

Convert old Flash object using a SWF widget to play mp3s into an equivalent HTML5 audio element , in the data file

Table of contents

Generalities.....	2
What do you have to convert?.....	2
How to make the conversion?.....	3
Explanation.....	5
Adaptation to your data file.....	6
How to write the Find phrase and the Replace phrase?.....	6
Examples.....	7
1) hbs players.....	7
2) P.Y.G. player.....	7
3) music player.....	8
4) dewplayer.....	8
5) mp3 player.....	9
6) player_mp3_maxi.....	9
7) Flash player from the interface.....	10
Warnings.....	10
What to do after conversion?.....	11

Generalities

What do you have to convert?

Here's an example (see [audio_HP6.jqz](#)) of an audio media inserted **in the JCloze 6 interface**:

```
<object type="application/x-shockwave-flash" data="hbs_mp3_player2but_narrow_slider.swf" width="250" height="75" style="vertical-align: middle;" >
<param name="type" value="application/x-shockwave-flash" />
<param name="src" value="hbs_mp3_player2but_narrow_slider.swf" />
<param name="data" value="hbs_mp3_player2but_narrow_slider.swf" />
<param name="codebase" value="hbs_mp3_player2but_narrow_slider.swf" />
<param name="FlashVars" value="TheSound=mysoundfile.mp3" />
<param name="allowScriptAccess" value="sameDomain" />
<param name="movie" value="hbs_mp3_player2but_narrow_slider.swf" />
<param name="loop" value="false" />
<param name="quality" value="high" />
<param name="wmode" value="transparent" /></object>
```

in the data file (jqz, jcl etc.) it is written using HTML code:

```
&#x003C;object type="application/x-shockwave-flash" data="hbs_mp3_player2but_narrow_slider.swf" width="250" height="75" style="vertical-align: middle;" &#x003E;
&#x003C;param name="type" value="application/x-shockwave-flash" /&#x003E;
&#x003C;param name="src" value="hbs_mp3_player2but_narrow_slider.swf" /&#x003E;
&#x003C;param name="data" value="hbs_mp3_player2but_narrow_slider.swf" /&#x003E;
&#x003C;param name="codebase" value="hbs_mp3_player2but_narrow_slider.swf" /&#x003E;
&#x003C;param name="FlashVars" value="TheSound=mysoundfile.mp3" /&#x003E;
&#x003C;param name="allowScriptAccess" value="sameDomain" /&#x003E;
&#x003C;param name="movie" value="hbs_mp3_player2but_narrow_slider.swf" /&#x003E;
&#x003C;param name="loop" value="false" /&#x003E;
&#x003C;param name="quality" value="high" /&#x003E;
&#x003C;param name="wmode" value="transparent" /&#x003E;&#x003C;/object&#x003E;
```

Note:

" is the HTML code for " "
& is the HTML code for & "&
&#x003C; is the HTML code for < "&
&#x003E; is the HTML code for > "&

And you need to convert all this into:

```
&amp;#x003C;audio controls=&quot;controls&quot;&amp;#x003E;&amp;#x003C;source src=&quot;mysoundfile .mp3&quot;  
type=&quot;audio/mp3&quot;/&amp;#x003E;&amp;#x003C;/audio&amp;#x003E;
```

which is equivalent to:

```
<audio controls="controls"><source src="mysoundfile.mp3" type="audio/mp3"></audio>
```

Of course you want to convert all the audio objects of a file and do this conversion for every file in different folders.

How to make the conversion?

- You need a text editor with a search and replace function that accepts regular expressions and can do batch processing. I recommend [Notepad++](#) which is a very good and very useful software. It's a Windows software but it works with Wine on Linux. And it's governed by GNU General Public License.

- The method (please try it with audio_HP6.jqz, because it may not be adapted to your own data file – see page 6):

Open Notepad++.

1- *Ctrl+Shift+F* to open the *Find in Files* window,

2- Enter the Find and Replace phrases (see the explanation page 5),

Find: &#x003C;object.+?TheSound= ([^&]+) .+?&#x003C;/object&#x003E;

(all on one line)

Replace: ;&#x003C;audio controls="controls"&#x003E;&#x003C;source src="\${1}"
type="audio/mp3"/&#x003E;&#x003C;/audio&#x003E;

(all on one line)

3- Add a filter to select only the files you need to treat: *.j* would work in your case (it will treat only the files with an extension beginning with j),

