Best Practices Report
October 27, 2011

Submitted by Riley McLaughlin

Project

Voices of the First People: Creating Website Access for the Presentation of Audio and Video Recordings from the Vi Hilbert Collection, a project of Northwest Heritage Resources and the Ethnomusicology Archives of the University of Washington.

The project was to design and build an on-line application that allows for the selection and playing of a specific group of audio and video recordings from the Vi Hilbert Collection. Users would be able to select recordings by category and by name. Audio and video playback would be controllable by users, and each recording would be accompanied by the display of its title and accompanying descriptive text. The application would be installed on a website hosted on the Ethnomusicology Archives at the School of Music, University of Washington, and would be easily updatable by the staff at the archives.

Early planning

The first planning meeting at the University of Washington established the basic requirements of the project: that user-selected audio and video clips from the archive would be easily accessed and played from the website, and that the clips would be searchable by categories and possibly other criteria. Users would be able to control the playback of the files, including volume, video picture size and play/pause functions. The audio and video would be provided to the consultant on CDs and DVDs by the archive staff and the consultant would prepare them as necessary for use on the website (including converting files to the appropriate file-types and optimizing them for efficient web streaming). The original intention was to build the application using Flash, which has become a standard for cross-platform access to audio and video through all the major browsers.

As the project developed, it became clear that it would be important to make the content available to the segments of our audience who use mobile devices – smart phones, iPads and other internet-connected digital tablets. Many of these devices do not support Flash and the decision was made to build the application with non-proprietary internet technologies that are now available in modern browsers on all platforms, including the mobile devices. This approach would allow much wider access to the Vi Hilbert collection as well as provide greater flexibility in updating the application and adding new features in the future without having to return to the original Flash code and recompile it into Flash Player .swf files.

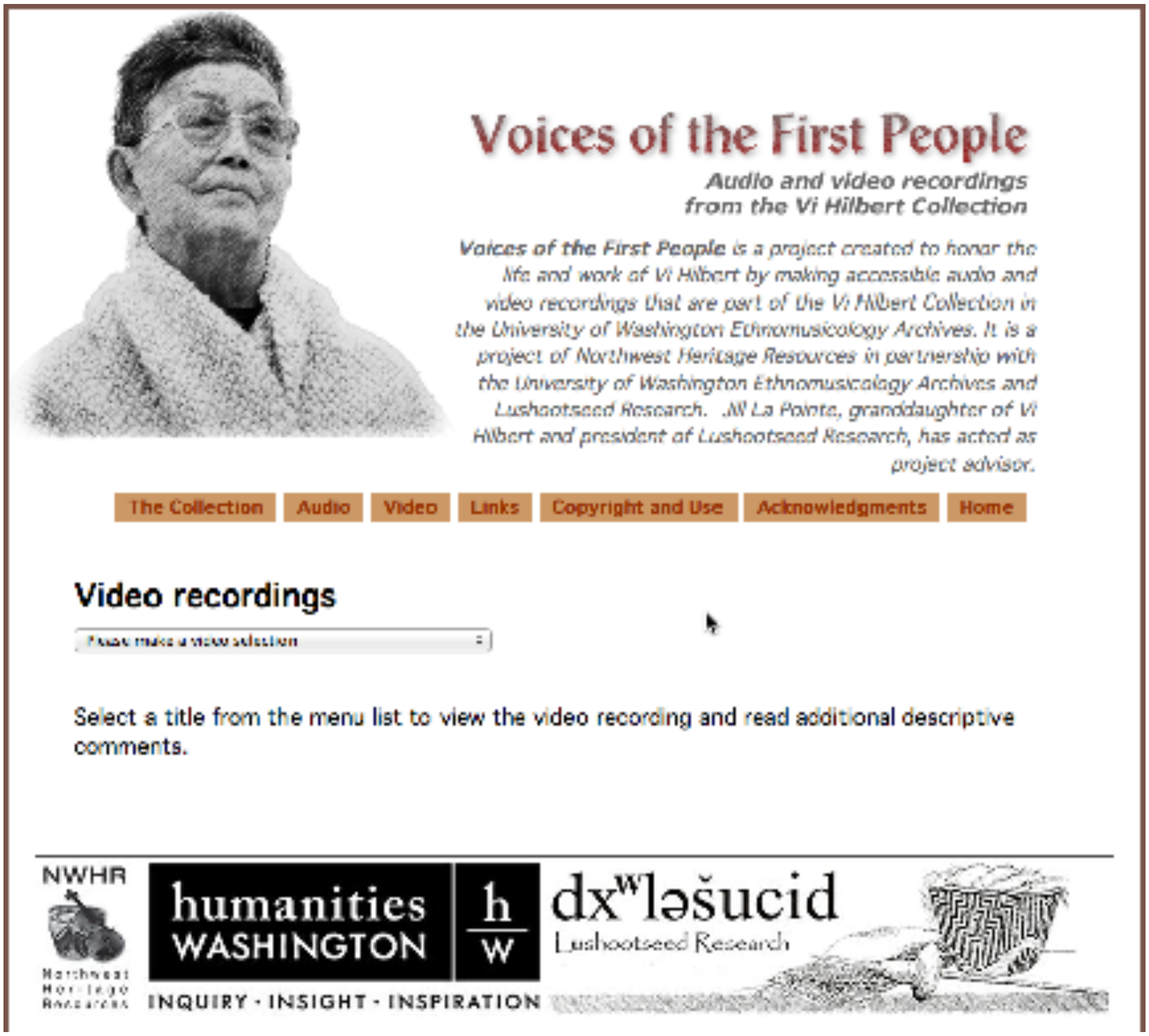
Work was then done to assess the feasibility of building the application using the audio and video capabilities currently available with the latest version of HTML. Though some of new features of HTML5 are not yet consistently supported by all the major browsers, *all* the modern browsers do support the core capabilities of HTML5, including interactive native audio and video playback. These A/V capabilities require no special plug-ins (such as the Flash player), and offer a more efficient and effective means of providing this content.

