

The Challenge of Audio and Video

Audio and video bitstreams can be very large, and take a long time to download, even on a fairly fast network. Yet users have been conditioned to expect nearly instant access to them on other services. How can we provide these voluminous documents in a way that makes them pleasant to use?

Who's interested?

Add your contact info. here if you want to join the discussion, whether you have a suggestion or not.

[Mark H. Wood](#),
[Dsalo](#),
[Byhisdeeds](#),
[John Davison](#),
[Pkeane](#),
[Jstimaman](#),
[Canderson34](#),
[ejansson](#),
[chodge5](#),
[Joseph John](#)
[Kshepherd](#)
[Colleen Greene](#)

[Brian Davis](#)

Issues

Flash video callouts

Flash video can apparently act much like HTML, including relative links to external resources such as images and soundfiles. The existing HTML-management code (primary bitstream etc.) might be extensible to cover this situation.

Flash's .flv file format allows for a progressive download of video data, which provides a sometimes acceptable alternative to streaming servers for smaller files.

Ideas

Wrapper document pointing to a streaming service

I've experimented briefly with storing as one bitstream of a DSpace item a minimal [SMIL](#) wrapper which points to the content on a streaming service. Like so:

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<smil xmlns="http://www.w3.org/2001/SMIL20/Language">

  <body>
    <video src='rtsp://129.79.6.46:554/vic/david_lewis_20060228.rm' />
  </body>

</smil>
```

For preservation purposes, it would be good to deposit a copy of the material in DSpace itself as well. That way you have a "slow" copy associated with metadata, and a "fast" copy that streaming clients can access quickly.

DSpace 1.5 XMLUI and Progressive FLV Video Download

This experiment is described on the [DSpace 1.5 XMLUI FLV Video Progressive Download](#) page.

Discussion elsewhere

http://sourceforge.net/mailarchive/forum.php?thread_name=592efc80804020606n5ca2ec6dj3190cced758b55c7%40mail.gmail.com&forum_name=dspace-tech