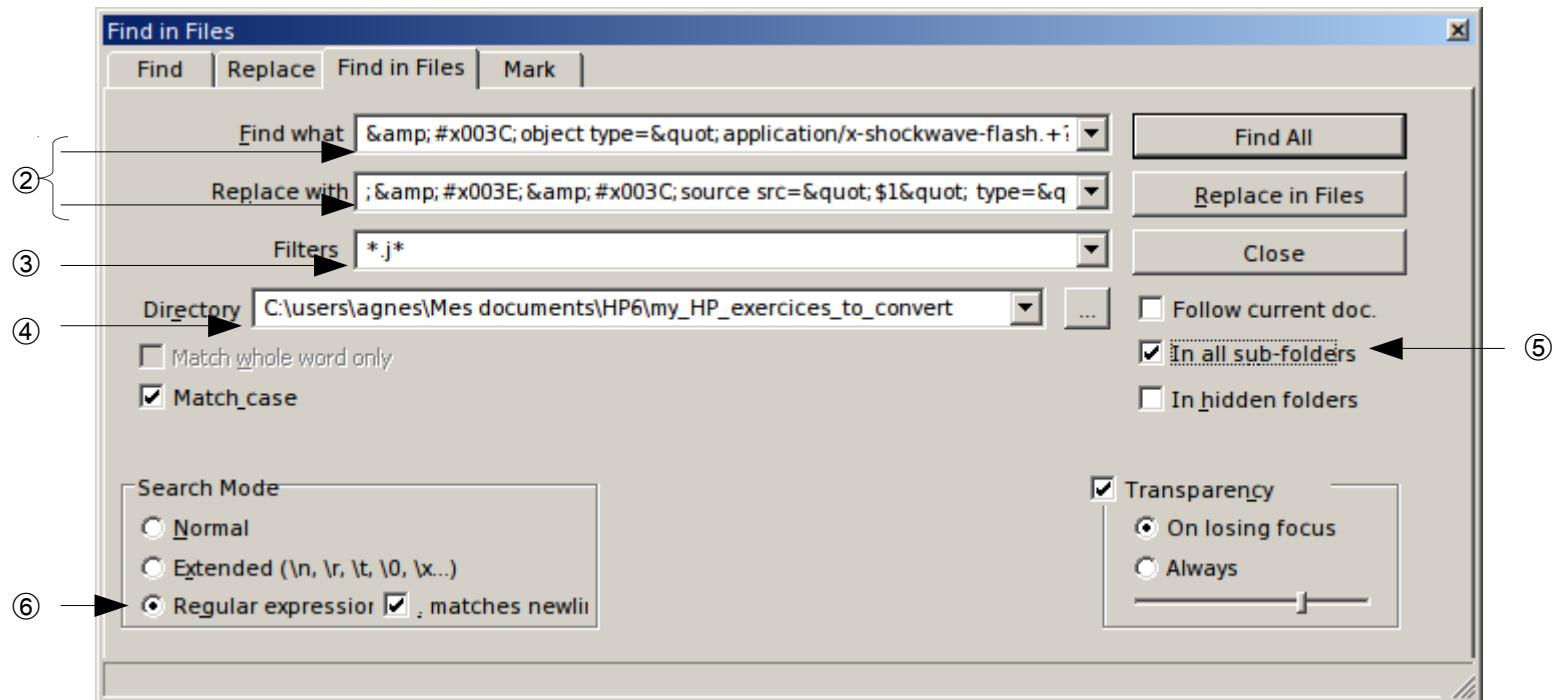
4- Select the folder that contains the files (possibly in sub-folders) you want to replace text in,

5- Select the option *In all sub-folders*, if it is what you want (but you can prefer to work folder by folder),

6- Make sure you select the options: *Regular expression* and *Matches newlines*,

7- [Optional] Hit **Find All** to be sure that the correct parts are effectively found in the files.

8- Hit **Replace in Files**.



Note : a good practice would be to test the Find and Replace with only one or two files in a test folder to be sure that everything is correct.

Explanation

Data file:

```
&#x003C;object type=&quot;application/x-shockwave-flash&quot; data=&quot;hbs_mp3_player2but_narrow_slider.swf&quot;
width=&quot;250&quot; height=&quot;75&quot; style=&quot;vertical-align: middle;&quot; &#x003E;
&#x003C;param name=&quot;type&quot; value=&quot;application/x-shockwave-flash&quot; /&#x003E;
&#x003C;param name=&quot;src&quot; value=&quot;hbs_mp3_player2but_narrow_slider.swf&quot; /&#x003E;
&#x003C;param name=&quot;data&quot; value=&quot;hbs_mp3_player2but_narrow_slider.swf&quot; /&#x003E;
&#x003C;param name=&quot;codebase&quot; value=&quot;hbs_mp3_player2but_narrow_slider.swf&quot; /&#x003E;
&#x003C;param name=&quot;FlashVars&quot; value=&quot;TheSound=mysoundfile.mp3&quot; /&#x003E;
&#x003C;param name=&quot;allowScriptAccess&quot; value=&quot;sameDomain&quot; /&#x003E;
&#x003C;param name=&quot;movie&quot; value=&quot;hbs_mp3_player2but_narrow_slider.swf&quot; /&#x003E;
&#x003C;param name=&quot;loop&quot; value=&quot;false&quot; /&#x003E;
&#x003C;param name=&quot;quality&quot; value=&quot;high&quot; /&#x003E;
&#x003C;param name=&quot;wmode&quot; value=&quot;transparent&quot; /&#x003E;&#x003C;/object&#x003E;
```