HTML5

HTML is a language for structuring and presenting content for the World Wide Web. HTML5 is the most recent implementation of the language and adds many new syntactical features. The new features key to our application are the <video> and <audio> tags, making it easy to add multi-media content without having to resort to proprietary plugins such as Flash. HTML5 is supported by mobile devices and tablets.

Our application centers around a page that offers a selection of video recordings and a second page offering audio recordings. Other pages in the site offer text and photos using standard html approaches.

The video page, shown below, presents a drop-down menu containing the list of recordings organized into useful categories. When a selection is made, the video is shown on the same page along with its title and caption. All video selections are made and presented through the single html page.



Voices of the First People

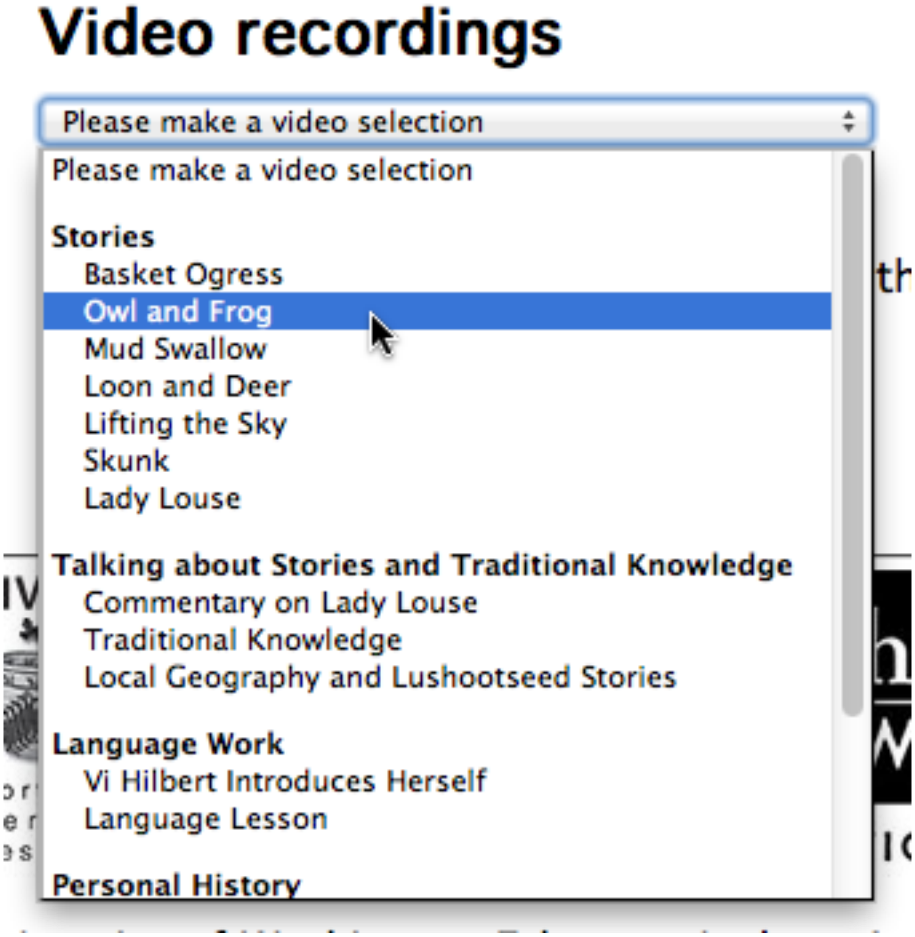
Audio and video recordings from the Vi Hilbert Collection

