Note: This sheet shows the changes that would need to be applied to each existing element and attribute in order to move it from EAD3 to EAD 4.0, while the sheets "Elements" and "Attributes" show the changes applied to each element and attribute in terms of its definition within the EAD schema. This sheet here therefore only refers to existing elements, but does not cover elements newly added to EAD 4.0.
Furthermore, this sheet only includes the actual changes. Anything that stays the same (e.g. an element or attribute that isn't renamed or replaced or doesn't have camelCasing applied or when an element retains the same attributes in EAD 4.0 as it had in EAD3), won't be mentioned as such.

General attributes' transformation for all elements that have existed in EAD3 already:
If @audience is used in one or more elements, add @audienceEncoding with the default value "EASList" to control
Replace @altrender with @xhtml:style and add the XHTML namespace to the element using @altrender; include a comment to confirm that @altrender was indeed used to indicate alternative formatting
Rename @lang to @languageOfElement; make sure @languageEncoding exists in control; if it doesn't, add the attribute with the default value "otherLanguageEncoding"
Rename @script to @scriptOfElement; make sure @scriptEncoding exists in control; if it doesn't, add the attribute with the default value "otherScriptEncoding"

General transformation for attributes of elements that are removed:
If a parent element is removed from EAD 4.0, but there is a transformation route for its child elements and if parent and child elements share attributes, check whether the attributes exist with the child element already; if the child elements include the attributes already, remove the attributes of the parent element along with the element itself; if the child elements do not include the attributes already, move the attributes of the parent element to each of its child elements without them

Element in EAD3	**Element in EAD 4.0**	**Transformation route for element and its content (draft)**	**Transformation route for attributes used with the element (draft)**
abbr	shortCode (referringString)	Transform conventiondeclaration/abbr, localtypedeclaration/abbr, and righsdeclaration/abbr to conventionDeclaration/shortCode, localTypeDeclaration/shortCode, and righsDeclaration/shortCode Transform abbr to referringString where XHTML isn't applied Transform abbr to xhtml:abbr where XHTML is applied	Add the value of @expan into shortCode or referringString (in brackets after the content of abbr itself, e.g. <abbr expan="Alphabet">ABC</abbr> becomes <shortCode>ABC (Alphabet)</shortcode>) Add the value of @expan to xhtml:abbr@title
abstract	abstract	Transform abbr, expan, persname, corpname, famname, geogname, name, occupation, subject, genreform, function, title, date, footnote, num (if existing) into referringString (for details see rows 4, 48, 52, 71, 72, 75, 78, 79, 80, 105, 108, 110, 119, 153, 160) Transform emph, foreign, quote (if existing) into span (for details see rows 65, 76, 134) Transform ptr, ref (if existing) into reference (for details see rows 128, 136) Remove lb Keep only the content of any sub-elements used in abstract and include a comment about any data potentially encoded in their attributes; values of attributes also available with referringString, span, and reference will of course be moved over to these three elements accordingly	Remove @encodinganalog and include a comment about the removal; if any parent element in EAD3 indicates a relatedencoding according to MARC21, add the MARC21 namespace and use the attribute @marc21:tag in abstract to include the value of @encodinganalog (xmlns:marc21="http://www.loc.gov/MARC21/slim" marc21:tag="...") Remove @label and include comment about the removal; an option could be to suggest adding the XHTML namespace and using the attribute @title in abstract to include the value of @label (xmlns:xhtml="http://www.w3.org/1999/xhtml" xhtml:title="...")
accessrestrict	accessConditions	Rename to accessConditions If only used with p, keep as is If used with p and any other formatting elements (i.e. blockquote, chronlist, head, list, table), move all into formattingExtension while adapting to XHTML where necessary (see rows 24, 40, 82, 96, 117, 155 for details) If used with nesting, move the sub-elements of the nested accessrestrict into accessConditions/formattingExtension to recreate the complete structure of the element alongside the sub-elements of the unnested (or parent) accessrestrict; in that case, even if the unnested (or parent) accessrestrict is only used with p, those p elements would also be used into accessConditions/formattingExtension	If @localtype is used, apply camelCasing to change to @localType; create a localTypeDeclaration with @id in control and add the default content "List of local types for conditions governing access" to its sub-element reference; then add @localTypeDeclarationReference to accessConditions pointing to the @id of said localTypeDeclaration; if a localTypeDeclaration with this content in its sub-element reference already exists, point to this instead of creating another one) Remove @encodinganalog and include a comment about the removal; if any parent element in EAD3 indicates a relatedencoding according to MARC21, add the MARC21 namespace and use the attribute @marc21:tag in accessConditions to include the value of @encodinganalog (xmlns:marc21="http://www.loc.gov/MARC21/slim" marc21:tag="...")
accruals	accruals	If only used with p, keep as is If used with p and any other formatting elements (i.e. blockquote, chronlist, head, list, table), move all into formattingExtension while adapting to XHTML where necessary (see rows 24, 40, 82, 96, 117, 155 for details) If used with nesting, move the sub-elements of the nested accruals into accruals/formattingExtension to recreate the complete structure of the element alongside the sub-elements of the unnested (or parent) accruals; in that case, even if the unnested (or parent) accessrestrict is only used with p, those p elements would also be used into accessConditions/formattingExtension; in that case, even if the unnested (or parent) accruals is only used with p, those p elements would also be used into accruals/formattingExtension	If @localtype is used, apply camelCasing to change to @localType; create a localTypeDeclaration with @id in control and add the default content "List of local types for accruals" to its sub-element reference; then add @localTypeDeclarationReference to accruals pointing to the @id of said localTypeDeclaration; if a localTypeDeclaration with this content in its sub-element reference already exists, point to this instead of creating another one) Remove @encodinganalog and include a comment about the removal; if any parent element in EAD3 indicates a relatedencoding according to MARC21, add the MARC21 namespace and use the attribute @marc21:tag in accruals to include the value of @encodinganalog (xmlns:marc21="http://www.loc.gov/MARC21/slim" marc21:tag="...")

Note: This sheet shows the changes that would need to be applied to each existing element and attribute in order to move it from EAD3 to EAD 4.0, while the sheets "Elements" and "Attributes" show the changes applied to each element and attribute in terms of its definition within the EAD schema. This sheet here therefore only refers to existing elements, but does not cover elements newly added to EAD 4.0. Furthermore, this sheet only includes the actual changes. Anything that stays the same (e.g. an element or attribute that isn't renamed or replaced or doesn't have camelCasing applied or when an element retains the same attributes in EAD 4.0 as it had in EAD3), won't be mentioned as such.		**General attributes' transformation for all elements that have existed in EAD3 already:** If @audience is used in one or more elements, add @audienceEncoding with the default value "EASList" to control Replace @altrender with @xhtml:style and add the XHTML namespace to the element using @altrender; include a comment to confirm that @altrender was indeed used to indicate alternative formatting Rename @lang to @languageOfElement; make sure @languageEncoding exists in control; if it doesn't, add the attribute with the default value "otherLanguageEncoding" Rename @script to @scriptOfElement; make sure @scriptEncoding exists in control; if it doesn't, add the attribute with the default value "otherScriptEncoding" **General transformation for attributes of elements that are removed:** If a parent element is removed from EAD 4.0, but there is a transformation route for its child elements and if parent and child elements share attributes, check whether the attributes exist with the child element already; if the child elements include the attributes already, remove the attributes of the parent element along with the element itself; if the child elements do not include the attributes already, move the attributes of the parent element to each of its child elements without them	
Element in EAD3	**Element in EAD 4.0**	**Transformation route for element and its content (draft)**	**Transformation route for attributes used with the element (draft)**
acqinfo	sourceOfAcquisition	Rename to sourceOfAcquisition If only used with p, keep as is If used with p and any other formatting elements (i.e. blockquote, chronlist, head, list, table), move all into formattingExtension while adapting to XHTML where necessary (see rows 24, 40, 82, 96, 117, 155 for details) If used with nesting, move the sub-elements of the nested acqinfo into sourceOfAcquisition/formattingExtension to recreate the complete structure of the element alongside the sub-elements of the unnested (or parent) acqinfo; in that case, even if the unnested (or parent) acqinfo is only used with p, those p elements would also be used into sourceOfAcquisition/formattingExtension	If @localtype is used, apply camelCasing to change to @localType; create a localTypeDeclaration with @id in control and add the default content "List of local types for sources of acquisition" to its sub-element reference; then add @localTypeDeclarationReference to sourceOfAcquisition pointing to the @id of said localTypeDeclaration; if a localTypeDeclaration with this content in its sub-element reference already exists, point to this instead of creating another one) Remove @encodinganalog and include a comment about the removal; if any parent element in EAD3 indicates a relatedencoding according to MARC21, add the MARC21 namespace and use the attribute @marc21:tag in sourceOfAcquisition to include the value of @encodinganalog (xmlns:marc21=" http://www.loc.gov/MARC21/slim" marc21:tag="...")
address	address	Apply camelCoding to sub-element addressLine If used in publicationstmt, move each publicationstmt/address into a separate findAidDesc/place/address If used in repository, concatenate the contents of each repository/address/addressline in agent/placeName and include a comment about this concatenation	In the case of concatenating the contents of address' sub-element(s) addressline into agent/placeName, move @audience and @id into placeName Also move address@lang and @script to placeName@languageOfElement and @scriptOfElement
addressline	addressLine	Apply camelCasing Transform abbr, expan (if existing) into referringString (for details see rows 4, 71) Transform emph, foreign (if existing) into span (for details see rows 65, 76) Transform ptr, ref (if existing) into reference (for details see rows 128, 136) Remove lb Keep only the content of any sub-elements used in container and include a comment about any data potentially encoded in their attributes; values of attributes also available with referringString, span, and reference will of course be moved over to these three elements accordingly	Move the value of localtype, if existing, into @addressLineType; add @addressLineTypeEncoding to control with the value "otherAddressLineTypeEncoding"; include a comment that those other address line types should be specified using a conventionDeclaration
agencycode	agencyCode	Apply camelCasing	Remove @encodinganalog and include a comment about the removal; if any parent element in EAD3 indicates a relatedencoding according to MARC21, add the MARC21 namespace and use the attribute @marc21:tag in agencyCode to include the value of @encodinganalog (xmlns:marc21="http://www.loc.gov/MARC21/slim" marc21:tag="...") Remove @localtype and include comment about the removal

Note: This sheet shows the changes that would need to be applied to each existing element and attribute in order to move it from EAD3 to EAD 4.0, while the sheets "Elements" and "Attributes" show the changes applied to each element and attribute in terms of its definition within the EAD schema. This sheet here therefore only refers to existing elements, but does not cover elements newly added to EAD 4.0.
Furthermore, this sheet only includes the actual changes. Anything that stays the same (e.g. an element or attribute that isn't renamed or replaced or doesn't have camelCasing applied or when an element retains the same attributes in EAD 4.0 as it had in EAD3), won't be mentioned as such.

General attributes' transformation for all elements that have existed in EAD3 already:
If @audience is used in one or more elements, add @audienceEncoding with the default value "EASList" to control
Replace @altrender with @xhtml:style and add the XHTML namespace to the element using @altrender; include a comment to confirm that @altrender was indeed used to indicate alternative formatting
Rename @lang to @languageOfElement; make sure @languageEncoding exists in control; if it doesn't, add the attribute with the default value "otherLanguageEncoding"
Rename @script to @scriptOfElement; make sure @scriptEncoding exists in control; if it doesn't, add the attribute with the default value "otherScriptEncoding"

General transformation for attributes of elements that are removed:
If a parent element is removed from EAD 4.0, but there is a transformation route for its child elements and if parent and child elements share attributes, check whether the attributes exist with the child element already; if the child elements include the attributes already, remove the attributes of the parent element along with the element itself; if the child elements do not include the attributes already, move the attributes of the parent element to each of its child elements without them

Element in EAD3	**Element in EAD 4.0**	**Transformation route for element and its content (draft)**	**Transformation route for attributes used with the element (draft)**
agencyname	agencyName	Apply camelCasing Move all instances of agencyName to appear before any otherAgencyCode (if existing)	Remove @encodinganalog and include a comment about the removal; if any parent element in EAD3 indicates a relatedencoding according to MARC21, add the MARC21 namespace and use the attribute @marc21:tag in agencyName to include the value of @encodinganalog (xmlns:marc21="http://www.loc.gov/MARC21/slim" marc21:tag="...") Remove @localtype and include comment about the removal
agent	agent	Move the content of agent into its new sub-element agentName Add a sub-element agentType including the value of maintenanceevent/agenttype@value	Remove @encodinganalog and include a comment about the removal; if any parent element in EAD3 indicates a relatedencoding according to MARC21, add the MARC21 namespace and use the attribute @marc21:tag in agent to include the value of @encodinganalog (xmlns:marc21="http://www.loc.gov/MARC21/slim" marc21:tag="...")
agenttype	agentType	Apply camelCasing Move maintenanceevent/agenttype into maintenanceEvent/agent/agentType (where that same agent also holds a parallel agentName element including the content previously held in agent directly)	Remove @value, using the value of agenttype@value as content of agentType directly Remove @encodinganalog and include a comment about the removal; if any parent element in EAD3 indicates a relatedencoding according to MARC21, add the MARC21 namespace and use the attribute @marc21:tag in agentType to include the value of @encodinganalog (xmlns:marc21="http://www.loc.gov/MARC21/slim" marc21:tag="...")
altformavail	formAvailable	Rename to formAvailable and apply camelCasing Make sure that this new formAvailable element sits within a form**s**Available plural element added directly to archDesc respectively the relevant numbered or unnumbered c element (if formsAvailable already exists on this level of description, add the new formAvailable to it after any already existing formAvailable elements) If only used with p, keep as is If used with p and any other formatting elements (i.e. blockquote, chronlist, head, list, table), move all into formattingExtension while adapting to XHTML where necessary (see rows 24, 40, 82, 96, 117, 155 for details) If used with nesting, move the sub-elements of the nested altformavail into formAvailable/formattingExtension to recreate the complete structure of the element alongside the sub-elements of the unnested (or parent) altformavail; in that case, even if the unnested (or parent) altformavail is only used with p, those p elements would also be used into formAvailable/formattingExtension	If @localtype is used, apply camelCasing to change to @localType; create a localTypeDeclaration with @id in control and add the default content "List of local types for alternative forms available" to its sub-element reference; then add @localTypeDeclarationReference to formAvailable pointing to the @id of said localTypeDeclaration; if a localTypeDeclaration with this content in its sub-element reference already exists, point to this instead of creating another one) Remove @encodinganalog and include a comment about the removal; if any parent element in EAD3 indicates a relatedencoding according to MARC21, add the MARC21 namespace and use the attribute @marc21:tag in formAvailable to include the value of @encodinganalog (xmlns:marc21="http://www.loc.gov/MARC21/slim" marc21:tag="...")

Note: This sheet shows the changes that would need to be applied to each existing element and attribute in order to move it from EAD3 to EAD 4.0, while the sheets "Elements" and "Attributes" show the changes applied to each element and attribute in terms of its definition within the EAD schema. This sheet here therefore only refers to existing elements, but does not cover elements newly added to EAD 4.0.
Furthermore, this sheet only includes the actual changes. Anything that stays the same (e.g. an element or attribute that isn't renamed or replaced or doesn't have camelCasing applied or when an element retains the same attributes in EAD 4.0 as it had in EAD3), won't be mentioned as such.

General attributes' transformation for all elements that have existed in EAD3 already:
If @audience is used in one or more elements, add @audienceEncoding with the default value "EASList" to control
Replace @altrender with @xhtml:style and add the XHTML namespace to the element using @altrender; include a comment to confirm that @altrender was indeed used to indicate alternative formatting
Rename @lang to @languageOfElement; make sure @languageEncoding exists in control; if it doesn't, add the attribute with the default value "otherLanguageEncoding"
Rename @script to @scriptOfElement; make sure @scriptEncoding exists in control; if it doesn't, add the attribute with the default value "otherScriptEncoding"

General transformation for attributes of elements that are removed:
If a parent element is removed from EAD 4.0, but there is a transformation route for its child elements and if parent and child elements share attributes, check whether the attributes exist with the child element already; if the child elements include the attributes already, remove the attributes of the parent element along with the element itself; if the child elements do not include the attributes already, move the attributes of the parent element to each of its child elements without them

Element in EAD3	**Element in EAD 4.0**	**Transformation route for element and its content (draft)**	**Transformation route for attributes used with the element (draft)**
appraisal	appraisal	If only used with p, keep as is If used with p and any other formatting elements (i.e. blockquote, chronlist, head, list, table), move all into formattingExtension while adapting to XHTML where necessary (see rows 24, 40, 82, 96, 117, 155 for details) If used with nesting, move the sub-elements of the nested appraisal into appraisal/formattingExtension to recreate the complete structure of the element alongside the sub-elements of the unnested (or parent) appraisal; in that case, even if the unnested (or parent) appraisal is only used with p, those p elements would also be used into appraisal/formattingExtension	If @localtype is used, apply camelCasing to change to @localType; create a localTypeDeclaration with @id in control and add the default content "List of local types for appraisal" to its sub-element reference; then add @localTypeDeclarationReference to appraisal pointing to the @id of said localTypeDeclaration; if a localTypeDeclaration with this content in its sub-element reference already exists, point to this instead of creating another one) Remove @encodinganalog and include a comment about the removal; if any parent element in EAD3 indicates a relatedencoding according to MARC21, add the MARC21 namespace and use the attribute @marc21:tag in appraisal to include the value of @encodinganalog (xmlns:marc21="http://www.loc.gov/MARC21/slim" marc21:tag="...")
archdesc	archDesc	Apply camelCasing Rename sub-element did to identificationData If one of archdesc/did/origination, archdesc/did/repository, archdesc/controlaccess/corpname (or famname, name, persname), archdesc/index/indexentry/corpname (or famname, name, persname), archdesc/index/indexentry/namegrp/corpname (or famname, name, persname) exist, create an agents sub-element to archDesc If one of archdesc/did/dao, archdesc/did/daoset, archdesc/altformavail, archdesc/originalsloc exist, create a formsAvailable sub-element to archDesc If one of archdesc/controlaccess/function, archdesc/index/indexentry/function, archdesc/index/indexentry/namegrp/function exist, create a functions sub-element to archDesc Rename sub-element dsc to descriptionOfComponents and make sure that it always appears after all other sub-elements of archDesc Remove altformavail, index, legalstatus, originalsloc, relations from archDesc (see rows 15, 86, 95, 112, 140 for details) Remove any repeated dsc elements from archDesc (see row 61 for details) Apply camelCasing and name changes to sub-elements as necessary (see relevant rows for details on each sub-element separately)	Add @levelEncoding to control If @level (with archDesc, but also with all c elements in the same EAD XML file) is used with any value other than "otherlevel", use @levelEncoding with value "EASList" If @level is used with value "otherlevel" (either with archDesc or at least one of the c elements in the same EAD XML file), use @levelEncoding with value "otherLevelEncoding" and create a conventionDeclaration sub-element to control with reference saying "Definition of other level encodings for archDesc and components" If @level is used with value "otherlevel" and @otherlevel exists in parallel, move the value of @otherlevel into @level Remove @relatedencoding and @encodinganalog and include a comment about the removal If either archDesc itself or ead indicates a relatedencoding according to MARC21, add the MARC21 namespace and use the attribute @marc21:tag in archDesc to include the value of @encodinganalog (xmlns:marc21="http://www.loc.gov/MARC21/slim" marc21:tag="...") If @localtype is used, apply camelCasing to change to @localType; create a localTypeDeclaration with @id in control and add the default content "List of local types for the highest arch description level" to its sub-element reference; then add @localTypeDeclarationReference to archDesc pointing to the @id of said localTypeDeclaration

Note: This sheet shows the changes that would need to be applied to each existing element and attribute in order to move it from EAD3 to EAD 4.0, while the sheets "Elements" and "Attributes" show the changes applied to each element and attribute in terms of its definition within the EAD schema. This sheet here therefore only refers to existing elements, but does not cover elements newly added to EAD 4.0.
Furthermore, this sheet only includes the actual changes. Anything that stays the same (e.g. an element or attribute that isn't renamed or replaced or doesn't have camelCasing applied or when an element retains the same attributes in EAD 4.0 as it had in EAD3), won't be mentioned as such.

General attributes' transformation for all elements that have existed in EAD3 already:
If @audience is used in one or more elements, add @audienceEncoding with the default value "EASList" to control
Replace @altrender with @xhtml:style and add the XHTML namespace to the element using @altrender; include a comment to confirm that @altrender was indeed used to indicate alternative formatting
Rename @lang to @languageOfElement; make sure @languageEncoding exists in control; if it doesn't, add the attribute with the default value "otherLanguageEncoding"
Rename @script to @scriptOfElement; make sure @scriptEncoding exists in control; if it doesn't, add the attribute with the default value "otherScriptEncoding"

General transformation for attributes of elements that are removed:
If a parent element is removed from EAD 4.0, but there is a transformation route for its child elements and if parent and child elements share attributes, check whether the attributes exist with the child element already; if the child elements include the attributes already, remove the attributes of the parent element along with the element itself; if the child elements do not include the attributes already, move the attributes of the parent element to each of its child elements without them

Element in EAD3	**Element in EAD 4.0**	**Transformation route for element and its content (draft)**	**Transformation route for attributes used with the element (draft)**
archref	(reference)	Transform bibliography/archref into publicationNote/p/reference Transform otherfindaid/archref into otherFindAid/p/reference Transform relatedmaterial/archref into relatedMaterial/p/reference Transform separatedmaterial/archref into separatedMaterial/p/reference For archref's sub-elements: Transform abbr, expan, persname, corpname, famname, geogname, name, occupation, subject, genreform, function, title, date, footnote, num (if existing) into reference/referringString (for details see rows 4, 48, 52, 71, 72, 75, 78, 79, 80, 105, 108, 110, 119, 153, 160) Transform emph, foreign, quote (if existing) into reference/span (for details see rows 65, 76, 134) Move the value of ptr@href into reference@href (see row 128 for more details on transforming ptr) Move the value of archref/ref@href into reference@href (see row 136 for details on transforming ref) Remove lb Keep only the content of any sub-elements used in archref and include a comment about any data potentially encoded in their attributes; values of attributes also available with referringString, span, and reference will of course be moved over to these three elements accordingly	Remove @encodinganalog and include a comment about the removal; if any parent element in EAD3 indicates a relatedencoding according to MARC21, add the MARC21 namespace to reference and use the attribute @marc21:tag in reference to include the value of @encodinganalog (xmlns:marc21="http://www.loc.gov/MARC21/slim" marc21:tag="...")
arrangement	arrangement	If only used with p, keep as is If used with p and any other formatting elements (i.e. blockquote, chronlist, head, list, table), move all into formattingExtension while adapting to XHTML where necessary (see rows 24, 40, 82, 96, 117, 155 for details) If used with nesting, move the sub-elements of the nested arrangement into arrangement/formattingExtension to recreate the complete structure of the element alongside the sub-elements of the unnested (or parent) arrangement; in that case, even if the unnested (or parent) arrangement is only used with p, those p elements would also be used into arrangement/formattingExtension	If @localtype is used, apply camelCasing to change to @localType; create a localTypeDeclaration with @id in control and add the default content "List of local types for arrangement" to its sub-element reference; then add @localTypeDeclarationReference to arrangement pointing to the @id of said localTypeDeclaration; if a localTypeDeclaration with this content in its sub-element reference already exists, point to this instead of creating another one) Remove @encodinganalog and include a comment about the removal; if any parent element in EAD3 indicates a relatedencoding according to MARC21, add the MARC21 namespace and use the attribute @marc21:tag in arrangement to include the value of @encodinganalog (xmlns:marc21="http://www.loc.gov/MARC21/slim" marc21:tag="...")

Note: This sheet shows the changes that would need to be applied to each existing element and attribute in order to move it from EAD3 to EAD 4.0, while the sheets "Elements" and "Attributes" show the changes applied to each element and attribute in terms of its definition within the EAD schema. This sheet here therefore only refers to existing elements, but does not cover elements newly added to EAD 4.0.

Furthermore, this sheet only includes the actual changes. Anything that stays the same (e.g. an element or attribute that isn't renamed or replaced or doesn't have camelCasing applied or when an element retains the same attributes in EAD 4.0 as it had in EAD3), won't be mentioned as such.

General attributes' transformation for all elements that have existed in EAD3 already:

If @audience is used in one or more elements, add @audienceEncoding with the default value "EASList" to control

Replace @altrender with @xhtml:style and add the XHTML namespace to the element using @altrender; include a comment to confirm that @altrender was indeed used to indicate alternative formatting

Rename @lang to @languageOfElement; make sure @languageEncoding exists in control; if it doesn't, add the attribute with the default value "otherLanguageEncoding"

Rename @script to @scriptOfElement; make sure @scriptEncoding exists in control; if it doesn't, add the attribute with the default value "otherScriptEncoding"

General transformation for attributes of elements that are removed:

If a parent element is removed from EAD 4.0, but there is a transformation route for its child elements and if parent and child elements share attributes, check whether the attributes exist with the child element already; if the child elements include the attributes already, remove the attributes of the parent element along with the element itself; if the child elements do not include the attributes already, move the attributes of the parent element to each of its child elements without them

Element in EAD3	**Element in EAD 4.0**	**Transformation route for element and its content (draft)**	**Transformation route for attributes used with the element (draft)**
author	(agent)	Move the content of each author into a separate findAidDesc/agent/agentName and add a parallel findAidDesc/agent/agentRole with the text "Author" Keep only the content of any sub-elements (m.mixed.basic) used in author and include a comment about any data potentially encoded in their attributes; e.g. <author><abbr>ABC</abbr> (<expan>Alphabet Company</expan>) <foreign>à la mode</foreign> in the <ref>123 project</ref></author> becomes <agent><agentName>ABC (Alphabet) à la mode in the 123 project</agentName><agentRole>author</agentRole></agent>	Move author@localtype to agent@localType; create a localTypeDeclaration with @id in control and add the default content "List of local types for agents" to its sub-element reference; then add @localTypeDeclarationReference to agent pointing to the @id of said localTypeDeclaration; if a localTypeDeclaration with this content in its sub-element reference already exists, point to this instead of creating another one Remove @encodinganalog and include a comment about the removal; if any parent element in EAD3 indicates a relatedencoding according to MARC21, add the MARC21 namespace and use the attribute @marc21:tag in agent to include the value of @encodinganalog (xmlns:marc21="http://www.loc.gov/MARC21/slim" marc21:tag="...")
bibliography	publicationNote	Rename to publicationNote and apply camelCasing If only used with p, keep as is If used with p and any other formatting elements (i.e. blockquote, chronlist, head, list, table), move all into formattingExtension while adapting to XHTML where necessary (see rows 24, 40, 82, 96, 117, 155 for details) If used with archref and/or bibref, replace these elements with p/reference when in a pure EAD 4.0 encoding or with the appropriate XHTML encoding when used together with formattingExtension (see rows 18 and 22 for details); include a comment about reviewing this and considering an encoding using relations instead If used with nesting, move the sub-elements of the nested bibliography into publicationNote/formattingExtension to recreate the complete structure of the element alongside the sub-elements of the unnested (or parent) bibliography; in that case, even if the unnested (or parent) bibliography is only used with p, those p elements would also be used into publicationNote/formattingExtension	If @localtype is used, apply camelCasing to change to @localType; create a localTypeDeclaration with @id in control and add the default content "List of local types for publication notes" to its sub-element reference; then add @localTypeDeclarationReference to publicationNote pointing to the @id of said localTypeDeclaration; if a localTypeDeclaration with this content in its sub-element reference already exists, point to this instead of creating another one) Remove @encodinganalog and include a comment about the removal; if any parent element in EAD3 indicates a relatedencoding according to MARC21, add the MARC21 namespace and use the attribute @marc21:tag in publicationNote to include the value of @encodinganalog (xmlns:marc21="http://www.loc.gov/MARC21/slim" marc21:tag="...")

