

## 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.

An example of this tag could be:

```
<setup interface="APE" projectReturn="http://project.return.url/goes/here" />
```

The setup should not require an element or any children to run.

### 3.1 Attributes

- `interface` - Mandatory. Defaults to APE, otherwise use to load any of the available interfaces
- `projectReturn` - Mandatory. Specify the URL to return the test results. If null client will generate XML locally and prompt user to return the file.

### 3.2 Elements

None

## 4 Tracks tag

There should be one tracks tag, inside which each track is specified as children. The track tag can help to generalise certain objects.

### 4.1 Attributes

- `hostURL` - Optional. 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.v` 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. 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 allows for checking if the sample rates match.
- **sampleRateExplicit** - Optional. A true/false to determine if the sample rate given in the attribute **sampleRate** must be matched by the system. If the web audio API does not match, a browser alert is called and the rest of the interface loading is discarded. Media files are also not downloaded until this check can be corrected.

## 4.2 Elements

Contain the track tags.

## 5 Track tag

This must reside as children in the tracks tag. There must be one track tag per track to load. The track tag must contain the URL to load.

### 5.1 Attributes

- **url** - Mandatory. 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.

## 6 Example

Here is an example XML structure

```
<?xml version="1.0" encoding="utf-8"?>
<BrowserEvalProjectDocument>
  <setup interface="APE" projectReturn="null" />
  <tracks hostURL="example_eval/" sampleRate="44100"
    sampleRateExplicit="true">
    <track url="0.wav"/>
    <track url="1.wav"/>
    <track url="2.wav"/>
    <track url="3.wav"/>
    <track url="4.wav"/>
    <track url="5.wav"/>
    <track url="6.wav"/>
    <track url="7.wav"/>
    <track url="8.wav"/>
    <track url="9.wav"/>
    <track url="10.wav"/>
  </tracks>
</BrowserEvalProjectDocument>
```