***Voices of the First People** is a project created to honor the life and work of Vi Hilbert by making accessible audio and video recordings that are part of the Vi Hilbert Collection in the University of Washington Ethnomusicology Archives. It is a project of Northwest Heritage Resources in partnership with the University of Washington Ethnomusicology Archives and Lushootseed Research. Jill La Pointe, granddaughter of Vi Hilbert and president of Lushootseed Research, has acted as project advisor.*

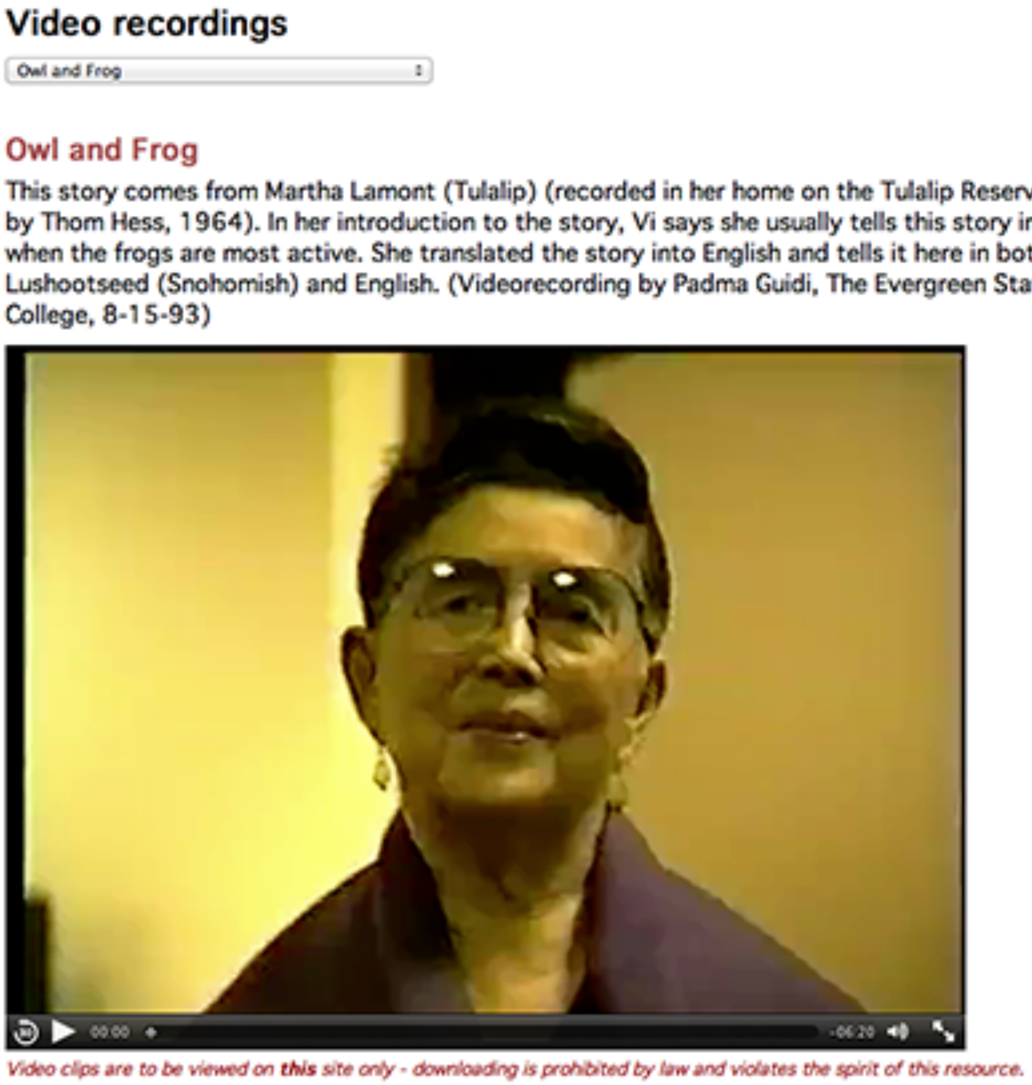
Video recordings

Select a title from the menu list to view the video recording and read additional descriptive comments.