Note: This sheet shows the changes that would need to be applied to each existing element and attribute in order to move it from EAD3 to EAD 4.0, while the sheets "Elements" and "Attributes" show the changes applied to each element and attribute in terms of its definition within the EAD schema. This sheet here therefore only refers to existing elements, but does not cover elements newly added to EAD 4.0.
Furthermore, this sheet only includes the actual changes. Anything that stays the same (e.g. an element or attribute that isn't renamed or replaced or doesn't have camelCasing applied or when an element retains the same attributes in EAD 4.0 as it had in EAD3), won't be mentioned as such.

General attributes' transformation for all elements that have existed in EAD3 already:
If @audience is used in one or more elements, add @audienceEncoding with the default value "EASList" to control
Replace @altrender with @xhtml:style and add the XHTML namespace to the element using @altrender; include a comment to confirm that @altrender was indeed used to indicate alternative formatting
Rename @lang to @languageOfElement; make sure @languageEncoding exists in control; if it doesn't, add the attribute with the default value "otherLanguageEncoding"
Rename @script to @scriptOfElement; make sure @scriptEncoding exists in control; if it doesn't, add the attribute with the default value "otherScriptEncoding"

General transformation for attributes of elements that are removed:
If a parent element is removed from EAD 4.0, but there is a transformation route for its child elements and if parent and child elements share attributes, check whether the attributes exist with the child element already; if the child elements include the attributes already, remove the attributes of the parent element along with the element itself; if the child elements do not include the attributes already, move the attributes of the parent element to each of its child elements without them

Element in EAD3	**Element in EAD 4.0**	**Transformation route for element and its content (draft)**	**Transformation route for attributes used with the element (draft)**
bibref	(reference)	Transform bibliography/bibref into publicationNote/p/reference Transform otherfindaid/bibref into otherFindAid/p/reference Transform relatedmaterial/bibref into relatedMaterial/p/reference Transform separatedmaterial/bibref into separatedMaterial/p/reference For archref's sub-elements: Transform abbr, expan, persname, corpname, famname, geogname, name, occupation, subject, genreform, function, title, date, footnote, num (if existing) into reference/referringString (for details see rows 4, 48, 52, 71, 72, 75, 78, 79, 80, 105, 108, 110, 119, 153, 160) Transform emph (if existing) into reference/span (for details see rows 65, 76, 134) Move the value of ptr@href into reference@href (see row 128 for more details on transforming ptr) Move the value of archref/ref@href into reference@href (see row 136 for details on transforming ref) Remove lb Keep only the content of any sub-elements used in bibref and include a comment about any data potentially encoded in their attributes; values of attributes also available with referringString, span, and reference will of course be moved over to these three elements accordingly	Remove @encodinganalog and include a comment about the removal; if any parent element in EAD3 indicates a relatedencoding according to MARC21, add the MARC21 namespace to reference and use the attribute @marc21:tag in reference to include the value of @encodinganalog (xmlns:marc21="http://www.loc.gov/MARC21/slim" marc21:tag="...")
bioghist	biogHist	Apply camelCasing If only used with p, keep as is If used with p and any other formatting elements (i.e. blockquote, chronlist, head, list, table), move all into formattingExtension while adapting to XHTML where necessary (see rows 24, 40, 82, 96, 117, 155 for details) If used with nesting, move the sub-elements of the nested bioghist into biogHist/formattingExtension to recreate the complete structure of the element alongside the sub-elements of the unnested (or parent) bioghist; in that case, even if the unnested (or parent) bioghist is only used with p, those p elements would also be used into biogHist/formattingExtension	If @localtype is used, apply camelCasing to change to @localType; create a localTypeDeclaration with @id in control and add the default content "List of local types for biographic history" to its sub-element reference; then add @localTypeDeclarationReference to biogHist pointing to the @id of said localTypeDeclaration; if a localTypeDeclaration with this content in its sub-element reference already exists, point to this instead of creating another one) Remove @encodinganalog and include a comment about the removal; if any parent element in EAD3 indicates a relatedencoding according to MARC21, add the MARC21 namespace and use the attribute @marc21:tag in biogHist to include the value of @encodinganalog (xmlns:marc21="http://www.loc.gov/MARC21/slim" marc21:tag="...")
blockquote	(formattingExtension)	Transform blockquote to xhtml:blockquote Transform the sub-elements p to xhtml:p, list to @xhtml:ul, @xhtml:ol, or @xhtml:dl (depending whether or not @listtype is used and with which value), chronlist and table to @xhtml:table (see more details for these sub-elements in rows 40, 96, 117, 155)	Transform @audience="internal" into e.g. @xhtml:hidden (while removing the attribute if the value "external" is used) Transform @id into @xhtml:id Transform @altrender into @xhtml:style Transform @lang into @xhtml:lang Integrate @script with the value of @lang in @xhtml:lang by using ietf-bcp-47 values (or remove the attribute)

Note: This sheet shows the changes that would need to be applied to each existing element and attribute in order to move it from EAD3 to EAD 4.0, while the sheets "Elements" and "Attributes" show the changes applied to each element and attribute in terms of its definition within the EAD schema. This sheet here therefore only refers to existing elements, but does not cover elements newly added to EAD 4.0. Furthermore, this sheet only includes the actual changes. Anything that stays the same (e.g. an element or attribute that isn't renamed or replaced or doesn't have camelCasing applied or when an element retains the same attributes in EAD 4.0 as it had in EAD3), won't be mentioned as such.		**General attributes' transformation for all elements that have existed in EAD3 already:** If @audience is used in one or more elements, add @audienceEncoding with the default value "EASList" to control Replace @altrender with @xhtml:style and add the XHTML namespace to the element using @altrender; include a comment to confirm that @altrender was indeed used to indicate alternative formatting Rename @lang to @languageOfElement; make sure @languageEncoding exists in control; if it doesn't, add the attribute with the default value "otherLanguageEncoding" Rename @script to @scriptOfElement; make sure @scriptEncoding exists in control; if it doesn't, add the attribute with the default value "otherScriptEncoding" **General transformation for attributes of elements that are removed:** If a parent element is removed from EAD 4.0, but there is a transformation route for its child elements and if parent and child elements share attributes, check whether the attributes exist with the child element already; if the child elements include the attributes already, remove the attributes of the parent element along with the element itself; if the child elements do not include the attributes already, move the attributes of the parent element to each of its child elements without them	
Element in EAD3	**Element in EAD 4.0**	**Transformation route for element and its content (draft)**	**Transformation route for attributes used with the element (draft)**
c	c	Rename sub-element did to identificationData If one of c/did/origination, c/did/repository, c/controlaccess/corpname (or famname, name, persname), c/index/indexentry/corpname (or famname, name, persname), c/index/indexentry/namegrp/corpname (or famname, name, persname) exist, create an agents sub-element to c If one of c/did/dao, c/did/daoset, c/altformavail, c/originalsloc exist, create a formsAvailable sub-element to c If one of c/controlaccess/function, c/index/indexentry/function, c/index/indexentry/namegrp/function exist, create a functions sub-element to c Remove altformavail, index, legalstatus, originalsloc, relations from c (see rows 15, 86, 95, 112, 140 for details) Remove any repeated dsc elements from c (see row 61 for details) Apply camelCasing and name changes to sub-elements as necessary (see relevant rows for details on each sub-element separately)	If @level is used, make sure @levelEncoding has been added to control (should already have happened when transforming archDesc) If @level (in all c elements) is used with any value other than "otherlevel", use @levelEncoding with value "EASList" If @level is used with value "otherlevel" (with least one of the c elements in the same EAD XML file), use @levelEncoding with value "otherLevelEncoding" and create a conventionDeclaration sub-element to control with reference saying "Definition of other level encodings for archDesc and components" If @level is used with value "otherlevel" and @otherlevel exists in parallel, move the value of @otherlevel into @level Remove @encodinganalog and include a comment about the removal; if either archDesc or ead indicates a relatedencoding according to MARC21, add the MARC21 namespace and use the attribute @marc21:tag in c to include the value of @encodinganalog (xmlns:marc21="http://www.loc.gov/MARC21/slim" marc21:tag="...") If @localtype is used, apply camelCasing to change to @localType; create a localTypeDeclaration with @id in control and add the default content "List of local types for component description level" to its sub-element reference; then add @localTypeDeclarationReference to c pointing to the @id of said localTypeDeclaration
c01	c01	Rename sub-element did to identificationData If one of c/did/origination, c/did/repository, c/controlaccess/corpname (or famname, name, persname), c/index/indexentry/corpname (or famname, name, persname), c/index/indexentry/namegrp/corpname (or famname, name, persname) exist, create an agents sub-element to c If one of c/did/dao, c/did/daoset, c/altformavail, c/originalsloc exist, create a formsAvailable sub-element to c If one of c/controlaccess/function, c/index/indexentry/function, c/index/indexentry/namegrp/function exist, create a functions sub-element to c Remove altformavail, index, legalstatus, originalsloc, relations from c01 (see rows 15, 86, 95, 112, 140 for details) Remove any repeated dsc elements from c01 (see row 61 for details) Apply camelCasing and name changes to sub-elements as necessary (see relevant rows for details on each sub-element separately)	If @level is used, make sure @levelEncoding has been added to control (should already have happened when transforming archDesc) If @level (in all c01 elements) is used with any value other than "otherlevel", use @levelEncoding with value "EASList" If @level is used with value "otherlevel" (with least one of the c01 elements in the same EAD XML file), use @levelEncoding with value "otherLevelEncoding" and create a conventionDeclaration sub-element to control with reference saying "Definition of other level encodings for archDesc and components" If @level is used with value "otherlevel" and @otherlevel exists in parallel, move the value of @otherlevel into @level Remove @encodinganalog and include a comment about the removal; if either archDesc or ead indicates a relatedencoding according to MARC21, add the MARC21 namespace and use the attribute @marc21:tag in c01 to include the value of @encodinganalog (xmlns:marc21="http://www.loc.gov/MARC21/slim" marc21:tag="...") If @localtype is used, apply camelCasing to change to @localType; create a localTypeDeclaration with @id in control and add the default content "List of local types for component description level" to its sub-element reference; then add @localTypeDeclarationReference to c01 pointing to the @id of said localTypeDeclaration

Note: This sheet shows the changes that would need to be applied to each existing element and attribute in order to move it from EAD3 to EAD 4.0, while the sheets "Elements" and "Attributes" show the changes applied to each element and attribute in terms of its definition within the EAD schema. This sheet here therefore only refers to existing elements, but does not cover elements newly added to EAD 4.0. Furthermore, this sheet only includes the actual changes. Anything that stays the same (e.g. an element or attribute that isn't renamed or replaced or doesn't have camelCasing applied or when an element retains the same attributes in EAD 4.0 as it had in EAD3), won't be mentioned as such.		**General attributes' transformation for all elements that have existed in EAD3 already:** If @audience is used in one or more elements, add @audienceEncoding with the default value "EASList" to control Replace @altrender with @xhtml:style and add the XHTML namespace to the element using @altrender; include a comment to confirm that @altrender was indeed used to indicate alternative formatting Rename @lang to @languageOfElement; make sure @languageEncoding exists in control; if it doesn't, add the attribute with the default value "otherLanguageEncoding" Rename @script to @scriptOfElement; make sure @scriptEncoding exists in control; if it doesn't, add the attribute with the default value "otherScriptEncoding" **General transformation for attributes of elements that are removed:** If a parent element is removed from EAD 4.0, but there is a transformation route for its child elements and if parent and child elements share attributes, check whether the attributes exist with the child element already; if the child elements include the attributes already, remove the attributes of the parent element along with the element itself; if the child elements do not include the attributes already, move the attributes of the parent element to each of its child elements without them	
Element in EAD3	**Element in EAD 4.0**	**Transformation route for element and its content (draft)**	**Transformation route for attributes used with the element (draft)**
c02	c02	Rename sub-element did to identificationData If one of c/did/origination, c/did/repository, c/controlaccess/corpname (or famname, name, persname), c/index/indexentry/corpname (or famname, name, persname), c/index/indexentry/namegrp/corpname (or famname, name, persname) exist, create an agents sub-element to c If one of c/did/dao, c/did/daoset, c/altformavail, c/originalsloc exist, create a formsAvailable sub-element to c If one of c/controlaccess/function, c/index/indexentry/function, c/index/indexentry/namegrp/function exist, create a functions sub-element to c Remove altformavail, index, legalstatus, originalsloc, relations from c02 (see rows 15, 86, 95, 112, 140 for details) Remove any repeated dsc elements from c02 (see row 61 for details) Apply camelCasing and name changes to sub-elements as necessary (see relevant rows for details on each sub-element separately)	If @level is used, make sure @levelEncoding has been added to control (should already have happened when transforming archDesc) If @level (in all c02 elements) is used with any value other than "otherlevel", use @levelEncoding with value "EASList" If @level is used with value "otherlevel" (with least one of the c02 elements in the same EAD XML file), use @levelEncoding with value "otherLevelEncoding" and create a conventionDeclaration sub-element to control with reference saying "Definition of other level encodings for archDesc and components" If @level is used with value "otherlevel" and @otherlevel exists in parallel, move the value of @otherlevel into @level Remove @encodinganalog and include a comment about the removal; if either archDesc or ead indicates a relatedencoding according to MARC21, add the MARC21 namespace and use the attribute @marc21:tag in c02 to include the value of @encodinganalog (xmlns:marc21="http://www.loc.gov/MARC21/slim" marc21:tag="...") If @localtype is used, apply camelCasing to change to @localType; create a localTypeDeclaration with @id in control and add the default content "List of local types for component description level" to its sub-element reference; then add @localTypeDeclarationReference to c02 pointing to the @id of said localTypeDeclaration
c03	c03	Rename sub-element did to identificationData If one of c/did/origination, c/did/repository, c/controlaccess/corpname (or famname, name, persname), c/index/indexentry/corpname (or famname, name, persname), c/index/indexentry/namegrp/corpname (or famname, name, persname) exist, create an agents sub-element to c If one of c/did/dao, c/did/daoset, c/altformavail, c/originalsloc exist, create a formsAvailable sub-element to c If one of c/controlaccess/function, c/index/indexentry/function, c/index/indexentry/namegrp/function exist, create a functions sub-element to c Remove altformavail, index, legalstatus, originalsloc, relations from c03 (see rows 15, 86, 95, 112, 140 for details) Remove any repeated dsc elements from c03 (see row 61 for details) Apply camelCasing and name changes to sub-elements as necessary (see relevant rows for details on each sub-element separately)	If @level is used, make sure @levelEncoding has been added to control (should already have happened when transforming archDesc) If @level (in all c03 elements) is used with any value other than "otherlevel", use @levelEncoding with value "EASList" If @level is used with value "otherlevel" (with least one of the c03 elements in the same EAD XML file), use @levelEncoding with value "otherLevelEncoding" and create a conventionDeclaration sub-element to control with reference saying "Definition of other level encodings for archDesc and components" If @level is used with value "otherlevel" and @otherlevel exists in parallel, move the value of @otherlevel into @level Remove @encodinganalog and include a comment about the removal; if either archDesc or ead indicates a relatedencoding according to MARC21, add the MARC21 namespace and use the attribute @marc21:tag in c03 to include the value of @encodinganalog (xmlns:marc21="http://www.loc.gov/MARC21/slim" marc21:tag="...") If @localtype is used, apply camelCasing to change to @localType; create a localTypeDeclaration with @id in control and add the default content "List of local types for component description level" to its sub-element reference; then add @localTypeDeclarationReference to c03 pointing to the @id of said localTypeDeclaration

Note: This sheet shows the changes that would need to be applied to each existing element and attribute in order to move it from EAD3 to EAD 4.0, while the sheets "Elements" and "Attributes" show the changes applied to each element and attribute in terms of its definition within the EAD schema. This sheet here therefore only refers to existing elements, but does not cover elements newly added to EAD 4.0.
Furthermore, this sheet only includes the actual changes. Anything that stays the same (e.g. an element or attribute that isn't renamed or replaced or doesn't have camelCasing applied or when an element retains the same attributes in EAD 4.0 as it had in EAD3), won't be mentioned as such.

General attributes' transformation for all elements that have existed in EAD3 already:
If @audience is used in one or more elements, add @audienceEncoding with the default value "EASList" to control
Replace @altrender with @xhtml:style and add the XHTML namespace to the element using @altrender; include a comment to confirm that @altrender was indeed used to indicate alternative formatting
Rename @lang to @languageOfElement; make sure @languageEncoding exists in control; if it doesn't, add the attribute with the default value "otherLanguageEncoding"
Rename @script to @scriptOfElement; make sure @scriptEncoding exists in control; if it doesn't, add the attribute with the default value "otherScriptEncoding"

General transformation for attributes of elements that are removed:
If a parent element is removed from EAD 4.0, but there is a transformation route for its child elements and if parent and child elements share attributes, check whether the attributes exist with the child element already; if the child elements include the attributes already, remove the attributes of the parent element along with the element itself; if the child elements do not include the attributes already, move the attributes of the parent element to each of its child elements without them

Element in EAD3	**Element in EAD 4.0**	**Transformation route for element and its content (draft)**	**Transformation route for attributes used with the element (draft)**
c04	c04	Rename sub-element did to identificationData If one of c/did/origination, c/did/repository, c/controlaccess/corpname (or famname, name, persname), c/index/indexentry/corpname (or famname, name, persname), c/index/indexentry/namegrp/corpname (or famname, name, persname) exist, create an agents sub-element to c If one of c/did/dao, c/did/daoset, c/altformavail, c/originalsloc exist, create a formsAvailable sub-element to c If one of c/controlaccess/function, c/index/indexentry/function, c/index/indexentry/namegrp/function exist, create a functions sub-element to c Remove altformavail, index, legalstatus, originalsloc, relations from c04 (see rows 15, 86, 95, 112, 140 for details) Remove any repeated dsc elements from c04 (see row 61 for details) Apply camelCasing and name changes to sub-elements as necessary (see relevant rows for details on each sub-element separately)	If @level is used, make sure @levelEncoding has been added to control (should already have happened when transforming archDesc) If @level (in all c04 elements) is used with any value other than "otherlevel", use @levelEncoding with value "EASList" If @level is used with value "otherlevel" (with least one of the c04 elements in the same EAD XML file), use @levelEncoding with value "otherLevelEncoding" and create a conventionDeclaration sub-element to control with reference saying "Definition of other level encodings for archDesc and components" If @level is used with value "otherlevel" and @otherlevel exists in parallel, move the value of @otherlevel into @level Remove @encodinganalog and include a comment about the removal; if either archDesc or ead indicates a relatedencoding according to MARC21, add the MARC21 namespace and use the attribute @marc21:tag in c04 to include the value of @encodinganalog (xmlns:marc21="http://www.loc.gov/MARC21/slim" marc21:tag="...") If @localtype is used, apply camelCasing to change to @localType; create a localTypeDeclaration with @id in control and add the default content "List of local types for component description level" to its sub-element reference; then add @localTypeDeclarationReference to c04 pointing to the @id of said localTypeDeclaration
c05	c05	Rename sub-element did to identificationData If one of c/did/origination, c/did/repository, c/controlaccess/corpname (or famname, name, persname), c/index/indexentry/corpname (or famname, name, persname), c/index/indexentry/namegrp/corpname (or famname, name, persname) exist, create an agents sub-element to c If one of c/did/dao, c/did/daoset, c/altformavail, c/originalsloc exist, create a formsAvailable sub-element to c If one of c/controlaccess/function, c/index/indexentry/function, c/index/indexentry/namegrp/function exist, create a functions sub-element to c Remove altformavail, index, legalstatus, originalsloc, relations from c05 (see rows 15, 86, 95, 112, 140 for details) Remove any repeated dsc elements from c05 (see row 61 for details) Apply camelCasing and name changes to sub-elements as necessary (see relevant rows for details on each sub-element separately)	If @level is used, make sure @levelEncoding has been added to control (should already have happened when transforming archDesc) If @level (in all c05 elements) is used with any value other than "otherlevel", use @levelEncoding with value "EASList" If @level is used with value "otherlevel" (with least one of the c05 elements in the same EAD XML file), use @levelEncoding with value "otherLevelEncoding" and create a conventionDeclaration sub-element to control with reference saying "Definition of other level encodings for archDesc and components" If @level is used with value "otherlevel" and @otherlevel exists in parallel, move the value of @otherlevel into @level Remove @encodinganalog and include a comment about the removal; if either archDesc or ead indicates a relatedencoding according to MARC21, add the MARC21 namespace and use the attribute @marc21:tag in c05 to include the value of @encodinganalog (xmlns:marc21="http://www.loc.gov/MARC21/slim" marc21:tag="...") If @localtype is used, apply camelCasing to change to @localType; create a localTypeDeclaration with @id in control and add the default content "List of local types for component description level" to its sub-element reference; then add @localTypeDeclarationReference to c05 pointing to the @id of said localTypeDeclaration

Note: This sheet shows the changes that would need to be applied to each existing element and attribute in order to move it from EAD3 to EAD 4.0, while the sheets "Elements" and "Attributes" show the changes applied to each element and attribute in terms of its definition within the EAD schema. This sheet here therefore only refers to existing elements, but does not cover elements newly added to EAD 4.0.
Furthermore, this sheet only includes the actual changes. Anything that stays the same (e.g. an element or attribute that isn't renamed or replaced or doesn't have camelCasing applied or when an element retains the same attributes in EAD 4.0 as it had in EAD3), won't be mentioned as such.

General attributes' transformation for all elements that have existed in EAD3 already:
If @audience is used in one or more elements, add @audienceEncoding with the default value "EASList" to control
Replace @altrender with @xhtml:style and add the XHTML namespace to the element using @altrender; include a comment to confirm that @altrender was indeed used to indicate alternative formatting
Rename @lang to @languageOfElement; make sure @languageEncoding exists in control; if it doesn't, add the attribute with the default value "otherLanguageEncoding"
Rename @script to @scriptOfElement; make sure @scriptEncoding exists in control; if it doesn't, add the attribute with the default value "otherScriptEncoding"

General transformation for attributes of elements that are removed:
If a parent element is removed from EAD 4.0, but there is a transformation route for its child elements and if parent and child elements share attributes, check whether the attributes exist with the child element already; if the child elements include the attributes already, remove the attributes of the parent element along with the element itself; if the child elements do not include the attributes already, move the attributes of the parent element to each of its child elements without them

Element in EAD3	**Element in EAD 4.0**	**Transformation route for element and its content (draft)**	**Transformation route for attributes used with the element (draft)**
c06	c06	Rename sub-element did to identificationData If one of c/did/origination, c/did/repository, c/controlaccess/corpname (or famname, name, persname), c/index/indexentry/corpname (or famname, name, persname), c/index/indexentry/namegrp/corpname (or famname, name, persname) exist, create an agents sub-element to c If one of c/did/dao, c/did/daoset, c/altformavail, c/originalsloc exist, create a formsAvailable sub-element to c If one of c/controlaccess/function, c/index/indexentry/function, c/index/indexentry/namegrp/function exist, create a functions sub-element to c Remove altformavail, index, legalstatus, originalsloc, relations from c06 (see rows 15, 86, 95, 112, 140 for details) Remove any repeated dsc elements from c06 (see row 61 for details) Apply camelCasing and name changes to sub-elements as necessary (see relevant rows for details on each sub-element separately)	If @level is used, make sure @levelEncoding has been added to control (should already have happened when transforming archDesc) If @level (in all c06 elements) is used with any value other than "otherlevel", use @levelEncoding with value "EASList" If @level is used with value "otherlevel" (with least one of the c06 elements in the same EAD XML file), use @levelEncoding with value "otherLevelEncoding" and create a conventionDeclaration sub-element to control with reference saying "Definition of other level encodings for archDesc and components" If @level is used with value "otherlevel" and @otherlevel exists in parallel, move the value of @otherlevel into @level Remove @encodinganalog and include a comment about the removal; if either archDesc or ead indicates a relatedencoding according to MARC21, add the MARC21 namespace and use the attribute @marc21:tag in c06 to include the value of @encodinganalog (xmlns:marc21="http://www.loc.gov/MARC21/slim" marc21:tag="...") If @localtype is used, apply camelCasing to change to @localType; create a localTypeDeclaration with @id in control and add the default content "List of local types for component description level" to its sub-element reference; then add @localTypeDeclarationReference to c06 pointing to the @id of said localTypeDeclaration
c07	c07	Rename sub-element did to identificationData If one of c/did/origination, c/did/repository, c/controlaccess/corpname (or famname, name, persname), c/index/indexentry/corpname (or famname, name, persname), c/index/indexentry/namegrp/corpname (or famname, name, persname) exist, create an agents sub-element to c If one of c/did/dao, c/did/daoset, c/altformavail, c/originalsloc exist, create a formsAvailable sub-element to c If one of c/controlaccess/function, c/index/indexentry/function, c/index/indexentry/namegrp/function exist, create a functions sub-element to c Remove altformavail, index, legalstatus, originalsloc, relations from c07 (see rows 15, 86, 95, 112, 140 for details) Remove any repeated dsc elements from c07 (see row 61 for details) Apply camelCasing and name changes to sub-elements as necessary (see relevant rows for details on each sub-element separately)	If @level is used, make sure @levelEncoding has been added to control (should already have happened when transforming archDesc) If @level (in all c07 elements) is used with any value other than "otherlevel", use @levelEncoding with value "EASList" If @level is used with value "otherlevel" (with least one of the c07 elements in the same EAD XML file), use @levelEncoding with value "otherLevelEncoding" and create a conventionDeclaration sub-element to control with reference saying "Definition of other level encodings for archDesc and components" If @level is used with value "otherlevel" and @otherlevel exists in parallel, move the value of @otherlevel into @level Remove @encodinganalog and include a comment about the removal; if either archDesc or ead indicates a relatedencoding according to MARC21, add the MARC21 namespace and use the attribute @marc21:tag in c07 to include the value of @encodinganalog (xmlns:marc21="http://www.loc.gov/MARC21/slim" marc21:tag="...") If @localtype is used, apply camelCasing to change to @localType; create a localTypeDeclaration with @id in control and add the default content "List of local types for component description level" to its sub-element reference; then add @localTypeDeclarationReference to c07 pointing to the @id of said localTypeDeclaration

Note: This sheet shows the changes that would need to be applied to each existing element and attribute in order to move it from EAD3 to EAD 4.0, while the sheets "Elements" and "Attributes" show the changes applied to each element and attribute in terms of its definition within the EAD schema. This sheet here therefore only refers to existing elements, but does not cover elements newly added to EAD 4.0.
Furthermore, this sheet only includes the actual changes. Anything that stays the same (e.g. an element or attribute that isn't renamed or replaced or doesn't have camelCasing applied or when an element retains the same attributes in EAD 4.0 as it had in EAD3), won't be mentioned as such.

