Revision notes

Introduction

The draft for EAD 4.0 (published on 19 April 2024) is the result of a major overhaul of EAD3 version 1.1.1.

- This revision looks at **aligning** EAD **with** its sibling standard, the **Encoded Archival Context - Corporate Bodies, Persons, and Families (EAC-CPF)**, to enable a smoother side-by-side use of both, as well as with other standards often used in relation to EAD. This not only relates to the encoding options, but also to the **terminology used in the names of elements and attributes**, which have been adapted to use a more generally applicable and accessible language. The alignment specifically with EAC-CPF also lays a lot of groundwork for other changes applied in EAD 4.0, which is the reason why these changes make a significant part of what is described below.
- While EAD 4.0 continues following ISAD(G) in terms of the elements of archival description that it includes, the revision also approaches **EAD's relation to Records in Contexts (RiC)**, version 1.0, which was released in December 2023. RiC replaces the existing ICA standards such as the General International Standard Archival Description (ISAD(G)), for which EAD currently provides an XML-based implementation approach. Re-evaluating EAD's relation to the ICA standard hence becomes pivotal. The Encoded Archival Standards (EAS) may, as a whole, in their future versions (among which EAD 4.0) be considered an XML-based representation of the RiC Conceptual Model alongside the RiC Ontology.
- Furthermore, this revision looks at how EAD can better support all the developments in terms of **Linked Open (Usable) Data**, which makes the Internet tick nowadays. While EAD3 already does this to some extent, possibilities for improvements have been brought to the table during conversations with the community throughout the past years.

Version control

The EAD3 schema was first released in August 2015 in version 1.0.0, with a minor release in April 2018 (version 1.1.0) and a batch release in December 2019 (version 1.1.1). Following the semantic versioning scheme introduced as part of the Technical Subcommittee on Encoded Archival Standards' (TS-EAS) rolling revision cycle (established in December 2019), this new major version is going to be EAD 4.0.

EAD 4.0 is a major version that will not be backwards compatible. Future minor revisions of EAD 4.0 with added functionality in a backwards compatible manner will become EAD 4.1 resp. EAD 4.2 etc. and patch releases with backwards compatible bug fixes will become EAD 4.0.1 resp. EAD 4.1.1 etc.

Deprecated elements and attributes of EAD3

EAD3 was released in a deprecated and an undeprecated version. EAD 4.0 has now removed the deprecated elements and attributes completely. This includes:

- <bibseries>
- <descgrp>
- <div>
- <extent>
- <frontmatter>
- <imprint>
- <runner>
- <titlepage>
- @placement
- @tpattern

Furthermore, the revision only took the deprecated content models into account for:

- <physdesc>, i.e. without <extent>, <physfacet>, or <dimensions>, without the access elements and without <date>, <footnote>, <num>, or <quote>;
- <unittitle>, i.e. without <imprint>, <bibseries>, <edition>, or <unitdate>.

Spelling and naming of elements and attributes

Following its sibling standard EAC-CPF, EAD 4.0 will introduce camelcase spelling for element and attribute names consisting of two or more words. This also adheres to best practices in XML definitions and makes the XML tagging easier to read for the human eye, especially considering the broad international community of EAD users, many of whom do not speak English as their first language. Better readability also supports better recognisability of an element or attribute’s name’s origin and makes EAD easier to learn.

This change also provided the opportunity to review the names of elements and attributes more generally for better alignment with the terminology used in related standards such as ISAD(G), RiC, or Describing Archives: A Content Standard (DACS), but also in collection management systems that support EAD as an exchange format. Similarly, this brought the chance to review abbreviated names of elements and attributes and to consider more concrete and more easily understandable names. It should be noted that not all abbreviated names of elements and attributes have been changed, as TS-EAS felt that some of these have been established and well-understood for two decades now.

List of removed, renamed or replaced elements and attributes

Elements

In total, the number of elements is down to 119 in EAD 4.0 from 166 in EAD3.

Of the 166 elements in EAD3, only 1 has been removed without substitution, while 69 have been replaced by or integrated with other elements. Furthermore, for 46 elements camelcase spelling has been applied and 13 additional elements have been renamed. Of these, 7 would have changed name anyway due to the introduction of camelcase spelling, 1 was renamed for alignment with EAC-CPF and 1 (namely <script>) needed to be renamed to avoid potential security vulnerabilities. The table below groups all of these elements into four sections and lists them alphabetically, including their counterparts in EAD 4.0.

For the group of replaced elements, some EAD3 elements might be listed more than once, in case different replacements apply depending on the context in which they are used (e.g. depending on whether <corpname> is used as sub-element of <origination> or as a mixed content element). Furthermore, only the element replacing the wrapper element in certain contexts might be named, but no detailed path for the replacement of its sub-elements (e.g. for some sub-elements of <filedesc> such as <controlnote> or <edition> only <findAidDesc> is named as the replacing element).

Removed elements	
In EAD3	**In EAD 4.0**
lb (when used as mixed content; for other contexts see “Replaced elements” below)	not included
localcontrol	not included
Replaced elements	
In EAD3	**In EAD 4.0**
altformavail, dao, daoset, originalsloc	formAvailable
archref, bibref, citation, ptr, sourceentry	reference
author, corpname, famname, name, origination, persname, publisher, repository, sponsor	agent
blockquote, chronitem, chronitemset, chronlist, colspec, defitem, entry, event, head01, head02, head03, item, label, lb, list, listhead, row, table, tbody, tgroup, thead	formattingExtension (with XHTML elements)
controlaccess, index, indexentry	subjectHeadings
controlnote, edition, editionstmt, notestmt, publicationstmt, representation, seriesstmt, titlestmt	findAidDesc
datesingle	date
emph	span
genreform, occupation	subject
eventtype	@maintenanceEventType

geogname	place resp. place/placeName (depending on the context)
maintenancestatus	@maintenanceStatus
namegrp	agent, function, place, publicationNote, subject (depending on the sub-element of namegrp)
num (as sub-element of publicationstmt or seriesstmt), subtitle, titleproper	title
ptrgrp	@target
publicationstatus	@publicationStatus
relationentry	targetEntity
When used as mixed content: abbr, corpname, date, expan, famname, footnote, foreign, function, genreform, geogname, name, num, occupation, persname, quote, subject, title	referringString
Elements with camelcase spelling applied	
In EAD3	**In EAD 4.0**
addressline, agencycode, agencyname, agenttype, archdesc, bioghist, conventiondeclaration, custodhist, daterange, dateset, descriptivenote, eventdatetime, eventdescription, fileplan, fromdate, geographiccoordinates, languagedeclaration, languageset, legalstatus, localtypedeclaration, maintenanceagency, mainteanceevent, maintenancehistory, materialspec, objectxmlwrap, otheragencycode, otherfindaid, otherrecordid, physdesc, physdescset, physdescstructured, physfacet, physloc, prefercite, processinfo, recordid, relatedmaterial, rightsdeclaration, scopecontent, separatedmaterial, unitdate, unitdatestructured, unitid, unittitle, unittype	addressLine, agencyCode, agencyName, agentType, archDesc, biogHist, conventionDeclaration, custodHist, dateRange, dateSet, descriptiveNote, eventDateTime, eventDescription, filePlan, fromDate, geographicCoordinates, languageDeclaration, languageSet, legalStatus, localTypeDeclaration, maintenanceAgency, mainteanceEvent, maintenanceHistory, materialSpec, objectXMLWrap, otherAgencyCode, otherFindAid, otherRecordId, physDesc, physDescSet, physDescStructured, physFacet, physLoc, preferCite, processinfo, recordId, relatedMaterial, rightsDeclaration, scopeContent, separatedMaterial, unitDate, unitDateStructured, unitId, unitTitle, unitType
Renamed elements	
In EAD3	**In EAD 4.0**
abbr	shortCode (in conventionDeclaration, localTypeDeclaration, rightsDeclaration)

