**Migration Tool:**

* bibliography notes didn’t migrate. We had 22 bibliographies in 34 files and not a single one made the migration.
* @target attribute not updated, even though the @id elements are changed during the migration process. Example: @ref685 in the AT became ref685\_qpv in ArchivesSpace. When this is exported in the EAD, it becomes @aspace\_ref685\_qpv, but any targets in the rest of the file still try to reference “ref685,” which no longer exists.
* @id on archdesc/did/abstract not copied over to ASpace. Not sure how pervasive this is, but it doesn’t look like any of the notes have @id to reference, which is definitely different than how the AT would allow you to make references to, for example, one of three different bioghist notes in a series component.
* Can the migration tool add the <p> elements? ASpace seems to require those. Here’s an example:

<scopecontent>
 <head>Description of the Papers</head>
 <p>The collection comprises…

 Some time after the collector gathered…

 Because the collector separated these letters…
 </p>
</scopecontent>

Just one big paragraph! This should be 3 paragraphs, though.

* Namespace prefixes. Since ASpace uses the xlink prefix, can the migration tool update all instances of “ns2:” to “xlink:”?
* Internal only attributes not migrated properly. I didn’t test this thoroughly, but I created one test record that had two abstracts… one marked internal only, the other not. Both were migrated as “published”.
* I didn’t test this thoroughly yet, but not every field in a Repository record made the migration. Example: most of the address information didn’t migrate. This isn’t that big of a deal, though, since we’d only need to fix this once for each repository. So, this should be considered a very low priority fix.

**ArchiveSpace EAD export problems:**

* Accessrestrict notes are not exported! (This problem still continues with version 1.0.7)
* Legal status notes are not exported. (This problem still continues with version 1.0.7)
* Index notes: The ref element needs to be exported after the “item value”. Right now, it’s exporting as the first element of indexentry, and this results in an invalid EAD file.
* Doesn’t always include the xlink namespace prefix when it should, resulting in invalid EAD files. I haven’t had the time to locate when and where these problems occur yet.
* “Finding Aid Title” field is not exported (instead, the collection “Title” is repeated in its place).
* “Finding Aid Note” is not exported as part of the eadheader.
* “Finding Aid Date” is not exported as part of the eadheader.
* @id and @parent are not included on container elements in the EAD export, but they were with the AT. There is NO way to group instances any longer once an EAD file is created. This is a problem if we want to use multiple instances at any one level of description.
* @relator is used instead of @role. This might only be done at the collection-level, though. Still, it makes for invalid EAD records.
* <list numeration="null" type="ordered">

This is invalid EAD, since @numeration cannot equal something like null. Instead, this attribute should be left off entirely… or, ArchivesSpace should drop @type=’ordered’. In the AT, you could not create a list without @type, so there was no way to really specify an “unordered” list. Therefore, in the AT, I’ve always considered unordered lists to be those lists that have @type=”ordered” but no @numeration attribute.

However ArchivesSpace wants to handle this (and I hope that it improves upon how the AT did), it should \*not\* export anything like @numeration=”null” in an EAD file, since this invalidates the EAD.

* ArchivesSpace will export a note, like scopecontent, by wrapping a <p> element around it if and only if that note does not start with another XML element. Example export problem (note the lack of a paragraph element):

 <scopecontent><head>Scope and Contents</head><emph render="italic">The Maker's Hand</emph>…

I presume that this is to counter when a multi-paragraph note would start with the <p> element. But this causes problems, as illustrated above, and no paragraph at all is exported. Can ArchivesSpace handle paragraph tagging the same way as the AT (i.e., that was the only element that you didn’t have to “wrap in tag”), or can we move to a more fully-realized WYSIWG type of interface where we don’t need to wrap things in emph tags just to make them bold, etc.???

* Mixed content doesn’t export very well where ArchivesSpace doesn’t expect said mixed content. On top of that, there’s a strange bug that affects how certain characters are encoded upon export. Here are two problems that I noticed with mixed content elements in unitittle fields:
1. When a component has a container element, mixed content is exported like so in the unittitle element:

<unittitle>&amp;lt;emph&amp;gt;Carnet Pass Holder&amp;lt;/emph&amp;gt;</unittitle>

1. When a component does not have a container element, mixed content is exported like this:

<unittitle>&lt;emph render="bold"&gt;London Transport Museum ticket&lt;/emph&gt;</unittitle>

Note the difference between &amp;lt and &lt; in the text. In any event, both exports are wrong, since these should be exported as XML, not text.

That said, an archivist should still be able to include either the “<” or “>” characters in a text node, if needed. However, only the “less than” symbol would need to be translated to a character encoding like &lt; (in other words, it’s fine to keep the > as is in any XML text node).

**ArchivesSpace editing interface / database problems:**

* No place to create a finding aid subtitle. The migration tool moves this over to the “Finding Aid Title” field, which is the best place for it to go, but alas, this field isn’t even exported in the EAD right now.
* Wrap in tag is not available in a variety of places that it should be, especially the unittitle field (which often needs to have a nested title element, for example).
* Wrap in tag problem: with extref, for example, you can add @target (which doesn’t belong), but you cannot add @xlink:href (which does belong). Haven’t checked the rest of the mixed content elements, but this is a problem.
* Would it be possible to attach user defined fields to an archival\_object? Right now, it looks like you can only use the user defined fields for accessions, resources, and digital objects (and I wonder why components and/or instances were left out of this list).