General attributes' transformation for all elements that have existed in EAD3 already:
If @audience is used in one or more elements, add @audienceEncoding with the default value "EASList" to control
Replace @altrender with @xhtml:style and add the XHTML namespace to the element using @altrender; include a comment to confirm that @altrender was indeed used to indicate alternative formatting
Rename @lang to @languageOfElement; make sure @languageEncoding exists in control; if it doesn't, add the attribute with the default value "otherLanguageEncoding"
Rename @script to @scriptOfElement; make sure @scriptEncoding exists in control; if it doesn't, add the attribute with the default value "otherScriptEncoding"

General transformation for attributes of elements that are removed:
If a parent element is removed from EAD 4.0, but there is a transformation route for its child elements and if parent and child elements share attributes, check whether the attributes exist with the child element already; if the child elements include the attributes already, remove the attributes of the parent element along with the element itself; if the child elements do not include the attributes already, move the attributes of the parent element to each of its child elements without them

Element in EAD3	**Element in EAD 4.0**	**Transformation route for element and its content (draft)**	**Transformation route for attributes used with the element (draft)**
c08	c08	Rename sub-element did to identificationData If one of c/did/origination, c/did/repository, c/controlaccess/corpname (or famname, name, persname), c/index/indexentry/corpname (or famname, name, persname), c/index/indexentry/namegrp/corpname (or famname, name, persname) exist, create an agents sub-element to c If one of c/did/dao, c/did/daoset, c/altformavail, c/originalsloc exist, create a formsAvailable sub-element to c If one of c/controlaccess/function, c/index/indexentry/function, c/index/indexentry/namegrp/function exist, create a functions sub-element to c Remove altformavail, index, legalstatus, originalsloc, relations from c08 (see rows 15, 86, 95, 112, 140 for details) Remove any repeated dsc elements from c08 (see row 61 for details) Apply camelCasing and name changes to sub-elements as necessary (see relevant rows for details on each sub-element separately)	If @level is used, make sure @levelEncoding has been added to control (should already have happened when transforming archDesc) If @level (in all c08 elements) is used with any value other than "otherlevel", use @levelEncoding with value "EASList" If @level is used with value "otherlevel" (with least one of the c08 elements in the same EAD XML file), use @levelEncoding with value "otherLevelEncoding" and create a conventionDeclaration sub-element to control with reference saying "Definition of other level encodings for archDesc and components" If @level is used with value "otherlevel" and @otherlevel exists in parallel, move the value of @otherlevel into @level Remove @encodinganalog and include a comment about the removal; if either archDesc or ead indicates a relatedencoding according to MARC21, add the MARC21 namespace and use the attribute @marc21:tag in c08 to include the value of @encodinganalog (xmlns:marc21="http://www.loc.gov/MARC21/slim" marc21:tag="...") If @localtype is used, apply camelCasing to change to @localType; create a localTypeDeclaration with @id in control and add the default content "List of local types for component description level" to its sub-element reference; then add @localTypeDeclarationReference to c08 pointing to the @id of said localTypeDeclaration
c09	c09	Rename sub-element did to identificationData If one of c/did/origination, c/did/repository, c/controlaccess/corpname (or famname, name, persname), c/index/indexentry/corpname (or famname, name, persname), c/index/indexentry/namegrp/corpname (or famname, name, persname) exist, create an agents sub-element to c If one of c/did/dao, c/did/daoset, c/altformavail, c/originalsloc exist, create a formsAvailable sub-element to c If one of c/controlaccess/function, c/index/indexentry/function, c/index/indexentry/namegrp/function exist, create a functions sub-element to c Remove altformavail, index, legalstatus, originalsloc, relations from c09 (see rows 15, 86, 95, 112, 140 for details) Remove any repeated dsc elements from c09 (see row 61 for details) Apply camelCasing and name changes to sub-elements as necessary (see relevant rows for details on each sub-element separately)	If @level is used, make sure @levelEncoding has been added to control (should already have happened when transforming archDesc) If @level (in all c09 elements) is used with any value other than "otherlevel", use @levelEncoding with value "EASList" If @level is used with value "otherlevel" (with least one of the c09 elements in the same EAD XML file), use @levelEncoding with value "otherLevelEncoding" and create a conventionDeclaration sub-element to control with reference saying "Definition of other level encodings for archDesc and components" If @level is used with value "otherlevel" and @otherlevel exists in parallel, move the value of @otherlevel into @level Remove @encodinganalog and include a comment about the removal; if either archDesc or ead indicates a relatedencoding according to MARC21, add the MARC21 namespace and use the attribute @marc21:tag in c09 to include the value of @encodinganalog (xmlns:marc21="http://www.loc.gov/MARC21/slim" marc21:tag="...") If @localtype is used, apply camelCasing to change to @localType; create a localTypeDeclaration with @id in control and add the default content "List of local types for component description level" to its sub-element reference; then add @localTypeDeclarationReference to c09 pointing to the @id of said localTypeDeclaration

Note: This sheet shows the changes that would need to be applied to each existing element and attribute in order to move it from EAD3 to EAD 4.0, while the sheets "Elements" and "Attributes" show the changes applied to each element and attribute in terms of its definition within the EAD schema. This sheet here therefore only refers to existing elements, but does not cover elements newly added to EAD 4.0.
Furthermore, this sheet only includes the actual changes. Anything that stays the same (e.g. an element or attribute that isn't renamed or replaced or doesn't have camelCasing applied or when an element retains the same attributes in EAD 4.0 as it had in EAD3), won't be mentioned as such.

General attributes' transformation for all elements that have existed in EAD3 already:
If @audience is used in one or more elements, add @audienceEncoding with the default value "EASList" to control
Replace @altrender with @xhtml:style and add the XHTML namespace to the element using @altrender; include a comment to confirm that @altrender was indeed used to indicate alternative formatting
Rename @lang to @languageOfElement; make sure @languageEncoding exists in control; if it doesn't, add the attribute with the default value "otherLanguageEncoding"
Rename @script to @scriptOfElement; make sure @scriptEncoding exists in control; if it doesn't, add the attribute with the default value "otherScriptEncoding"

General transformation for attributes of elements that are removed:
If a parent element is removed from EAD 4.0, but there is a transformation route for its child elements and if parent and child elements share attributes, check whether the attributes exist with the child element already; if the child elements include the attributes already, remove the attributes of the parent element along with the element itself; if the child elements do not include the attributes already, move the attributes of the parent element to each of its child elements without them

Element in EAD3	Element in EAD 4.0	Transformation route for element and its content (draft)	Transformation route for attributes used with the element (draft)
c10	c10	Rename sub-element did to identificationData If one of c/did/origination, c/did/repository, c/controlaccess/corpname (or famname, name, persname), c/index/indexentry/corpname (or famname, name, persname), c/index/indexentry/namegrp/corpname (or famname, name, persname) exist, create an agents sub-element to c If one of c/did/dao, c/did/daoset, c/altformavail, c/originalsloc exist, create a formsAvailable sub-element to c If one of c/controlaccess/function, c/index/indexentry/function, c/index/indexentry/namegrp/function exist, create a functions sub-element to c Remove altformavail, index, legalstatus, originalsloc, relations from c10 (see rows 15, 86, 95, 112, 140 for details) Remove any repeated dsc elements from c10 (see row 61 for details) Apply camelCasing and name changes to sub-elements as necessary (see relevant rows for details on each sub-element separately)	If @level is used, make sure @levelEncoding has been added to control (should already have happened when transforming archDesc) If @level (in all c10 elements) is used with any value other than "otherlevel", use @levelEncoding with value "EASList" If @level is used with value "otherlevel" (with least one of the c10 elements in the same EAD XML file), use @levelEncoding with value "otherLevelEncoding" and create a conventionDeclaration sub-element to control with reference saying "Definition of other level encodings for archDesc and components" If @level is used with value "otherlevel" and @otherlevel exists in parallel, move the value of @otherlevel into @level Remove @encodinganalog and include a comment about the removal; if either archDesc or ead indicates a relatedencoding according to MARC21, add the MARC21 namespace and use the attribute @marc21:tag in c10 to include the value of @encodinganalog (xmlns:marc21="http://www.loc.gov/MARC21/slim" marc21:tag="...") If @localtype is used, apply camelCasing to change to @localType; create a localTypeDeclaration with @id in control and add the default content "List of local types for component description level" to its sub-element reference; then add @localTypeDeclarationReference to c10 pointing to the @id of said localTypeDeclaration
c11	c11	Rename sub-element did to identificationData If one of c/did/origination, c/did/repository, c/controlaccess/corpname (or famname, name, persname), c/index/indexentry/corpname (or famname, name, persname), c/index/indexentry/namegrp/corpname (or famname, name, persname) exist, create an agents sub-element to c If one of c/did/dao, c/did/daoset, c/altformavail, c/originalsloc exist, create a formsAvailable sub-element to c If one of c/controlaccess/function, c/index/indexentry/function, c/index/indexentry/namegrp/function exist, create a functions sub-element to c Remove altformavail, index, legalstatus, originalsloc, relations from c11 (see rows 15, 86, 95, 112, 140 for details) Remove any repeated dsc elements from c11 (see row 61 for details) Apply camelCasing and name changes to sub-elements as necessary (see relevant rows for details on each sub-element separately)	If @level is used, make sure @levelEncoding has been added to control (should already have happened when transforming archDesc) If @level (in all c11 elements) is used with any value other than "otherlevel", use @levelEncoding with value "EASList" If @level is used with value "otherlevel" (with least one of the c11 elements in the same EAD XML file), use @levelEncoding with value "otherLevelEncoding" and create a conventionDeclaration sub-element to control with reference saying "Definition of other level encodings for archDesc and components" If @level is used with value "otherlevel" and @otherlevel exists in parallel, move the value of @otherlevel into @level Remove @encodinganalog and include a comment about the removal; if either archDesc or ead indicates a relatedencoding according to MARC21, add the MARC21 namespace and use the attribute @marc21:tag in c11 to include the value of @encodinganalog (xmlns:marc21="http://www.loc.gov/MARC21/slim" marc21:tag="...") If @localtype is used, apply camelCasing to change to @localType; create a localTypeDeclaration with @id in control and add the default content "List of local types for component description level" to its sub-element reference; then add @localTypeDeclarationReference to c11 pointing to the @id of said localTypeDeclaration

Note: This sheet shows the changes that would need to be applied to each existing element and attribute in order to move it from EAD3 to EAD 4.0, while the sheets "Elements" and "Attributes" show the changes applied to each element and attribute in terms of its definition within the EAD schema. This sheet here therefore only refers to existing elements, but does not cover elements newly added to EAD 4.0. Furthermore, this sheet only includes the actual changes. Anything that stays the same (e.g. an element or attribute that isn't renamed or replaced or doesn't have camelCasing applied or when an element retains the same attributes in EAD 4.0 as it had in EAD3), won't be mentioned as such.		**General attributes' transformation for all elements that have existed in EAD3 already:** If @audience is used in one or more elements, add @audienceEncoding with the default value "EASList" to control Replace @altrender with @xhtml:style and add the XHTML namespace to the element using @altrender; include a comment to confirm that @altrender was indeed used to indicate alternative formatting Rename @lang to @languageOfElement; make sure @languageEncoding exists in control; if it doesn't, add the attribute with the default value "otherLanguageEncoding" Rename @script to @scriptOfElement; make sure @scriptEncoding exists in control; if it doesn't, add the attribute with the default value "otherScriptEncoding" **General transformation for attributes of elements that are removed:** If a parent element is removed from EAD 4.0, but there is a transformation route for its child elements and if parent and child elements share attributes, check whether the attributes exist with the child element already; if the child elements include the attributes already, remove the attributes of the parent element along with the element itself; if the child elements do not include the attributes already, move the attributes of the parent element to each of its child elements without them	
Element in EAD3	**Element in EAD 4.0**	**Transformation route for element and its content (draft)**	**Transformation route for attributes used with the element (draft)**
c12	c12	Rename sub-element did to identificationData If one of c/did/origination, c/did/repository, c/controlaccess/corpname (or famname, name, persname), c/index/indexentry/corpname (or famname, name, persname), c/index/indexentry/namegrp/corpname (or famname, name, persname) exist, create an agents sub-element to c If one of c/did/dao, c/did/daoset, c/altformavail, c/originalsloc exist, create a formsAvailable sub-element to c If one of c/controlaccess/function, c/index/indexentry/function, c/index/indexentry/namegrp/function exist, create a functions sub-element to c Remove altformavail, index, legalstatus, originalsloc, relations from c12 (see rows 15, 86, 95, 112, 140 for details) Remove any repeated dsc elements from c12 (see row 61 for details) Apply camelCasing and name changes to sub-elements as necessary (see relevant rows for details on each sub-element separately)	If @level is used, make sure @levelEncoding has been added to control (should already have happened when transforming archDesc) If @level (in all c12 elements) is used with any value other than "otherlevel", use @levelEncoding with value "EASList" If @level is used with value "otherlevel" (with least one of the c12 elements in the same EAD XML file), use @levelEncoding with value "otherLevelEncoding" and create a conventionDeclaration sub-element to control with reference saying "Definition of other level encodings for archDesc and components" If @level is used with value "otherlevel" and @otherlevel exists in parallel, move the value of @otherlevel into @level Remove @encodinganalog and include a comment about the removal; if either archDesc or ead indicates a relatedencoding according to MARC21, add the MARC21 namespace and use the attribute @marc21:tag in c12 to include the value of @encodinganalog (xmlns:marc21="http://www.loc.gov/MARC21/slim" marc21:tag="...") If @localtype is used, apply camelCasing to change to @localType; create a localTypeDeclaration with @id in control and add the default content "List of local types for component description level" to its sub-element reference; then add @localTypeDeclarationReference to c12 pointing to the @id of said localTypeDeclaration
chronitem	(formattingExtension)	Transform each chronitem into a separate xhtml:table/tr The use of chronitem's sub-elements will define whether there are two or three cells per row If all chronitem elements within a chronlist only use one of the date elements (datesingle, daterange or dateset) and the mandatory event element, there will be two cells per row If there is at least one chronitem element that also uses the optional geogname element, there will be three cells per row, leaving the third cell empty in those cases where there is no geogname If chronitem is used with dateset, the content of each date element respectively each combination of daterange/fromdate and daterange/todate elements will be separated by an xhtml:br element within the same xhtml:table/tr/td element If chronitem is used with chronitemset to enable several event or geogname elements, the contents of all event elements will be added in the same xhtml:table/tr/td element, separated by an xhtml:br element; similarly, the contents of all geogname elements will be added in the same xhtml:table/tr/td element, again separated by an xhtml:br element	Transform @audience="internal" into e.g. @xhtml:hidden (while removing the attribute if the value "external" is used) Transform @id into @xhtml:id Transform @altrender into @xhtml:style Transform @lang into @xhtml:lang Integrate @script with the value of @lang in @xhtml:lang by using ietf-bcp-47 values (or remove the attribute) Transform @localtype into @xhtml:title

Note: This sheet shows the changes that would need to be applied to each existing element and attribute in order to move it from EAD3 to EAD 4.0, while the sheets "Elements" and "Attributes" show the changes applied to each element and attribute in terms of its definition within the EAD schema. This sheet here therefore only refers to existing elements, but does not cover elements newly added to EAD 4.0.
Furthermore, this sheet only includes the actual changes. Anything that stays the same (e.g. an element or attribute that isn't renamed or replaced or doesn't have camelCasing applied or when an element retains the same attributes in EAD 4.0 as it had in EAD3), won't be mentioned as such.

General attributes' transformation for all elements that have existed in EAD3 already:
If @audience is used in one or more elements, add @audienceEncoding with the default value "EASList" to control
Replace @altrender with @xhtml:style and add the XHTML namespace to the element using @altrender; include a comment to confirm that @altrender was indeed used to indicate alternative formatting
Rename @lang to @languageOfElement; make sure @languageEncoding exists in control; if it doesn't, add the attribute with the default value "otherLanguageEncoding"
Rename @script to @scriptOfElement; make sure @scriptEncoding exists in control; if it doesn't, add the attribute with the default value "otherScriptEncoding"

General transformation for attributes of elements that are removed:
If a parent element is removed from EAD 4.0, but there is a transformation route for its child elements and if parent and child elements share attributes, check whether the attributes exist with the child element already; if the child elements include the attributes already, remove the attributes of the parent element along with the element itself; if the child elements do not include the attributes already, move the attributes of the parent element to each of its child elements without them

Element in EAD3	**Element in EAD 4.0**	**Transformation route for element and its content (draft)**	**Transformation route for attributes used with the element (draft)**
chronitemset	(formattingExtension)	Concatenate the contents of all event sub-elements of chronitemset in the same xhtml:table/tr/td element within the xhtml:table/tr element created for the parent chronitem element Separate the content of each event element by an xhtml:br element within that same xhtml:table/tr/td element Similarly, concatenate the contents of all geogname sub-elements of chronitemset in the same xhtml:table/tr/td element within the xhtml:table/tr element created for the parent chronitem element Separate the content of each geogname element by an xhtml:br element within that same xhtml:table/tr/td element	Transform @audience="internal" into e.g. @xhtml:hidden (while removing the attribute if the value "external" is used) Transform @id into @xhtml:id Transform @altrender into @xhtml:style Transform @lang into @xhtml:lang Integrate @script with the value of @lang in @xhtml:lang by using ietf-bcp-47 values (or remove the attribute)
chronlist	(formattingExtension)	Transform chronlist to xhtml:table Transform the sub-elements head into e.g. xhtml:h3, listhead/head01, listhead/head02, and listhead/head03 into one xhtml:table/tr/th each within the same xhtml:table/tr, each chronitem into a separate xhtml:tr within the same xhtml:table (see details in row 38)	Transform @audience="internal" into e.g. @xhtml:hidden (while removing the attribute if the value "external" is used) Transform @id into @xhtml:id Transform @altrender into @xhtml:style Transform @lang into @xhtml:lang Integrate @script with the value of @lang in @xhtml:lang by using ietf-bcp-47 values (or remove the attribute) Transform @localtype into @xhtml:title Remove @encodinganalog
citation	(reference)	Replace citation with reference Replace citation/abbr, /expan (if existing) with reference/referrringString Replace citation/emph, /foreign (if existing) with reference/span Remove citation/lb and include a comment about the removal Move citation/ptr, /ref (if existing) out of citation and replace them with a sibling reference **Note:** in EAD3, citation is only used as a sub-element of conventiondeclaration, localtypedeclaration, and rightsdeclaration, where there will be a one-to-one replacement with reference in EAD 4.0	Apply camelCasing to @linkRole and @linkTitle (if existing) If @actuate, @arcole, @show have been used, add the XLink namespace to reference and add @xlink:actuate, @xlink:arcrole, @xlink:show to include the attributes' values (xmlns:xlink="http://www.loc.gov/standards/xlink/xlink.xsd" xlink:show="..." xlink:actuate="..." xlink:arcrole="...") Remove @encodinganalog and include a comment about the removal; if any parent element in EAD3 indicates a relatedencoding according to MARC21, add the MARC21 namespace and use the attribute @marc21:tag in reference to include the value of @encodinganalog (xmlns:marc21="http://www.loc.gov/MARC21/slim" marc21:tag="...") Remove @lastdatetimeverified and include a comment about the removal
colspec	(formattingExtension)		Use relevant @xhtml attributes in the context of table

Note: This sheet shows the changes that would need to be applied to each existing element and attribute in order to move it from EAD3 to EAD 4.0, while the sheets "Elements" and "Attributes" show the changes applied to each element and attribute in terms of its definition within the EAD schema. This sheet here therefore only refers to existing elements, but does not cover elements newly added to EAD 4.0. Furthermore, this sheet only includes the actual changes. Anything that stays the same (e.g. an element or attribute that isn't renamed or replaced or doesn't have camelCasing applied or when an element retains the same attributes in EAD 4.0 as it had in EAD3), won't be mentioned as such.		**General attributes' transformation for all elements that have existed in EAD3 already:** If @audience is used in one or more elements, add @audienceEncoding with the default value "EASList" to control Replace @altrender with @xhtml:style and add the XHTML namespace to the element using @altrender; include a comment to confirm that @altrender was indeed used to indicate alternative formatting Rename @lang to @languageOfElement; make sure @languageEncoding exists in control; if it doesn't, add the attribute with the default value "otherLanguageEncoding" Rename @script to @scriptOfElement; make sure @scriptEncoding exists in control; if it doesn't, add the attribute with the default value "otherScriptEncoding" **General transformation for attributes of elements that are removed:** If a parent element is removed from EAD 4.0, but there is a transformation route for its child elements and if parent and child elements share attributes, check whether the attributes exist with the child element already; if the child elements include the attributes already, remove the attributes of the parent element along with the element itself; if the child elements do not include the attributes already, move the attributes of the parent element to each of its child elements without them	
Element in EAD3	**Element in EAD 4.0**	**Transformation route for element and its content (draft)**	**Transformation route for attributes used with the element (draft)**
container	container	Transform abbr, expan (if existing) into referringString (for details see rows 4, 71) Transform emph, foreign (if existing) into span (for details see rows 65, 76) Transform ptr, ref (if existing) into reference (for details see rows 128, 136) Remove lb Keep only the content of any sub-elements used in container and include a comment about any data potentially encoded in their attributes; values of attributes also available with referringString, span, and reference will of course be moved over to these three elements accordingly	Apply camelCasing to @containerId Remove @encodinganalog and include a comment about the removal; if any parent element in EAD3 indicates a relatedencoding according to MARC21, add the MARC21 namespace and use the attribute @marc21:tag in container to include the value of @encodinganalog (xmlns:marc21="http://www.loc.gov/MARC21/slim" marc21:tag="...") Remove @label and include comment about the removal; an option could be to suggest adding the XHTML namespace and using the attribute @title in abstract to include the value of @label (xmlns:xhtml="http://www.w3.org/1999/xhtml" xhtml:title="...")

Note: This sheet shows the changes that would need to be applied to each existing element and attribute in order to move it from EAD3 to EAD 4.0, while the sheets "Elements" and "Attributes" show the changes applied to each element and attribute in terms of its definition within the EAD schema. This sheet here therefore only refers to existing elements, but does not cover elements newly added to EAD 4.0.
Furthermore, this sheet only includes the actual changes. Anything that stays the same (e.g. an element or attribute that isn't renamed or replaced or doesn't have camelCasing applied or when an element retains the same attributes in EAD 4.0 as it had in EAD3), won't be mentioned as such.

General attributes' transformation for all elements that have existed in EAD3 already:
If @audience is used in one or more elements, add @audienceEncoding with the default value "EASList" to control
Replace @altrender with @xhtml:style and add the XHTML namespace to the element using @altrender; include a comment to confirm that @altrender was indeed used to indicate alternative formatting
Rename @lang to @languageOfElement; make sure @languageEncoding exists in control; if it doesn't, add the attribute with the default value "otherLanguageEncoding"
Rename @script to @scriptOfElement; make sure @scriptEncoding exists in control; if it doesn't, add the attribute with the default value "otherScriptEncoding"

General transformation for attributes of elements that are removed:
If a parent element is removed from EAD 4.0, but there is a transformation route for its child elements and if parent and child elements share attributes, check whether the attributes exist with the child element already; if the child elements include the attributes already, remove the attributes of the parent element along with the element itself; if the child elements do not include the attributes already, move the attributes of the parent element to each of its child elements without them

Element in EAD3	**Element in EAD 4.0**	**Transformation route for element and its content (draft)**	**Transformation route for attributes used with the element (draft)**
control	control	Adapt sequence of sub-elements to recordId, maintenanceAgency, maintenanceHistory, sources (if existing), followed by any other sub-elements (if existing) Remove filedesc and representation from control (turned into findAidDesc, see details in rows 73 and 142) Remove maintenancestatus and publicationstatus (if existing) from control (turn into attributes @maintenanceStatus and @publicationStatus, see details in column D) Apply camelCasing to sub-elements recordId, maintenanceAgeny, maintenanceHistory, as well as the following (if existing): conventionDeclaration, languageDeclaration, localTypeDeclaration, otherRecordId, rightsDeclaration Remove localcontrol from control and include a comment about the removal	Add @maintenanceStatus including the value of maintenancestatus@value and @maintenanceStatusEncoding with the value "EASList" If publicationstatus exists, add @publicationStatus including the value of publicationstatus@value and @publicationStatusEncoding with the value "EASList" Add @maintenanceEventTypeEncoding, @descriptionOfComponentsTypeEncoding (if dsc exists), @levelEncoding, @physDescStructuredTypeEncoding (if physdescstructured exists) Use value "EASList" with @maintenanceEventTypeEncoding Use value "EASList" also with @descriptionOfComponentsTypeEncoding, @levelEncoding, @physDescStructuredTypeEncoding as long as @dsctype, @level, and @physdescstructuredtype are not use with "otherdsctype", "otherlevel", or "otherphysdescstructuredtype" respectively; otherwise use them with the alternative "other...Encoding" value In case any of these three attributes is used with the "other...Encoding" value, create a conventionDeclaration sub-element to control with reference saying "Definition of other [name of attribute] encodings for [name of element(s)]" If addressline exists anywhere in the EAD XML file and is used with @localtype, add @addressLineEncoding with value "otherAddressLineTypeEncoding"; create a conventionDeclaration sub-element to control with reference saying "Definition of other addressLineType encodings for addressLine" If @audience exists anywhere in the EAD XML file, add @audienceEncoding with the value "EASList" If dao, daoset, or physdescstructured exist anywhere in the EAD XML file and are used with @coverage, add @coverageEncoding with the value "EASList" If unitdate or unitdatestructured exist anywhere in the EAD XML file and are used with @unitdatetype, add @unitDateTypeEncoding with the value "EASList" Rename @langencoding to @languageEncoding if it exists Apply camelCasing to @countryEncoding, @dateEncoding, @repositoryEncoding, @scriptEncoding if they exist Rename the attribute values "iso639-2b" to "iso639-2" and "otherlangencoding" to "otherLanguageEncoding" if used Apply camelCasing to the values "otherCountryEncoding", "otherDateEncoding", "otherRepositoryEncoding", "otherScriptEncoding" if used Remove @relatedencodig and @encodinganalog If either control itself or ead ndicates a relatedencoding according to MARC21, add the MARC21 namespace and use the attribute @marc21:tag in control to include the value of @encodinganalog (xmlns:marc21="http://www.loc.gov/MARC21/slim" marc21:tag="...")

Note: This sheet shows the changes that would need to be applied to each existing element and attribute in order to move it from EAD3 to EAD 4.0, while the sheets "Elements" and "Attributes" show the changes applied to each element and attribute in terms of its definition within the EAD schema. This sheet here therefore only refers to existing elements, but does not cover elements newly added to EAD 4.0.
Furthermore, this sheet only includes the actual changes. Anything that stays the same (e.g. an element or attribute that isn't renamed or replaced or doesn't have camelCasing applied or when an element retains the same attributes in EAD 4.0 as it had in EAD3), won't be mentioned as such.