accessrestrict	accessConditions
acqinfo	sourceOfAcquisition
bibliography	publicationNote
did and didnote	identificationData and identificationDataNote
dsc	descriptionOfComponents
filedesc	findAidDesc
langmaterial	languageOfMaterial
odd	otherDescriptiveInfo
phystech	physicalOrTechnicalRequirements
script	writingSystem
userestrict	useConditions

Attributes

In total, the number of attributes is down to 66 in EAD 4.0 from 85 in EAD3.

Of the 85 attributes in EAD3, 3 have been removed without substitution.

There are 28 more attributes that have been removed, but can be replaced by alternative encodings in other namespaces. This mainly refers to those 23 attributes used with formatting elements or for display settings, which would now be encoded using alternatives from the XHTML namespace. Furthermore, this group includes 3 attributes originally coming from the XLink namespace, for which this would now be the suggested encoding again, and the attribute pair of @relatedencoding and @encodinganalog, for which one can now make use of the new option to include attributes from other namespaces and e.g. directly use the MARC21 namespace instead.

In addition, 14 attributes have been replaced by or integrated with other attributes resp. elements of EAD 4.0. Besides these replacements, for 19 attributes camelcase spelling has been applied and 7 additional attributes have been renamed. Of these 7, 6 were renamed for alignment with EAC-CPF.

The table below groups all of these attributes into five sections and lists them alphabetically, including their counterparts in EAD 4.0 where applicable.

Removed attributes	
In EAD3	**In EAD 4.0**
@entityref and @xpointer	not included
@lastdatetimeverified	not included
Attributes replaced by encoding from other namespaces	

In EAD3	**In EAD 4.0**
@abbr, @align, @althead, @altrender, @char, @charoff, @colname, @colnum, @cols, @colsep, @colwidth, @expan, @frame, @label, @listtype, @mark, @morerows, @nameend, @namest, @numeration, @pgwide, @rowsep, @valign	XHTML encoding (mostly @xhtml:style, but also e.g. @xhtml:title for @label or with expanded forms of abbreviated terms in combination with xhtml:abbr)
@actuate, @arcrole, @show	@xlink:actuate, @xlink:arcrole, @xlink:show
@encodinganalog, @relatedencoding	Depending on the value in @relatedencoding, the respective namespace is added, while the value of @encodinganalog is moved to a suitable attribute of said namespace
Replaced attributes	
In EAD3	**In EAD 4.0**
@daotype and @otherdaotype	targetType for @daotype="borndigital" relationType for @daotype="derived", "unknown" and "otherdaotype" (together with the value of @otherdaotype)
@instanceurl	findAidDesc@href
@normal	@standardDate
@otherdsctype	@descriptionOfComponentsType (when @descriptionOfComponentsTypeEncoding is set to "otherDescriptionOfComponents-TypeEncoding"
@otherdsctype	@descriptionOfComponentsType (when @descriptionOfComponentsTypeEncoding is set to "otherDescriptionOfComponents-TypeEncoding"
@otherlevel	@level (when @levelEncoding is set to "otherLevelEncoding")
@otherphysdescstructuredtype	@physDescStructuredType (when @physDescStructuredTypeEncoding is set to "otherphysDescStructuredTypeEncoding"
@otherrelationtype	relationType
@relationtype	Determines whether the corresponding relation element is transformed into a relatedMaterial/relation, agent, or function

@relator	targetRole
@render	@style
@rules	conventionDeclaration/reference> (with a @conventionDeclarationReference pointing from the element previously using @rules to the @id attribute of the conventionDeclaration element)
@transliteration	conventionDeclaration/reference (with a @conventionDeclarationReference pointing from the element previously using @transliteration to the @id attribute of the conventionDeclaration element)
@value	@maintenanceStatus, @publicationStatus, @maintenanceEventType, or agentType (depending on the context in which @value was previously used)
Attributes with camelcase spelling applied	
In EAD3	**In EAD 4.0**
@containerid, @coordinatesystem, @countrycode, @countryencoding, @datechar, @dateencoding, @linkrole, @linktitle, @localtype, @notafter, @notbefore, @physdescstructuredtype, @repositorycode, @repositoryencoding, @scriptcode, @scriptencoding, @standarddate, @standarddatetime, @unitdatetype	@containerId, @coordinateSystem, @countryCode, @countryEncoding, @dateChar, @dateEncoding, @linkRole, @linkTitle, @localType, @notAfter, @notBefore, @physDescStructuredType, @repositoryCode, @repositoryEncoding, @scriptCode, @scriptEncoding, @standardDate, @standardDateTime, @unitDateType
Renamed attributes	
In EAD3	**In EAD 4.0**
@dsctype	@descriptionOfComponentsType (following the renaming of the related element dsc to descriptionOfComponents)
@identifier and @source	@valueURI and @vocabularySource (following the naming applied in EAC-CPF as well as ensuring precision and avoiding confusion compared to @id and source)
@lang, @langcode, @langencoding and @script	@languageOfElement, @languageCode, @languageEncoding, @scriptOfElement (following the naming applied in EAC-CPF)

List of new elements and attributes

18 new elements and 25 new attributes have been added to EAD 4.0. The latter group includes 13 new optional attributes within the <control> element to specify the values used for encoding certain types of information, which were previously prescribed by the schema and have now been relaxed towards easier use of national or institutional code lists.