The selection is made from a list organized by categories:



The html uses <div> tags to present the video. When a selection is made, the video is retrieved from the server and displayed in the "player" div and the title and captions are shown in their own mediaTitle and caption divs:



Video recordings

Owl and Frog

Owl and Frog

This story comes from Martha Lamont (Tulalip) (recorded in her home on the Tulalip Reservation by Thom Hess, 1964). In her introduction to the story, Vi says she usually tells this story in April, when the frogs are most active. She translated the story into English and tells it here in both Lushootseed (Snohomish) and English. (Videorecording by Padma Guidi, The Evergreen State College, 8-15-93)

*Video clips are to be viewed on **this** site only - downloading is prohibited by law and violates the spirit of this resource.*

When new selections are made, the appropriate text and video replace any existing text and media, allowing the user to view selections in any order without leaving the page. The audio page works in the same way, presenting user-controllable audio selections with titles and captions.

Audio and video file types

Though the different browsers support playing audio and video, they do not all support the same file formats. The reasons for this situation are not our concern, but to successfully deliver the media content to all our users, we needed to convert our original audio and video files into types that will work for any browser the user chooses.

The major browsers support one or more of three video file types: MPEG4 (.mp4/H.264), supported by Safari, Google Chrome and Internet Explorer 9; Ogg Vorbis (.ogg or .ogv), supported by Firefox and Opera; and .webm (supported by Google

Chrome, Firefox and Opera). The browsers all support one of two audio file types: .mp3, supported by Safari, Google Chrome and Internet Explorer 9; and .ogg, supported by Firefox.

For our project, we converted each video file into .mp4/H.264, .ogg, and .webm formats. The HTML5 code manages the selection of the appropriate file type based on which browser is in use, making this aspect transparent to the user.

There are a number of applications, some of them available for free, that can be used to convert the original video and audio files into the required types. Our project used the Hewbo Video Converter (free from http://www.hewbo.com/free-video-converter.html) and the Miro Video Converter (free from http://www.mirovideoconverter.com/) to convert the video files. We used Roxio's Toast Titanium to convert audio files to mp3 and OGG Vorbis formats.

Flash Fallback

While recent Safari, Firefox, Chrome and Opera browsers support audio and video tags, Internet Explorer only began supporting these HTML5 elements in version 9. Many Internet Explorer users are still using IE8 and earlier versions, and it was necessary to provide our media content to these users (and those of other older browsers). We employed a "Flash fallback" approach that resorts to delivering the content through a generic Flash player if the browser cannot handle the HTML5 code or file types. The player can play .mp4 /H.264 video files and mp3 audio files, so the fallback works with files already converted and available.

HTML5 audio and video tags

HTML5 audio and video elements are straightforward. The simple form is:

```
<video controls>

        <source src=”video/videoFile.mp4”>

        <source src=”video/videoFile.ogg”>

        <source src=”video/videoFile.webm”>

        <p>This browser doesn’t support the video tag.</p>

</video>
```

When the browser encounters this code, it steps through each line looking for a file type it can use. If it doesn't recognize the file type in the line, it looks in the next line. When it finds a file type it can use, it displays the video with a controller. If the browser does not find a file it can use or if it does not support the <video> tag, it processes the last line in the tag, telling the user the browser doesn't support the code it found. We can replace the last line in the above example with code to show the mp4 file with a flash player. The Flash .swf file we use is provided on-line from a free Adobe service through a URL inserted in the code.