General attributes' transformation for all elements that have existed in EAD3 already:
If @audience is used in one or more elements, add @audienceEncoding with the default value "EASList" to control
Replace @altrender with @xhtml:style and add the XHTML namespace to the element using @altrender; include a comment to confirm that @altrender was indeed used to indicate alternative formatting
Rename @lang to @languageOfElement; make sure @languageEncoding exists in control; if it doesn't, add the attribute with the default value "otherLanguageEncoding"
Rename @script to @scriptOfElement; make sure @scriptEncoding exists in control; if it doesn't, add the attribute with the default value "otherScriptEncoding"

General transformation for attributes of elements that are removed:
If a parent element is removed from EAD 4.0, but there is a transformation route for its child elements and if parent and child elements share attributes, check whether the attributes exist with the child element already; if the child elements include the attributes already, remove the attributes of the parent element along with the element itself; if the child elements do not include the attributes already, move the attributes of the parent element to each of its child elements without them

Element in EAD3	**Element in EAD 4.0**	**Transformation route for element and its content (draft)**	**Transformation route for attributes used with the element (draft)**
controlaccess	subjectHeadings	Rename to subjectHeadings If only used with p, keep as is If used with p and any other formatting elements (i.e. blockquote, chronlist, head, list, table), move all into formattingExtension while adapting to XHTML where necessary (see rows 24, 40, 82, 96, 117, 155 for details) Move controlaccess/subject/part into subjectHeadings/subject/term (see row 153 for details on subject) Turn corpname, famname, name, and persname into the entity element agent (see rows 48, 72, 105, 119 for details) Turn function into the entity element function (see row 78 for details) Turn geogname into the entity element place (see row 80 for details) Move title into publicationNote (or turn it into subjectHeadings/subject) (see row 160 for details) Turn gerneform and occupation into subjectHeadings/subject (see rows 79, 110 for details) If used with nesting, move any formatting sub-elements of the nested controlaccess into subjectHeadingst/formattingExtension to recreate the complete structure of the element alongside the sub-elements of the unnested (or parent) controlaccess; in that case, even if the unnested (or parent) controlaccess is only used with p, those p elements would also be used into subjectHeadings/formattingExtension; any access sub-elements of the nested controlaccess get the same transformation as for those of the unnested (or parent) controlaccess	If @localtype is used, apply camelCasing to change to @localType; create a localTypeDeclaration with @id in control and add the default content "List of local types for subject headings" to its sub-element reference; then add @localTypeDeclarationReference to subjectHeadings pointing to the @id of said localTypeDeclaration; if a localTypeDeclaration with this content in its sub-element reference already exists, point to this instead of creating another one) Remove @encodinganalog and include a comment about the removal; if any parent element in EAD3 indicates a relatedencoding according to MARC21, add the MARC21 namespace and use the attribute @marc21:tag in subjectHeadings to include the value of @encodinganalog (xmlns:marc21="http://www.loc.gov/MARC21/slim" marc21:tag="...")
controlnote	(findAidDesc)	Create a findAidDesc/formattingExtension for all controlnote elements and adapt their m.blocks content to XHTML encoding (see rows 24, 40, 96, 117, 155 for the m.blocks elements p, blockquote, chronlist, list, table for details)	Remove @encodinganalog and include a comment about the removal; if any parent element indicates a relatedencoding according to MARC21, add the MARC21 namespace and use the attribute @marc21:tag in formattingExtentions to include the value of @encodinganalog (xmlns:marc21="http://www.loc.gov/MARC21/slim" marc21:tag="...") Remove @localtype and include a comment about the removal As formattingExtension itself doesn't include @languageOfElement and @scriptOfElement, remove @script; for @lang, check wether controlnote or its sub-elements include @lang; if they don't, create an @xml:lang for each new element in the XHTML namespace that uses the value of controlnote@lang
conventiondeclaration	conventionDeclaration	Apply camelCasing Rename sub-elements abbr to shortCode and citation to reference Apply camelCasing to sub-element descriptiveNote Make sure sub-element reference comes first, followed by shortCode (if existing) and descriptiveNote (if existing)	Remove @encodinganalog and include a comment about the removal; if control or ead indicates a relatedencoding according to MARC21, add the MARC21 namespace and use the attribute @marc21:tag in conventionDeclaration to include the value of @encodinganalog (xmlns:marc21="http://www.loc.gov/MARC21/slim" marc21:tag="...") Remove @localtype and include a comment about the removal

Note: This sheet shows the changes that would need to be applied to each existing element and attribute in order to move it from EAD3 to EAD 4.0, while the sheets "Elements" and "Attributes" show the changes applied to each element and attribute in terms of its definition within the EAD schema. This sheet here therefore only refers to existing elements, but does not cover elements newly added to EAD 4.0. Furthermore, this sheet only includes the actual changes. Anything that stays the same (e.g. an element or attribute that isn't renamed or replaced or doesn't have camelCasing applied or when an element retains the same attributes in EAD 4.0 as it had in EAD3), won't be mentioned as such.		**General attributes' transformation for all elements that have existed in EAD3 already:** If @audience is used in one or more elements, add @audienceEncoding with the default value "EASList" to control Replace @altrender with @xhtml:style and add the XHTML namespace to the element using @altrender; include a comment to confirm that @altrender was indeed used to indicate alternative formatting Rename @lang to @languageOfElement; make sure @languageEncoding exists in control; if it doesn't, add the attribute with the default value "otherLanguageEncoding" Rename @script to @scriptOfElement; make sure @scriptEncoding exists in control; if it doesn't, add the attribute with the default value "otherScriptEncoding" **General transformation for attributes of elements that are removed:** If a parent element is removed from EAD 4.0, but there is a transformation route for its child elements and if parent and child elements share attributes, check whether the attributes exist with the child element already; if the child elements include the attributes already, remove the attributes of the parent element along with the element itself; if the child elements do not include the attributes already, move the attributes of the parent element to each of its child elements without them	
Element in EAD3	**Element in EAD 4.0**	**Transformation route for element and its content (draft)**	**Transformation route for attributes used with the element (draft)**
corpname	(agent) (referringString)	When corpname is used as sub-element of origination, repository, controlaccess, indexentry, or namegrp, transform it to agent within a new agents plural element added directly to archDesc respectively the relevant numbered or unnumbered c element (if agents already exists on this level of description, add the new agent to it after any already existing agent elements); in the case of corpname being used as a sub-element of origination or repository, that agent element will be created in the transformation of origination respectively repository Move the content of the part sub-element(s) of corpname into agentName; if corpname includes several part elements, concatenate these into the same agentName element and include a comment about this concatenation Create a parallel agentType sub-element and add the default content "Corporate Body" Create a parallel agentRole sub-element and add the value of the attribute @relator (if that was used with corpname); otherwise add the default content "Creator" (if corpname was a sub-element of origination) or "Repository" (if corpname was as sub-element of repository); do NOT add agentRole if corpname was a sub-element of controlaccess, indexentry, or namegrp NOR was used with @relator, but include a comment about agentRole being available for further definition When corpname is used as a mixed content element in ref, archref, bibref, abstract, unittitle, physfacet, or p, transform it to referringString in reference, abstract, unitTitle, physFacet and p respectively Move the content of the part sub-element(s) of corpname into referringString; if corpname includes several part elements, concatenate these into the same referringString element and include a comment about this concatenation When corpname is used as a mixed content element in event, item, or entry, find appropriate XHTML encoding to capture its information, e.g. the XHTML element <span> within the XHTML elements <li>, <td>, or <dt> together with the XHTML @title attribute including the default value "Name of a corporate body"	Move the value of corpname@identifier to agent@valueURI or referringString@valueURI Move the value of corpname@source to agent@vocabularySource or referringString@vocabularySource Move the value of corpname@rules to a newly created conventionDeclaration with @id in control and add its value to the conventionDeclaration sub-element reference; then add @conventionDeclarationReference to agent or referringString pointing to the @id of said conventionDeclaration; if a conventionDeclaration with this content in its sub-element reference already exists, point to this instead of creating another one Remove @normal and include a comment about its removal If @relator is used in the context of origination, repository, controlaccess, indexentry, or namegrp, move its value into an agentRole element within the newly created agent element; if corpname@relator is used in any other context, remove the attribute and include a comment about its removal If corpname@localtype is used, apply camelCasing to change to agent@localType or referringString@localType; create a localTypeDeclaration with @id in control and add the default content "List of local types for agents" (in the case of transforming corpname to agent) or "List of local types for corporate names" (in the case of transforming to referringString) to its sub-element reference; then add @localTypeDeclarationReference to agent or referringString pointing to the @id of said localTypeDeclaration; if a localTypeDeclaration with this content in its sub-element reference already exists, point to this instead of creating another one) Remove @encodinganalog and include a comment about the removal; if any parent element in EAD3 indicates a relatedencoding according to MARC21, add the MARC21 namespace and use the attribute @marc21:tag in agent or referringString to include the value of @encodinganalog (xmlns:marc21="http://www.loc.gov/MARC21/slim" marc21:tag="...")

Note: This sheet shows the changes that would need to be applied to each existing element and attribute in order to move it from EAD3 to EAD 4.0, while the sheets "Elements" and "Attributes" show the changes applied to each element and attribute in terms of its definition within the EAD schema. This sheet here therefore only refers to existing elements, but does not cover elements newly added to EAD 4.0. Furthermore, this sheet only includes the actual changes. Anything that stays the same (e.g. an element or attribute that isn't renamed or replaced or doesn't have camelCasing applied or when an element retains the same attributes in EAD 4.0 as it had in EAD3), won't be mentioned as such.		**General attributes' transformation for all elements that have existed in EAD3 already:** If @audience is used in one or more elements, add @audienceEncoding with the default value "EASList" to control Replace @altrender with @xhtml:style and add the XHTML namespace to the element using @altrender; include a comment to confirm that @altrender was indeed used to indicate alternative formatting Rename @lang to @languageOfElement; make sure @languageEncoding exists in control; if it doesn't, add the attribute with the default value "otherLanguageEncoding" Rename @script to @scriptOfElement; make sure @scriptEncoding exists in control; if it doesn't, add the attribute with the default value "otherScriptEncoding" **General transformation for attributes of elements that are removed:** If a parent element is removed from EAD 4.0, but there is a transformation route for its child elements and if parent and child elements share attributes, check whether the attributes exist with the child element already; if the child elements include the attributes already, remove the attributes of the parent element along with the element itself; if the child elements do not include the attributes already, move the attributes of the parent element to each of its child elements without them	
Element in EAD3	**Element in EAD 4.0**	**Transformation route for element and its content (draft)**	**Transformation route for attributes used with the element (draft)**
custodhist	custodHist	Apply camelCasing If only used with p, keep as is If used with p and any other formatting elements (i.e. blockquote, chronlist, head, list, table), move all into formattingExtension while adapting to XHTML where necessary (see rows 24, 40, 82, 96, 117, 155 for details) If used with nesting, move the sub-elements of the nested custodhist into custodHist/formattingExtension to recreate the complete structure of the element alongside the sub-elements of the unnested (or parent) custodhist; in that case, even if the unnested (or parent) custodhist is only used with p, those p elements would also be used into custodHist/formattingExtension	If @localtype is used, apply camelCasing to change to @localType; create a localTypeDeclaration with @id in control and add the default content "List of local types for custodial history" to its sub-element reference; then add @localTypeDeclarationReference to custodHist pointing to the @id of said localTypeDeclaration; if a localTypeDeclaration with this content in its sub-element reference already exists, point to this instead of creating another one) Remove @encodinganalog and include a comment about the removal; if any parent element in EAD3 indicates a relatedencoding according to MARC21, add the MARC21 namespace and use the attribute @marc21:tag in custodHist to include the value of @encodinganalog (xmlns:marc21="http://www.loc.gov/MARC21/slim" marc21:tag="...")

Note: This sheet shows the changes that would need to be applied to each existing element and attribute in order to move it from EAD3 to EAD 4.0, while the sheets "Elements" and "Attributes" show the changes applied to each element and attribute in terms of its definition within the EAD schema. This sheet here therefore only refers to existing elements, but does not cover elements newly added to EAD 4.0.
Furthermore, this sheet only includes the actual changes. Anything that stays the same (e.g. an element or attribute that isn't renamed or replaced or doesn't have camelCasing applied or when an element retains the same attributes in EAD 4.0 as it had in EAD3), won't be mentioned as such.

General attributes' transformation for all elements that have existed in EAD3 already:
If @audience is used in one or more elements, add @audienceEncoding with the default value "EASList" to control
Replace @altrender with @xhtml:style and add the XHTML namespace to the element using @altrender; include a comment to confirm that @altrender was indeed used to indicate alternative formatting
Rename @lang to @languageOfElement; make sure @languageEncoding exists in control; if it doesn't, add the attribute with the default value "otherLanguageEncoding"
Rename @script to @scriptOfElement; make sure @scriptEncoding exists in control; if it doesn't, add the attribute with the default value "otherScriptEncoding"

General transformation for attributes of elements that are removed:
If a parent element is removed from EAD 4.0, but there is a transformation route for its child elements and if parent and child elements share attributes, check whether the attributes exist with the child element already; if the child elements include the attributes already, remove the attributes of the parent element along with the element itself; if the child elements do not include the attributes already, move the attributes of the parent element to each of its child elements without them

Element in EAD3	**Element in EAD 4.0**	**Transformation route for element and its content (draft)**	**Transformation route for attributes used with the element (draft)**
dao	(formAvailable)	Transform dao into formAvailable within a new form**s**Available plural element added directly to archDesc respectively the relevant numbered or unnumbered c element (if form**s**Available already exists on this level of description, add the new formAvailable to it after any already existing formAvailable elements) When transforming several dao from the same level of description, keep the formAvailable elements in the same sequence as the transformed dao elements Move any p elements of the sub-element descriptivenote (if existing) directly into formAvailable If these p elements used any attributes, move the values of @id and @audience to the respective attributes of the new p elements, move the values of @lang and @script to @languageOfElement and @scriptOfElement in the new p elements, and remove @altrender (include a comment about any data potentially encoded in this attribute) If descriptivenote used any attributes and ***only if dao itself did NOT use any of these***, move the values of @id and @audience to the respective attributes of the new formAvailable element, move the values of @lang and @script to @languageOfElement and @scriptOfElement in the new formAvailable element, and remove @altrender and @encodinganalog (include a comment about any data potentially encoded in these attributes); if dao itself used any of these attributes already, simply remove the attributes of descriptivenote and include an according comment about any data potentially encoded there)	Move the values of @id and @audience from dao into the same attributes in formAvailable Move the values of @lang and @script from dao into the attributes @languageOfElement and @scripOfElement in formAvailable Remove the attribute @altrender and include a comment about the data potentially available encoded there) Remove @encodinganalog and include a comment about the removal; if any parent element in EAD3 indicates a relatedencoding according to MARC21, add the MARC21 namespace and use the attribute @marc21:tag in formAvailable to include the value of @encodinganalog (xmlns:marc21="http://www.loc.gov/MARC21/slim" marc21:tag="...") Move dao@localtype to formAvailable@localType; create a localTypeDeclaration with @id in control and add the default content "List of local types for digital objects" to its sub-element reference; then add @localTypeDeclarationReference to formAvailable pointing to the @id of said localTypeDeclaration; if a localTypeDeclaration with this content in its sub-element reference already exists, point to this instead of creating another one Remove @label and include comment about the removal; an option could be to suggest adding the XHTML namespace and using the attribute @title in abstract to include the value of @label (xmlns:xhtml="http://www.w3.org/1999/xhtml" xhtml:title="...") Move the value of @href from dao to @valueURI in formAvailable Create relations as a sub-element of formAvailable with its own required sub-element relation/targetEntity/part Move the value of the attribute @linktitle from dao (if existing) to this part element (if @linktitle does not exist, either repeat the value of @identifier from dao (if existing) as content of part or the value of @href from dao; if none of these attributes exist, add the default content "Digital object" Move the value of the attribute @linkrole from dao (if existing) to a targetRole element created next to the above targetEntity element If @actuate, @arcole, @show have been used, add the XLink namespace to formAvailable and add @xlink:actuate, @xlink:arcrole, @xlink:show to include the attributes' values (xmlns:xlink="http://www.loc.gov/standards/xlink/xlink.xsd" xlink:show="..." xlink:actuate="..." xlink:arcrole="...") Move the value of the attribute @identifier from dao (if existing) to @valueURI of the targetEntity element create above Remove the attributes @xpointer and @entityref and include a comment about any data potentially encoded in these attributes) If dao used the attribute @daotype with the value "borndigital", add the sub-element targetType next to the targetEntity element created above and use "born digital" as its content; if dao used the attribute @daotype with one of the values "derived", "unknown" or "otherdaotype", add the relationType next to that targetEntity element instead and use "derived" or "unknown" as its content; in the case of @daotype="otherdaotype", use the value of @otherdaotype as content of this relationType element Move the value of the attribute @coverage from dao to the same attribute in formAvailable (add @coverageEncoding with the value "EASList" to control if not existing already)

Note: This sheet shows the changes that would need to be applied to each existing element and attribute in order to move it from EAD3 to EAD 4.0, while the sheets "Elements" and "Attributes" show the changes applied to each element and attribute in terms of its definition within the EAD schema. This sheet here therefore only refers to existing elements, but does not cover elements newly added to EAD 4.0. Furthermore, this sheet only includes the actual changes. Anything that stays the same (e.g. an element or attribute that isn't renamed or replaced or doesn't have camelCasing applied or when an element retains the same attributes in EAD 4.0 as it had in EAD3), won't be mentioned as such.		**General attributes' transformation for all elements that have existed in EAD3 already:** If @audience is used in one or more elements, add @audienceEncoding with the default value "EASList" to control Replace @altrender with @xhtml:style and add the XHTML namespace to the element using @altrender; include a comment to confirm that @altrender was indeed used to indicate alternative formatting Rename @lang to @languageOfElement; make sure @languageEncoding exists in control; if it doesn't, add the attribute with the default value "otherLanguageEncoding" Rename @script to @scriptOfElement; make sure @scriptEncoding exists in control; if it doesn't, add the attribute with the default value "otherScriptEncoding" **General transformation for attributes of elements that are removed:** If a parent element is removed from EAD 4.0, but there is a transformation route for its child elements and if parent and child elements share attributes, check whether the attributes exist with the child element already; if the child elements include the attributes already, remove the attributes of the parent element along with the element itself; if the child elements do not include the attributes already, move the attributes of the parent element to each of its child elements without them	
Element in EAD3	**Element in EAD 4.0**	**Transformation route for element and its content (draft)**	**Transformation route for attributes used with the element (draft)**
daoset	(formsAvailable)	Transform daoset into formsAvailable added directly to archDesc respectively the relevant numbered or unnumbered c element Transform each of the dao elements included in daoset into a formAvailable element within this new form**s**Available element (see row 50 for details) If form**s**Available already exists on this level of description, only do the transformation from dao within daoset to formAvailable and add the new formAvailable elements after any already existing formAvailable elements within that form**s**Available plural element Keep the formAvailable elements in the same sequence as the transformed dao elements Move descriptivenote (if existing) to form**s**Available and apply camelCasing (see row 57 for details)	Move the values of @id and @audience from daoset into the same attributes in formsAvailable Move the values of @lang and @script from daoset into the attributes @languageOfElement and @scripOfElement in formsAvailable Remove the attribute @altrender and include a comment about the data potentially available encoded there) Move daoset@localtype to formsAvailable@localType; create a localTypeDeclaration with @id in control and add the default content "List of local types for digital objects" to its sub-element reference; then add @localTypeDeclarationReference to formsAvailable pointing to the @id of said localTypeDeclaration; if a localTypeDeclaration with this content in its sub-element reference already exists, point to this instead of creating another one Remove @encodinganalog and include a comment about the removal; if any parent element in EAD3 indicates a relatedencoding according to MARC21, add the MARC21 namespace and use the attribute @marc21:tag in formsAvailable to include the value of @encodinganalog (xmlns:marc21="http://www.loc.gov/MARC21/slim" marc21:tag="...") Remove @label and include comment about the removal; an option could be to suggest adding the XHTML namespace and using the attribute @title in abstract to include the value of @label (xmlns:xhtml="http://www.w3.org/1999/xhtml" xhtml:title="...") Move the value of the attribute @coverage from dao to the same attribute in formAvailable (add @coverageEncoding with the value "EASList" to control if not existing already) *Remove @base and add the base URI/URL to the values of each @valueURI attribute in either formAvailable or formAvailable/relations/relation/targetEntity created from the dao elements within daoset (include a comment about this change and about confirming that these new absolute URIs/URLs are functionable; see #108)*

Note: This sheet shows the changes that would need to be applied to each existing element and attribute in order to move it from EAD3 to EAD 4.0, while the sheets "Elements" and "Attributes" show the changes applied to each element and attribute in terms of its definition within the EAD schema. This sheet here therefore only refers to existing elements, but does not cover elements newly added to EAD 4.0.
Furthermore, this sheet only includes the actual changes. Anything that stays the same (e.g. an element or attribute that isn't renamed or replaced or doesn't have camelCasing applied or when an element retains the same attributes in EAD 4.0 as it had in EAD3), won't be mentioned as such.

General attributes' transformation for all elements that have existed in EAD3 already:
If @audience is used in one or more elements, add @audienceEncoding with the default value "EASList" to control
Replace @altrender with @xhtml:style and add the XHTML namespace to the element using @altrender; include a comment to confirm that @altrender was indeed used to indicate alternative formatting
Rename @lang to @languageOfElement; make sure @languageEncoding exists in control; if it doesn't, add the attribute with the default value "otherLanguageEncoding"
Rename @script to @scriptOfElement; make sure @scriptEncoding exists in control; if it doesn't, add the attribute with the default value "otherScriptEncoding"

General transformation for attributes of elements that are removed:
If a parent element is removed from EAD 4.0, but there is a transformation route for its child elements and if parent and child elements share attributes, check whether the attributes exist with the child element already; if the child elements include the attributes already, remove the attributes of the parent element along with the element itself; if the child elements do not include the attributes already, move the attributes of the parent element to each of its child elements without them

Element in EAD3	**Element in EAD 4.0**	**Transformation route for element and its content (draft)**	**Transformation route for attributes used with the element (draft)**
date	date (referringString)	Keep only the content of any sub-elements used in date and include a comment about any data potentially encoded in their attributes If used as publicationstmt/date transform to findAidDesc/date If used as mixed content, transform to referringString where XHTML isn't applied and to xhtml:time where XHTML is applied	Move value of publicationstmt/date@normal to findAidDesc/date@standardDate Remove @encodinganalog and include a comment about the removal; if any parent element in EAD3 indicates a relatedencoding according to MARC21, add the MARC21 namespace and use the attribute @marc21:tag in date or referringString to include the value of @encodinganalog (xmlns:marc21="http://www.loc.gov/MARC21/slim" marc21:tag="...") Apply camelCasing to @localType; create a localTypeDeclaration with @id in control and add the default content "List of local types for single dates" to its sub-element reference; then add @localTypeDeclarationReference to date or referringString pointing to the @id of said localTypeDeclaration; if a localTypeDeclaration with this content in its sub-element reference already exists, point to this instead of creating another one In case of a transformation to xhtml:time, move the value of date@normal into xhtml:time@datetime
daterange	dateRange	Apply camelCasing to dateRange and its sub-elements fromDate and toDate If daterange did not include any sub-elements, add an empty fromDate to dateRange If daterange included fromdate and todate, make sure that fromDate comes before toDate within dateRange	Apply camelCasing to @localType; create a localTypeDeclaration with @id in control and add the default content "List of local types for date ranges" to its sub-element reference; then add @localTypeDeclarationReference to dateRange pointing to the @id of said localTypeDeclaration; if a localTypeDeclaration with this content in its sub-element reference already exists, point to this instead of creating another one
dateset	dateSet	Apply camelCasing to dateSet and its sub-element dateRange Replace datesingle with date as a sub-element of dateSet (see row 55 for details)	Apply camelCasing to @localType; create a localTypeDeclaration with @id in control and add the default content "List of local types for date sets" to its sub-element reference; then add @localTypeDeclarationReference to dateSet pointing to the @id of said localTypeDeclaration; if a localTypeDeclaration with this content in its sub-element reference already exists, point to this instead of creating another one
datesingle	(date)	Replace unitdatestructured/datesingle, dateset/datesingle, and relation/datesingle with unitDateStructured/date, dateSet/date, and relation/date For the transformation of datesingle as used in chronitem see row 38	Apply camelCasing to @standardDate, @notAfter, @notBefore when transforming datesingle to date Apply camelCasing to @localType when transforming datesingle to date; create a localTypeDeclaration with @id in control and add the default content "List of local types for single dates" to its sub-element reference; then add @localTypeDeclarationReference to date pointing to the @id of said localTypeDeclaration; if a localTypeDeclaration with this content in its sub-element reference already exists, point to this instead of creating another one

Note: This sheet shows the changes that would need to be applied to each existing element and attribute in order to move it from EAD3 to EAD 4.0, while the sheets "Elements" and "Attributes" show the changes applied to each element and attribute in terms of its definition within the EAD schema. This sheet here therefore only refers to existing elements, but does not cover elements newly added to EAD 4.0.
Furthermore, this sheet only includes the actual changes. Anything that stays the same (e.g. an element or attribute that isn't renamed or replaced or doesn't have camelCasing applied or when an element retains the same attributes in EAD 4.0 as it had in EAD3), won't be mentioned as such.

General attributes' transformation for all elements that have existed in EAD3 already:
If @audience is used in one or more elements, add @audienceEncoding with the default value "EASList" to control
Replace @altrender with @xhtml:style and add the XHTML namespace to the element using @altrender; include a comment to confirm that @altrender was indeed used to indicate alternative formatting
Rename @lang to @languageOfElement; make sure @languageEncoding exists in control; if it doesn't, add the attribute with the default value "otherLanguageEncoding"
Rename @script to @scriptOfElement; make sure @scriptEncoding exists in control; if it doesn't, add the attribute with the default value "otherScriptEncoding"

General transformation for attributes of elements that are removed:
If a parent element is removed from EAD 4.0, but there is a transformation route for its child elements and if parent and child elements share attributes, check whether the attributes exist with the child element already; if the child elements include the attributes already, remove the attributes of the parent element along with the element itself; if the child elements do not include the attributes already, move the attributes of the parent element to each of its child elements without them

Element in EAD3	**Element in EAD 4.0**	**Transformation route for element and its content (draft)**	**Transformation route for attributes used with the element (draft)**
defitem	(formattingExtension)	If used within a list with @listtype="deflist" with**out** listhead, transform each list/defitem/label into an xhtml:dt and and each list/defitem/item into an xhtml:dd within that same xhtml:dl created for the parent list If used within a list with @listtype="deflist" **with** listhead, transform each list/defitem/label into the first xhtml:table/tr/td and and each list/defitem/item into the second xhtml:table/tr/td within that same xhtml:table/tr within the xhtml:table created for the parent list	Transform @audience="internal" into e.g. @xhtml:hidden (while removing the attribute if the value "external" is used) Transform @id into @xhtml:id Transform @altrender into @xhtml:style Transform @lang into @xhtml:lang Integrate @script with the value of @lang in @xhtml:lang by using ietf-bcp-47 values (or remove the attribute)
descriptivenote	descriptiveNote	Apply camelCasing	Remove @encodinganalog and include a comment about the removal; if any parent element in EAD3 indicates a relatedencoding according to MARC21, add the MARC21 namespace and use the attribute @marc21:tag in descriptiveNote to include the value of @encodinganalog (xmlns:marc21="http://www.loc.gov/MARC21/slim" marc21:tag="...")
did	identificationData	Rename from did to identificationData Move abstract (if existing) out identificationData and add to scopeContent (if there's any indication, e.g. via an @encodinganalog, that abstract belongs more appropriately to any other sibling element of scopeContent, move it there) Remove dao and daoset (if existing) from identificationData, transform them into formAvailable within formsAvailable Remove origination and repository (if existing) from identificationData, transform them into agent within agents Rename sub-elements didnote and langmaterial (if existing) to identificationDataNote and languageOfMaterial respectively Apply camelCasing to remaining sub-elements materialSpec, physDescSet, physDesc, physDescStructured, physLoc, unitDate, unitDateStructured, unitId, unitTitle (if existing)	Remove @encodinganalog and include a comment about the removal; if any parent element indicates a relatedencoding according to MARC21, add the MARC21 namespace and use the attribute @marc21:tag in identificationData to include the value of @encodinganalog (xmlns:marc21="http://www.loc.gov/MARC21/slim" marc21:tag="...")
didnote	identificationDataNote	Rename from didnote to identificationDataNote Move the content of didnote into a new p sub-element to identificationDataNote Transform abbr, expan from didnote (if existing) into referringString in p (for details see rows 4, 71) Transform emph, foreign from didnote (if existing) into span in p (for details see row 65) Transform ptr, ref from didnote (if existing) into reference in p (for details see rows 128, 136) Remove lb Keep only the content of any sub-elements used in didnote and include a comment about any data potentially encoded in their attributes; values of attributes also available with referringString, span, and reference will of course be moved over to these three elements accordingly	Remove @localtype and include a comment about the removal Remove @encodinganalog and include a comment about the removal; if any parent element in EAD3 indicates a relatedencoding according to MARC21, add the MARC21 namespace and use the attribute @marc21:tag in identificationDataNote to include the value of @encodinganalog (xmlns:marc21="http://www.loc.gov/MARC21/slim" marc21:tag="...") Remove @label and include comment about the removal; an option could be to suggest adding the XHTML namespace and using the attribute @title in abstract to include the value of @label (xmlns:xhtml="http://www.w3.org/1999/xhtml" xhtml:title="...")