New elements	
Element name	**Origin and context in EAD 4.0**
agentName, agentRole	agent (shared elements with EAC-CPF)
agents, formsAvailable, functions, places	emphasized entity elements in archDesc and the unnumbered and numbered c elements (functions and places are shared elements with EAC-CPF)
citedRange	source (shared element with EAC-CPF)
contact, contactLine, place, placeName, placeRole, placeType	place (all are shared elements with EAC-CPF with the exception of placeType, which would be suggested to be added to EAC-CPF in a next revision)
formattingExtension	used as a choice next to simple p elements in all elements that previously supported the block elements blockquote, chronlist, head, list, and table (would be suggested to be added to EAC-CPF in a next revision)
referringString	used alongside reference and span as a reduced mixed content model (would be suggested to be added to EAC-CPF in a next revision)
relationType, targetRole, targetType	relation (shared elements with EAC-CPF apart from targetType, which is currently an attribute in EAC-CPF and would be suggested to be made an element in a next revision)
New attributes	
Attribute name	**Origin and context in EAD 4.0**
@addressLinetype, @contactLineType	sub-elements address/addresLine and contact/contacLine of place (shared attributes with EAC-CPF)
@addressLineTypeEncoding,	control; can be used either with the value “EASList” (if the corresponding attributes in

@audienceEncoding, @contactLineTypeEncoding, @coverageEncoding, @descriptionOfComponentsTypeEncoding, @detailLevelEncoding, @levelEncoding, @maintenanceEventTypeEncoding, @maintenanceStatusEncoding, @physDescStructuredTypeEncoding, @publicationStatusEncoding, @statusEncoding, @unitDateTypeEncoding	the archival description use the previously predefined values, e.g. “class”, “collection”, “file”, “fonds” etc. for @level) or with the value “other[...]Encoding” (of the corresponding attributes in the archival description use other values in addition, e.g. if @level=”otherlevel” @otherlevel=”dossier” was used previously) (mostly shared attributes with EAC-CPF apart from @coverageEncoding, @descriptionOfComponentsEncoding, @levelEncoding, @physDescStructuredTypeEncoding, @unitDateEncoding)
@conventionDeclarationReference, @maintenanceEventReference, @sourceReference	available with all elements used outside of the control element (shared attributes with EAC-CPF)
@detailLevel, @maintenanceStatus, @publicationStatus	control (shared attributes with EAC-CPF)
@localTypeDeclarationReference	available with all elements that allow for the @localType attribute (shared attribute with EAC-CPF)
@status	available with agencyCode and otherAgencyCode as well as date, fromDate, toDate and unitDate (shared attribute with EAC-CPF)
@style	available with span and title (shared attribute with EAC-CPF)
@vocabularySourceURI	available with all elements that allow for @valueURI and @vocabularySource (shared attribute with EAC-CPF)

Common schema principles

To support and facilitate EAD schema usage a few general principles have been defined, which are also applied to the EAC-CPF schema.

Elements

Elements are ordered within their parent elements as follows:

1. required, not repeatable elements
2. required, repeatable elements
3. optional, not repeatable elements
4. optional repeatable elements

The only exceptions of this rule are <descriptiveNote> and <objectXMLWrap>, which are not repeatable, but still always will be used after all other sub-elements of their respective parent elements.

Furthermore, the current EAD schema has two concepts to bundle elements:

1. **Plural elements**

   Plural elements serve as wrapper elements for one or more singular elements of the same kind, i.e. the new elements <agents>, <formsAvailable>, <functions>, and <places>, plus the element <relations>.

   Plural elements are optional and not repeatable and they must contain at least one singular element, but can include unlimited additional singular elements, plus a descriptive note. Even if only one single singular element is used, this needs to be wrapped by a plural element.

   A new singular element within its plural wrapper is needed for a new entry.

   Translations or other variations can be given repeating the relevant sub-element of the singular element.

   E.g.

   ```
   <agents>
     <agent valueURI=”https://viaf.org/viaf/100170140” vocabularySource=”VIAF”>
       <agentName>Martin Luther King Jr.</agentName>
     </agent>
     <agent valueURI=”https://viaf.org/viaf/49401907” vocabularySource=”VIAF”>
       <agentName>Bayard Rustin</agentName>
     </agent>
     <agent valueURI=”https://viaf.org/viaf/52524593” vocabularySource=”VIAF”>
       <agentName>Ella Baker</agentName>
     </agent>
     <agent valueURI=”https://www.wikidata.org/wiki/Q2959520”
     vocabularySource=”Wikidata”>
       <agentName>Charles Kenzie Steele</agentName>
     </agent>
   </agents>
   ```

   but

   ```
   <agents>
     <agent valueURI=”https://viaf.org/viaf/74033181/” vocabularySource=”VIAF”>
       <agentName>Olympe Audouard</agentName>
       <agentName localType=”birthName”>Félicité-Olympe de Jouval</agentName>
       <agentName languageOfElement=”rus” scriptOfElement=”Cyrl”>
       Олимпи Одуар</agentName
     </agent>
   </agents>
   ```

2. **Element sets**

   Element sets bundle elements with different concepts/information. Examples include <dateSet> and <languageSet>.

Attributes

Next to specified EAD attributes, all elements can contain any additional attributes from any other namespace.

There are three global attributes for all elements:

1. @id to refer to the element,
2. @target to point to another element within the EAD instance (not available in <ead>),
3. @audience to control the visibility of an element.

In addition, all elements with the exception of <formattingExtension> and <objectXMLWrap> include the optional attributes @languageOfElement and @scriptOfElement to provide information of the language and script of the respective element's content, thereby supporting multilingual archival descriptions.

Furthermore, all elements used outside of <control> allow for the optional attributes @conventionDeclarationReference, @maintenanceEventReference, and @sourceReference. These enable the inclusion of evidence-based assertions by referencing the source supporting the information provided in a specific element, the agent responsible for adding an assertion and the date when an assertion was added or revised, and the rules used for formulating the assertion.

Data types, removal of predefined values for attributes, and other changes

Restrictions on attribute values were removed for the following cases:

- @audience, @coverage, @descriptionOfComponentsType, @level, @physDescStructuredType, @unitDateType had the predefined values removed; when used, the corresponding @...Encoding attribute in control should be added to indicate whether the values for these attributes have been taken from the suggested "EASList" or whether an extension or alternative list is applied;
- @calendar, @certainty, @containerId, @countryCode, @era, @languageCode, @languageOfElement, @scriptCode, and @scriptOfElement were relaxed from data type NMTOKEN to token;
- @coverage, @level with archDesc, @physDescStructuredType, and @unitDateType were made optional;
- @href and @valueURI changed their data type from token to anyURI; while stricter in general, most values that comply to "token" should still comply to "anyURI";
- @standardDateTime had the constraint to dates and times before or within the year 2099 removed;
- @target was broadened from data type IDREF to IDREFS.

Schema alignment with EAC-CPF

TS-EAS is responsible for maintaining two archival encoding standards: EAD and EAC-CPF. The user communities of both standards overlap; therefore, the alignment of both schemas, where possible, shall ease the standards' maintenance, usage, teaching, and learning.

In EAD3, none of its 166 elements were actually shared with EAC-CPF 2.0, even though 48 elements (i.e. 29%) used the same names (camelcase spelling aside) in both schemas. However, there were smaller to more significant differences in how these elements were designed and used. EAD3 did share 27 out of 85 attributes (i.e. 32%) with EAC-CPF 2.0, yet again with a difference in spelling and - to some extent - in data types.