Find: `<object.+?TheSound=([^&]+).+?</object>`

Explanation:

`<object`: matches the characters `<object`

`.`: matches *any character*

`+?`: repeat minimal number of times (till `TheSound=` is found)

`TheSound=`: matches the characters `TheSound=`

`([^&]+)`: matches *any character which is not &*, as many times as possible

and keeps the result (here `mysoundfile.mp3`) in a variable \$1 which will be used in the Replace phrase.

`.+?`: matches *any character*, repeat minimal number of times (till `</object>` is found)

`</object>`: matches the characters `</object>`

So all the text from `<object` to `</object>` will be found and `mysoundfile.mp3` is kept in a variable.

Replace: `<audio controls="controls"><source src="$1" type="audio/mp3"/></audio>`

Explanation:

`<audio controls="controls"><source src="`: writes `<audio controls="controls"><source src="`

`$1`: writes the value of `$1` which is `mysoundfile.mp3`

`" type="audio/mp3"/></audio>`: writes `" type="audio/mp3"/></audio>`

So `<audio controls="controls"><source src="mysoundfile.mp3" type="audio/mp3"/></audio>` will be written, which is equivalent to:

`<audio controls="controls"><source src="mysoundfile.mp3" type="audio/mp3"></audio>`

Adaptation to your data file

How to write the Find phrase and the Replace phrase?

There were a lot of ways to add audio objects in HP6.

You will have to **adapt the Find and Replace phrases** to your situation. With the explanation above you should be able to do it.

The important points are:

- All the codes begin with `<object` (equivalent to `<object`) and end with `</object>` (equivalent to `</object>`)
- You must be able to find specific characters just **before** and **after** the **soundfile name**.

Examples

Lots of Hot Potatoes users have used players from [Stan Bogdanov's site](#), or the flash player proposed in the interface . Here's the Find and Replace phrases for the most used players.

1) hbs players

There were a lot of : hbs_mp3_player2but.swf, hbs_mp3_player2but_narrow_slider, , etc.

The soundfile name is always preceded by TheSound= and always followed by &, so the Find phrase and Replace phrase explained above (page 5) will work:

Find phrase: `<object.+?TheSound=([^\&]+).+?</object>`

Replace phrase: `<audio controls="controls"><source src="$1" type="audio/mp3"/></audio>`

2) P.Y.G. player

Usually the inserted code looks like:

```
<object type="application/x-shockwave-flash" data="pyg_player.swf?file=your_audio_file.mp3&autolaunch=false" width="220"
height="30" style="vertical-align: middle;">
<param name="movie" value="pyg_player.swf?file=your_audio_file.mp3&autolaunch=false" />
<param name="FlashVars" value="my_BackgroundColor=0xE6E6FA" />
<a href="javascript:void(0);" onclick="play('your_audio_file.mp3', '128');">Click here</a>
<a href="http://www.macromedia.com/go/getflashplayer">You need Flash! Go get it!</a>
</object>
```

The soundfile name is always preceded by pyg_player.swf?file= and followed by &, so the Find and Replace phrases should be:

Find phrase: `<object.+?pyg_player.swf?file=([^\&]+).+?</object>`

Replace phrase: `<audio controls="controls"><source src="$1" type="audio/mp3"/></audio>`

3) music player

Usually the inserted code looks like:

```
<object type="application/x-shockwave-flash" data="musicplayer_button.swf?
&amp;song_url=mysoundfile.mp3&amp;b_bgcolor=E6E6FA&amp;b_fgcolor=E6E6FA&amp;b_colors=000000,000000,aa0000&amp;"
width="17" height="17" style="vertical-align:middle;"><param name="movie" value="musicplayer_button.swf?
&amp;song_url=mysoundfile.mp3&amp;b_bgcolor=E6E6FA&amp;b_fgcolor=E6E6FA&amp;b_colors=000000,000000,b80000&amp;"
/></object>
```

The soundfile name is always preceded by `song_url=` and followed by `&`, so the Find and Replace phrases should be:

Find phrase: `&#x003C;object.+?song_url=(\[&]+).+?&#x003C;/object&#x003E;`

Replace phrase: `&#x003C;audio controls="controls"&#x003E;&#x003C;source src="$1"
type="audio/mp3"/&#x003E;&#x003C;/audio&#x003E;`