Note: This sheet shows the changes that would need to be applied to each existing element and attribute in order to move it from EAD3 to EAD 4.0, while the sheets "Elements" and "Attributes" show the changes applied to each element and attribute in terms of its definition within the EAD schema. This sheet here therefore only refers to existing elements, but does not cover elements newly added to EAD 4.0. Furthermore, this sheet only includes the actual changes. Anything that stays the same (e.g. an element or attribute that isn't renamed or replaced or doesn't have camelCasing applied or when an element retains the same attributes in EAD 4.0 as it had in EAD3), won't be mentioned as such.	**General attributes' transformation for all elements that have existed in EAD3 already:** If @audience is used in one or more elements, add @audienceEncoding with the default value "EASList" to control Replace @altrender with @xhtml:style and add the XHTML namespace to the element using @altrender; include a comment to confirm that @altrender was indeed used to indicate alternative formatting Rename @lang to @languageOfElement; make sure @languageEncoding exists in control; if it doesn't, add the attribute with the default value "otherLanguageEncoding" Rename @script to @scriptOfElement; make sure @scriptEncoding exists in control; if it doesn't, add the attribute with the default value "otherScriptEncoding" **General transformation for attributes of elements that are removed:** If a parent element is removed from EAD 4.0, but there is a transformation route for its child elements and if parent and child elements share attributes, check whether the attributes exist with the child element already; if the child elements include the attributes already, remove the attributes of the parent element along with the element itself; if the child elements do not include the attributes already, move the attributes of the parent element to each of its child elements without them

Element in EAD3	**Element in EAD 4.0**	**Transformation route for element and its content (draft)**	**Transformation route for attributes used with the element (draft)**
dimensions	dimensions	Transform abbr, expan (if existing) into referringString (for details see rows 4, 71) Transform emph, foreign (if existing) into span (for details see rows 65, 76) Transform ptr, ref (if existing) into reference (for details see rows 128, 136) Remove lb Keep only the content of any sub-elements used in dimensions and include a comment about any data potentially encoded in their attributes; values of attributes also available with referringString, span, and reference will of course be moved over to these three elements accordingly If nested dimensions is used, move the nested dimensions to being a sibling element instead of a sub-element of the unnested (or parent) dimensions and include a comment for this to be reviewed	If @localtype is used, apply camelCasing to change to @localType; create a localTypeDeclaration with @id in control and add the default content "List of local types for dimensions" to its sub-element reference; then add @localTypeDeclarationReference to dimensions pointing to the @id of said localTypeDeclaration; if a localTypeDeclaration with this content in its sub-element reference already exists, point to this instead of creating another one) Remove @encodinganalog and include a comment about the removal; if any parent element in EAD3 indicates a relatedencoding according to MARC21, add the MARC21 namespace and use the attribute @marc21:tag in dimentsions to include the value of @encodinganalog (xmlns:marc21="http://www.loc.gov/MARC21/slim" marc21:tag="...")
dsc	descriptionOfComponents	Rename to descriptionOfComponents If only used with p, keep as is If used with p and any other formatting elements (i.e. blockquote, chronlist, head, list, table, thead), move all into formattingExtension while adapting to XHTML where necessary (see rows 24, 40, 82, 96, 117, 155, 159 for details)	Rename attribute @dsctype to @descriptionOfComponentsType Remove attribute @otherdsctype and move its value into @descriptionOfComponentsType Add @descriptionOfComponentsTypeEncoding to control with the value "otherDescriptionOfComponentsTypeEncoding" in the case of @dsctype="otherdsctype"; for all other values in @dsctype add the value "EASList" to @descriptionOfComponentsTypeEncoding Remove @encodinganalog and include a comment about the removal; if any parent element in EAD3 indicates a relatedencoding according to MARC21, add the MARC21 namespace and use the attribute @marc21:tag in descriptionOfComponents to include the value of @encodinganalog (xmlns:marc21="http://www.loc.gov/MARC21/slim" marc21:tag="...")
ead	ead	Apply camelCasing to sub-element archDesc Add one findAidDesc sub-element (between control and archDesc) to capture the information held in control/filedesc Add another findAidDesc element for control/recordid@instanceurl (if existing) and one for each control/representation (if existing)	Remove @relatedencoding and include a comment about the removal; if ead indicates a relatedencoding according to MARC21, this will be captured by adding the MARC21 namespace and using the attribute @marc21:tag with the relevant values in each sub-element using an @encodinganalog in this sense
edition	(findAidDesc)	Move the content of each edition into a p in the XHTML namespace within the same findAidDesc/formattingExtension Transform abbr to xhtml:abbr Transform expan into the value of an attribute @title in xhtml:abbr Transform emph@render to xhtml:em@style Transform lb to xhtml:br (ensuring that this comes with a closing tag) Transform ptr and ref to xhtml:a Transform foreign to xhtml:span (e.g. using @xhtml:title to capture the name of the former EAD element)	Transform edition@localtype to xhtml:p@title Remove @encodinganalog and include a comment about the removal As formattingExtension itself doesn't include @languageOfElement and @scriptOfElement, remove @script; for @lang, check wether the edition sub-elements include @lang and move its value to an @xml:lang for each new p in the XHTML namespace

Note: This sheet shows the changes that would need to be applied to each existing element and attribute in order to move it from EAD3 to EAD 4.0, while the sheets "Elements" and "Attributes" show the changes applied to each element and attribute in terms of its definition within the EAD schema. This sheet here therefore only refers to existing elements, but does not cover elements newly added to EAD 4.0.
Furthermore, this sheet only includes the actual changes. Anything that stays the same (e.g. an element or attribute that isn't renamed or replaced or doesn't have camelCasing applied or when an element retains the same attributes in EAD 4.0 as it had in EAD3), won't be mentioned as such.

General attributes' transformation for all elements that have existed in EAD3 already:
If @audience is used in one or more elements, add @audienceEncoding with the default value "EASList" to control
Replace @altrender with @xhtml:style and add the XHTML namespace to the element using @altrender; include a comment to confirm that @altrender was indeed used to indicate alternative formatting
Rename @lang to @languageOfElement; make sure @languageEncoding exists in control; if it doesn't, add the attribute with the default value "otherLanguageEncoding"
Rename @script to @scriptOfElement; make sure @scriptEncoding exists in control; if it doesn't, add the attribute with the default value "otherScriptEncoding"

General transformation for attributes of elements that are removed:
If a parent element is removed from EAD 4.0, but there is a transformation route for its child elements and if parent and child elements share attributes, check whether the attributes exist with the child element already; if the child elements include the attributes already, remove the attributes of the parent element along with the element itself; if the child elements do not include the attributes already, move the attributes of the parent element to each of its child elements without them

Element in EAD3	**Element in EAD 4.0**	**Transformation route for element and its content (draft)**	**Transformation route for attributes used with the element (draft)**
editionstmt	(findAidDesc)	Remove editionstmt (for the transformation of its child elements edition and p see rows 63 and 117)	
emph	span	Transform to span where XHTML isn't applied; keep only the content of any sub-elements used in emph and include a comment about any data potentially encoded in their attributes Transform to xhtml:em where XHTML is applied	Replace @render with @style Transform emph@render to xhtml:em@style
entry	(formattingExtension)	If used in tbody/row, transform to xhtml:tr/td If used in thead/row, transform to xhtml:tr/th Transform abbr to xhtml:abbr Transform expan to the value of an attribute @title in xhtml:abbr Transform date to xhtml:time Transform quote to xhtml:q Transform emph@render to xhtml:em@style Transform lb to xhtml:br (ensuring that this comes with a closing tag) Transform ptr and ref to xhtml:a Transform foreign, persname, corpname, famname, geogname, name, occupation, subject, genreform, function, title, footnote, num to xhtml:span (e.g. using @xhtml:title to capture the name of the former EAD element)	Transform @audience="internal" into e.g. @xhtml:hidden (while removing the attribute if the value "external" is used) Transform @id into @xhtml:id Transform @altrender into @xhtml:style Transform @lang into @xhtml:lang Integrate @script with the value of @lang in @xhtml:lang by using ietf-bcp-47 values (or remove the attribute) Integrate @colname, @namest, @nameend, @morerows, @colsep, @rowsep, @align, @char, @charoff, @valign with @xhtml:style as applicable
event	(formattingExtension)	If used as a single sub-element in chronitem, transform event to xhtml:table/tr/td within the xhtml:table/tr for the parent chronitem If used with several other event elements within chronitemset, create an xhtml: table/tr/td element for the chronitemset and concatenate the contents of all event sub-elements within that same xhtml:table/tr/td; separate the content of each event element by an xhtml:br element in that case Transform abbr to xhtml:abbr Transform expan into the value of an attribute @title in xhtml:abbr Transform date to xhtml:time Transform quote to xhtml:q Transform emph@render to xhtml:em@style Transform lb to xhtml:br (ensuring that this comes with a closing tag) Transform ptr and ref to xhtml:a Transform foreign, persname, corpname, famname, geogname, name, occupation, subject, genreform, function, title, footnote, num to xhtml:span (e.g. using @xhtml:title to capture the name of the former EAD element)	Transform @audience="internal" into e.g. @xhtml:hidden (while removing the attribute if the value "external" is used) Transform @id into @xhtml:id Transform @altrender into @xhtml:style Transform @lang into @xhtml:lang Integrate @script with the value of @lang in @xhtml:lang by using ietf-bcp-47 values (or remove the attribute) Transform @localtype into @xhtml:title

Note: This sheet shows the changes that would need to be applied to each existing element and attribute in order to move it from EAD3 to EAD 4.0, while the sheets "Elements" and "Attributes" show the changes applied to each element and attribute in terms of its definition within the EAD schema. This sheet here therefore only refers to existing elements, but does not cover elements newly added to EAD 4.0.
Furthermore, this sheet only includes the actual changes. Anything that stays the same (e.g. an element or attribute that isn't renamed or replaced or doesn't have camelCasing applied or when an element retains the same attributes in EAD 4.0 as it had in EAD3), won't be mentioned as such.

General attributes' transformation for all elements that have existed in EAD3 already:
If @audience is used in one or more elements, add @audienceEncoding with the default value "EASList" to control
Replace @altrender with @xhtml:style and add the XHTML namespace to the element using @altrender; include a comment to confirm that @altrender was indeed used to indicate alternative formatting
Rename @lang to @languageOfElement; make sure @languageEncoding exists in control; if it doesn't, add the attribute with the default value "otherLanguageEncoding"
Rename @script to @scriptOfElement; make sure @scriptEncoding exists in control; if it doesn't, add the attribute with the default value "otherScriptEncoding"

General transformation for attributes of elements that are removed:
If a parent element is removed from EAD 4.0, but there is a transformation route for its child elements and if parent and child elements share attributes, check whether the attributes exist with the child element already; if the child elements include the attributes already, remove the attributes of the parent element along with the element itself; if the child elements do not include the attributes already, move the attributes of the parent element to each of its child elements without them

Element in EAD3	**Element in EAD 4.0**	**Transformation route for element and its content (draft)**	**Transformation route for attributes used with the element (draft)**
eventdatetime	eventDateTime	Apply camelCasing	Apply camelCasing to @standardDateTime Remove @encodinganalog and include a comment about the removal; if any parent element indicates a relatedencoding according to MARC21, add the MARC21 namespace and use the attribute @marc21:tag in eventDateTime to include the value of @encodinganalog (xmlns:marc21="http://www.loc.gov/MARC21/slim" marc21:tag="...")
eventdescription	eventDescription	Apply camelCasing	Remove @encodinganalog and include a comment about the removal; if any parent element indicates a relatedencoding according to MARC21, add the MARC21 namespace and use the attribute @marc21:tag in eventDescription to include the value of @encodinganalog (xmlns:marc21="http://www.loc.gov/MARC21/slim" marc21:tag="...") Remove @localtype and include a comment about the removal
eventtype	@maintenanceEventType	Remove eventtype from maintenanceEvent	Add @maintenanceEventType to maintenanceEvent, moving the value of eventtype@value into maintenanceEvent@maintenanceEventType Add @maintenanceEventTypeEncoding with the default value "EASList" to control
expan	(referringString)	Transform to referringString where XHTML isn't applied Transform to xhtml:abbr@title where XHTML is applied	Add the value of @abbr into referringString (in brackets after the content of expan itself, e.g. <expan abbr="ABC">Alphabet</expan> becomes <referringString>Alphabet (ABC)</referringString>) Add the value of @abbr as content of the xhtml:abbr that has the content of expan in its @xhtml:title, i.e. <expan abbr="ABC">Alphabet</expan> becomes <xhtml:abbr title="Alphabet">ABC</xhtml:abbr>

Note: This sheet shows the changes that would need to be applied to each existing element and attribute in order to move it from EAD3 to EAD 4.0, while the sheets "Elements" and "Attributes" show the changes applied to each element and attribute in terms of its definition within the EAD schema. This sheet here therefore only refers to existing elements, but does not cover elements newly added to EAD 4.0. Furthermore, this sheet only includes the actual changes. Anything that stays the same (e.g. an element or attribute that isn't renamed or replaced or doesn't have camelCasing applied or when an element retains the same attributes in EAD 4.0 as it had in EAD3), won't be mentioned as such.		**General attributes' transformation for all elements that have existed in EAD3 already:** If @audience is used in one or more elements, add @audienceEncoding with the default value "EASList" to control Replace @altrender with @xhtml:style and add the XHTML namespace to the element using @altrender; include a comment to confirm that @altrender was indeed used to indicate alternative formatting Rename @lang to @languageOfElement; make sure @languageEncoding exists in control; if it doesn't, add the attribute with the default value "otherLanguageEncoding" Rename @script to @scriptOfElement; make sure @scriptEncoding exists in control; if it doesn't, add the attribute with the default value "otherScriptEncoding" **General transformation for attributes of elements that are removed:** If a parent element is removed from EAD 4.0, but there is a transformation route for its child elements and if parent and child elements share attributes, check whether the attributes exist with the child element already; if the child elements include the attributes already, remove the attributes of the parent element along with the element itself; if the child elements do not include the attributes already, move the attributes of the parent element to each of its child elements without them	
Element in EAD3	**Element in EAD 4.0**	**Transformation route for element and its content (draft)**	**Transformation route for attributes used with the element (draft)**
famname	(agent) (referringString)	When famname is used as sub-element of origination, repository, controlaccess, indexentry, or namegrp, transform it to agent within a new agents plural element added directly to archDesc respectively the relevant numbered or unnumbered c element (if agents already exists on this level of description, add the new agent to it after any already existing agent elements); in the case of famname being used as a sub-element of origination or repository, that agent element will be created in the transformation of origination respectively repository Move the content of the part sub-element(s) of famname into agentName; if famname includes several part elements, concatenate these into the same agentName element and include a comment about this concatenation Create a parallel agentType sub-element and add the default content "Family" Create a parallel agentRole sub-element and add the value of the attribute @relator (if that was used with famname); otherwise add the default content "Creator" (if famname was a sub-element of origination) or "Repository" (if famname was as sub-element of repository); do NOT add agentRole if famname was a sub-element of controlaccess, indexentry, or namegrp NOR was used with @relator, but include a comment about agentRole being available for further definition When famname is used as a mixed content element in ref, archref, bibref, abstract, unittitle, physfacet, or p, transform it to referringString in reference, abstract, unitTitle, physFacet and p respectively Move the content of the part sub-element(s) of famname into referringString; if famname includes several part elements, concatenate these into the same referringString element and include a comment about this concatenation When famname is used as a mixed content element in event, item, or entry, find appropriate XHTML encoding to capture its information, e.g. the XHTML element <span> within the XHTML elements <li>, <td>, or <dt> together with the XHTML @title attribute including the default value "Name of a family"	Move the value of famname@identifier to agent@valueURI or referringString@valueURI Move the value of famname@source to agent@vocabularySource or referringString@vocabularySource Move the value of famname@rules to a newly created conventionDeclaration with @id in control and add its value to the conventionDeclaration sub-element reference; then add @conventionDeclarationReference to agent or referringString pointing to the @id of said conventionDeclaration; if a conventionDeclaration with this content in its sub-element reference already exists, point to this instead of creating another one Remove @normal and include a comment about its removal If @relator is used in the context of origination, repository, controlaccess, indexentry, or namegrp, move its value into an agentRole element within the newly created agent element; if famname@relator is used in any other context, remove the attribute and include a comment about its removal If famname@localtype is used, apply camelCasing to change to agent@localType or referringString@localType; create a localTypeDeclaration with @id in control and add the default content "List of local types for agents" (in the case of transforming famname to agent) or "List of local types for corporate names" (in the case of transforming to referringString) to its sub-element reference; then add @localTypeDeclarationReference to agent or referringString pointing to the @id of said localTypeDeclaration; if a localTypeDeclaration with this content in its sub-element reference already exists, point to this instead of creating another one) Remove @encodinganalog and include a comment about the removal; if any parent element in EAD3 indicates a relatedencoding according to MARC21, add the MARC21 namespace and use the attribute @marc21:tag in agent or referringString to include the value of @encodinganalog (xmlns:marc21="http://www.loc.gov/MARC21/slim" marc21:tag="...")
filedesc	findAidDesc	Move filedesc out of control and create findAidDesc as a sibling element between control and archDesc Have findAidDesc include at least one title sub-element to capture filedesc/titlestmt/titleproper Remove titlestmt, editionstmt, publicationstmt, seriesstmt, notestmt (see rows 64, 107, 131, 148, 162 for details on these elements' transformation routes)	Remove @encodinganalog and include a comment about the removal; if control or ead indicates a relatedencoding according to MARC21, add the MARC21 namespace and use the attribute @marc21:tag in findAidDesc to include the value of @encodinganalog (xmlns:marc21="http://www.loc.gov/MARC21/slim" marc21:tag="...")

Note: This sheet shows the changes that would need to be applied to each existing element and attribute in order to move it from EAD3 to EAD 4.0, while the sheets "Elements" and "Attributes" show the changes applied to each element and attribute in terms of its definition within the EAD schema. This sheet here therefore only refers to existing elements, but does not cover elements newly added to EAD 4.0.
Furthermore, this sheet only includes the actual changes. Anything that stays the same (e.g. an element or attribute that isn't renamed or replaced or doesn't have camelCasing applied or when an element retains the same attributes in EAD 4.0 as it had in EAD3), won't be mentioned as such.

General attributes' transformation for all elements that have existed in EAD3 already:
If @audience is used in one or more elements, add @audienceEncoding with the default value "EASList" to control
Replace @altrender with @xhtml:style and add the XHTML namespace to the element using @altrender; include a comment to confirm that @altrender was indeed used to indicate alternative formatting
Rename @lang to @languageOfElement; make sure @languageEncoding exists in control; if it doesn't, add the attribute with the default value "otherLanguageEncoding"
Rename @script to @scriptOfElement; make sure @scriptEncoding exists in control; if it doesn't, add the attribute with the default value "otherScriptEncoding"

General transformation for attributes of elements that are removed:
If a parent element is removed from EAD 4.0, but there is a transformation route for its child elements and if parent and child elements share attributes, check whether the attributes exist with the child element already; if the child elements include the attributes already, remove the attributes of the parent element along with the element itself; if the child elements do not include the attributes already, move the attributes of the parent element to each of its child elements without them

Element in EAD3	**Element in EAD 4.0**	**Transformation route for element and its content (draft)**	**Transformation route for attributes used with the element (draft)**
fileplan	filePlan	Apply camelCasing If only used with p, keep as is If used with p and any other formatting elements (i.e. blockquote, chronlist, head, list, table), move all into formattingExtension while adapting to XHTML where necessary (see rows 24, 40, 82, 96, 117, 155 for details) If used with nesting, move the sub-elements of the nested fileplan into filePlan/formattingExtension to recreate the complete structure of the element alongside the sub-elements of the unnested (or parent) fileplan; in that case, even if the unnested (or parent) fileplan is only used with p, those p elements would also be used into fileplan/formattingExtension	If @localtype is used, apply camelCasing to change to @localType; create a localTypeDeclaration with @id in control and add the default content "List of local types for file plan" to its sub-element reference; then add @localTypeDeclarationReference to biogHist pointing to the @id of said localTypeDeclaration; if a localTypeDeclaration with this content in its sub-element reference already exists, point to this instead of creating another one) Remove @encodinganalog and include a comment about the removal; if any parent element in EAD3 indicates a relatedencoding according to MARC21, add the MARC21 namespace and use the attribute @marc21:tag in filePlan to include the value of @encodinganalog (xmlns:marc21="http://www.loc.gov/MARC21/slim" marc21:tag="...")
footnote	(referringString)	Concatenate the contents of footnote into referringString If used in a context where transformation of potential parent elements of footnote to formattingExtension is applicable (i.e. within p, item, event, or entry), transform footnote to xhtml:blockquote and its sub-elements p to xhtml:p, list to @xhtml:ul, @xhtml:ol, or @xhtml:dl (depending whether or not @listtype is used and with which value), chronlist and table to @xhtml:table (see more details for these sub-elements in rows 40, 96, 117, 155) instead of concatenating their contents	Move footnote@localtype into referringString@localType If @actuate or @show have been used, add the XLink namespace to referringString and add @xlink:actuate and @xlink:show to include the attributes' values (xmlns:xlink="http://www.loc.gov/standards/xlink/xlink.xsd" xlink:show="..." xlink:actuate="...")
foreign	(span)	Transform to span where XHTML isn't applied; keep only the content of any sub-elements used in emph and include a comment about any data potentially encoded in their attributes Transform to xhtml:span where XHTML is applied (e.g. using @xhtml:title to capture the name of the former EAD element)	Replace @render with @style respectively @xhtml:style depending on the transformation context
fromdate	fromDate	Apply camelCasing	Apply camelCasing to @standardDate, @notAfter, @notBefore Apply camelCasing to @localType; create a localTypeDeclaration with @id in control and add the default content "List of local types for single dates" to its sub-element reference; then add @localTypeDeclarationReference to fromDate pointing to the @id of said localTypeDeclaration; if a localTypeDeclaration with this content in its sub-element reference already exists, point to this instead of creating another one

Note: This sheet shows the changes that would need to be applied to each existing element and attribute in order to move it from EAD3 to EAD 4.0, while the sheets "Elements" and "Attributes" show the changes applied to each element and attribute in terms of its definition within the EAD schema. This sheet here therefore only refers to existing elements, but does not cover elements newly added to EAD 4.0. Furthermore, this sheet only includes the actual changes. Anything that stays the same (e.g. an element or attribute that isn't renamed or replaced or doesn't have camelCasing applied or when an element retains the same attributes in EAD 4.0 as it had in EAD3), won't be mentioned as such.		**General attributes' transformation for all elements that have existed in EAD3 already:** If @audience is used in one or more elements, add @audienceEncoding with the default value "EASList" to control Replace @altrender with @xhtml:style and add the XHTML namespace to the element using @altrender; include a comment to confirm that @altrender was indeed used to indicate alternative formatting Rename @lang to @languageOfElement; make sure @languageEncoding exists in control; if it doesn't, add the attribute with the default value "otherLanguageEncoding" Rename @script to @scriptOfElement; make sure @scriptEncoding exists in control; if it doesn't, add the attribute with the default value "otherScriptEncoding" **General transformation for attributes of elements that are removed:** If a parent element is removed from EAD 4.0, but there is a transformation route for its child elements and if parent and child elements share attributes, check whether the attributes exist with the child element already; if the child elements include the attributes already, remove the attributes of the parent element along with the element itself; if the child elements do not include the attributes already, move the attributes of the parent element to each of its child elements without them	
Element in EAD3	**Element in EAD 4.0**	**Transformation route for element and its content (draft)**	**Transformation route for attributes used with the element (draft)**
function	function (referringString)	When function is used as sub-element of controlaccess, indexentry, or namegrp, transform it to function within a new function**s** plural element added directly to archDesc respectively the relevant numbered or unnumbered c element (if functions already exists on this level of description, add the new function to it after any already existing function elements) Move the content of the part sub-element(s) of function into term; if function includes several part elements, concatenate these into the same term element and include a comment about this concatenation Create a targetType sub-element and add the default content "Function" Create a targetRole sub-element and add the value of the attribute @relator (if that was used with function); otherwise include a comment about targetRole being available for further definition When function is used as a mixed content element in ref, archref, bibref, abstract, unittitle, physfacet, or p, transform it to referringString in reference, abstract, unitTitle, physFacet and p respectively Move the content of the part sub-element(s) of function into referringString; if function includes several part elements, concatenate these into the same referringString element and include a comment about this concatenation When function is used as a mixed content element in event, item, or entry, find appropriate XHTML encoding to capture its information, e.g. the XHTML element <span> within the XHTML elements <li>, <td>, or <dt> together with the XHTML @title attribute including the default value "Name of a function"	Move the value of function@identifier to function@valueURI or referringString@valueURI Move the value of function@source to function@vocabularySource or referringString@vocabularySource Move the value of function@rules to a newly created conventionDeclaration with @id in control and add its value to the conventionDeclaration sub-element reference; then add @conventionDeclarationReference to function or referringString pointing to the @id of said conventionDeclaration; if a conventionDeclaration with this content in its sub-element reference already exists, point to this instead of creating another one Remove @normal and include a comment about its removal If @relator is used in the context of controlaccess, indexentry, or namegrp, move its value into a targetRole element within the newly created function element; if function@relator is used in any other context, remove the attribute and include a comment about its removal If @localtype is used, apply camelCasing to change to function@localType or referringString@localType; create a localTypeDeclaration with @id in control and add the default content "List of local types for functions" to its sub-element reference; then add @localTypeDeclarationReference to function or referringString pointing to the @id of said localTypeDeclaration; if a localTypeDeclaration with this content in its sub-element reference already exists, point to this instead of creating another one) Remove @encodinganalog and include a comment about the removal; if any parent element in EAD3 indicates a relatedencoding according to MARC21, add the MARC21 namespace and use the attribute @marc21:tag in function or referringString to include the value of @encodinganalog (xmlns:marc21="http://www.loc.gov/MARC21/slim" marc21:tag="...")

Note: This sheet shows the changes that would need to be applied to each existing element and attribute in order to move it from EAD3 to EAD 4.0, while the sheets "Elements" and "Attributes" show the changes applied to each element and attribute in terms of its definition within the EAD schema. This sheet here therefore only refers to existing elements, but does not cover elements newly added to EAD 4.0. Furthermore, this sheet only includes the actual changes. Anything that stays the same (e.g. an element or attribute that isn't renamed or replaced or doesn't have camelCasing applied or when an element retains the same attributes in EAD 4.0 as it had in EAD3), won't be mentioned as such.		**General attributes' transformation for all elements that have existed in EAD3 already:** If @audience is used in one or more elements, add @audienceEncoding with the default value "EASList" to control Replace @altrender with @xhtml:style and add the XHTML namespace to the element using @altrender; include a comment to confirm that @altrender was indeed used to indicate alternative formatting Rename @lang to @languageOfElement; make sure @languageEncoding exists in control; if it doesn't, add the attribute with the default value "otherLanguageEncoding" Rename @script to @scriptOfElement; make sure @scriptEncoding exists in control; if it doesn't, add the attribute with the default value "otherScriptEncoding" **General transformation for attributes of elements that are removed:** If a parent element is removed from EAD 4.0, but there is a transformation route for its child elements and if parent and child elements share attributes, check whether the attributes exist with the child element already; if the child elements include the attributes already, remove the attributes of the parent element along with the element itself; if the child elements do not include the attributes already, move the attributes of the parent element to each of its child elements without them	
Element in EAD3	**Element in EAD 4.0**	**Transformation route for element and its content (draft)**	**Transformation route for attributes used with the element (draft)**
genreform	(subject) (referringString)	When genreform is used as sub-element of controlaccess, indexentry, or namegrp, transform it to subject within subjectHeadings (with integrates controlaccess and index) Move the content of the part sub-element(s) of genreform into subject/term; if genreform includes several part elements, concatenate these into the same term element and include a comment about this concatenation Create a targetType element parallel to subject/term and add the default content "genre or form" Create a targetRole element parallel to subject/term and add the value of the attribute @relator (if that was used with genreform); otherwise include a comment about targetRole being available for further definition When genreform is used as a mixed content element in ref, archref, bibref, abstract, unittitle, physfacet, or p, transform it to referringString in reference, abstract, unitTitle, physFacet and p respectively Move the content of the part sub-element(s) of genreform into referringString; if genreform includes several part elements, concatenate these into the same referringString element and include a comment about this concatenation When genreform is used as a mixed content element in event, item, or entry, find appropriate XHTML encoding to capture its information, e.g. the XHTML element <span> within the XHTML elements <li>, <td>, or <dt> together with the XHTML @title attribute including the default value "Name of a genre or form"	Move the value of genreform@identifier to subject@valueURI or referringString@valueURI Move the value of genreform@source to subject@vocabularySource or referringString@vocabularySource Move the value of genreform@rules to a newly created conventionDeclaration with @id in control and add its value to the conventionDeclaration sub-element reference; then add @conventionDeclarationReference to subject or referringString pointing to the @id of said conventionDeclaration; if a conventionDeclaration with this content in its sub-element reference already exists, point to this instead of creating another one Remove @normal and include a comment about its removal If @relator is used in the context of controlaccess, indexentry, or namegrp, move its value into a targetRole element within the newly created subject element; if genreform@relator is used in any other context, remove the attribute and include a comment about its removal If genreform@localtype is used, apply camelCasing to change to subject@localType or referringString@localType; create a localTypeDeclaration with @id in control and add the default content "List of local types for genre or form" to its sub-element reference; then add @localTypeDeclarationReference to subject or referringString pointing to the @id of said localTypeDeclaration; if a localTypeDeclaration with this content in its sub-element reference already exists, point to this instead of creating another one) Remove @encodinganalog and include a comment about the removal; if any parent element in EAD3 indicates a relatedencoding according to MARC21, add the MARC21 namespace and use the attribute @marc21:tag in subject or referringString to include the value of @encodinganalog (xmlns:marc21="http://www.loc.gov/MARC21/slim" marc21:tag="...")