In the current draft for EAD 4.0, on the other hand, 47 of 119 elements (i.e. 40%) are shared with and defined in exactly the same way as in EAC-CPF 2.0, with 14 more elements being likely to be added and adapted in a future revision of EAC-CPF, bringing the total percentage in EAD 4.0 to 51%. Furthermore, EAD 4.0 shares 34 of its 65 attributes (i.e. 52%) with EAC-CPF 2.0, with 15 more attributes being likely to be added and adapted in a future revision of EAC-CPF, bringing the total percentage in EAD 4.0 to 74%.

Shared elements *(elements printed in italics indicate those which would require further updates in EAC-CPF to reestablish alignment; each of these indicates in brackets whether the element would need to be adapted in EAC-CPF or added to EAC-CPF)*
abstract (adapt), address, addressLine, agencyCode, agencyName, *agent (adapt)*, *agentName (add)*, *agentRole (add)*, *agentType (adapt)*, *biogHist (adapt)*, citedRange, contact, contactLine, control, conventionDeclaration, date, dateRange, dateSet, descriptiveNote, eventDateTime, eventDescription, *formattingExtension (add)*, fromDate, function, functions, geographicCoordinates, *head (adapted)*, language, languageDeclaration, legalStatus, localTypeDeclaration, maintenanceAgency, maintenanceEvent, maintenanceHistory, objectXMLWrap, otherAgencyCode, otherRecordId, *p (adapt)*, part, *place (adapt)*, placeName, placeRole, places, *placeType (add)*, recordId, reference, *referringString (add)*, *relation (adapt)*, relations, relationType, rightsDeclaration, shortCode, source, sources, span, targetRole, targetEntity, *targetType (adapted)*, term, toDate, writingSystem
Shared attributes *(attributes printed in italics indicate those which would require further updates in EAC-CPF to reestablish alignment; each of these indicates in brackets whether the attribute would need to be adapted in EAC-CPF or added to EAC-CPF)*
addressLineType (adapt), *addressLineTypeEncoding (add)*, *audience (adapt)*, *audienceEncoding (add)*, base, calendar, certainty, *contactLineType (adapt)*, *contactLineTypeEncoding (add)*, conventionDeclarationReference, coordinateSystem, countryCode, countryEncoding, dateEncoding, *detailLevel (adapt)*, *detailLevelEncoding (add)*, era, href, id, languageCode, languageEncoding, languageOfElement, linkRole, linkTitle, localType, localTypeDeclarationReference, maintenanceEventReference, *maintenanceEventTypeEncoding (add)*, *maintenanceStatus (adapt)*, *maintenanceStatusEncoding (add)*, notAfter, notBefore, *publicationStatus (adapt)*, *publicationStatusEncoding (add)*, repositoryEncoding, scriptCode, scriptEncoding, scriptOfElement, sourceReference, standardDate, standardDateTime, *status (adapt)*, *statusEncoding (add)*, style, target, unit, valueURI, vocabularySource, vocabularySourceURI

Modified encoding concepts

The following section details the most important modifications with regard to some general encoding concepts provided in EAD 4.0 compared to its predecessor version EAD3.

Linking and referencing

For linking internally within an EAD instance and referencing externally to other sources, coherent basics have been defined.

General internal referencing

The attribute @target has now been enabled with all elements of EAD 4.0 with the exception of the root element <ead>. The attribute can be used to reference any other EAD element via its @id attribute. In addition, @target now has the data type IDREFS and can hold references to several other elements at the same time.

Specific internal referencing

The more specific internal references between <container> elements have been kept and follow the same approach. Via the @parent attribute with data type IDREFS in any <container> element one or more other <container> elements can be referenced via their @containerId attributes. For the element <physLoc>, which - in EAD3 - allowed for the @parent attribute as well, the general referencing via @target is applied in EAD 4.0.

Furthermore, there are three new attributes to link directly to declarations of rules and conventions, maintenance events, and sources. These specific reference attributes (@conventionDeclarationReference, @maintenanceEventReference, and @sourceReference) have been adopted from EAC-CPF. They are optionally available in all elements used outside of <control> and are defined with data type IDREFS to provide multiple link targets, if necessary.

The attribute @conventionDeclarationReference resp. the element <conventionDeclaration>, to which the attribute points, also provides a new encoding option in lieu of the attributes @rules and @transliteration, which have been removed from EAD 4.0.

In the same way as @conventionDeclarationReference, @mainteanceEventReference, and @sourceReference, the new attribute @localTypeDeclarationReference points to a <localTypeDeclaration> within <control> to declare the list of values used for the @localType attribute in the context of a specific element. Nearly half of EAD 4.0’s elements allow for including locally defined types as required, hence declaring these types is highly recommended to retain interoperability.

Referencing external vocabularies, thesauri, or authority files

To support interoperability with linked data EAD 4.0 now includes the attribute group of @valueURI, @vocabularySource and @vocabularySourceURI. @valueURI replaces @identifier to avoid confusion with the general @id attribute mentioned above; @vocabularySource replaces @source following the TS-EAS design principle #4, according to which the same name should not be reused for both an attribute and an element.

All three attributes of this group have been adopted from and are shared with EAC-CPF. They serve as optional attributes in all elements that can contain any kind of entity, identifier, code, role or type etc. where links to vocabularies, thesauri, or authority files might be useful. EAD 4.0 thereby extends the linked data support significantly from 16 to 52 elements.

EAD 4.0 elements with @valueURI, @vocabularySource, @vocabularySourceURI
accessConditions, agencyCode, agencyName, agent, agentName, agentRole, agentType, archDesc, c, c01 to c12, container, conventionDeclaration, formAvailable, legalStatus, localTypeDeclaration, maintenanceAgency, materialSpec, otherAgencyCode, otherFindAid, otherRecordId, physFacet, place, placeName, placeRole, placeType, publicationNote, referringString, relatedMaterial, relationType, rightsDeclaration, separatedMaterial, source, subject, targetEntity, targetRole, targetType, title, unitId unitTitle, unitType, useConditions

Generic external referencing

When creating links to generic external resources (like any website, preferably with a persistent URI) the element <reference>, renamed from <ref>, with the optional attribute @href should be used. To connect this with what has been said in the previous section: if a reference to a Wikidata ID shall be given, use @valueURI, but if a link to a Wikipedia page is provided, use the element <reference> with @href.

The element <reference> is available in some elements directly, where it replaces <citation> resp. <sourceentry>. Additionally, <reference> can be used in various elements as a mixed content sub-element. In these contexts, it also integrates the previous EAD3 element <ptr>.

Furthermore, <reference> is available in a series of elements via their sub-element <p>. This includes the elements <otherFindAid>, <publicationNote>, <relatedMaterial>, and <separatedMaterial>, where <reference> integrates the previous EAD3 elements <archref> and <bibref>. Last, the use of <reference> can be initiated in some elements via their sub-element <descriptiveNote><p>.

Where more than one of these encoding options is available, e.g. <reference> and <descriptiveNote> as sub-elements of the same EAD element, it is recommended to use the most direct encoding.