In the Vi Hilbert project, the video file is selected by the user from a drop-down menu, and the <video> tag code is built on-the-fly by a "setVideo" function which receives the selected filename when the choice is made. The function uses the filename to set variables to the complete filenames and paths, then builds the <video> tag and inserts it into the player <div> in the html.

```
function setVideo(fileName) {
      //set variables to hold the three filenames and types
      var filepath1 = "video/" + fileName + ".mp4";
      var filepath2 = "video/" + fileName + ".ogg";
      var filepath3 = "video/" + fileName + ".webm";
      var type1 = 'video/mp4';
      var type2 = 'video/ogg';
      var type3 = 'video/webm';
      //build the html text and insert into the player div
      //create the video controls with flash fallback code as the
      //last item
      $('#player').html('<video controls width="654"
height="480"><source src="' + filepath1 + '" type="' + type1 + '"
/><source src="' + filepath2 + '" type="' + type2 + '" /><source
src="' + filepath3 + '" type="' + type3 + '" /><object
width="654" height="480"> <param name="movie"
value="http://fpdownload.adobe.com/strobe/FlashMediaPlayback.swf"
></param><param name="flashvars"
value="src=http%3A%2F%2Fmusic.washington.edu%2Fdev%2Fethno%2F'+
filepath1
+'&poster=http%3A%2F%2Fmusic.washington.edu%2Fdev%2Fethno%2Fimage
s%2FVideoPoster.jpg"></param><param name="allowFullScreen"
value="true"></param><param name="allowscriptaccess"
value="always"></param><embed
src="http://fpdownload.adobe.com/strobe/FlashMediaPlayback.swf"
type="application/x-shockwave-flash" allowscriptaccess="always"
allowfullscreen="true" width="654" height="480"
flashvars="src=http%3A%2F%2Fmusic.washington.edu%2Fdev%2Fethno%2F
'+ filepath1
+'&poster=http%3A%2F%2Fmusic.washington.edu%2Fdev%2Fethno%2Fimage
s%2FVideoPoster.jpg"></embed></object><p>(Users viewing with
Internet Explorer 8 and earlier may experience a pause as the
video loads before it will begin playing.)</p></video>');
}
```

Interactive content selection

The interactive searching and selection components of the application were written with JavaScript, using a combination of technologies known as AJAX (Asynchronous JavaScript and XML). The AJAX methods (available in the open source jQuery library found at api.jquery.com/jQuery.ajax) allow interactivity between the HTML page and the server such that users can make selections and see server responses without requiring distracting page reloads. Content is brought to the user immediately as it wouof be in a stand-alone application (or an application running in the Flash Player). JavaScript code and AJAX methods are supported by all the major browsers.

The data required by the application is kept in XML (Extensible Markup Language) documents, one for audio content and one for video. These files hold the required data for each clip – title, description, search terms and file names. The XML file is a plain text document that organizes content into a simple structure readable by the browser. XML is created and edited with a plain text editor, and content managers are able to easily change and make additions to the content by editing this document and adding any new clips to be shown. The use of XML allows building the application without requiring a server-side database, making the project more portable and adaptable for different servers.

When the video html page loads, a javaScript function creates an object to hold caption text indexed for retrieval when selections are made. It then gets the XML file and parses it, loading the caption object with caption text and loading a drop-down menu with categories, titles and clip file names.

The drop-down menu is assigned a change function which is invoked when a selection is made. This function displays the title and caption in the appropriate divs and calls the setVideo function that uses the clip file name to display the selected video in the browser.