Note: This sheet shows the changes that would need to be applied to each existing element and attribute in order to move it from EAD3 to EAD 4.0, while the sheets "Elements" and "Attributes" show the changes applied to each element and attribute in terms of its definition within the EAD schema. This sheet here therefore only refers to existing elements, but does not cover elements newly added to EAD 4.0. Furthermore, this sheet only includes the actual changes. Anything that stays the same (e.g. an element or attribute that isn't renamed or replaced or doesn't have camelCasing applied or when an element retains the same attributes in EAD 4.0 as it had in EAD3), won't be mentioned as such.		**General attributes' transformation for all elements that have existed in EAD3 already:** If @audience is used in one or more elements, add @audienceEncoding with the default value "EASList" to control Replace @altrender with @xhtml:style and add the XHTML namespace to the element using @altrender; include a comment to confirm that @altrender was indeed used to indicate alternative formatting Rename @lang to @languageOfElement; make sure @languageEncoding exists in control; if it doesn't, add the attribute with the default value "otherLanguageEncoding" Rename @script to @scriptOfElement; make sure @scriptEncoding exists in control; if it doesn't, add the attribute with the default value "otherScriptEncoding" **General transformation for attributes of elements that are removed:** If a parent element is removed from EAD 4.0, but there is a transformation route for its child elements and if parent and child elements share attributes, check whether the attributes exist with the child element already; if the child elements include the attributes already, remove the attributes of the parent element along with the element itself; if the child elements do not include the attributes already, move the attributes of the parent element to each of its child elements without them	
Element in EAD3	**Element in EAD 4.0**	**Transformation route for element and its content (draft)**	**Transformation route for attributes used with the element (draft)**
geogname	place (referringString)	When geogname is used as sub-element of relation, replace relation/geogname with relation/place Move the content of the part sub-element(s) of geogname into place/placeName; if geogname includes several part elements, concatenate these into the same placeName element and include a comment about this concatenation Move geogname/geographiccoordinates to place/geographicCoordinates and applying camelCasing in this context Transform geogname@relator (if used) into place/placeRole; otherwise include a comment about targetRole being available for further definition When geogname is used as sub-element of controlaccess, indexentry, or namegrp, transform it to place within a new place**s** plural element added directly to archDesc respectively the relevant numbered or unnumbered c element (if places already exists on this level of description, add the new place to it after any already existing place elements) Move the content of the part sub-element(s) of geogname into placeName; if geogname includes several part elements, concatenate these into the same placeName element and include a comment about this concatenation Create a parallel placeRole sub-element and add the value of the attribute @relator (if that was used with geogname); otherwise do NOT add placeRole, but include a comment about placeRole being available for further definition When geogname is used as a mixed content element in ref, archref, bibref, abstract, unittitle, physfacet, or p, transform it to referringString in reference, abstract, unitTitle, physFacet and p respectively Move the content of the part sub-element(s) of geogname into referringString; if geogname includes several part elements, concatenate these into the same referringString element and include a comment about this concatenation When geogname is used as sub-element of chronitem or chronitemset or when it is used as a mixed content element in event, item, or entry, find appropriate XHTML encoding to capture its information, e.g. the XHTML element <span> within the XHTML elements <li>, <td>, or <dt> together with the XHTML @title attribute including the default value "Name of a place"	Move the value of geogname@identifier to place@valueURI or referringString@valueURI Move the value of geogname@source to place@vocabularySource or referringString@vocabularySource Move the value of geogname@rules to a newly created conventionDeclaration with @id in control and add its value to the conventionDeclaration sub-element reference; then add @conventionDeclarationReference to place or referringString pointing to the @id of said conventionDeclaration; if a conventionDeclaration with this content in its sub-element reference already exists, point to this instead of creating another one Remove @normal and include a comment about its removal If @relator is used in the context of controlaccess, indexentry, or namegrp, move its value into a placeRole element within the newly created place element; if geogname@relator is used in any other context, remove the attribute and include a comment about its removal If geogname@localtype is used, apply camelCasing to change to place@localType or referringString@localType; create a localTypeDeclaration with @id in control and add the default content "List of local types for places" (in the case of transforming name to place) or "List of local types for place names" (in the case of transforming to referringString) to its sub-element reference; then add @localTypeDeclarationReference to place or referringString pointing to the @id of said localTypeDeclaration; if a localTypeDeclaration with this content in its sub-element reference already exists, point to this instead of creating another one) Remove @encodinganalog and include a comment about the removal; if any parent element in EAD3 indicates a relatedencoding according to MARC21, add the MARC21 namespace and use the attribute @marc21:tag in place or referringString to include the value of @encodinganalog (xmlns:marc21="http://www.loc.gov/MARC21/slim" marc21:tag="...")
geographiccoordinates	geographicCoordinates	Apply camelCasing	Apply camelCasing to the required attribute @coordinateSystem

Note: This sheet shows the changes that would need to be applied to each existing element and attribute in order to move it from EAD3 to EAD 4.0, while the sheets "Elements" and "Attributes" show the changes applied to each element and attribute in terms of its definition within the EAD schema. This sheet here therefore only refers to existing elements, but does not cover elements newly added to EAD 4.0.
Furthermore, this sheet only includes the actual changes. Anything that stays the same (e.g. an element or attribute that isn't renamed or replaced or doesn't have camelCasing applied or when an element retains the same attributes in EAD 4.0 as it had in EAD3), won't be mentioned as such.

General attributes' transformation for all elements that have existed in EAD3 already:
If @audience is used in one or more elements, add @audienceEncoding with the default value "EASList" to control
Replace @altrender with @xhtml:style and add the XHTML namespace to the element using @altrender; include a comment to confirm that @altrender was indeed used to indicate alternative formatting
Rename @lang to @languageOfElement; make sure @languageEncoding exists in control; if it doesn't, add the attribute with the default value "otherLanguageEncoding"
Rename @script to @scriptOfElement; make sure @scriptEncoding exists in control; if it doesn't, add the attribute with the default value "otherScriptEncoding"

General transformation for attributes of elements that are removed:
If a parent element is removed from EAD 4.0, but there is a transformation route for its child elements and if parent and child elements share attributes, check whether the attributes exist with the child element already; if the child elements include the attributes already, remove the attributes of the parent element along with the element itself; if the child elements do not include the attributes already, move the attributes of the parent element to each of its child elements without them

Element in EAD3	Element in EAD 4.0	Transformation route for element and its content (draft)	Transformation route for attributes used with the element (draft)
head	head	When head is used in did (now: identificationData) and the numbered and unnumbered c elements, keep head in itself and: Transform abbr, expan (if existing) into referringString (for details see rows 4, 71) Transform emph, foreign (if existing) into span (for details see rows 65, 76) Transform ptr, ref (if existing) into reference (for details see rows 128, 136) Remove lb Keep only the content of any sub-elements used in dimensions and include a comment about any data potentially encoded in their attributes; values of attributes also available with referringString, span, and reference will of course be moved over to these three elements accordingly When head is used in any other elements (i.e. the ones now allowing for formattingExtension), transform head to e.g. xhtml:h2; in that case: Transform abbr to xhtml:abbr Transform expan into the value of an attribute @title in xhtml:abbr Transform emph@render to xhtml:em@style Transform lb to xhtml:br (ensuring that this comes with a closing tag) Transform ptr and ref to xhtml:a Transform foreign to xhtml:span (e.g. using @xhtml:title to capture the name of the former EAD element)	Remove @althead and include a comment about its removal; where head is transformed to xhtml:h2 include a suggestion of adding an xhtml:h6 or an xhtml:div to include the value of @althead and create a running header
head01	(formattingExtension)	If used with chronlist/listhead or list/listhead, transform to the first xhtml:table/tr/th within the xhtml:table/tr created for the parent listhead If used with index, consider splitting into two or three repeated subjectHeadings elements and using head01 within formattingExtension of the first subjectHeadings as e.g. xhtml:h3 Transform abbr to xhtml:abbr Transform expan into the value of an attribute @title in xhtml:abbr Transform emph@render to xhtml:em@style Transform lb to xhtml:br (ensuring that this comes with a closing tag) Transform ptr and ref to xhtml:a Transform foreign to xhtml:span (e.g. using @xhtml:title to capture the name of the former EAD element)	Transform @audience="internal" into e.g. @xhtml:hidden (while removing the attribute if the value "external" is used) Transform @id into @xhtml:id Transform @altrender into @xhtml:style Transform @lang into @xhtml:lang Integrate @script with the value of @lang in @xhtml:lang by using ietf-bcp-47 values (or remove the attribute)

Note: This sheet shows the changes that would need to be applied to each existing element and attribute in order to move it from EAD3 to EAD 4.0, while the sheets "Elements" and "Attributes" show the changes applied to each element and attribute in terms of its definition within the EAD schema. This sheet here therefore only refers to existing elements, but does not cover elements newly added to EAD 4.0.
Furthermore, this sheet only includes the actual changes. Anything that stays the same (e.g. an element or attribute that isn't renamed or replaced or doesn't have camelCasing applied or when an element retains the same attributes in EAD 4.0 as it had in EAD3), won't be mentioned as such.

General attributes' transformation for all elements that have existed in EAD3 already:
If @audience is used in one or more elements, add @audienceEncoding with the default value "EASList" to control
Replace @altrender with @xhtml:style and add the XHTML namespace to the element using @altrender; include a comment to confirm that @altrender was indeed used to indicate alternative formatting
Rename @lang to @languageOfElement; make sure @languageEncoding exists in control; if it doesn't, add the attribute with the default value "otherLanguageEncoding"
Rename @script to @scriptOfElement; make sure @scriptEncoding exists in control; if it doesn't, add the attribute with the default value "otherScriptEncoding"

General transformation for attributes of elements that are removed:
If a parent element is removed from EAD 4.0, but there is a transformation route for its child elements and if parent and child elements share attributes, check whether the attributes exist with the child element already; if the child elements include the attributes already, remove the attributes of the parent element along with the element itself; if the child elements do not include the attributes already, move the attributes of the parent element to each of its child elements without them

Element in EAD3	**Element in EAD 4.0**	**Transformation route for element and its content (draft)**	**Transformation route for attributes used with the element (draft)**
head02	(formattingExtension)	If used with chronlist/listhead or list/listhead, transform to the second xhtml:table/tr/th within the xhtml:table/tr created for the parent listhead If used with index, consider splitting into two or three repeated subjectHeadings elements and using head02 within formattingExtension of the second subjectHeadings as e.g. xhtml:h3 Transform abbr to xhtml:abbr Transform expan to the value of an attribute @title in xhtml:abbr Transform emph@render to xhtml:em@style Transform lb to xhtml:br (ensuring that this comes with a closing tag) Transform ptr and ref to xhtml:a Transform foreign to xhtml:span (e.g. using @xhtml:title to capture the name of the former EAD element)	Transform @audience="internal" into e.g. @xhtml:hidden (while removing the attribute if the value "external" is used) Transform @id into @xhtml:id Transform @altrender into @xhtml:style Transform @lang into @xhtml:lang Integrate @script with the value of @lang in @xhtml:lang by using ietf-bcp-47 values (or remove the attribute)
head03	(formattingExtension)	If used with chronlist/listhead, transform to the third xhtml:table/tr/th within the xhtml:table/tr created for the parent listhead If used with index, consider splitting into two or three repeated subjectHeadings elements and using head03 within formattingExtension of the third subjectHeadings as e.g. xhtml:h3 Transform abbr to xhtml:abbr Transform expan into the value of an attribute @title in xhtml:abbr Transform emph@render to xhtml:em@style Transform lb to xhtml:br (ensuring that this comes with a closing tag) Transform ptr and ref to xhtml:a Transform foreign to xhtml:span (e.g. using @xhtml:title to capture the name of the former EAD element)	Transform @audience="internal" into e.g. @xhtml:hidden (while removing the attribute if the value "external" is used) Transform @id into @xhtml:id Transform @altrender into @xhtml:style Transform @lang into @xhtml:lang Integrate @script with the value of @lang in @xhtml:lang by using ietf-bcp-47 values (or remove the attribute)

Note: This sheet shows the changes that would need to be applied to each existing element and attribute in order to move it from EAD3 to EAD 4.0, while the sheets "Elements" and "Attributes" show the changes applied to each element and attribute in terms of its definition within the EAD schema. This sheet here therefore only refers to existing elements, but does not cover elements newly added to EAD 4.0.
Furthermore, this sheet only includes the actual changes. Anything that stays the same (e.g. an element or attribute that isn't renamed or replaced or doesn't have camelCasing applied or when an element retains the same attributes in EAD 4.0 as it had in EAD3), won't be mentioned as such.

General attributes' transformation for all elements that have existed in EAD3 already:
If @audience is used in one or more elements, add @audienceEncoding with the default value "EASList" to control
Replace @altrender with @xhtml:style and add the XHTML namespace to the element using @altrender; include a comment to confirm that @altrender was indeed used to indicate alternative formatting
Rename @lang to @languageOfElement; make sure @languageEncoding exists in control; if it doesn't, add the attribute with the default value "otherLanguageEncoding"
Rename @script to @scriptOfElement; make sure @scriptEncoding exists in control; if it doesn't, add the attribute with the default value "otherScriptEncoding"

General transformation for attributes of elements that are removed:
If a parent element is removed from EAD 4.0, but there is a transformation route for its child elements and if parent and child elements share attributes, check whether the attributes exist with the child element already; if the child elements include the attributes already, remove the attributes of the parent element along with the element itself; if the child elements do not include the attributes already, move the attributes of the parent element to each of its child elements without them

Element in EAD3	Element in EAD 4.0	Transformation route for element and its content (draft)	Transformation route for attributes used with the element (draft)
index	(subjectHeadings)	Integrate with subjectHeadings If only used with p, keep as is If used with p and any other formatting elements (i.e. blockquote, chronlist, head, list, table), move all into formattingExtension while adapting to XHTML where necessary (see rows 24, 40, 82, 96, 117, 155 for details) The same applies, if index is used with listhead (and its sub-elements head01, head02, head03) (see row 97 for details) If index/indexentry is used with a single access element or with the sub-element namegrp including repeated access elements, move them into subjectHeadings/subject, into the entity elements agent or function or place, or into publicationNote as applicable (see row 87 for details) If index/indexentry is used with a single ptr or a single ref or a ptrgrp including repeated reference elements, move the IDs referenced in the @target attribute (s) of ptr and ref into the @target attribute of the new element now including the access term (see row 87 for details) If used with nesting, move any formatting sub-elements of the nested controlaccess into subjectHeadingst/formattingExtension to recreate the complete structure of the element alongside the sub-elements of the unnested (or parent) controlaccess; in that case, even if the unnested (or parent) controlaccess is only used with p, those p elements would also be used into subjectHeadings/formattingExtension; any access sub-elements of the nested index get the same transformation as for those of the unnested (or parent) index	If @localtype is used, apply camelCasing to change to @localType; create a localTypeDeclaration with @id in control and add the default content "List of local types for subject headings" to its sub-element reference; then add @localTypeDeclarationReference to subjectHeadings pointing to the @id of said localTypeDeclaration; if a localTypeDeclaration with this content in its sub-element reference already exists, point to this instead of creating another one) Remove @encodinganalog and include a comment about the removal; if any parent element in EAD3 indicates a relatedencoding according to MARC21, add the MARC21 namespace and use the attribute @marc21:tag in subjectHeadings to include the value of @encodinganalog (xmlns:marc21="http://www.loc.gov/MARC21/slim" marc21:tag="...")

Note: This sheet shows the changes that would need to be applied to each existing element and attribute in order to move it from EAD3 to EAD 4.0, while the sheets "Elements" and "Attributes" show the changes applied to each element and attribute in terms of its definition within the EAD schema. This sheet here therefore only refers to existing elements, but does not cover elements newly added to EAD 4.0.
Furthermore, this sheet only includes the actual changes. Anything that stays the same (e.g. an element or attribute that isn't renamed or replaced or doesn't have camelCasing applied or when an element retains the same attributes in EAD 4.0 as it had in EAD3), won't be mentioned as such.

General attributes' transformation for all elements that have existed in EAD3 already:
If @audience is used in one or more elements, add @audienceEncoding with the default value "EASList" to control
Replace @altrender with @xhtml:style and add the XHTML namespace to the element using @altrender; include a comment to confirm that @altrender was indeed used to indicate alternative formatting
Rename @lang to @languageOfElement; make sure @languageEncoding exists in control; if it doesn't, add the attribute with the default value "otherLanguageEncoding"
Rename @script to @scriptOfElement; make sure @scriptEncoding exists in control; if it doesn't, add the attribute with the default value "otherScriptEncoding"

General transformation for attributes of elements that are removed:
If a parent element is removed from EAD 4.0, but there is a transformation route for its child elements and if parent and child elements share attributes, check whether the attributes exist with the child element already; if the child elements include the attributes already, remove the attributes of the parent element along with the element itself; if the child elements do not include the attributes already, move the attributes of the parent element to each of its child elements without them

Element in EAD3	**Element in EAD 4.0**	**Transformation route for element and its content (draft)**	**Transformation route for attributes used with the element (draft)**
indexentry	(subjectHeadings)	Move indexentry/subject/part as well as indexentry/namegrp/subject/part into subjectHeadings/subject/term (see row 153 for details on subject) Turn indexentry/corpname, indexentry/famname, indexentry/name, and indexentry/persname as well as indexentry/namegrp/corpname, indexentry/namegrp/famname, indexentry/namegrp/name, and indexentry/namegrp/persname into the entity element agent (see rows 48, 72, 105, 119 for details) Turn indexentry/function as well as indexentry/namegrp/function into the entity element function (see row 78 for details) Turn indexentry/geogname as well as indexentry/namegrp/geogname into the entity element place (see row 80 for details) Move indexentry/title as well as indexentry/namegrp/title into publicationNote (or turn it into subjectHeadings/subject) (see row 160 for details) Turn indexentry/gerneform and indexentry/occupation as well as indexentry/namegrp/gerneform and indexentry/namegrp/occupation into subjectHeadings/subject (see rows 79, 110 for details) If used with a single ptr or a single ref or a ptrgrp including repeated reference elements, move the IDs referenced in the @target attribute(s) of ptr and ref into the @target attribute of the new element now including the access term; in the case of ptrgrp, this could also lead to several IDs referenced in the same @target attribute; include a comment about this transformation (in case ptr and ref were used in other ways or included other attributes or - in the case of ref - other content) If used with nesting, any sub-elements of the nested indexentry are treated in the same way as those of the unnested (or parent) index	

Note: This sheet shows the changes that would need to be applied to each existing element and attribute in order to move it from EAD3 to EAD 4.0, while the sheets "Elements" and "Attributes" show the changes applied to each element and attribute in terms of its definition within the EAD schema. This sheet here therefore only refers to existing elements, but does not cover elements newly added to EAD 4.0.
Furthermore, this sheet only includes the actual changes. Anything that stays the same (e.g. an element or attribute that isn't renamed or replaced or doesn't have camelCasing applied or when an element retains the same attributes in EAD 4.0 as it had in EAD3), won't be mentioned as such.

General attributes' transformation for all elements that have existed in EAD3 already:
If @audience is used in one or more elements, add @audienceEncoding with the default value "EASList" to control
Replace @altrender with @xhtml:style and add the XHTML namespace to the element using @altrender; include a comment to confirm that @altrender was indeed used to indicate alternative formatting
Rename @lang to @languageOfElement; make sure @languageEncoding exists in control; if it doesn't, add the attribute with the default value "otherLanguageEncoding"
Rename @script to @scriptOfElement; make sure @scriptEncoding exists in control; if it doesn't, add the attribute with the default value "otherScriptEncoding"

General transformation for attributes of elements that are removed:
If a parent element is removed from EAD 4.0, but there is a transformation route for its child elements and if parent and child elements share attributes, check whether the attributes exist with the child element already; if the child elements include the attributes already, remove the attributes of the parent element along with the element itself; if the child elements do not include the attributes already, move the attributes of the parent element to each of its child elements without them

Element in EAD3	**Element in EAD 4.0**	**Transformation route for element and its content (draft)**	**Transformation route for attributes used with the element (draft)**
item	(formattingExtension)	If used with a list without @listtype or with @listtype="unordered" or with @listtype="ordered", transform to xhtml:li If used within a list with @listtype="deflist" without listhead, transform to an xhtml:dd If used within a list with @listtype="deflist" with listhead, transform to the second xhtml:table/tr/td in the xhtml:table/tr created for the parent defitem Transform abbr to xhtml:abbr Transform expan into the value of an attribute @title in xhtml:abbr Transform date to xhtml:time Transform quote to xhtml:q Transform emph@render to xhtml:em@style Transform lb to xhtml:br (ensuring that this comes with a closing tag) Transform ptr and ref to xhtml:a Transform foreign, persname, corpname, famname, geogname, name, occupation, subject, genreform, function, title, footnote, num to xhtml:span (e.g. using @xhtml:title to capture the name of the former EAD element)	Transform @audience="internal" into e.g. @xhtml:hidden (while removing the attribute if the value "external" is used) Transform @id into @xhtml:id Transform @altrender into @xhtml:style Transform @lang into @xhtml:lang Integrate @script with the value of @lang in @xhtml:lang by using ietf-bcp-47 values (or remove the attribute)
label	(formattingExtension)	If used within a list with @listtype="deflist" with**out** listhead, transform to an xhtml:dt If used within a list with @listtype="deflist" **with** listhead, transform to the first xhtml:table/tr/td in the xhtml:table/tr created for the parent defitem Transform abbr to xhtml:abbr Transform expan into the value of an attribute @title in xhtml:abbr Transform emph@render to xhtml:em@style Transform lb to xhtml:br (ensuring that this comes with a closing tag) Transform ptr and ref to xhtml:a Transform foreign to xhtml:span (e.g. using @xhtml:title to capture the name of the former EAD element)	Transform @audience="internal" into e.g. @xhtml:hidden (while removing the attribute if the value "external" is used) Transform @id into @xhtml:id Transform @altrender into @xhtml:style Transform @lang into @xhtml:lang Integrate @script with the value of @lang in @xhtml:lang by using ietf-bcp-47 values (or remove the attribute)
langmaterial	languageOfMaterial	Rename to languageOfMaterial Apply camelCasing to sub-elements languageSet and descriptiveNote (if existing)	Remove @encodinganalog and include a comment about the removal; if any parent element in EAD3 indicates a relatedencoding according to MARC21, add the MARC21 namespace and use the attribute @marc21:tag in languageOfMaterial to include the value of @encodinganalog (xmlns:marc21=" http://www.loc.gov/MARC21/slim" marc21:tag="...") Remove @label and include comment about the removal; an option could be to suggest adding the XHTML namespace and using the attribute @title in abstract to include the value of @label (xmlns:xhtml="http://www.w3.org/1999/xhtml" xhtml:title="...")

Note: This sheet shows the changes that would need to be applied to each existing element and attribute in order to move it from EAD3 to EAD 4.0, while the sheets "Elements" and "Attributes" show the changes applied to each element and attribute in terms of its definition within the EAD schema. This sheet here therefore only refers to existing elements, but does not cover elements newly added to EAD 4.0.
Furthermore, this sheet only includes the actual changes. Anything that stays the same (e.g. an element or attribute that isn't renamed or replaced or doesn't have camelCasing applied or when an element retains the same attributes in EAD 4.0 as it had in EAD3), won't be mentioned as such.

General attributes' transformation for all elements that have existed in EAD3 already:
If @audience is used in one or more elements, add @audienceEncoding with the default value "EASList" to control
Replace @altrender with @xhtml:style and add the XHTML namespace to the element using @altrender; include a comment to confirm that @altrender was indeed used to indicate alternative formatting
Rename @lang to @languageOfElement; make sure @languageEncoding exists in control; if it doesn't, add the attribute with the default value "otherLanguageEncoding"
Rename @script to @scriptOfElement; make sure @scriptEncoding exists in control; if it doesn't, add the attribute with the default value "otherScriptEncoding"

General transformation for attributes of elements that are removed:
If a parent element is removed from EAD 4.0, but there is a transformation route for its child elements and if parent and child elements share attributes, check whether the attributes exist with the child element already; if the child elements include the attributes already, remove the attributes of the parent element along with the element itself; if the child elements do not include the attributes already, move the attributes of the parent element to each of its child elements without them

Element in EAD3	Element in EAD 4.0	Transformation route for element and its content (draft)	Transformation route for attributes used with the element (draft)
language	language		Rename @langcode to @languageCode and apply camelCasing Remove @encodinganalog and include a comment about the removal; if any parent element in EAD3 indicates a relatedencoding according to MARC21, add the MARC21 namespace and use the attribute @marc21:tag in language to include the value of @encodinganalog (xmlns:marc21="http://www.loc.gov/MARC21/slim" marc21:tag="...") Remove @label and include comment about the removal; an option could be to suggest adding the XHTML namespace and using the attribute @title in abstract to include the value of @label (xmlns:xhtml="http://www.w3.org/1999/xhtml" xhtml:title="...")
languagedeclaration	languageDeclaration	Apply camelCasing Remove sub-elements language and script Apply camelCasing to sub-element descriptiveNote (if existing)	Move @langcode and @scriptcode (if existing) into languagedeclaration directly Rename @langcode to @languageCode and apply camelCasing to @scriptCode (optional) into languageDeclaration If @langcode does not exist, include @languageCode (as this is required) with the value "mul" (multiple languages) in languageDeclaration and ensure that @languageEncoding in control points to ISO 639-2 (if it doesn't exist already); include a comment that the actual language code should be added Remove @encodinganalog and include a comment about the removal; if control or ead indicates a relatedencoding according to MARC21, add the MARC21 namespace and use the attribute @marc21:tag in languageDeclaration to include the value of @encodinganalog (xmlns:marc21="http://www.loc.gov/MARC21/slim" marc21:tag="...")
languageset	languageSet	Apply camelCasing Rename sub-element script to writingSystem Apply camelCasing to sub-element descriptiveNote	Remove @encodinganalog and include a comment about the removal; if control or ead indicates a relatedencoding according to MARC21, add the MARC21 namespace and use the attribute @marc21:tag in languageDeclaration to include the value of @encodinganalog (xmlns:marc21="http://www.loc.gov/MARC21/slim" marc21:tag="...")
lb	(formattingExtension)	Remove where XHTML isn't applied Transform to xhtml:br where XHTML is applied	

Note: This sheet shows the changes that would need to be applied to each existing element and attribute in order to move it from EAD3 to EAD 4.0, while the sheets "Elements" and "Attributes" show the changes applied to each element and attribute in terms of its definition within the EAD schema. This sheet here therefore only refers to existing elements, but does not cover elements newly added to EAD 4.0. Furthermore, this sheet only includes the actual changes. Anything that stays the same (e.g. an element or attribute that isn't renamed or replaced or doesn't have camelCasing applied or when an element retains the same attributes in EAD 4.0 as it had in EAD3), won't be mentioned as such.		**General attributes' transformation for all elements that have existed in EAD3 already:** If @audience is used in one or more elements, add @audienceEncoding with the default value "EASList" to control Replace @altrender with @xhtml:style and add the XHTML namespace to the element using @altrender; include a comment to confirm that @altrender was indeed used to indicate alternative formatting Rename @lang to @languageOfElement; make sure @languageEncoding exists in control; if it doesn't, add the attribute with the default value "otherLanguageEncoding" Rename @script to @scriptOfElement; make sure @scriptEncoding exists in control; if it doesn't, add the attribute with the default value "otherScriptEncoding" **General transformation for attributes of elements that are removed:** If a parent element is removed from EAD 4.0, but there is a transformation route for its child elements and if parent and child elements share attributes, check whether the attributes exist with the child element already; if the child elements include the attributes already, remove the attributes of the parent element along with the element itself; if the child elements do not include the attributes already, move the attributes of the parent element to each of its child elements without them	
Element in EAD3	**Element in EAD 4.0**	**Transformation route for element and its content (draft)**	**Transformation route for attributes used with the element (draft)**
legalstatus	legalStatus	Apply camelCasing Move legalStatus into identificationData Add an empty term sub-element (and include a comment to indicate that it would be recommended to add a term identifying the legal status of the records) If legalstatus/p, legalstatus/list/item, or legalstatus/table/tgroup/tbody/row/entry are used and include a date element as mixed content, move this to being a sub-element of legalStatus and include a comment about this needing to be checked Move (the first) date information included in a legalstatus/chronlist/chronitem to being a direct sub-element of legalStatus (making datesingle to date and applying camelCasing to daterange with fromdate and todate and to dateset as applicable); include a comment about this needing to be checked If legalstatus/p, legalstatus/list/item, or legalstatus/table/tgroup/tbody/row/entry are used and include a geogname element as mixed content, move this to being a sub-element of legalStatus and include a comment about this needing to be checked Move geogname(s) included in a legalstatus/chronlist/chronitem or a legalstatus/chronlist/chronitem/chronitemset to being a direct sub-element of legalStatus and include a comment about this needing to be checked If only used with p, move all p elements into a legalStatus/descriptiveNote (see row 117 for details for transformation of p itself) If used with p and any other formatting elements (i.e. blockquote, chronlist, head, list, table), remove all and include a comment about the removal in this specific context, e.g. with a recommendation of reformatting any textual description with simple p elements only If used with nesting, move the nested legalstatus into being a separate, repeated identficitationData/legalStatus treating its sub-elements as described above	If @localtype is used, apply camelCasing to change to @localType; create a localTypeDeclaration with @id in control and add the default content "List of local types for legal statuses" to its sub-element reference; then add @localTypeDeclarationReference to legalStatus pointing to the @id of said localTypeDeclaration; if a localTypeDeclaration with this content in its sub-element reference already exists, point to this instead of creating another one) Remove @encodinganalog and include a comment about the removal; if any parent element in EAD3 indicates a relatedencoding according to MARC21, add the MARC21 namespace and use the attribute @marc21:tag in legalStatus to include the value of @encodinganalog (xmlns:marc21="http://www.loc.gov/MARC21/slim" marc21:tag="...")