EAD 4.0 elements with <reference>
Elements with <reference> as direct sub-element
conventionDeclaration, localTypeDeclaration, rightsDeclaration (instead of <citation>); source (instead of <sourceentry>)
Elements with <reference> as mixed content sub-element
abstract, container, dimensions, eventDescription, head, materialSpec, p, physDesc, physFacet, physLoc, unitDate, unitId, unitTitle
Elements with <p><reference> as sub-element
accessConditions, accruals, appraisal, arrangement, biogHist, custodHist, filePlan, otherDescriptiveInfo, otherFindAid, physicalOrTechnicalRequirements, preferCite, processInfo, publicationNote, relatedMaterial, scopeContent, separatedMaterial, sourceOfAcquisition, subjectHeadings, useConditions, plus descriptionOfComponents, formAvailable, identificationDataNote

Elements with <descriptiveNote><p><reference> as sub-element
agent, agents, conventionDeclaration, formsAvailable, function, functions, languageDeclaration, languageOfMaterial, languageSet, legalStatus, localTypeDeclaration, maintenanceAgency, physDescStructured, place, places, relation, relations, rightsDeclaration, source, sources, subject

The attribute @href, optional in <reference>, is accompanied by two more optional attributes: @linkRole and @linkTitle. This group of attributes is also available with <contactLine>, <findAidDesc>, and <source> to link to generic external resources.

The attribute @lastdatetimeverified, which was available in some linking elements of EAD3, has been removed as it is underused and it seems a relic of early internet days.

Base URL for external referencing

Some wrapper elements in EAD include the optional attribute @base to encode a base URL to be applied to all external links encoded in that element and its sub-elements. E.g. if the base URL is https://www.example.com/terms/persons/ and the URIs provided in the @valueURI attribute of three agent elements are 1234, 5678, and 9012, the complete URLs to see the referenced information about these three agents are:

- https://www.example.com/terms/persons/1234
- https://www.example.com/terms/persons/5678
- https://www.example.com/terms/persons/9012

@base is available with archDesc, c, c01 to c12, control, ead, relations, and sources.

Formatting

Next to the general objective of providing a standardized, machine-readable format to encode archival descriptions for data management, publication, discovery, and exchange, EAD also stems from the desire to find a digital format for representing archival finding aids. As such, EAD always included longer narrative texts, which asked for formatting options to assist readers with easily ascertaining the text.

While EAD 4.0 still wants to support this, it shifts its focus a little more from documents to data with regard to what is enabled in the EAD 4.0 schema itself. All elements that previously allowed for an array of formatting elements such as generic lists, chronological lists, definition lists, or tables, now come with a choice between repeated <p> elements for a simple structure or the new element <formattingExtension>. The latter can include elements from another namespace and is specifically intended to enable the encoding with XHTML elements. With this, EAD 4.0 aligns its formatting options with the standard used more generally in web publications and often found in default exports of narrative texts from databases.

Similarly, attributes that were specifically aimed at the display and formatting of information rather than the encoding of information per se, have been removed resp. users are recommended to employ the new option of including attributes from other namespaces if display is indeed something that they wish to encode as part of their EAD instances. This e.g. includes the replacement of @label with @xthml:title.

Keeping within the Encoded Archival Standards namespace, the element <span> is adopted by EAD 4.0 from EAC-CPF and replaces <emph> as a mixed content element in certain contexts along the optional attribute @style. <span> is used to format words or phrases for linguistic effect. Values of the attribute @style should conform to W3C CSS and offer hence a broader palette of options than the attribute @render did in EAD3. @style is furthermore available with the element <title>.

EAD 4.0 elements with <formattingExtension>
accessConditions, accruals, appraisal, arrangement, biogHist, custodHist, descriptionOfComponents, findAidDesc, filePlan, formAvailable, otherDescriptiveInfo, otherFindAid, physicalOrTechnicalRequirements, preferCite, processInfo, publicationNote, relatedMaterial, scopeContent, separatedMaterial, sourceOfAcquisition, subjectHeadings, useConditions

Mixed content

Mixed content means elements which can contain text themselves, but also allow for the use of sub-elements mixed with that text for in-line tagging. While this enables “encoding in situ”, mixed content also makes any import and export, any data transformation, and any data exchange more complex, especially when different variations of mixed content are possible. EAD 4.0 still retains the option for mixed content, but it reduces the number of mixed content models to a single one with only three possible elements:

- <span> to emphasize a word or phrase within a longer text by applying a different text formatting;
  <span> integrates <emph>, <foreign>, and <quote> in mixed content contexts (outside of <formattingExtension>)
- <reference> to include links to generic external resources;
  <reference> integrates <ptr> and <ref> in mixed content contexts (outside of <formattingExtension>)
- <referringString> to indicate a term or name or code, which might identify an entity within a longer text, e.g. the name of a person or a place;
  <referringString> integrates <abbr>, <expan>, <corpname>, <famname>, <function>, <genreform>, <geogname>, <name>, <occupation>, <persname>, <subject>, <title>, <date>, <footnote>, and <num> in mixed content contexts (outside of <formattingExtension>).

EAD 4.0 elements allowing mixed content
abstract, container, dimensions, eventDescription, head, materialSpec, p, physDesc, physFacet, physLoc, reference (only includes span and referringString), unitDate, unitId, unitTitle

Nesting

The concept of nesting refers to using an element as a sub-element of itself, thereby - in theory - creating an endless hierarchy with an XML document. EAD 4.0 has removed the concept of nesting from its elements apart from one exception, the unnumbered <c> element. For the longer narrative elements such as <biogHist> or <scopeContent>, however, it is recommended to make use of the new element <formattingExtension>, if nesting is mainly used for structuring those narrations, or to simply repeat the element, if parallel approaches e.g. for <appraisal> or <arrangement> are to be described.

Encoding of date and time information

Date encoding is possible for single dates, time spans, and complex dates. Depending on the context, only one of these options might be available or there might be a choice between two or all three of them. While EAD3 already used the elements <daterange> and <dateset> in a similar way to EAC-CPF, it included two elements for single dates: <datesingle> when used as one option in a two-way or three-way choice, and <date> when used in a mixed content context.

EAD 4.0 has combined these two single date elements into one, the <date> element as a shared element with EAC-CPF, given that both elements - within their respective contexts - effectively fulfilled the same use case. The updated <date> element, same as <fromDate> and <toDate> as sub-elements of <dateRange>, now includes a set of 7 optional and date specific attributes. @standardDate, @notBefore, and @notAfter can be used with values according to ISO 8601 or some other date encoding standard. If using an alternative date encoding standard, it is highly recommended to specify this by using the @dateEncoding attribute in <control> with the value “otherDateEncoding” and by including more information in a <conventionDeclaration> element. When using ISO 8601, EAD 4.0 now refers to the standard’s latest version of 2019, including the usage of the Extended Date/Time Format (EDTF) to qualify a normalized date as uncertain (?), approximate (~), or uncertain and approximate (%).