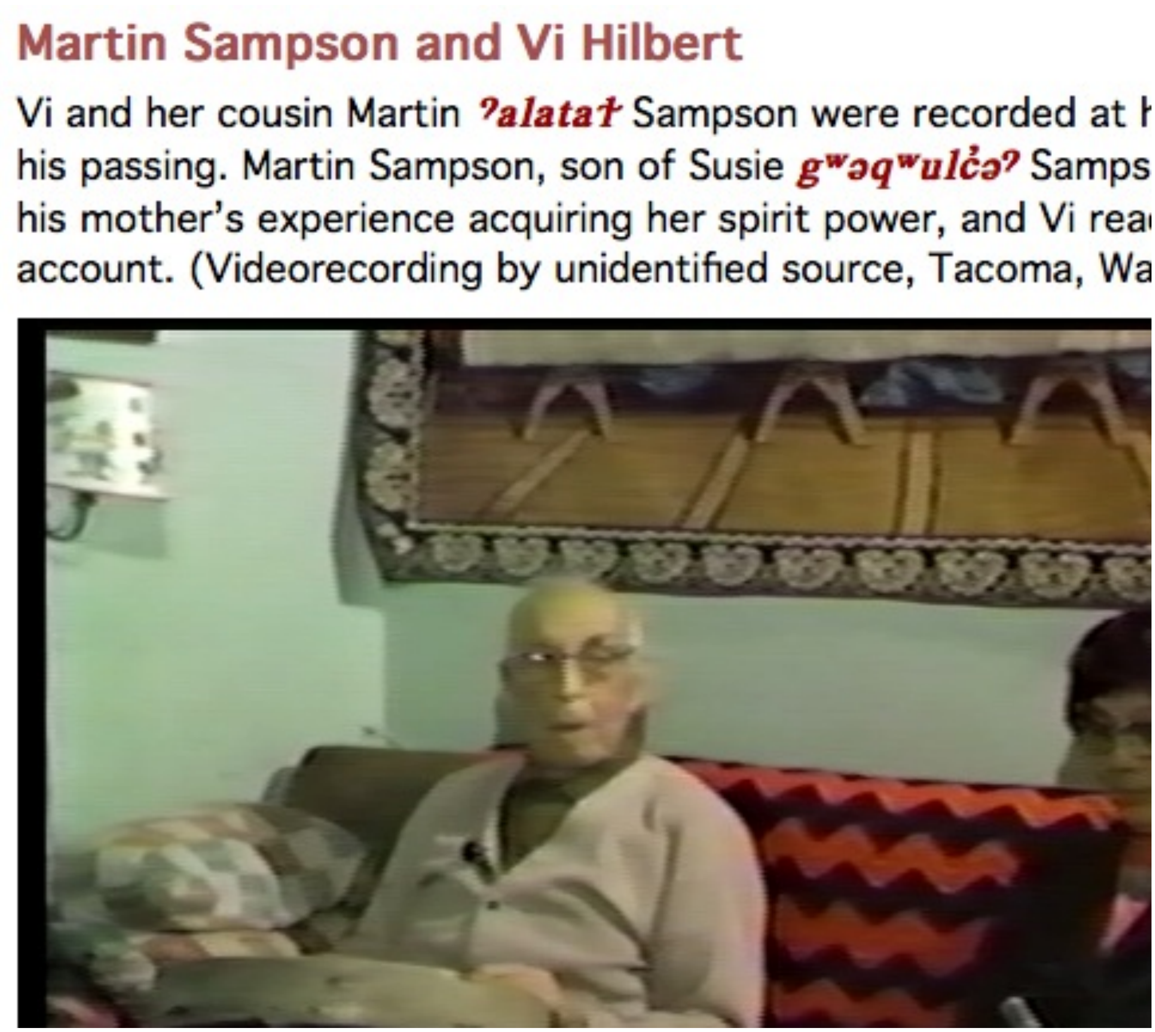
The audio page performs the same functions using the audio XML data file, and manages the audio file types. Audio is played natively in the recent browsers and in the case of browsers that don't support the html5 <audio> tag, is played through the same Flash fallback methods and .swf file as the video.

Lushootseed language support

An additional advantage of HTML5 and CSS3 is the ability to embed fonts into the pages. Our project includes some text that appears in the native language Lushootseed, which uses a font not found on most user's machines. Previous methods of handling special fonts usually required rendering text in those fonts as graphics. In our application, the Lushootseed content is included in the plain text XML documents and needed the same flexibility in handling as the rest of the text.

HTML5 supports embedded fonts through the @font-face selector. Text can be rendered in special fonts by storing the fonts on the server in a folder with the other site files, and using CSS selectors to style the text with the stored fonts. As with the media files, different browsers require different font formats: true-type (.ttf) for most browsers and Embedded OpenType (.eot) for Internet Explorer. Code for managing this works like the media code. Including the different font files on the server and letting browsers use the appropriate types works well.