Note: This sheet shows the changes that would need to be applied to each existing element and attribute in order to move it from EAD3 to EAD 4.0, while the sheets "Elements" and "Attributes" show the changes applied to each element and attribute in terms of its definition within the EAD schema. This sheet here therefore only refers to existing elements, but does not cover elements newly added to EAD 4.0. Furthermore, this sheet only includes the actual changes. Anything that stays the same (e.g. an element or attribute that isn't renamed or replaced or doesn't have camelCasing applied or when an element retains the same attributes in EAD 4.0 as it had in EAD3), won't be mentioned as such.		**General attributes' transformation for all elements that have existed in EAD3 already:** If @audience is used in one or more elements, add @audienceEncoding with the default value "EASList" to control Replace @altrender with @xhtml:style and add the XHTML namespace to the element using @altrender; include a comment to confirm that @altrender was indeed used to indicate alternative formatting Rename @lang to @languageOfElement; make sure @languageEncoding exists in control; if it doesn't, add the attribute with the default value "otherLanguageEncoding" Rename @script to @scriptOfElement; make sure @scriptEncoding exists in control; if it doesn't, add the attribute with the default value "otherScriptEncoding" **General transformation for attributes of elements that are removed:** If a parent element is removed from EAD 4.0, but there is a transformation route for its child elements and if parent and child elements share attributes, check whether the attributes exist with the child element already; if the child elements include the attributes already, remove the attributes of the parent element along with the element itself; if the child elements do not include the attributes already, move the attributes of the parent element to each of its child elements without them	
Element in EAD3	**Element in EAD 4.0**	**Transformation route for element and its content (draft)**	**Transformation route for attributes used with the element (draft)**
list	(formattingExtension)	Transform list to xhtml:ul when used without @listtype or with @listtype=" unordered" and to xhtml:ol when used with @listtype="ordered" Transform the sub-elements head into e.g. xhtml:h3 and item into xhmtl:li When list is used with @listtype="deflist" with**out** listhead, transform list to xhtml:dl; in this case, transform each list/defitem/label into an xhtml:dt and and each list/defitem/item into an xhtml:dd within that same xhtml:dl When list is used with @listtype="deflist" **with** listhead, transform list to xhtml: table with two columns; in this case, transform listhead/head01 and listhead/head02 into one xhtml:table/tr/th each within the same xhtml:table/tr; furthermore, transform each list/defitem/label into the first xhtml:table/tr/td and and each list/defitem/item into the second xhtml:table/tr/td within that same xhtml:table/tr	Transform @audience="internal" into e.g. @xhtml:hidden (while removing the attribute if the value "external" is used) Transform @id into @xhtml:id Transform @altrender into @xhtml:style Transform @lang into @xhtml:lang Integrate @script with the value of @lang in @xhtml:lang by using ietf-bcp-47 values (or remove the attribute) Integrate @mark and @numeration into @style @listtype is only used to determine the transformation route for list, but is not kept in itself
listhead	(formattingExtension)	If used with chronlist or list, transform listhead into an xhtml:table/tr and include listhead/head01 and listhead/head02 plus - in the context of chronlist - potentially listhead/head03 into one xhtml:table/tr/th each within that same xhtml:table/tr If used with index, consider splitting into two or three repeated subjectHeadings elements and using listhead's subelements head01, head02, and head03 within formattingExtension of each subjectHeadings as e.g. xhtml:h3	Transform @audience="internal" into e.g. @xhtml:hidden (while removing the attribute if the value "external" is used) Transform @id into @xhtml:id Transform @altrender into @xhtml:style Transform @lang into @xhtml:lang Integrate @script with the value of @lang in @xhtml:lang by using ietf-bcp-47 values (or remove the attribute)
localcontrol	*removed*	Remove localcontrol without any transformation to replace it with something else; include a comment about the removal	
localtypedeclaration	localTypeDeclaration	Apply camelCasing Rename sub-elements abbr to shortCode and citation to reference Apply camelCasing to sub-element descriptiveNote Make sure sub-element reference comes first, followed by shortCode (if existing) and descriptiveNote (if existing)	Remove @encodinganalog and include a comment about the removal; if control or ead indicates a relatedencoding according to MARC21, add the MARC21 namespace and use the attribute @marc21:tag in localTypeDeclaration to include the value of @encodinganalog (xmlns:marc21="http://www.loc. gov/MARC21/slim" marc21:tag="...")
maintenanceagency	maintenanceAgency	Apply camelCasing Apply camelCasing to sub-elements agencyCode (if existing), agencyName, otherAgencyCode (if existing), descriptiveNote (if existing) Make sure that any otherAgencyCode (if existing) appears after all instance of agencyName	Apply camelCasing to @countryCode (if existing); make sure that @countryEncoding exists in control in this case Remove @encodinganalog and include a comment about the removal; if control or ead indicates a relatedencoding according to MARC21, add the MARC21 namespace and use the attribute @marc21:tag in maintenanceAgency to include the value of @encodinganalog (xmlns:marc21="http://www.loc. gov/MARC21/slim" marc21:tag="...")

Note: This sheet shows the changes that would need to be applied to each existing element and attribute in order to move it from EAD3 to EAD 4.0, while the sheets "Elements" and "Attributes" show the changes applied to each element and attribute in terms of its definition within the EAD schema. This sheet here therefore only refers to existing elements, but does not cover elements newly added to EAD 4.0.
Furthermore, this sheet only includes the actual changes. Anything that stays the same (e.g. an element or attribute that isn't renamed or replaced or doesn't have camelCasing applied or when an element retains the same attributes in EAD 4.0 as it had in EAD3), won't be mentioned as such.

General attributes' transformation for all elements that have existed in EAD3 already:
If @audience is used in one or more elements, add @audienceEncoding with the default value "EASList" to control
Replace @altrender with @xhtml:style and add the XHTML namespace to the element using @altrender; include a comment to confirm that @altrender was indeed used to indicate alternative formatting
Rename @lang to @languageOfElement; make sure @languageEncoding exists in control; if it doesn't, add the attribute with the default value "otherLanguageEncoding"
Rename @script to @scriptOfElement; make sure @scriptEncoding exists in control; if it doesn't, add the attribute with the default value "otherScriptEncoding"

General transformation for attributes of elements that are removed:
If a parent element is removed from EAD 4.0, but there is a transformation route for its child elements and if parent and child elements share attributes, check whether the attributes exist with the child element already; if the child elements include the attributes already, remove the attributes of the parent element along with the element itself; if the child elements do not include the attributes already, move the attributes of the parent element to each of its child elements without them

Element in EAD3	**Element in EAD 4.0**	**Transformation route for element and its content (draft)**	**Transformation route for attributes used with the element (draft)**
maintenanceevent	maintenanceEvent	Apply camelCasing Remove eventtype Apply camelCasing to agentType and move the element into maintenanceEvent/agent Apply camelCasing to sub-elements eventDateTime and eventDescription Change sequence of elements so that agent comes before eventDateTime and then eventDescription (if existing)	Add @maintenanceEventType to maintenanceEvent, moving the value of eventtype@value into maintenanceEvent@maintenanceEventType Add @maintenanceEventTypeEncoding with the default value "EASList" to control Remove @encodinganalog and include a comment about the removal; if control or ead indicates a relatedencoding according to MARC21, add the MARC21 namespace and use the attribute @marc21:tag in maintenanceEvent to include the value of @encodinganalog (xmlns:marc21="http://www.loc.gov/MARC21/slim" marc21:tag="...")
maintenancehistory	maintenanceHistory	Apply camelCasing Also apply camelCasing to sub-element maintenanceEvent	Remove @encodinganalog and include a comment about the removal; if control or ead indicates a relatedencoding according to MARC21, add the MARC21 namespace and use the attribute @marc21:tag in maintenanceHistory to include the value of @encodinganalog (xmlns:marc21="http://www.loc.gov/MARC21/slim" marc21:tag="...")
maintenancestatus	@maintenanceStatus	Remove maintenancestatus from control	Add @maintenanceStatus to control, moving the value of maintenancestatus@value into control@maintenanceStatus Add @maintenanceStatusEncoding with the default value "EASList" to control
materialspec	materialSpec	Apply camelCasing Transform abbr, expan (if existing) into referringString (for details see rows 4, 71) Transform emph, foreign (if existing) into span (for details see rows 65, 76) Transform ptr, ref (if existing) into reference (for details see rows 128, 136) Remove lb Keep only the content of any sub-elements used in container and include a comment about any data potentially encoded in their attributes; values of attributes also available with referringString, span, and reference will of course be moved over to these three elements accordingly	Remove @encodinganalog and include a comment about the removal; if any parent element in EAD3 indicates a relatedencoding according to MARC21, add the MARC21 namespace and use the attribute @marc21:tag in materialSpec to include the value of @encodinganalog (xmlns:marc21="http://www.loc.gov/MARC21/slim" marc21:tag="...") Remove @label and include comment about the removal; an option could be to suggest adding the XHTML namespace and using the attribute @title in abstract to include the value of @label (xmlns:xhtml="http://www.w3.org/1999/xhtml" xhtml:title="...")

Note: This sheet shows the changes that would need to be applied to each existing element and attribute in order to move it from EAD3 to EAD 4.0, while the sheets "Elements" and "Attributes" show the changes applied to each element and attribute in terms of its definition within the EAD schema. This sheet here therefore only refers to existing elements, but does not cover elements newly added to EAD 4.0.
Furthermore, this sheet only includes the actual changes. Anything that stays the same (e.g. an element or attribute that isn't renamed or replaced or doesn't have camelCasing applied or when an element retains the same attributes in EAD 4.0 as it had in EAD3), won't be mentioned as such.

General attributes' transformation for all elements that have existed in EAD3 already:
If @audience is used in one or more elements, add @audienceEncoding with the default value "EASList" to control
Replace @altrender with @xhtml:style and add the XHTML namespace to the element using @altrender; include a comment to confirm that @altrender was indeed used to indicate alternative formatting
Rename @lang to @languageOfElement; make sure @languageEncoding exists in control; if it doesn't, add the attribute with the default value "otherLanguageEncoding"
Rename @script to @scriptOfElement; make sure @scriptEncoding exists in control; if it doesn't, add the attribute with the default value "otherScriptEncoding"

General transformation for attributes of elements that are removed:
If a parent element is removed from EAD 4.0, but there is a transformation route for its child elements and if parent and child elements share attributes, check whether the attributes exist with the child element already; if the child elements include the attributes already, remove the attributes of the parent element along with the element itself; if the child elements do not include the attributes already, move the attributes of the parent element to each of its child elements without them

Element in EAD3	Element in EAD 4.0	Transformation route for element and its content (draft)	Transformation route for attributes used with the element (draft)
name	(agent) (referringString)	When name is used as sub-element of origination, repository, controlaccess, indexentry, or namegrp, transform it to agent within a new agents plural element added directly to archDesc respectively the relevant numbered or unnumbered c element (if agents already exists on this level of description, add the new agent to it after any already existing agent elements); in the case of name being used as a sub-element of origination or repository, that agent element will be created in the transformation of origination respectively repository Move the content of the part sub-element(s) of name into agentName; if name includes several part elements, concatenate these into the same agentName element and include a comment about this concatenation Create a parallel agentType sub-element and add the default content "Corporate Body" Create a parallel agentRole sub-element and add the value of the attribute @relator (if that was used with name); otherwise do NOT add agentRole, but include a comment about agentRole being available for further definition When name is used as a mixed content element in ref, archref, bibref, abstract, unittitle, physfacet, or p, transform it to referringString in reference, abstract, unitTitle, physFacet and p respectively Move the content of the part sub-element(s) of name into referringString; if name includes several part elements, concatenate these into the same referringString element and include a comment about this concatenation When name is used as a mixed content element in event, item, or entry, find appropriate XHTML encoding to capture its information, e.g. the XHTML element <span> within the XHTML elements <li>, <td>, or <dt> together with the XHTML @title attribute including the default value "Name of an agent"	Move the value of name@identifier to agent@valueURI or referringString@valueURI Move the value of name@source to agent@vocabularySource or referringString@vocabularySource Move the value of name@rules to a newly created conventionDeclaration with @id in control and add its value to the conventionDeclaration sub-element reference; then add @conventionDeclarationReference to agent or referringString pointing to the @id of said conventionDeclaration; if a conventionDeclaration with this content in its sub-element reference already exists, point to this instead of creating another one Remove @normal and include a comment about its removal If @relator is used in the context of origination, repository, controlaccess, indexentry, or namegrp, move its value into an agentRole element within the newly created agent element; if name@relator is used in any other context, remove the attribute and include a comment about its removal If name@localtype is used, apply camelCasing to change to agent@localType or referringString@localType; create a localTypeDeclaration with @id in control and add the default content "List of local types for agents" (in the case of transforming name to agent) or "List of local types for corporate names" (in the case of transforming to referringString) to its sub-element reference; then add @localTypeDeclarationReference to agent or referringString pointing to the @id of said localTypeDeclaration; if a localTypeDeclaration with this content in its sub-element reference already exists, point to this instead of creating another one) Remove @encodinganalog and include a comment about the removal; if any parent element in EAD3 indicates a relatedencoding according to MARC21, add the MARC21 namespace and use the attribute @marc21:tag in agent or referringString to include the value of @encodinganalog (xmlns:marc21="http://www.loc.gov/MARC21/slim" marc21:tag="...")

Note: This sheet shows the changes that would need to be applied to each existing element and attribute in order to move it from EAD3 to EAD 4.0, while the sheets "Elements" and "Attributes" show the changes applied to each element and attribute in terms of its definition within the EAD schema. This sheet here therefore only refers to existing elements, but does not cover elements newly added to EAD 4.0.
Furthermore, this sheet only includes the actual changes. Anything that stays the same (e.g. an element or attribute that isn't renamed or replaced or doesn't have camelCasing applied or when an element retains the same attributes in EAD 4.0 as it had in EAD3), won't be mentioned as such.

General attributes' transformation for all elements that have existed in EAD3 already:
If @audience is used in one or more elements, add @audienceEncoding with the default value "EASList" to control
Replace @altrender with @xhtml:style and add the XHTML namespace to the element using @altrender; include a comment to confirm that @altrender was indeed used to indicate alternative formatting
Rename @lang to @languageOfElement; make sure @languageEncoding exists in control; if it doesn't, add the attribute with the default value "otherLanguageEncoding"
Rename @script to @scriptOfElement; make sure @scriptEncoding exists in control; if it doesn't, add the attribute with the default value "otherScriptEncoding"

General transformation for attributes of elements that are removed:
If a parent element is removed from EAD 4.0, but there is a transformation route for its child elements and if parent and child elements share attributes, check whether the attributes exist with the child element already; if the child elements include the attributes already, remove the attributes of the parent element along with the element itself; if the child elements do not include the attributes already, move the attributes of the parent element to each of its child elements without them

Element in EAD3	**Element in EAD 4.0**	**Transformation route for element and its content (draft)**	**Transformation route for attributes used with the element (draft)**
namegrp	(subject) (agent) (function) (place) (publicationNote)	Move indexentry/namegrp/subject/part into subjectHeadings/subject/term (see row 153 for details on subject) Turn indexentry/namegrp/corpname, indexentry/namegrp/famname, indexentry/namegrp/name, and indexentry/namegrp/persname into the entity element agent (see rows 48, 72, 105, 119 for details) Turn indexentry/namegrp/function into the entity element function (see row 78 for details) Turn indexentry/namegrp/geogname into the entity element place (see row 80 for details) Move indexentry/namegrp/title into publicationNote (or turn it into subjectHeadings/subject) (see row 160 for details) Turn indexentry/namegrp/gerneform and indexentry/namegrp/occupation into subjectHeadings/subject (see rows 79, 110 for details)	
notestmt	(findAidDesc)	Remove notestmt (for the transformation of its child element controlnote see row 46)	
num	(referringString) (title@localType="publication-number"/part) (title@localType="series-number"/part)	Move content of each num in ref, abstract, unittitle, physfacet, archref, and bibref into a separate referringString in reference (for ref, archref, and bibref), abstract, unitTitle, and physFacet Move content of each num in p into a separate referringString in p, provided this is kept as the EAD element p Transform num when used in p, event, item, entry in the context of moving these elements to the XHTML namespace into xhtml:span (e.g. using @xhtml:title to capture the name of the former EAD element) Move the content of each publicationstmt/num and each seriesstmt/num into a separate findAidDesc/title/part Keep only the content of any sub-elements (m.mixed.basic) used in num and include a comment about any data potentially encoded in their attributes; e.g. <num><abbr>1st</abbr> (<expan>First</expan>) publication in <foreign>Deutsch</foreign> translated by <ref>Goethe-Institut</ref></num> becomes <title><part>1st (First) publication in Deutsch translated by Goethe-Institut</part></title>	Remove @encodinganalog and include a comment about the removal; if any parent element indicates a relatedencoding according to MARC21, add the MARC21 namespace and use the attribute @marc21:tag in referringString respectively in title (for the transformation from the publicationstmt and seriesstmt contexts) to include the value of @encodinganalog (xmlns:marc21="http://www.loc.gov/MARC21/slim" marc21:tag="...") If publicationstmt/num doesn't use @localtype already add a @localType with value "publication-number"; if seriesstmt/num doesn't use @localtype already add a @localType with value "series-number"; if @localtype is used already, apply camelCasing to change to @localType (in either case, create a localTypeDeclaration with @id in control and add the default content "List of local types for titles" to its sub-element <reference>; then add @localTypeDeclarationReference to the new title element pointing to the @id of said localTypeDeclaration; if a localTypeDeclaration with this content in its sub-element reference already exists, point to this instead of creating another one)
objectxmlwrap	objectXMLWrap	Apply camelCasing and make sure, objectXMLWrap is used at the end of the sequence of sub-elements within its respective parent elements	Remove @lang and @script

Note: This sheet shows the changes that would need to be applied to each existing element and attribute in order to move it from EAD3 to EAD 4.0, while the sheets "Elements" and "Attributes" show the changes applied to each element and attribute in terms of its definition within the EAD schema. This sheet here therefore only refers to existing elements, but does not cover elements newly added to EAD 4.0.
Furthermore, this sheet only includes the actual changes. Anything that stays the same (e.g. an element or attribute that isn't renamed or replaced or doesn't have camelCasing applied or when an element retains the same attributes in EAD 4.0 as it had in EAD3), won't be mentioned as such.

General attributes' transformation for all elements that have existed in EAD3 already:
If @audience is used in one or more elements, add @audienceEncoding with the default value "EASList" to control
Replace @altrender with @xhtml:style and add the XHTML namespace to the element using @altrender; include a comment to confirm that @altrender was indeed used to indicate alternative formatting
Rename @lang to @languageOfElement; make sure @languageEncoding exists in control; if it doesn't, add the attribute with the default value "otherLanguageEncoding"
Rename @script to @scriptOfElement; make sure @scriptEncoding exists in control; if it doesn't, add the attribute with the default value "otherScriptEncoding"

General transformation for attributes of elements that are removed:
If a parent element is removed from EAD 4.0, but there is a transformation route for its child elements and if parent and child elements share attributes, check whether the attributes exist with the child element already; if the child elements include the attributes already, remove the attributes of the parent element along with the element itself; if the child elements do not include the attributes already, move the attributes of the parent element to each of its child elements without them

Element in EAD3	**Element in EAD 4.0**	**Transformation route for element and its content (draft)**	**Transformation route for attributes used with the element (draft)**
occupation	(subject) (referringString)	When occupation is used as sub-element of controlaccess, indexentry, or namegrp, transform it to subject within subjectHeadings (with integrates controlaccess and index) Move the content of the part sub-element(s) of occupation into subject/term; if occupation includes several part elements, concatenate these into the same term element and include a comment about this concatenation Create a targetType element parallel to subject/term and add the default content "occupation" Create a targetRole element parallel to subject/term and add the value of the attribute @relator (if that was used with occupation); otherwise include a comment about targetRole being available for further definition When occupation is used as a mixed content element in ref, archref, bibref, abstract, unittitle, physfacet, or p, transform it to referringString in reference, abstract, unitTitle, physFacet and p respectively Move the content of the part sub-element(s) of occupation into referringString; if occupation includes several part elements, concatenate these into the same referringString element and include a comment about this concatenation When occupation is used as a mixed content element in event, item, or entry, find appropriate XHTML encoding to capture its information, e.g. the XHTML element <span> within the XHTML elements <li>, <td>, or <dt> together with the XHTML @title attribute including the default value "Name of an occupation"	Move the value of occupation@identifier to subject@valueURI or referringString@valueURI Move the value of occupation@source to subject@vocabularySource or referringString@vocabularySource Move the value of occupation@rules to a newly created conventionDeclaration with @id in control and add its value to the conventionDeclaration sub-element reference; then add @conventionDeclarationReference to subject or referringString pointing to the @id of said conventionDeclaration; if a conventionDeclaration with this content in its sub-element reference already exists, point to this instead of creating another one Remove @normal and include a comment about its removal If @relator is used in the context of controlaccess, indexentry, or namegrp, move its value into a targetRole element within the newly created subject element; if occupation@relator is used in any other context, remove the attribute and include a comment about its removal If occupation@localtype is used, apply camelCasing to change to subject@localType or referringString@localType; create a localTypeDeclaration with @id in control and add the default content "List of local types for occupations" to its sub-element reference; then add @localTypeDeclarationReference to subject or referringString pointing to the @id of said localTypeDeclaration; if a localTypeDeclaration with this content in its sub-element reference already exists, point to this instead of creating another one) Remove @encodinganalog and include a comment about the removal; if any parent element in EAD3 indicates a relatedencoding according to MARC21, add the MARC21 namespace and use the attribute @marc21:tag in subject or referringString to include the value of @encodinganalog (xmlns:marc21="http://www.loc.gov/MARC21/slim" marc21:tag="...")
odd	otherDescriptiveInfo	Rename to otherDescriptiveInfo If only used with p, keep as is If used with p and any other formatting elements (i.e. blockquote, chronlist, head, list, table), move all into formattingExtension while adapting to XHTML where necessary (see rows 24, 40, 82, 96, 117, 155 for details) If used with nesting, move the sub-elements of the nested odd into otherDescriptiveInfo/formattingExtension to recreate the complete structure of the element alongside the sub-elements of the unnested (or parent) odd; in that case, even if the unnested (or parent) odd is only used with p, those p elements would also be used into otherDescriptiveInfo/formattingExtension	If @localtype is used, apply camelCasing to change to @localType; create a localTypeDeclaration with @id in control and add the default content "List of local types for other descriptive information" to its sub-element reference; then add @localTypeDeclarationReference to otherDescriptiveInfo pointing to the @id of said localTypeDeclaration; if a localTypeDeclaration with this content in its sub-element reference already exists, point to this instead of creating another one) Remove @encodinganalog and include a comment about the removal; if any parent element in EAD3 indicates a relatedencoding according to MARC21, add the MARC21 namespace and use the attribute @marc21:tag in otherDescriptiveInfo to include the value of @encodinganalog (xmlns:marc21="http://www.loc.gov/MARC21/slim" marc21:tag="...")

Note: This sheet shows the changes that would need to be applied to each existing element and attribute in order to move it from EAD3 to EAD 4.0, while the sheets "Elements" and "Attributes" show the changes applied to each element and attribute in terms of its definition within the EAD schema. This sheet here therefore only refers to existing elements, but does not cover elements newly added to EAD 4.0.
Furthermore, this sheet only includes the actual changes. Anything that stays the same (e.g. an element or attribute that isn't renamed or replaced or doesn't have camelCasing applied or when an element retains the same attributes in EAD 4.0 as it had in EAD3), won't be mentioned as such.

General attributes' transformation for all elements that have existed in EAD3 already:
If @audience is used in one or more elements, add @audienceEncoding with the default value "EASList" to control
Replace @altrender with @xhtml:style and add the XHTML namespace to the element using @altrender; include a comment to confirm that @altrender was indeed used to indicate alternative formatting
Rename @lang to @languageOfElement; make sure @languageEncoding exists in control; if it doesn't, add the attribute with the default value "otherLanguageEncoding"
Rename @script to @scriptOfElement; make sure @scriptEncoding exists in control; if it doesn't, add the attribute with the default value "otherScriptEncoding"

General transformation for attributes of elements that are removed:
If a parent element is removed from EAD 4.0, but there is a transformation route for its child elements and if parent and child elements share attributes, check whether the attributes exist with the child element already; if the child elements include the attributes already, remove the attributes of the parent element along with the element itself; if the child elements do not include the attributes already, move the attributes of the parent element to each of its child elements without them

Element in EAD3	Element in EAD 4.0	Transformation route for element and its content (draft)	Transformation route for attributes used with the element (draft)
originalsloc	(formAvailable)	Rename to formAvailable and apply camelCasing Make sure that this new formAvailable element sits within a form**s**Available plural element added directly to archDesc respectively the relevant numbered or unnumbered c element (if formsAvailable already exists on this level of description, add the new formAvailable to it after any already existing formAvailable elements) If only used with p, keep as is If used with p and any other formatting elements (i.e. blockquote, chronlist, head, list, table), move all into formattingExtension while adapting to XHTML where necessary (see rows 24, 40, 82, 96, 117, 155 for details) If used with nesting, move the sub-elements of the nested originalsloc into formAvailable/formattingExtension to recreate the complete structure of the element alongside the sub-elements of the unnested (or parent) originalsloc; in that case, even if the unnested (or parent) originalsloc is only used with p, those p elements would also be used into formAvailable/formattingExtension	If @localtype is used, apply camelCasing to change to @localType; create a localTypeDeclaration with @id in control and add the default content "List of local types for location of originals" to its sub-element reference; then add @localTypeDeclarationReference to formAvailable pointing to the @id of said localTypeDeclaration; if a localTypeDeclaration with this content in its sub-element reference already exists, point to this instead of creating another one) Remove @encodinganalog and include a comment about the removal; if any parent element in EAD3 indicates a relatedencoding according to MARC21, add the MARC21 namespace and use the attribute @marc21:tag in originalsLoc to include the value of @encodinganalog (xmlns:marc21="http://www.loc.gov/MARC21/slim" marc21:tag="...")
origination	(agent)	Transform origination into agent within a new agents plural element added directly to archDesc respectively the relevant numbered or unnumbered c element (if agents already exists on this level of description, add the new agent to it after any already existing agent elements) Create a separate agent element for each corpname, famname, name, and/or persname element in the same origination element Create an agentName sub-element in each agent and move the content of the part sub-element(s) of origination's corpname, famname, name, and/or persname sub-elements into agentName; if a corpname, famname, name, and/or persname element includes several part elements, concatenate these into the same agentName element and include a comment about this concatenation Create an agentType sub-element in each agent and add the default content "Corporate Body" (if origination was used with corpname), "Family" (if origination was used with famname), or "Person" (if origination was used with persname); in case origination was used with name, do NOT create an agentType sub-element In all cases, create an agentRole sub-element in each agent and add the default content "Creator" When transforming several origination elements from the same level of description, keep the agent elements in the same sequence as the transformed origination elements	Move origination/localtype to agent@localType; create a localTypeDeclaration with @id in control and add the default content "List of local types for agents" to its sub-element reference; then add @localTypeDeclarationReference to agent pointing to the @id of said localTypeDeclaration; if a localTypeDeclaration with this content in its sub-element reference already exists, point to this instead of creating another one Remove @encodinganalog and include a comment about the removal; if any parent element in EAD3 indicates a relatedencoding according to MARC21, add the MARC21 namespace and use the attribute @marc21:tag in agent to include the value of @encodinganalog (xmlns:marc21="http://www.loc.gov/MARC21/slim" marc21:tag="...") Remove @label and include comment about the removal; an option could be to suggest adding the XHTML namespace and using the attribute @title in abstract to include the value of @label (xmlns:xhtml="http://www.w3.org/1999/xhtml" xhtml:title="...")