4) dewplayer

Usually the inserted code looks like:

```
<object type="application/x-shockwave-flash" data="dewplayer.swf" width="200" height="20" id="dewplayer" name="dewplayer">
<param name="movie" value="dewplayer.swf" />
<param name="flashvars" value="mp3=mysoundfile.mp3" />
<param name="wmode" value="transparent" />
</object>
```

The soundfile name is always preceded by `mp3=` and followed by `"` (html code: `"`), so the Find and Replace phrases should be:

Find phrase: `&#x003C;object.+?mp3=(\[&]+).+?&#x003C;/object&#x003E;`

Replace phrase: `&#x003C;audio controls="controls"&#x003E;&#x003C;source src="$1"
type="audio/mp3"/&#x003E;&#x003C;/audio&#x003E;`

5) mp3 player

Usually the inserted code looks like:

```
<object type="application/x-shockwave-flash" data="mp3player.swf?&mp3=mysoundfile.mp3" width="180" height="18" style="vertical-align:middle;"><param name="movie" value="mp3player.swf" /><param name="FlashVars" value="mp3=mysoundfile.mp3"></object>
```

The soundfile name is always preceded by mp3= and followed by " (html code: ";) , so the Find and Replace phrases should be:

Find phrase: `<object.+?mp3=([^\&|+).+?</object>`

Replace phrase: `<audio controls="controls"><source src="$1" type="audio/mp3"/></audio>`

6) player_mp3_maxi

Usually the inserted code looks like:

```
<object type="application/x-shockwave-flash" data="player_mp3_maxi.swf" width="51" height="20"><param name="wmode" value="transparent" /><param name="movie" value="player_mp3_maxi.swf" /><param name="FlashVars" value="mp3=mysoundfile.mp3&#amp;showstop=1&#amp;showinfo=1&#amp;width=51&#amp;showslider=0&#amp;buttonovercolor=9af000" /></object>
```

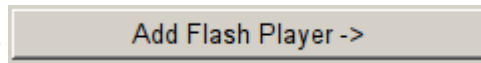
The soundfile name is always preceded by mp3= and followed by " (html code: ";) , so the Find and Replace phrases should be:

Find phrase: `<object.+?mp3=([^\&|+).+?</object>`

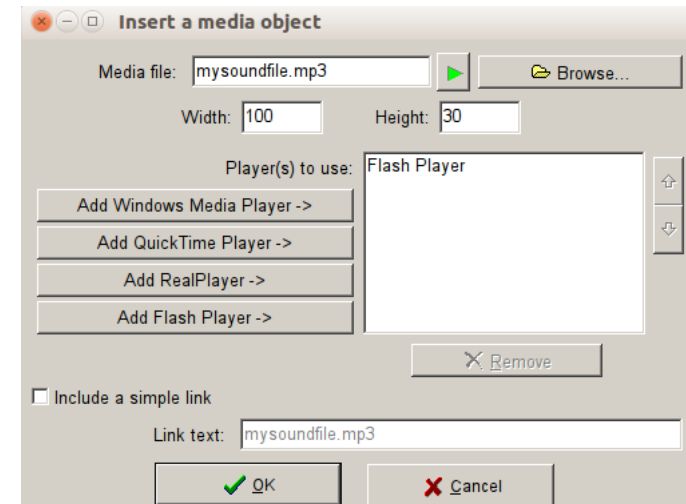
Replace phrase: `<audio controls="controls"><source src="$1" type="audio/mp3"/></audio>`

7) Flash player from the interface

For example if you have used



to insert an audio object:



Usually the inserted code is :

```
<object codebase="mysoundfile.mp3" type="application/x-shockwave-flash" width="100" height="30" data="mysoundfile.mp3"> <param name="movie" value="mysoundfile.mp3" /></object>
```

The soundfile name is always preceded by `codebase="` and followed by `"` (html code: `"` ;) , so the Find and Replace phrases should be:

Find phrase: `<object.+?codebase="([\^&]+).+?</object>`

Replace phrase: `<audio controls="controls"><source src="$1" type="audio/mp3"/></audio>`

Warnings

- First of all **make a backup copy of your files** and keep them in a safe folder.
- A little difference in your code (an added space for example) could lead to a complete mess in the converted data file. So always work on a copy and make tests with one or two files before converting all you files in batch mode.
- If you have used different SWF widgets in the same data file you can't use this method (unless the soundfile names are always preceded by the same characters – for example, if you have used different hbs players).
- If you have used a widget for audio files and a widget (the same or not) for video files you can't use this method.

What to do after conversion?

After the conversion of the data file you will **have to recreate every htm file**, one by one, using HP7. Hopefully it's quite repetitive: you can create a hundred of files per hour (it will be longer if you want to use an add-on for some of them).

So it's a lot of work but your old exercises will be up to date.