Text intended to be displayed in Lushootseed is written in English fonts in our XML files, but is preceded by the ^ character and followed by a ~ character. A small javaScript function was written to look through text to be displayed, and when the ^ is found, the "nativelanguage" styling rule is applied until the ~ is found and the font style reverts to English as in the following example:



Martin Sampson and Vi Hilbert

Vi and her cousin Martin ***ʔalataƚ*** Sampson were recorded at h
his passing. Martin Sampson, son of Susie ***gʷəqʷulč̓əʔ*** Samps
his mother's experience acquiring her spirit power, and Vi rea
account. (Videorecording by unidentified source, Tacoma, Wa

A note about copyright and downloadable files

While this method satisfies the need to provide access for a wide audience on multiple platforms, it's important to understand that playing media with HTML5 is not a streaming technology. The audio and video files are temporarily downloaded for playback, and with some browsers, the clips can be easily saved to the local machine. In some cases this raises a concern about copyrighted material, and there is a fair amount of discussion in the development community about whether or not the technology should be altered to handle copy-protected media. In our project we include a page about copyright and use, which includes the statement:

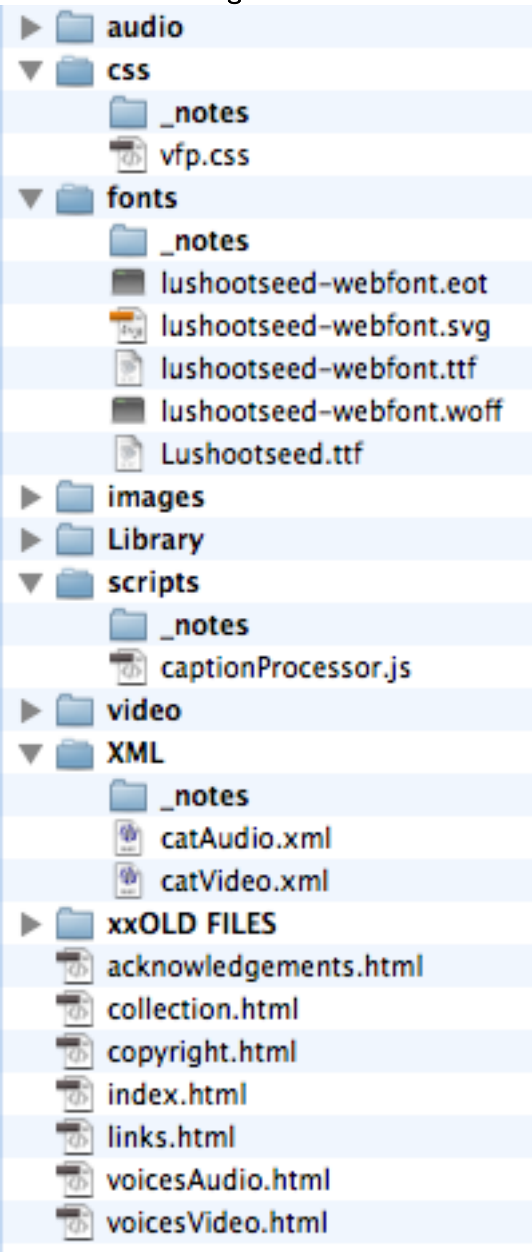
"The audio and video examples in the Vi Hilbert database are intended for educational and instructional use only. Downloading of these files is prohibited.
To obtain permission from the copyright holder for commercial use of any of these images, contact…"

In addition, we added text that appears immediately below each clip that states: Audio (or video) clips are to be heard on this site only - downloading is prohibited by law and violates the spirit of this resource.

There is no way to prevent the capture and saving of any audio or video if a site visitor is intent on doing this. Even the use of streaming media in Flash is susceptible to various applications designed to capture media as it streams through a machine. This is an issue that will continue to be debated for some time to come, but in our case, it was decided that the material was important enough to make it available now in the most accessible way possible.

The final product

Commented example code is provided with this report that shows how the html, css, javaScript and XML documents work together to manage the playing of media and handling of the native font in the site. The folder structure of the site is shown at the end of this document. Unless new features are added, the code doesn't require changes when adding, deleting or editing the media content of the site. The *Voices of the First People* site has been tested and launched, providing a non-proprietary and flexible solution to making this valuable content available to a large audience.



```
▶ audio
▼ css
      _notes
      vfp.css
▼ fonts
      _notes
      lushootseed-webfont.eot
      lushootseed-webfont.svg
      lushootseed-webfont.ttf
      lushootseed-webfont.woff
      Lushootseed.ttf
▶ images
▶ Library
▼ scripts
      _notes
      captionProcessor.js
▶ video
▼ XML
      _notes
      catAudio.xml
      catVideo.xml
▶ xxOLD FILES
  acknowledgements.html
  collection.html
  copyright.html
  index.html
  links.html
  voicesAudio.html
  voicesVideo.html
```