

1 Document

An XML file containing all project information to load and execute the project on the client. Certain interfaces are optional, however others are mandatory. This guide should reflect the changes in the XML project and keep track of the versions. Hopwfully this can remain simple!

2 Root

The XML root must be `<BrowserEvalProjectDocument>`. This should be sufficiently identifiable in both itself and in the JavaScript decoding as it will create an object called the root name.

There must also be a `<version>` tag which has the attribute `id` containing a numerical representation of the version. Currently everything in this document can be assumed to be version 1. If future updates or corrections are made post delivery this should give the flexibility to ensure past projects still work.

The root will also contain the following tags: setup and tracks.

3 Setup tag

The setup tag specifies certain global test settings including: the interface type to use, the project return location and any other setup instructions. Any general pre/post test questions must be specified in the relevant children tag. Any enabled metrics must also be specified in the metric child node.

3.1 Attributes

- **interface** - Mandatory, String. Defaults to APE, otherwise use to load any of the available interfaces. Currently only valid string is APE.
- **projectReturn** - Mandatory, String. Specify the URL to return the test results. If null client will generate XML locally and prompt user to return the file.
- **randomiseOrder** - Optional, default to false. Specify if the order of the test pages are to be randomised.
- **collectMetrics** - Deprecated. Optional, Boolean. Default to false. Determine if the test metrics should be collected. These include how long each test session took etc. The full metrics list can be modified in the 'metrics' tag.

3.2 Elements

None

4 Metric tag

A 'setup' node child tag, metrics must be declared in the setup tag. This takes a set of children 'metricEnable' to define which metrics to collect and present, for example `<metricEnable>testTimer </metricEnable>`. The interface may not be able to utilise all of these features. It

is up to the interface to determine whether to use the metric or not. For example, 'elementFlag-Moved' would not be usable in an AB test as there are no interface value objects.

4.1 **metricEnable tag**

This takes a single attribute to determine which metric to enable for collection. Some of these are a global, per track or per test instance.

- **testTimer** - Return the global test timer and test instance timers. Measures the time between the first start and final submit.
- **elementTimer** - Return the total time each audioElement in each test was listened too. Measures time between successive clicks on the track changer
- **elementTracker** - Return the initial position of each track
- **elementTrackerFull** - Return an enumerated pair of time and position. Track the entire movement of each element position. NOTE: Will override the elementTracker option above and throw an error into the browser console.
- **elementFlagListenedTo** - Return a boolean per element to see if the element was listened to
- **elementFlagMoved** - Return a boolean per element to see if the element slider was moved.
- **elementFlagComments** - Return a boolean per element to see if the element has comments.

5 **Interface tag**

This enables any interface options for each test page. Further interface tags in each audioHolder add further options. This takes option nodes only. Each option node takes a 'name' to determine what feature to enable. The following options are currently employed.

5.1 **Option nodes**

- **fragmentPlayed** - Enforce each fragment be partially played before finishing the page
- **fragmentFullPlayback** - Enforce each fragment to be fully played from start to end before finishing the page. Not enabled if an audioHolder reports it is to be looped playback.
- **fragmentMoved** - Enforce each fragment to be moved at least once from its starting position.
- **fragmentComments** - Enforce each fragment comment to have some text entered.
- **playhead** - Show the playhead object.
- **page-count** - Show the current test page number and the total number
- **scalerange** - Must also have min and max values between 0 and 100. Enforce that at least one fragment is below the min value and one fragment is above the max value before continuing.

6 **AudioHolder tag**

There should be one audioHolder tag for each test page, inside which each audioElement is specified as children. The audioHolder tag can help to generalise certain objects.

6.1 Attributes

- **id** - Mandatory, String. Give an ID string or number to identify the test in the result.
- **hostURL** - Optional, String. If all tracks are hosted from the same folder on a server, you can put in the lead here. For instance, if loading `http://test.com/tracks/track1.wav` and `http://test.com/tracks/track2.wav`, this could equal `http://test.com/tracks/` and the url attribute in the track tag can be `track1.wav` or `track2.wav`. Equally `http://test.com/` and then using `tracks/track1.wav` and `tracks/track2.wav` is valid.
- **sampleRate** - Optional, Number. If your test requires a specific sample rate, this should be set to the desired sample rate in Hertz. This does not set the browser to the correct sample rate, but forces the browser to check the sample rate matches. If this is undefined, no sample rate matching will occur.
- **randomiseOrder** - Optional, Boolean String. Defaults to false. Determine if the track order should be randomised. Must be true or false.
- **repeatCount** - Optional, Number. Defaults to 0 (ie: no repeats). The number of times a test should be repeated.
- **loop** - Optional, Boolean String. Defaults to false. Enable if audioElements should loop their playback or not.
- **elementComments** - Optional, Boolean String. Defaults to false. Enable to populate the test page with Comment Boxes linked to each fragment.

6.2 Elements

Contain the audioElements tags and the interfaceSetup tag.

7 audioElements tag

This must reside as children in the audioHolder tag. There must be one audioElement tag per sound sample to load into the test.

7.1 Attributes

- **id** - Mandatory, String. Must give a string or number to identify each audio element. This id is used in the output to identify each track once randomised.
- **url** - Mandatory, String. Contain the full URL to the track. If the Tracks tag hostURL is set, concatenate this tag with the hostURL attribute to obtain the full URL.
- **type** - Optional, String. Can be 'normal', 'anchor', 'reference' or 'outside-reference'. Default is for normal. Only one anchor can be specified per page. Only one reference can be specified per page. Only one outside-reference can be specified per page. If multiple audioelements have the same type in the same page, the browser console will explain the problem and the audioelements will be treated as 'normal' fragments.