Additionally, the level of confidence for the information given in <date>, <fromDate>, or <toDate> can be provided in textual form via the attribute @certainty, while approximation can also be indicated textually via the new attribute @status, which could be used with values such as “unknown” or “open”. Furthermore, all single date elements contain the optional attributes @calendar and @era to specify the date being encoded.

Regarding the wrapper elements for time spans and complex dates, the design for <dateSet> remains the same (at least two single dates or two time spans or one single date plus one time span), while <dateRange> now requires either <fromDate> or <toDate> to be present and expects <fromDate> to come first in case both sub-elements are used. Neither <fromDate>, nor <toDate> have to contain text, though.

Encoding of language information

Language information can be given in several places of an EAD instance depending on the context and the object of the attributed language. Next to literal information, it is recommended to provide language and script information in a normalized format for machine readability. While applied in different contexts across the same EAD instance, all such normalized information should follow the same standard.

As was already the case in EAD3, these standards can be defined in the <control> section using the attributes @languageEncoding (renamed from @langencoding) and @scriptEncoding. EAD 4.0 does not recommend the usage of a specific list of short codes for language and/or script names but facilitates the use of international standards like the ISO 639 code sets, the ISO 15924 standard and any other language and script codes, defined in <conventionDeclaration>. The ISO standards available for language encoding have been extended in comparison to EAD3 in as far as ISO standard 639-2 is not restricted to the version including bibliographic codes (ISO 639-2b) anymore. Furthermore, EAD 4.0 follows EAC-CPF 2.0 in explicitly proposing the usage of IETF language tags often used by computing standards.

Once defined, these language and script codes can be used in three main contexts:

- The language and writing system used in the EAD instance itself;
  Same as in EAD3, this can be defined on the element <languageDeclaration> within the <control> section. Following EAC-CPF 2.0 and a simplification of the schema, <languageDeclaration> now declares the language and script in which an EAD instance is written directly with the attributes @languageCode and @scriptCode, with @languageCode being required. The sub-elements <language> and <script> have been removed, but <languageDeclaration> can still have a descriptive note for further information.
- The language and writing system used in a specific EAD element;
  Next to the general declaration of the language and writing system used for the EAD instance as a whole, the language and the writing system of the content of each single text element can be specified by using the optional attributes @languageOfElement and @scriptOfElement (renamed from @lang and @script for greater distinction). Both attributes contain the language and script codes, following the once declared encoding standard.
  Using @languageOfElement and @scriptOfElement together with the repeatability of most text elements enables users to create multilingual EAD instances, e.g. by giving the same element once in English and once in Japanese. For this, languages and scripts for elements' content might be specified in a specific element or in the parent or wrapper element to summarize language and script information for an encoding section. Depending on how diverse the languages and script represented in an EAD instance are, it might be more feasible to encode the primary language of the EAD instance in <languageDeclaration> and to use the single element language attribute @languageOfElement only in cases where the content language differs from the primary language.
- The language of the materials described;
  The attributes @languageCode and @scriptCode as used with <languageDeclaration> in <control> also are available with the elements <language> and <writingSystem> (renamed from <script> to avoid security vulnerabilities) within the element <languageOfMaterial> (renamed from <langmaterial>) in the descriptive sections of an EAD instance.
  The element <language> is available directly within <languageOfMaterial> or within the grouping element <languageSet> and can contain the name of the language itself. Next to this, the attribute @languageCode can be used to provide the code for said language. The element <writingSystem> is only used within <languageSet> and, similar to <language>, can contain the name of the script or writing system present in the materials. The attribute @scriptCode can then be used to specify the code for said script.

When working with EAD instances or materials in different writing systems, it might also be suitable to include information about the rules followed for transliteration from one writing system into the other. The attribute @transliteration, only available with a few selected elements in EAD3, was removed from EAD 4.0 in the process of clearing up unused or redundant components. Instead, the new attribute @conventionDeclarationReference is now available with all text elements within the descriptive parts of an EAD instance. It can be used to refer to the element <conventionDeclaration> within <control>, which is used to specify the system, conventions, or rules applied to transliteration, e.g. ISO standard 9 for the transliteration of Cyrillic characters into Latin characters.

Encoding of place information

Places are, along with agents and functions, paramount contextual entities when describing archival materials. In EAD 4.0, the description of places has hence been elevated to a dedicated section for entity elements within the descriptive parts of an EAD instance. In this context, EAD 4.0 has adopted the plural wrapper element <places> from EAC-CPF 2.0, which has to contain at least one singular <place> element, but can contain as many <place> elements as applicable. Any controlled access headings referring to places, i.e. <geogname> within <controlaccess> or <index> in EAD3, will now be encoded as <place> within a <places> grouping element.

In addition, a <place> can be given to locate a relation (extending on the element <geogname>, which was available in EAD3 in this context) or different instances of finding aids for the materials being described with the renamed and restructured <findAidDesc> element. Furthermore, names of places can be encoded when describing the relation between the materials and an agent, a function, a legal status, or a subject.

The element <place> comes with a group of six central child elements; at least one of these has to be present and all of them are repeatable:

- <placeName> to provide the name of a geographical feature;
- <placeRole> to specify the role a place has in relation to the materials described, e.g. place of creation or place of access;
- <placeType> to differentiate places by their type, especially if the name of a place could be designated to several different types (e.g. implementing the feature codes used by GeoNames, such as “populated place” or “first-order administrative division”);
- <geographicCoordinates>, now a sibling element to <placeName> and requiring the attribute @coordinateSystem to specify the system used to express the geographic coordinates;
- <address>, used independent of encoding a place name in EAD3, to encode any physical address details within one or more <addressLine> sub-elements;
- <contact> to encode any digital contact details and following a similar content model as <address>, i.e. having at least one <contactLine> sub-element.

Element-specific changes

The following section looks at the most important modifications for specific elements and their sub-elements as available in EAD 4.0 compared to its predecessor version EAD3.

<control>

The control area is mainly updated for EAC-CPF alignment and to simplify the schema. The most significant change in this context is the removal of <filedesc> from <control> and the creation of the element <findAidDesc> in its stead as a sibling element of <control> and

<archDesc> (see more on <findAidDesc> below). In this context, the element <representation> is removed from <control> as well.

Sequence of sub-elements

The sequence of sub-elements in <control> has been adapted to follow the general principle described above, i.e. the mandatory <recordId>, <maintenanceAgency>, and <maintenanceHistory> come first, followed by the optional, not repeatable <sources>, followed by the optional and repeatable <conventionDeclaration>, <languageDeclaration>, <localTypeDeclaration>, <otherRecordId>, and <rightsDeclaration>.

Encoding attributes

Enabling more flexibility in the application of certain attributes throughout an EAD instance, EAD 4.0 introduces a set of new encoding attributes within the <control> element. These follow, in principle, the already existing attributes @countryEncoding, @dateEncoding, @languageEncoding, @repositoryEncoding, and @scriptEncoding. Instead of pointing to ISO standards, however, these new attributes point to the “EASList” for values used with a specific attribute next to offering the option to choose another encoding approach, either extending on the relevant EAS list or using one’s own list of values. The EAS lists are part of the schematron for EAD 4.0 (see the current version in the folder “internal-lists” on GitHub). Any other encoding lists should be declared using the <conventionDeclaration> element.

