User manual for iAH – appendix – streamline videos

Ernesto Spinak v.1.0 provisional – december 2011

❖ Introduction

The default interface for consulting ISIS databases in ABCD as well as in BVS is the application iAH from Bireme, which uses the software wxis.exe and the language IsisScript. This document explains how to incorporate video presentations in real time (streamline videos) by means javascript and adaptation of the formatting instructions in the PFT file. The explanations apply without distinction to versions 2.x and 3.x of iAH.[1]

A video embedded in a web page is called *inline video*. There are basically two ways of presenting the videos:

a) Upload to a provider like Youtube and then display or open from the ISIS database using a special PFT that contains the HTML instructions.
The limitation is that Youtube accepts videos up to a maximum size which is currently 10 minutes duration.

b) Store the videos on your own server and use a videoplayer application also on your own server. You also need to include javascript code and to embed in the PFT format the necessary instructions in HTML to run the videoplayer. In this case the limitation will be the disk space and bandwith available.

Using Youtube

Let us suppose that we have a database that contains records with descriptions of videos which we require to make available online in real time (*streamline*). In the database we have defined the subfield **v866^u** to hold the name of the video file. The field can have various additional subfields, for example the title or name of the video, size in bytes, etc.

Follow this procedure:

(a) Upload the video to Youtube,

(b) Enter Youtube and check the URL for access to this video. For example the video is called http://www.youtube.com/watch?v=EgtExoc_Zfk&feature=related
However, this is the URL that Youtube uses to display the video; the part of the name which interests us is **EgtExoc_Zfk**

(c) Insert this data in the subfield **v856^uEgtExoc_Zfk**

(d) Include in the display format (PFT) of your ISIS database one of the two alternatives which follow with HTML code that permits seeing the video in your results page.

d.1) Youtube iFrame

```
'<iframe width="420" height="345"
      src="http://www.youtube.com/embed/', v856^u, '">
</iframe>',
```

[1] v.3.x: http://bvsmodelo.bvsalud.org/download/abcd/manuals/ABCD-IAH-3.1.0_en-OfficialVersion.pdf
v.2.x: http://bvsmodelo.bvsalud.org/download/iah/IAH-ManualUsuario-en.pdf

Note the use of the single inverted commas **'**. The field **v856^u** does not need to contain the complete URL because the start of the address of the host is always fixed. The dimensions of the video are adjusted with the parameters **width=** and **height=.**

d.2) Youtube embedded

In a similar way

```
'<embed
      width="320" height="300"
      src="http://www.youtube.com/v/',v856^u, '"
      type="application/x-shockwave-flash">
</embed>'
```

Using a videoplayer on your server

Adobe Flash Player is the classic professional software for displaying videos. However, there exist free alternatives with sufficient possibilities to provide the basic necessities for display in an ISIS database.

In the following example we will use the application **JW Player**[2]. If you want to have more advanced displays than the examples shown, then read the manual[3] of JW Player where three different methods of including videos in a web page are explained with many parameters that add flexibility to your presentation.

We explain the simplest form of installation, which does not require specialised programming knowledge, only a little acquaintance with PFT formats in ISIS.

For the simple procedure you need to follow these steps:

- Arrange a server to host the videos.
- Install the application JW Player on the server where the application iAH is.
- Add a javascript function to the file **ahead.pft** of iAH.
- Modifiy the display format of the files (PFT) to include a set of HTML instructions for the execution of the javascript.

---

[2] http://www.longtailvideo.com/players/jw-flv-player/. The application JW Player has a Creative Commons licence, which authorises to use, modify and redistribute the script, but only for noncommercial purposes.
JW Player is offered in two versions. We recommend to download the version "*non-viral*" by removing the tick as shown in the figure below. If this is not unticked, the application will have an option at the end of the video display that appears like the figure on the right and needs programming for sending email, and also this plug-in is not available in other languages.





[3] http://www.longtailvideo.com/support/jw-player/jw-player-for-flash-v5/12538/supported-player-embed-methods

Video server

The video files can be stored in any domain: it is not necessary that they are where the iAH application is.

Bear in mind that the principal bottleneck will be the bandwidth available on this server, and the size of the files which are to be stored. For example, to store a collection of **.flv** videos of between 20 and 30 minutes duration needs some 200MB each, so to store 150 videos requires some 30 GB of space on the server.

The application can contain as an option an initial still image (*initial splash image*) for presentation in the space for videos before they start, which may be the logo of the institution. This image is recommended to be stored with the videos. For the sake of example we will call it "***preview.jpg***" but you can assign any name.

Installing JW Player

- Create a directory called "*flashplayer*" in the root of your site, for example: **../htdocs/flashplayer.**
- Decompress the package ***mediaplayer-viral.zip*** below the site and save in the directory just created the files **jwplayer.js, player.swf, swobject.js**
  Attention: the files of jwplayer should be downloaded by FTP as binaries.
- Give Execute rights to the browser for these files.

Insert a javascript function into **ahead.pft**

In iAH version 2.x it is included in .. **/cgi-bin/iah/<lang>/ahead.pft**
In iAH version 3.x it is included in . . /**htdocs/iah/scripts/<lang>/ahead.pft**

The code should be embedded in the section <head> .. </head> of the file **ahead.pft**:

```
'Content-type: text/html' /#
'<html>
 <head>' /
     '<title>'
          ...
'<script type="text/javascript"
      src="http://<host>/flashplayer/swfobject.js"></script>'/

'<script type="text/javascript">
      swfobject.registerObject("player2","9.0.98","http://
<host>/flashplayer/expressInstall.swf");
  </script>'/
          ...
'</head>
...
```

Display format PFT

In the example, the field v856 is used to store the names of the videos. Observe that the subfield **v856^u** contains only the physical name of the file, WITHOUT including the route (path) to the

server where it is held. The format supplies the rest of the URL; by this means the database is independent of the server and can be moved if necessary.

```
 if p(v856) then

 '<tr><td align="right" width="13%" valign="top" nowrap>
     <font size=2 face="Arial">',
         '<b>See video:</td>',
     '<td valign="top" width="97%"><font face="Arial" size=2>',
       mpl,

'<div align="center" ><br>',
(
'<object id="player2"
            classid="clsid:D27CDB6E-AE6D-11cf-96B8-444553540000"
             name="player2" width="300" height="225">
  <param name="movie" value="http://<host>/flashplayer/player.swf" />
  <param name="allowfullscreen" value="true" />
  <param name="wmode" value="transparent" />
  <param name="allowscriptaccess" value="always" />
  <param name="flashvars"
  value="file=http:// <host>/Videos/', v856^u,
  '.flv&amp;autostart=false&amp;controlbar=over&amp;image=http:// <host>/Videos/
preview.jpg" />

   <object type="application/x-shockwave-flash"
           data="http://<host>/flashplayer/player.swf"
           width="300" height="225">
    <param name="movie" value="http://<host>/flashplayer/player.swf" />
    <param name="allowfullscreen" value="true" />
    <p><a href="http://get.adobe.com/flashplayer">
                        Get Flash</a> to see this  player</p>
   </object>

</object><br><br>'
          )
 '</div>',
 '</td></tr>',

 fi,
```

❖ Acknowledgments

I have to thank Ing. Diego Muses, expert in IT, for help and cooperation in resolving this problem.

For the translation, Dr. Andrew Buxton (UK)