8 interface tag

This is contained within the audioHolder tag and outlines test instance specific requirements. These include the following children tags:

- 'title' - Contains the test title to be shown at the top of the page. Can only be one title node per interface.
- 'scale' - Takes the attribute position to be a value between 0 and 100 indicating where on the scale to place the text contained inside. Can be multiple scale tags per interface.
- 'option' - Can hold any of the option tags available in the setup tag. These will only be enabled for the page instance.

9 CommentQuestion tag

This allows another question and comment box to be presented on the page. The results of these are passed back in the results XML with both the comment and the question. An id must be set, otherwise the result is undefined. Also the type must be set as follows.

- 'type="text"' - Default type. Creates a text box on the page. The text is included as the element.
- 'type="radio"' - Create radio button entry. Multiple equally spaced entries per box. Only one entry can be selected. Each radio button is specified by an option tag. The tag must contain a name attribute, which will be the response if true. Optional text can be included as the element to label the box. Presented question is included in a statement node.
- 'type="checkbox"' - Create a checkbox entry. Multiple equally spaced entries per box, multiple can be selected. Each checkbox is specified by an option tag. The tag must contain a name attribute. Optional text can be included as the element to label the checkbox. Presented question is included in a statement node.

10 PreTest tag and PostTest tag

These are 1st level tags. The PreTest tag allows for the specifying of pre test instructions and questions. These appear as a pop-up style window with next buttons and other automatic GUI. The postTest tag allows for specifying post test instructions, questions and resources. These appear as a pop-up style window after the submit button is pressed.

PreTest and PostTag nodes can be included in the audioHolders (for pre and post for that test page) and in the setup node for pre-test before the first page, and post-test for after the last test.

10.1 Attributes

None.

10.2 Elements

Takes the following available tags to structure the pre and post test options. The order these are presented in the XML define the order they appear.

10.2.1 Statement

The statement tag simply prints the included string verbatim on a 'pop-up' window with a next button.

10.2.2 Question

This allows for a question to be asked pre/post the test. The response includes both the question asked and the response. The following attributes are used:

- **id** - Mandatory, String. Used to reference to the response.
- **mandatory** - Optional, String Boolean. Determine if this question must have some response. Defaults to false.
- **boxsize** - Optional, String. Defaults to normal. Allows 'small', 'normal', 'large' or 'huge'. This determines the size of the box entry. All entries are wrappable, so this does not determine the maximum size of the text response, but can be used to encourage (or discourage) long answers.

10.2.3 Number

Gives a number box entry defined with the following attributes:

- **id** - Mandatory, String. Used to reference to the response.
- **mandatory** - Optional, String Boolean. Determine if this question must have some response. Defaults to false.
- **min**, **max** - Optional, Number. Defaults to undefined. Used to bound the number response. If a number entered is below this, the pre/post sequence will not continued.

10.2.4 Radio

Create a set of radio boxes. Only one element can be returned as true. The radio node must have an id to reference for the output. The radio node also must have a statment node which will contain the text to show on the popup. Radio buttons are created using option nodes. Each node must have a name attribute to indentify which radio was selected in the response. The option node can also contain any text to link to the node.

10.2.5 Checkbox

Create a set of checkbox boxes. Multiple elements can be returned as true. The checkbox node must have an id to reference for the output. The checkbox node also must have a statment node which will contain the text to show on the popup. Checkbox buttons are created using option nodes. Each node must have a name attribute to indentify which radio was selected in the response. The option node can also contain any text to link to the node.

11 Example

Here is an example XML structure

```

<?xml version="1.0" encoding="utf-8"?>
<BrowserEvalProjectDocument>
  <setup interface="APE" projectReturn="null" randomiseOrder='true' colle
    <PreTest>
      <statement>Please listen to all mixes</statement>
      <question id="location" mandatory="true">Please enter y
    </PreTest>
    <PostTest>
      <statement>Thank you for taking this listening test.</s
      <question id="SessionID">Please enter your name.</quest
    </PostTest>
    <Metric>
      <metricEnable>testTimer</metricEnable>
      <metricEnable>elementTimer</metricEnable>
      <metricEnable>elementTracker</metricEnable>
      <metricEnable>elementFlagListenedTo</metricEnable>
      <metricEnable>elementFlagMoved</metricEnable>
    </Metric>
  </setup>
  <audioHolder id='0' hostURL="example_eval/" sampleRate="44100" randomis
    <interface>
      <title>Example Test Question</title>
      <scale position="0">Min</scale>
      <scale position="100">Max</scale>
      <scale position="50">Middle</scale>
      <scale position="20">20</scale>
    </interface>
    <audioElements url="0.wav" id="0"/>
    <audioElements url="1.wav" id="1"/>
    <audioElements url="2.wav" id="2"/>
    <audioElements url="3.wav" id="3"/>
    <audioElements url="4.wav" id="4"/>
    <audioElements url="5.wav" id="5"/>
    <audioElements url="6.wav" id="6"/>
    <audioElements url="7.wav" id="7"/>
    <audioElements url="8.wav" id="8"/>
    <audioElements url="9.wav" id="9"/>
    <audioElements url="10.wav" id="10"/>
    <CommentQuestion id='mixingExperiance'>What is your mixing exper
    <PreTest>
      <statement>Start the Test 3</statement>
    </PreTest>
    <PostTest>
      <statement>Please take a break before the next test</sta
      <question id="testComment">How did you find the test</q
    </PostTest>
  </audioHolder>
</BrowserEvalProjectDocument>

```