Encoding attributes in <control>	
Encoding attribute	**Attribute using values following the specified encoding scheme**
@addressLineTypeEncoding	@addressLineType
@audienceEncoding	@audience
@contactLineTypeEncoding	@contactLineType
@countryEncoding	@countryCode
@coverageEncoding	@coverage
@detailLevelEncoding	@detailLevel
@descriptionOfComponentsTypeEncoding	@descriptionOfComponentsType
@dateEncoding	@standardDate, @notAfter, @notBefore and @standardDateTime
@languageEncoding	@languageCode and @languageOfElement
@levelEncoding	@level
@maintenanceEventTypeEncoding	@maintenanceEventType
@maintenanceStatusEncoding	@maintenanceStatus
@physDescStructuredTypeEncoding	@physDescStructuredType
@publicationStatusEncoding	@publicationStatus
@repositoryEncoding	@repositoryCode
@scriptEncoding	@scriptCode and @scriptOfElement

@statusEncoding @unitDateTypeEncoding	@status @unitDateType

With this, the attributes listed on the right in the table above do not include any predefined value lists anymore in the EAD 4.0 schema itself. They also have all been made optional. Furthermore, instances of @other… attributes, e.g. @otherlevel, used in combination with some of these attributes in EAD3, are now incorporated with the main attribute when the encoding is set to “other…Encoding”. See the following (simplified) example for the attribute @level:

Encoding in EAD3	**Encoding in EAD 4.0**
<archdesc level=”collection”> [...] <dsc> <c level=”series”> [...] <c level=”otherlevel” otherlevel=”dossier”> [...] <c level=”otherlevel” otherlevel=”dossier”> [...] <c level=”otherlevel” otherlevel=”dossier”> [...] <c level=”otherlevel” otherlevel=”group”> [...] <c level=”file”> [...] <c level=”otherlevel” otherlevel=”volume”> [...] <c level=”otherlevel” otherlevel=”volume”> [...] <c level=”file”> [...]	<control levelEncoding=”otherLevelEncoding”> <conventionDeclaration> <reference>Extended level encoding based on EAS list</reference> <archDesc level=”collection”> [...] <dsc> <c level=”series”> [...] <c level=”dossier”> [...] <c level=”dossier”> [...] <c level=”dossier”> [...] <c level=”group”> [...] <c level=”file”> [...] <c level=”volume”> [...] <c level=”volume”> [...] <c level=”file”> [...]

Transformed elements

Upon evaluation of the elements and attributes, some elements were removed and superseded by attributes in their parent elements in EAC-CPF 2.0.

- <maintenanceStatus> becomes the optional attribute @maintenanceStatus in <control>;
- <publicationStatus> becomes the optional attribute @publicationStatus in <control>;
- <eventType> becomes the optional attribute @maintenanceEventType in <maintenanceEvent>;
- <agentType> becomes an optional sub-element of <agent>, next to <agentName>, <agentRole> and others, instead of being used as a sibling element of <agent>.

Other sub-elements

The content model of <maintenanceAgency> has been relaxed as far as its sub-element <agencyName> is not required anymore under all circumstances. Instead, either <agencyName> or <agencyCode> have to be present. If both are used, <agencyCode> is expected to appear before <agencyName>.

The elements <conventionDeclaration>, <localTypeDeclaration>, and <rightsDeclaration> now use the required sub-element <reference> (replacing <citation>) first and the optional sub-element <shortCode> (replacing <abbr>) second.

Furthermore, <conventionDeclaration> and <localTypeDeclaration> together with <maintenanceEvent> and <source> are emphasized in their usage by the introduction of the attributes @conventionDeclarationReference, @localTypeDeclarationReference, @maintenanceEventReference and @sourceReference available in the descriptive parts of an EAD instance. With these attributes it is possible to refer to various administrative aspects and principles defined in the <control> section such as:

- rules and conventions applied in the creation of the archival descriptions, such as rules for transliteration and rules for authorized, alternative, and preferred forms of names (<conventionDeclaration>);
- the source for specific assertion made as part of the archival description, especially in case of potentially having differing assertions in parallel to each other, e.g. with regard to the date(s) of creation of the materials being described (<source>);
- the agent responsible for including a specific assertion into the archival description and the date when this assertion was added (<maintenanceEvent>);
- lists of local types used with specific descriptive elements (<localTypeDeclaration>).

<findAidDesc>

<findAidDes> is a new sub-element of <ead>, sitting between <control> and <archDesc>. It replaces the element <filedesc>, which was a sub-element of <control> in EAD3. <findAidDesc> is meant to encode information about any potentially existing instances of the encoded archival description included in an EAD XML document. This includes analogue forms like a printed finding aid or an index card catalog as well as digital forms such as an HTML page or a PDF document (previously encoded with <representation> within <control>) or the EAD XML instance itself (previously only given in form of a link in the attribute @instanceurl of <recordid>). With this, <control> remains focused on administrative

information necessary for the management and maintenance of the EAD XML file, same as it is on EAC-CPF with regard to the EAC-CPF XML file.

Contrary to <filedesc>, <findAidDesc> is now optional within EAD 4.0. In addition, <findAidDesc> makes use of elements also applied in other parts of the EAD XML file, such as <agent>, <citedRange>, <date>, <formattingExtension>, <place>, and <title>, rather than having a series of elements only used in this specific context, which were functionally similar if not the same.

<archDesc> and <c> elements

The changes applied to <archDesc> and the unnumbered and numbered <c> elements mainly result from the changes applied to their sub-elements as described in the following chapters.

Identification data

The element <did> is renamed to <identificationData>.

A series of its sub-elements are removed and elevated to entity elements (see the next chapter). This includes the elements <dao> and <daoset>, which are integrated with the new element <formAvailable>, and the elements <origination> and <repository>, which are integrated with the extended element <agent>.

Apart from this, the element <abstract> is removed from <identificationData> and can now be used in the context of all narrative elements, while the element <legalStatus> is changed from being a narrative element to following the same content model as defined on EAC-CPF 2.0 and is now a sub-element of <identificationData>.

For the remaining sub-elements of <identificationData>, the main changes are:

- Renaming <didnote> and <langmaterial> to <identificationDataNote> and <languageOfMaterial>;
- <identificationDataNote> furthermore now follows the same content model as <descriptiveNote>, i.e. using the sub-element <p> to hold its content;
- Applying camelcase spelling to <materialSpec>, <physDescSet>, <physDesc>, <physDescStructured>, <physLoc>, <unitDate>, <unitDateStructured>, <unitId>, <unitTitle>;
- Applying the changes with regard to mixed content and referencing external vocabularies etc. as applicable;
- Applying the change from <datesingle> to <date> in the context of <unitDateStructured>.

