

This document describes the sound/audio architecture and the related structure map to be used in R1.1. In addition, recent updates include the ability to handle an optional transcript as a separate manifestation within the sound object. These notes will as well identify the changes to be made for R1.1 in order to support this object architecture. These notes reflect numerous discussions among members of sw_arch and the repository steering committee.

Sound Object Architecture

As in the IJS project and perhaps others using digital sound, multiple files may be created that are logically part of one information resource, for example a single interview. In the case of the IJS oral histories, one interview, captured in .wav form, can span as many as 10 or 11 CDs. To create the recommended architecture of Figure 1, we extract the .wav files from each CD and compress them as a .shn file. Shn (Shorten) is a file format used for lossless compression of CD-quality audio files (44.1 kHz 16-bit stereo PCM) and can produce files that are in the range of 50% of the original file size. If an interview spanned, for example, three CDs, we would then tar the resulting 3 .shn files into a single tar file which becomes the archival master.

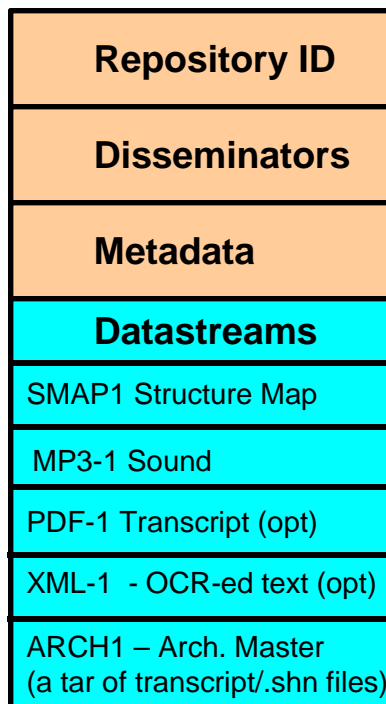


Figure 1 – Sound/Audio Object

For the sound presentation format, a single mp3 file is created from the .wav files as shown in Figure 1. This single mp3 can be presented in different ways depending on the structure map (see next section).

Structure Map

Initially, the structure map will be default and disseminator code will not be added to the object. We may, in later revisions, decide to implement a more sophisticated disseminator that would provide a “table of contents” like functionality for the sound object. In the IJS oral histories, there is very little logical structure to the interview. In this case, we will have a default structure map and the user will not be required to provide any structure in the WMS input process. A default disseminator will allow the user to play the sound file in 5, 10, or other generic time increments.

The capability for a custom structure map should be built into WMS in R1.1. The user should be able to provide a logical file structure, very similar to the table of contents for a book. A sample of a logical structure map for a sound file is shown below. A custom disseminator will provide the user the ability to select any logical segment to be played. For example, if the user selected “Highlights”, an mp3 file containing the segment of the oral history from 5 minutes to 13 minutes would be extracted and played (see div1.2 below).

```
<METS:structMap TYPE="logical">
<METS:div ID="div1" FORMAT="OralHistory" TYPE="audio"
LABEL="Music of Mary Lou Williams">

    <METS:div ID="div1.1" TYPE="" LABEL=" Music of Mary Lou
Williams " ORDER="0">
    <METS:fptr>
    <METS:area FILEID="" BEGIN="00:00" END="2:17:00" />
    </METS:fptr>
    </METS:div>

    <METS:div ID="div1.2" TYPE="" LABEL="Highlights"
ORDER="1">
    <METS:fptr>
    <METS:area FILEID="" BEGIN="05:00" END="13:00" />
    </METS:fptr>
    </METS:div>

    <METS:div ID="div1.3" TYPE="" LABEL=" Bringing Lazarus
Back from the Dead " ORDER="2">
    <METS:fptr>
    <METS:area FILEID="" BEGIN="00:00" END="52:00" />
    </METS:fptr>
    </METS:div>

    <METS:div ID="div1.4" TYPE="" LABEL="Tell Them Not to
Talk too Loud" ORDER="3">
    <METS:fptr>
    <METS:area FILEID="" BEGIN="53:00" END="1:39:00" />
    </METS:fptr>
    </METS:div>
```

```

<METS:div ID="div1.5" TYPE="" LABEL="I have a Dream"
ORDER="3">
<METS:fptr>
<METS:area FILEID="" BEGIN="1:40:00" END="2:17:00"/>
</METS:fptr>
</METS:div>

</METS:div>
</METS:structMap>

```

Text Transcripts for Audio Files

Some, perhaps many, of the sound objects may have a text transcript. It is the consensus that we consider this text file as another manifestation. For presentation purposes, the text transcript will be captured in PDF format and will be another datastream in the sound object (see above figure). The archival master for the transcript can be .txt, .rtf, or .xml and this file should be encapsulated in the tar file of the ARCH1 datastream. Note that the xml text that is used for full text searching can be either supplied as an upload or the text file can be generated automatically by having the pipeline run ocr on the images of the transcript (this pipeline capability to be included in R2.0). This approach is equivalent to how we are treating the ocr-ed text for books.

Architecture-related Metadata

The architecture related fields that must be included in the metadata are: 1) The typeOfResource is "Sound", 2) the objectArchitecture is "Audio", 3) objectArchitecture is replicated in the first level div TYPE attribute of the structure map (see below) and 4) the genre is "oral histories" for IJS. There may be other genres.

```

<METS:mets xmlns:METS=http://www.loc.gov/METS/
xmlns:loc="http://www.loc.gov/">
  <METS:structMap ID="1" TYPE="logical" LABEL="default">
    <METS:div ID="S1-1" ORDER="1" TYPE="audio">
</METS:div>
  </METS:structMap>
</METS:mets>

```

Datastream IDs

In the discussion of the sound/audio object architecture, we also addressed naming conventions for datastreams. We decided to move to a functional naming scheme in R1.1 for sound objects only (note: R2.0 will implement functional names for all objects). The generic approach is as follows: each datastream will have a functional name (e.g. smap, arch, mp3, pdf, etc) concatenated to a numeric representing the nth datastream of this type with an intervening hyphen, i.e. functional name || - || numeric. Examples are as follows: SMAP-1, ARCH-1, PDF-1, XML-1 MP3-1, etc. Note for R1.1 sound objects we need to retain the current naming for structure maps and archival masters – there will be no hyphen for these datastreams in R1.1. Figure 1 shows the correct names to be used in R1.1.