path on the ‘/’
character.

var media = urlBits[urlBits - 1];   // The
vimeo media id should be the last part of our
resulting array
```

T +44 (0)117 929 7333 E info@nomensa.com
F +44 (0)117 929 7543 W www.nomensa.com

```
$holder.player({  // Instantiate the player

      id : 'vimeo_player_' + i,     // ensure
that the index of the current element is
appended to the base id

      media : media,    // The media property
is assigned the id of the vimeo video that was
extracted from the link

      url :
'http://vimeo.com/moogaloop.swf?clip_id=',
      // Url that points to the vimeo player
swf

captions : captions_file // A reference to the
captions file is also provided.

}, vimeoconfig);  // Ensure that the
vimeoconfig is passed into the player method
call as the 2nd arg on instantiation

      });

});
```

T +44 (0)117 929 7333 **E** info@nomensa.com
F +44 (0)117 929 7543 **W** www.nomensa.com

nomensa
humanising technology

Using captions

The available APIs for online media players do not provide a way to turn captions on/off. The AMP uses a JavaScript based solution to solve this problem. In some circumstances JavaScript may not be a workable solution however.

To use captions with any configuration of the AMP, the following criteria should be met:

1. The captions file must be well formed XML and must be formatted correctly;

2. The captions file must be hosted on the same server/domain as the website that the media player is being integrated into. The reason for this is that AJAX is used to retrieve the associated captions file when the player is instantiated. If the captions file is hosted on a different domain many browsers will prevent the AJAX call from executing correctly. This is down to the 'Same origin policy' which is well documented online;

3. A path to the captions file must be provided as an argument to the AMP constructor on instantiation. This can be extracted from the page HTML or hard coded into the JavaScript initialisation call.

Example code: Caption formatting

The following shows an example of a well formed captions file:

```
<tt xmlns="http://www.w3.org/2006/10/ttaf1">

    <body>

      <div xml:id="captions">

<p begin="00:00:07" end="00:00:14">Hello, I'm
Léonie and in this video we're going to take a
look at ARIA document landmark roles.</p>

<p begin="00:00:14" end="00:00:20">Accessible
Rich Internet Applications (ARIA) is a new
specification from the W3C</p>

      </div>

    </body>
```

T +44 (0)117 929 7333 **E** info@nomensa.com
F +44 (0)117 929 7543 **W** www.nomensa.com

```
</tt>
```

Each of the tags within the XML file constitutes one discrete caption. Each of these tags should be given 2 attributes (a 'begin' and an 'end' time). These attributes should use the following format hours:minutes:seconds(hh:mm:ss) where hours minutes and seconds represent the time elapsed whilst playing the video.

For example, the following captions will not work:

- Hello, I'm Léonie and in this video we're going to take a look at ARIA document landmark roles.
- Hello, I'm Léonie and in this video we're going to take a look at ARIA document landmark roles.
- Hello, I'm Léonie and in this video we're going to take a look at ARIA document landmark roles.

T +44 (0)117 929 7333 **E** info@nomensa.com
F +44 (0)117 929 7543 **W** www.nomensa.com

Appendix A

Required properties

The following properties are required for the AMP to work.

`id`: A unique id for this player. It is essential that each call to generate a new player passes in a specific new id.

Default: 'media_player'

`url`: The relative path to the player.swf file from the file calling the player generation. This is only required for calls to the JW player (the player.swf is actually the JW player itself).

Default: 'http://www.youtube.com/apiplayer?enablejsapi=1&version=3&playerapiid='

`media`: The media file you want the player to load when generated. For YouTube and Vimeo players this is the unique id at the end of the URL.

The JW player on the other hand requires a full url or relative path to the media file that you wish to play.

For Example:

- To play a YouTube file located at http://www.youtube.com/watch?v=kXiGXcq4pqY the 'media' property should be assigned the value 'kXiGXcq4pqY';
- To play a vimeo file located at http://vimeo.com/3985019 the 'media' property should be assigned the value '3985019';
- For JW players the 'media' property should be assigned a value representing the relative path to the media file you want the player to load.

Default: '8LiQ-bLJaM4'

T +44 (0)117 929 7333 **E** info@nomensa.com
F +44 (0)117 929 7543 **W** www.nomensa.com

Optional properties

The following properties are not required, but can be used to alter the behaviour of the AMP.

`repeat`: This takes a Boolean value (either true or false) indicating whether or not to loop the video when it finishes.

Default: false

`captions`: This should be a relative path to a captions file for the video. Captioning should work for all players. It is worth noting that captions need to be in a specific format. For more information on this please refer to appendix B at the end of this document.

Default: null

`captionsOn`: This takes a Boolean value (either true or false) indicating whether or not to start the video with captioning turned on.

Default: false

`flashWidth`: The width of the flash player within the container. This changes the width of the visual screen of the flash player rather than the container. This defaults to 100%, but can be changed to other values such as px or different %'s.

Default: '100%'

`flashHeight`: The height of the flash player within the container.

Default: '300px'

`playerStyles`: A map of styles to apply to the player container. This map consists of key value pairs that map to a CSS property name and a CSS property value respectively. Any number of values can be assigned to this map.

Default: {

'height' : '100%',

'width' : '100%'

}

T +44 (0)117 929 7333 **E** info@nomensa.com
F +44 (0)117 929 7543 **W** www.nomensa.com

`sliderTimeout`: An integer representing the interval (in milliseconds) for updating the position of the slider within the slider bar.

Default: 350

`flashContainer`: The type of container that the flash/media player will be wrapped in. Can be 'div', 'span' or any other valid or appropriate html element. Note that the '<' and '>' characters should not be provided.

Default: 'span'

`playerContainer`: The type of container that the entire media player will be wrapped in. Can be 'div', 'span' or any other valid or appropriate html element. Note that the '<' and '>' characters should not be provided.

Default: 'span'

`Image`: A placeholder image to show before the video starts playing. This will not work with the YouTube or Vimeo players. The property should be assigned a relative path to the image.

Default: '' (an empty string)

`playerSkip`: An integer representing the amount of time in seconds that the fast-forward and rewind buttons skip through the video.

Default: 10

`volumeStep`: The amount by which to increase and decrease the volume of the video (as a percentage) when the volume controls are used. Expressed as an integer.

Default: 10

`useHtml5`: A Boolean flag determining whether or not HTML5 video will be used. If this is enabled and the browser in use supports HTML5 video playback for the specified file the media player will make use of the <video> or <audio> elements instead of flash. Note that this currently only works with the JW Player.

Default: true

`swfCallback`: A reference to a function that will be called when the swf file has been embedded into the page and has loaded. Only required for the JW Player

T +44 (0)117 929 7333 **E** info@nomensa.com
F +44 (0)117 929 7543 **W** www.nomensa.com

implementation of the AMP. We advise that this is not used unless you know what you are doing.

Default: null

`buttons`: A map of key value pairs denoting which buttons should be inserted into the players control bar. Each button takes a Boolean (true/false) value. If the button is given a true value it will appear in the control bar. If not it will be removed. The only exception to this rule is the toggle button. If this is set to true the play and pause buttons will be combined into one. If set to false both a play and a pause button will be provided.

Default: {

forward:true,

rewind:true,

toggle:true

}

T +44 (0)117 929 7333 **E** info@nomensa.com
F +44 (0)117 929 7543 **W** www.nomensa.com