Entity elements

EAD 4.0 introduces a set of elevated entity elements indicating these entities' importance when describing archival records and their contexts. With this, EAD 4.0 moves existing elements that represent the entities in question in EAD3 into a dedicated section between <identificationData> and the narrative elements. Furthermore, several elements representing the same type of entity are integrated into one entity element. Each entity type is represented by the pair of a plural and a repeatable singular element, a concept adopted from EAC-CPF:

- <agents> with its singular element <agent> integrating the elements <origination> and <repository> as well as the elements <corpname>, <famname>, <name>, and <persname> when used as sub-elements of <controlaccess> or <index> in EAD3;
- <formsAvailable> with its singular element <formAvailable> integrating the elements <dao>, <daoset>, <altformavail>, and <originalsloc> from EAD3;
- <functions> with its singular element <function> replacing the element <function> when used as a sub-element of <controlaccess> or <index> in EAD3;
- <places> with its singular element <place> replacing the element <geogname> when used as a sub-element of <controlaccess> or <index> in EAD3.

<agent>, previously only available within <maintenanceEvent> in <control>, will be used with a required <agentName> sub-element to pick up on predecessor elements such as <corpname>, <famname>, <name>, and <persname>. It will also allow for an <agentRole> and an <agentType> to be defined and will additionally include elements for encoding a date and place dimension as well as re-using the element <relationType> from the element <relation>. Next to being used within its plural wrapper element <agents>, the singular <agent> will remain available in <maintenanceEvent> and also forms an optional sub-element of <findAidDesc>.

<formAvailable> will allow for a textual description using the elements <abstract>, <p>, or <formattingExtension> to pick up on the narrative nature of its predecessor elements <altformavail> and <originalsloc> respectively the previous option of a descriptive note in <dao> and <daoset>. Furthermore, <formAvailable> will include <relations> as a whole to enable a more specific encoding of the relationship between the materials being described and its variety of instantiations.

<function> will be adapted to the element's definition in EAC-CPF 2.0 and require the sub-element <term> (instead of <part>). Furthermore, <function> allows for encoding a date and place dimension and re-uses the element <targetRole>, <targetType>, and <relationType> from the element <relation> (see more on encoding relations below).

For the details on the element <place> see the chapter on encoding places above.

All entity elements will additionally allow for the use of the three attributes @valueURI, @vocabularySource, and @vocabularySourceURI for referencing any external descriptions of the related entities. With this, the entity elements provide the opportunity for a relations model that can be extended step-by-step in the way an institution with archival holdings feels comfortable with and capable of. This approach allows progression:

- From providing literal values in the descriptive elements
- Via adding URIs to external resources providing more information about the related entities in the vocabulary attributes
- To describing the relationship between these related entities and the archival materials in <relations>, <relation> and its sub-elements directly in EAD.

Narrative elements

There is a series of general changes applicable to all narrative elements or groups of narrative elements. This includes:

- Applying camelCasing to <biogHist>, <custodHist>, <filePlan>, <otherFindAid>, <preferCite>, <processInfo>, <relatedMaterial>, <scopeContent>, and <separatedMaterial>

- Renaming <accessrestrict> to <accessConditions>, <acqinfo> to <sourceOfAcquisition>, <bibliography> to <publicationNote>, <controlaccess> to <subjectHeadings>, <odd> to <otherDescriptiveInfo>, <phystech> to <physicalOrTechnicalRequirements>, and <userestrict> to <useConditions>
- Adding <abstract> as an optional sub-element to all narrative elements
- Replacing <head> and the block elements <blockquote>, <chronlist>, <list>, and <table> with the new element <formattingExtension> and enabling a choice between simple <p> elements and more structured formatting via <formattingExtension>
- Removing the option to nest narrative elements within themselves

Furthermore, <altformavail> and <originalsloc> are integrated with the new entity element <formAvailable>, <index> is integrated with <subjectHeadings>, and <legalStatus> is moved into <identificationData>

Similarly to <formAvailable>, <relations> will also be included in those elements that represent archival as well as bibliographic resources that are related to the materials being described such as <otherFindAid>, <publicationNote>, <relatedMaterial>, and <separatedMaterial>. In addition, these four elements also will allow for the attributes @valueURI, @vocabularySource, and @vocabularySourceURI.

Those three attributes also will be available with <accessConditions> and <useConditions>. Furthermore, both elements now allow for the specification of a <dateRange> during with the conditions governing access resp. use apply.

As a last “relational” element, the new element <subjectHeadings> will allow for <abstract>, a choice between <p> and <formattingExtension> and one or more <subject> elements. A @localType attribute will be available to distinguish between different types - or usage scenarios - for <subjectHeadings>. The sub-element <subject> will follow the same content model as <function>, i.e. requiring one or more <term> sub-elements> and enabling the encoding of a date and place dimension, a <targetRole>, a <targetType>, and a <relationType>.

<dsc> becomes <descriptionOfComponents>

The element <dsc> as a sub-element of <archDesc> in EAD 4.0 is renamed to <descriptionOfComponents>. Its repeatability has been removed due to lack of examples for this usage type in practice. For multilingual scenarios it is - according to community feedback - more likely that either a complete copy of an EAD instance is created in a second language or that only specific elements are repeated in different languages.

Relations

Besides aligning the element <relations> and its sub-element <relation> with their definitions in EAC-CPF 2.0, EAD 4.0 removes the elements’ experimental status and fully integrates them into existing elements that represent related entities and their relationship to the materials being described. Instead of being a separate addition to the element <archDesc> and the numbered and unnumbered <c> elements as in EAD3, <relations> becomes more of a possible extension to existing elements.

Overall, the new encoding structure aims at simplifying relations and at achieving better interoperability:

- As is the case already in EAD3, <relations> still serves as a wrapper element for one or more single <relation> elements.
- Furthermore, <relations> can now include a descriptive note as applicable.
- Each <relation> element contains one required element <targetEntity> to identify the entity that is being targeted by the relation. This is different from EAD3, where the equivalent <relationentry> was only optional though allowed for repetition within the same <relation> element.
- Next to <targetEntity>, the singular <relation> element in EAD 4.0 includes an optional and repeatable <relationType> element, incorporating the required attribute @relationtype and its sibling attribute @otherrelationtype from EAD3.
- Furthermore, <relation> allows for optional <targetRole> and <targetType> elements, which are new possibilities in EAD 4.0. Together with <relationType> these two elements enable more precision when describing the relationship between an entity and the materials described. E.g. an agent can be specified as being of the (target) type “person”, having a general relation of type “creator” and a (target) role as “correspondent”.
- All four of these sub-elements of <relation> can contain text - <targetEntity> via its sub-element <part> - and have the attributes @valueURI, @vocabularySource and @vocabularySourceURI enabled to refer to external vocabularies, thesauri, and authority files as suitable and applicable.
- In addition, relation context can be given with date, place, and descriptive information in the according child elements of <relation>: <date>, <dateRange>, <dateSet>, <descriptiveNote>, <objectXMLWrap>, and <place>, which already was possible in EAD3.