Note: This sheet shows the changes that would need to be applied to each existing element and attribute in order to move it from EAD3 to EAD 4.0, while the sheets "Elements" and "Attributes" show the changes applied to each element and attribute in terms of its definition within the EAD schema. This sheet here therefore only refers to existing elements, but does not cover elements newly added to EAD 4.0.
Furthermore, this sheet only includes the actual changes. Anything that stays the same (e.g. an element or attribute that isn't renamed or replaced or doesn't have camelCasing applied or when an element retains the same attributes in EAD 4.0 as it had in EAD3), won't be mentioned as such.

General attributes' transformation for all elements that have existed in EAD3 already:
If @audience is used in one or more elements, add @audienceEncoding with the default value "EASList" to control
Replace @altrender with @xhtml:style and add the XHTML namespace to the element using @altrender; include a comment to confirm that @altrender was indeed used to indicate alternative formatting
Rename @lang to @languageOfElement; make sure @languageEncoding exists in control; if it doesn't, add the attribute with the default value "otherLanguageEncoding"
Rename @script to @scriptOfElement; make sure @scriptEncoding exists in control; if it doesn't, add the attribute with the default value "otherScriptEncoding"

General transformation for attributes of elements that are removed:
If a parent element is removed from EAD 4.0, but there is a transformation route for its child elements and if parent and child elements share attributes, check whether the attributes exist with the child element already; if the child elements include the attributes already, remove the attributes of the parent element along with the element itself; if the child elements do not include the attributes already, move the attributes of the parent element to each of its child elements without them

Element in EAD3	**Element in EAD 4.0**	**Transformation route for element and its content (draft)**	**Transformation route for attributes used with the element (draft)**
otheragencycode	otherAgencyCode	Apply camelCasing Move otherAgencyCode to appear after all existing instances of agencyName	Remove @encodinganalog and include a comment about the removal; if any parent element in EAD3 indicates a relatedencoding according to MARC21, add the MARC21 namespace and use the attribute @marc21:tag in otherAgencyCode to include the value of @encodinganalog (xmlns:marc21="http://www.loc.gov/MARC21/slim" marc21:tag="...") If @localtype is used, apply camelCasing to change to @localType; create a localTypeDeclaration with @id in control and add the default content "List of local types for other agency codes" to its sub-element reference; then add @localTypeDeclarationReference to otherAgencyCode pointing to the @id of said localTypeDeclaration; if a localTypeDeclaration with this content in its sub-element reference already exists, point to this instead of creating another one
otherfindaid	otherFindAid	Apply camelCasing If only used with p, keep as is If used with p and any other formatting elements (i.e. blockquote, chronlist, head, list, table), move all into formattingExtension while adapting to XHTML where necessary (see rows 24, 40, 82, 96, 117, 155 for details) If used with archref and/or bibref, replace these elements with p/reference when in a pure EAD 4.0 encoding or with the appropriate XHTML encoding when used together with formattingExtension (see rows 18 and 22 for details); include a comment about reviewing this and considering an encoding using relations instead If used with nesting, move the sub-elements of the nested otherfindaid into otherFindAid/formattingExtension to recreate the complete structure of the element alongside the sub-elements of the unnested (or parent) otherfindaid; in that case, even if the unnested (or parent) otherfindaid is only used with p, those p elements would also be used into otherFindAid/formattingExtension	If @localtype is used, apply camelCasing to change to @localType; create a localTypeDeclaration with @id in control and add the default content "List of local types for other finding aids" to its sub-element reference; then add @localTypeDeclarationReference to otherFindAid pointing to the @id of said localTypeDeclaration; if a localTypeDeclaration with this content in its sub-element reference already exists, point to this instead of creating another one Remove @encodinganalog and include a comment about the removal; if any parent element in EAD3 indicates a relatedencoding according to MARC21, add the MARC21 namespace and use the attribute @marc21:tag in otherFindAidto include the value of @encodinganalog (xmlns:marc21="http://www.loc.gov/MARC21/slim" marc21:tag="...")
otherrecordid	otherRecordId	Apply camelCasing	Remove @encodinganalog and include a comment about the removal; if control or ead indicates a relatedencoding according to MARC21, add the MARC21 namespace and use the attribute @marc21:tag in otherRecordId to include the value of @encodinganalog (xmlns:marc21="http://www.loc.gov/MARC21/slim" marc21:tag="...") If @localtype is used, apply camelCasing to change to @localType; create a localTypeDeclaration with @id in control and add the default content "List of local types for other record identifiers" to its sub-element reference; then add @localTypeDeclarationReference to otherRecordId pointing to the @id of said localTypeDeclaration; if a localTypeDeclaration with this content in its sub-element reference already exists, point to this instead of creating another one

Note: This sheet shows the changes that would need to be applied to each existing element and attribute in order to move it from EAD3 to EAD 4.0, while the sheets "Elements" and "Attributes" show the changes applied to each element and attribute in terms of its definition within the EAD schema. This sheet here therefore only refers to existing elements, but does not cover elements newly added to EAD 4.0.
Furthermore, this sheet only includes the actual changes. Anything that stays the same (e.g. an element or attribute that isn't renamed or replaced or doesn't have camelCasing applied or when an element retains the same attributes in EAD 4.0 as it had in EAD3), won't be mentioned as such.

General attributes' transformation for all elements that have existed in EAD3 already:
If @audience is used in one or more elements, add @audienceEncoding with the default value "EASList" to control
Replace @altrender with @xhtml:style and add the XHTML namespace to the element using @altrender; include a comment to confirm that @altrender was indeed used to indicate alternative formatting
Rename @lang to @languageOfElement; make sure @languageEncoding exists in control; if it doesn't, add the attribute with the default value "otherLanguageEncoding"
Rename @script to @scriptOfElement; make sure @scriptEncoding exists in control; if it doesn't, add the attribute with the default value "otherScriptEncoding"

General transformation for attributes of elements that are removed:
If a parent element is removed from EAD 4.0, but there is a transformation route for its child elements and if parent and child elements share attributes, check whether the attributes exist with the child element already; if the child elements include the attributes already, remove the attributes of the parent element along with the element itself; if the child elements do not include the attributes already, move the attributes of the parent element to each of its child elements without them

Element in EAD3	**Element in EAD 4.0**	**Transformation route for element and its content (draft)**	**Transformation route for attributes used with the element (draft)**
p	p	Transform abbr, expan, persname, corpname, famname, geogname, name, occupation, subject, genreform, function, title, date, footnote, num (if existing) into referringString (for details see rows 4, 48, 52, 71, 72, 75, 78, 79, 80, 105, 108, 110, 119, 153, 160) Transform emph, foreign, quote (if existing) into span (for details see rows 65, 76, 134) Transform ptr, ref (if existing) into reference (for details see rows 128, 136) Remove lb Keep only the content of any sub-elements used in abstract and include a comment about any data potentially encoded in their attributes; values of attributes also available with referringString, span, and reference will of course be moved over to these three elements accordingly If p is used with list as a sub-element, transform this into a p in the XHTML namespace in all contexts where formattingExtension is now available (all contexts where p is used except for descriptiveNote) - e.g. <p>Payment options include:<list><item>cash</item><item>credit card</item><item>vouchers.</item></list></p> is transformed into <p>Payment options include:<ul><li>cash</li><li>credit card</li><li>vouchers.</li></ul></p> If p is used with list as a sub-element in descriptiveNote, split the content of <p> and of the <item> sub-elements in its sub-element <list> across repeated <p> elements - e.g. <p>Payment options include:<list><item>cash</item><item>credit card</item><item>vouchers.</item></list></p> is transformed into <p>Payment options include:</p><p>cash</p><p>credit card</p><p>vouchers.</p> If p is used in editionstmt, publicationstmt, or seriesstmt (see rows 64, 131, and 148) or if it is used together with other block elements (e.g. in accessConditions, row 6, etc.) or if it is used within blockquote (see row 24), move the content of each p into a p in the XHTML namespace within formattingExtension in the relevant parent element In the XHTML namespace, abbr becomes xhtml:abbr, expan becomes the value of an attribute @title in xhtml:abbr, date becomes xhtml:time, quote becomes xhtml:q, emph@render becomes xhtml:em@style, lb becomes xhtml:br (ensuring that this comes with a closing tag), and ptr and ref become xhtml:a foreign, persname, corpname, famname, geogname, name, occupation, subject, genreform, function, title, footnote, num all become xhtml:span (e.g. using @xhtml:title to capture the name of the former EAD element) For transformation of p as a sub-element footnote see row 75	Only general attributes' transformation in the context of transforming ead3:p to ead4:p If p is transformed to xhtml:p, however, @audience="internal" could become @xhtml:hidden (while it would be removed if the value "external" is used), @id becomes @xhtml:id, @altrender becomes @xhtml:style, @lang becomes @xhtml:lang, @script could be integrated with the value of @lang in @xhtml:lang by using ietf-bcp-47 values (or would otherwise be removed)

Note: This sheet shows the changes that would need to be applied to each existing element and attribute in order to move it from EAD3 to EAD 4.0, while the sheets "Elements" and "Attributes" show the changes applied to each element and attribute in terms of its definition within the EAD schema. This sheet here therefore only refers to existing elements, but does not cover elements newly added to EAD 4.0. Furthermore, this sheet only includes the actual changes. Anything that stays the same (e.g. an element or attribute that isn't renamed or replaced or doesn't have camelCasing applied or when an element retains the same attributes in EAD 4.0 as it had in EAD3), won't be mentioned as such.		**General attributes' transformation for all elements that have existed in EAD3 already:** If @audience is used in one or more elements, add @audienceEncoding with the default value "EASList" to control Replace @altrender with @xhtml:style and add the XHTML namespace to the element using @altrender; include a comment to confirm that @altrender was indeed used to indicate alternative formatting Rename @lang to @languageOfElement; make sure @languageEncoding exists in control; if it doesn't, add the attribute with the default value "otherLanguageEncoding" Rename @script to @scriptOfElement; make sure @scriptEncoding exists in control; if it doesn't, add the attribute with the default value "otherScriptEncoding" **General transformation for attributes of elements that are removed:** If a parent element is removed from EAD 4.0, but there is a transformation route for its child elements and if parent and child elements share attributes, check whether the attributes exist with the child element already; if the child elements include the attributes already, remove the attributes of the parent element along with the element itself; if the child elements do not include the attributes already, move the attributes of the parent element to each of its child elements without them	
Element in EAD3	**Element in EAD 4.0**	**Transformation route for element and its content (draft)**	**Transformation route for attributes used with the element (draft)**
part	part	Keep only the content of any sub-elements (m.mixed.basic.date) used in part and include a comment about any data potentially encoded in their attributes; e. g. <part><abbr>ABC</abbr> (<expan>Alphabet Company</expan>) <foreign>" À la mode" </foreign><ref>Grants Programme</ref></part> becomes <part>ABC (Alphabet) "À la mode" Grants Programme</part>	Move the value of part@identifier to @valueURI of the (new) parent element of part (or the element it has been transformed into) in case this attribute doesn't exist there yet; otherwise, remove part@identifier and include a comment about its removal Move the value of part@source to @vocabularySource of the (new) parent element of part (or the element it has been transformed into) in case this attribute doesn't exist there yet; otherwise, remove part@source and include a comment about its removal Move the value of part@rules to a newly created conventionDeclaration with @id in control and add its value to the conventionDeclaration sub-element reference; then add @conventionDeclarationReference to part (or the element it has been transformed into) pointing to the @id of said conventionDeclaration; if a conventionDeclaration with this content in its sub-element reference already exists, point to this instead of creating another one If @localtype is used, apply camelCasing to change to part@localType (or @localType with the element part has been transformed into, in case this attribute doesn't exist there yet); create a localTypeDeclaration with @id in control and add the default content "List of local types for parts of names"; then add @localTypeDeclarationReference to part or the element it has been transformed into pointing to the @id of said localTypeDeclaration; if a localTypeDeclaration with this content in its sub-element reference already exists, point to this instead of creating another one) Remove @encodinganalog and include a comment about the removal; if any parent element in EAD3 indicates a relatedencoding according to MARC21, add the MARC21 namespace and use the attribute @marc21:tag in part to include the value of @encodinganalog (xmlns:marc21="http://www.loc.gov/MARC21/slim" marc21:tag="...")

Note: This sheet shows the changes that would need to be applied to each existing element and attribute in order to move it from EAD3 to EAD 4.0, while the sheets "Elements" and "Attributes" show the changes applied to each element and attribute in terms of its definition within the EAD schema. This sheet here therefore only refers to existing elements, but does not cover elements newly added to EAD 4.0.
Furthermore, this sheet only includes the actual changes. Anything that stays the same (e.g. an element or attribute that isn't renamed or replaced or doesn't have camelCasing applied or when an element retains the same attributes in EAD 4.0 as it had in EAD3), won't be mentioned as such.

General attributes' transformation for all elements that have existed in EAD3 already:
If @audience is used in one or more elements, add @audienceEncoding with the default value "EASList" to control
Replace @altrender with @xhtml:style and add the XHTML namespace to the element using @altrender; include a comment to confirm that @altrender was indeed used to indicate alternative formatting
Rename @lang to @languageOfElement; make sure @languageEncoding exists in control; if it doesn't, add the attribute with the default value "otherLanguageEncoding"
Rename @script to @scriptOfElement; make sure @scriptEncoding exists in control; if it doesn't, add the attribute with the default value "otherScriptEncoding"

General transformation for attributes of elements that are removed:
If a parent element is removed from EAD 4.0, but there is a transformation route for its child elements and if parent and child elements share attributes, check whether the attributes exist with the child element already; if the child elements include the attributes already, remove the attributes of the parent element along with the element itself; if the child elements do not include the attributes already, move the attributes of the parent element to each of its child elements without them

Element in EAD3	**Element in EAD 4.0**	**Transformation route for element and its content (draft)**	**Transformation route for attributes used with the element (draft)**
persname	(agent) (referringString)	When persname is used as sub-element of origination, repository, controlaccess, indexentry, or namegrp, transform it to agent within a new agents plural element added directly to archDesc respectively the relevant numbered or unnumbered c element (if agents already exists on this level of description, add the new agent to it after any already existing agent elements); in the case of persname being used as a sub-element of origination or repository, that agent element will be created in the transformation of origination respectively repository Move the content of the part sub-element(s) of persname into agentName; if persname includes several part elements, concatenate these into the same agentName element and include a comment about this concatenation Create a parallel agentType sub-element and add the default content "Person" Create a parallel agentRole sub-element and add the value of the attribute @relator (if that was used with persname); otherwise add the default content "Creator" (if persname was a sub-element of origination) or "Repository" (if persname was as sub-element of repository); do NOT add agentRole if persname was a sub-element of controlaccess, indexentry, or namegrp NOR was used with @relator, but include a comment about agentRole being available for further definition When persname is used as a mixed content element in ref, archref, bibref, abstract, unittitle, physfacet, or p, transform it to referringString in reference, abstract, unitTitle, physFacet and p respectively Move the content of the part sub-element(s) of persname into referringString; if persname includes several part elements, concatenate these into the same referringString element and include a comment about this concatenation When persname is used as a mixed content element in event, item, or entry, find appropriate XHTML encoding to capture its information, e.g. the XHTML element <span> within the XHTML elements <li>, <td>, or <dt> together with the XHTML @title attribute including the default value "Name of a person"	Move the value of persname@identifier to agent@valueURI or referringString@valueURI Move the value of persname@source to agent@vocabularySource or referringString@vocabularySource Move the value of persname@rules to a newly created conventionDeclaration with @id in control and add its value to the conventionDeclaration sub-element reference; then add @conventionDeclarationReference to agent or referringString pointing to the @id of said conventionDeclaration; if a conventionDeclaration with this content in its sub-element reference already exists, point to this instead of creating another one Remove @normal and include a comment about its removal If @relator is used in the context of origination, repository, controlaccess, indexentry, or namegrp, move its value into an agentRole element within the newly created agent element; if persname@relator is used in any other context, remove the attribute and include a comment about its removal If persname@localtype is used, apply camelCasing to change to agent@localType or referringString@localType; create a localTypeDeclaration with @id in control and add the default content "List of local types for agents" (in the case of transforming persname to agent) or "List of local types for corporate names" (in the case of transforming to referringString) to its sub-element reference; then add @localTypeDeclarationReference to agent or referringString pointing to the @id of said localTypeDeclaration; if a localTypeDeclaration with this content in its sub-element reference already exists, point to this instead of creating another one) Remove @encodinganalog and include a comment about the removal; if any parent element in EAD3 indicates a relatedencoding according to MARC21, add the MARC21 namespace and use the attribute @marc21:tag in agent or referringString to include the value of @encodinganalog (xmlns:marc21="http://www.loc.gov/MARC21/slim" marc21:tag="...")

<table>
<tr><td colspan="2"><b>Note:</b> This sheet shows the changes that would need to be applied to each existing element and attribute in order to move it from EAD3 to EAD 4.0, while the sheets "Elements" and "Attributes" show the changes applied to each element and attribute in terms of its definition within the EAD schema. This sheet here therefore only refers to existing elements, but does not cover elements newly added to EAD 4.0.
Furthermore, this sheet only includes the actual changes. Anything that stays the same (e.g. an element or attribute that isn't renamed or replaced or doesn't have camelCasing applied or when an element retains the same attributes in EAD 4.0 as it had in EAD3), won't be mentioned as such.</td><td colspan="2"><b>General attributes' transformation for all elements that have existed in EAD3 already:</b>
If @audience is used in one or more elements, add @audienceEncoding with the default value "EASList" to control
Replace @altrender with @xhtml:style and add the XHTML namespace to the element using @altrender; include a comment to confirm that @altrender was indeed used to indicate alternative formatting
Rename @lang to @languageOfElement; make sure @languageEncoding exists in control; if it doesn't, add the attribute with the default value "otherLanguageEncoding"
Rename @script to @scriptOfElement; make sure @scriptEncoding exists in control; if it doesn't, add the attribute with the default value "otherScriptEncoding"

<b>General transformation for attributes of elements that are removed:</b>
If a parent element is removed from EAD 4.0, but there is a transformation route for its child elements and if parent and child elements share attributes, check whether the attributes exist with the child element already; if the child elements include the attributes already, remove the attributes of the parent element along with the element itself; if the child elements do not include the attributes already, move the attributes of the parent element to each of its child elements without them</td></tr>
<tr><td></td><td></td><td></td><td></td></tr>
<tr><td><b>Element in EAD3</b></td><td><b>Element in EAD 4.0</b></td><td><b>Transformation route for element and its content (draft)</b></td><td><b>Transformation route for attributes used with the element (draft)</b></td></tr>
<tr><td>physdesc</td><td>physDesc</td><td>Apply camelCasing
Transform abbr, expan (if existing) into referringString (for details see rows 4, 71)
Transform emph, foreign (if existing) into span (for details see rows 65, 76)
Transform ptr, ref (if existing) into reference (for details see rows 128, 136)
Remove lb
Keep only the content of any sub-elements used in container and include a comment about any data potentially encoded in their attributes; values of attributes also available with referringString, span, and reference will of course be moved over to these three elements accordingly</td><td>If @localtype is used, apply camelCasing to change to @localType; create a localTypeDeclaration with @id in control and add the default content "List of local types for phys description" to its sub-element reference; then add @localTypeDeclarationReference to archDesc pointing to the @id of said localTypeDeclaration; if a localTypeDeclaration with this content in its sub-element reference already exists, point to this instead of creating another one
Remove @encodinganalog and include a comment about the removal; if any parent element in EAD3 indicates a relatedencoding according to MARC21, add the MARC21 namespace and use the attribute @marc21:tag in physDesc to include the value of @encodinganalog (xmlns:marc21="http://www.loc.gov/MARC21/slim" marc21:tag="...")
Remove @label and include comment about the removal; an option could be to suggest adding the XHTML namespace and using the attribute @title in abstract to include the value of @label (xmlns:xhtml="http://www.w3.org/1999/xhtml" xhtml:title="...")</td></tr>
<tr><td>physdescset</td><td>physDescSet</td><td>Apply camelCasing to physDescSet and its sub-element physDescStructured</td><td>Remove @encodinganalog and include a comment about the removal; if any parent element in EAD3 indicates a relatedencoding according to MARC21, add the MARC21 namespace and use the attribute @marc21:tag in physDescSet to include the value of @encodinganalog (xmlns:marc21="http://www.loc.gov/MARC21/slim" marc21:tag="...")
Remove @label and include comment about the removal; an option could be to suggest adding the XHTML namespace and using the attribute @title in abstract to include the value of @label (xmlns:xhtml="http://www.w3.org/1999/xhtml" xhtml:title="...")</td></tr>
</table>

Note: This sheet shows the changes that would need to be applied to each existing element and attribute in order to move it from EAD3 to EAD 4.0, while the sheets "Elements" and "Attributes" show the changes applied to each element and attribute in terms of its definition within the EAD schema. This sheet here therefore only refers to existing elements, but does not cover elements newly added to EAD 4.0.
Furthermore, this sheet only includes the actual changes. Anything that stays the same (e.g. an element or attribute that isn't renamed or replaced or doesn't have camelCasing applied or when an element retains the same attributes in EAD 4.0 as it had in EAD3), won't be mentioned as such.

General attributes' transformation for all elements that have existed in EAD3 already:
If @audience is used in one or more elements, add @audienceEncoding with the default value "EASList" to control
Replace @altrender with @xhtml:style and add the XHTML namespace to the element using @altrender; include a comment to confirm that @altrender was indeed used to indicate alternative formatting
Rename @lang to @languageOfElement; make sure @languageEncoding exists in control; if it doesn't, add the attribute with the default value "otherLanguageEncoding"
Rename @script to @scriptOfElement; make sure @scriptEncoding exists in control; if it doesn't, add the attribute with the default value "otherScriptEncoding"

General transformation for attributes of elements that are removed:
If a parent element is removed from EAD 4.0, but there is a transformation route for its child elements and if parent and child elements share attributes, check whether the attributes exist with the child element already; if the child elements include the attributes already, remove the attributes of the parent element along with the element itself; if the child elements do not include the attributes already, move the attributes of the parent element to each of its child elements without them

Element in EAD3	**Element in EAD 4.0**	**Transformation route for element and its content (draft)**	**Transformation route for attributes used with the element (draft)**
physdescstructured	physDescStructured	Apply camelCasing Apply camelCasing to sub-elements unitType, physFacet, and descriptiveNote	Apply camelCasing to attribute @physDescStructuredType Apply camelCasing to the values "materialType" and "spaceOccupied" for @physDescstructuredType (if used) If @physDescStructuredType is used with "carrier", "materialType", or "spaceOccupied", add @physDescStructuredTypeEncoding with value "EASList" to control If @physDescStructuredType is used with "otherphysdescstructuredtype", move the value of @otherphysdescstructuredtype into @physDescStructuredType and add @physDescStructuredTypeEncoding with value "otherPhysDescStructuredTypeEncoding" to control and create a conventionDeclaration sub-element to control with reference saying "Definition of other structured physical description types" Remove @encodinganalog and include a comment about the removal; if any parent element in EAD3 indicates a relatedencoding according to MARC21, add the MARC21 namespace and use the attribute @marc21:tag in physDescStructured to include the value of @encodinganalog (xmlns:marc21=" http://www.loc.gov/MARC21/slim" marc21:tag="...") Remove @label and include comment about the removal; an option could be to suggest adding the XHTML namespace and using the attribute @title in abstract to include the value of @label (xmlns:xhtml="http://www.w3.org/1999/xhtml" xhtml:title="...")
physfacet	physFacet	Apply camelCasing Transform abbr, expan, corpname, famname, function, genreform, geogname, name, occupation, persname, subject, title, date, footnote, num (if existing) into referringString (for details see rows 4, 48, 52, 71, 72, 75, 78, 79, 80, 105, 108, 110, 119, 153, 160) Transform emph, foreign, quote (if existing) into span (for details see rows 65, 76, 134) Transform ptr, ref (if existing) into reference (for details see rows 128, 136) Remove lb Keep only the content of any sub-elements used in physFacet and include a comment about any data potentially encoded in their attributes; values of attributes also available with referringString, span, and reference will of course be moved over to these three elements accordingly	Move the value of @identifier to @valueURI Move the value of @source to @vocabularySource Move the value of @rules to a newly created conventionDeclaration with @id in control and add its value to the conventionDeclaration sub-element reference; then add @conventionDeclarationReference to physFacet pointing to the @id of said conventionDeclaration; if a conventionDeclaration with this content in its sub-element reference already exists, point to this instead of creating another one If @localtype is used, apply camelCasing to change to @localType; create a localTypeDeclaration with @id in control and add the default content "List of local types for physical nature" to its sub-element reference; then add @localTypeDeclarationReference to physFacet pointing to the @id of said localTypeDeclaration; if a localTypeDeclaration with this content in its sub-element reference already exists, point to this instead of creating another one) Remove @encodinganalog and include a comment about the removal; if any parent element in EAD3 indicates a relatedencoding according to MARC21, add the MARC21 namespace and use the attribute @marc21:tag in physFacet to include the value of @encodinganalog (xmlns:marc21="http://www.loc.gov/MARC21/slim" marc21:tag="...")

Note: This sheet shows the changes that would need to be applied to each existing element and attribute in order to move it from EAD3 to EAD 4.0, while the sheets "Elements" and "Attributes" show the changes applied to each element and attribute in terms of its definition within the EAD schema. This sheet here therefore only refers to existing elements, but does not cover elements newly added to EAD 4.0.
Furthermore, this sheet only includes the actual changes. Anything that stays the same (e.g. an element or attribute that isn't renamed or replaced or doesn't have camelCasing applied or when an element retains the same attributes in EAD 4.0 as it had in EAD3), won't be mentioned as such.

General attributes' transformation for all elements that have existed in EAD3 already:
If @audience is used in one or more elements, add @audienceEncoding with the default value "EASList" to control
Replace @altrender with @xhtml:style and add the XHTML namespace to the element using @altrender; include a comment to confirm that @altrender was indeed used to indicate alternative formatting
Rename @lang to @languageOfElement; make sure @languageEncoding exists in control; if it doesn't, add the attribute with the default value "otherLanguageEncoding"
Rename @script to @scriptOfElement; make sure @scriptEncoding exists in control; if it doesn't, add the attribute with the default value "otherScriptEncoding"

General transformation for attributes of elements that are removed:
If a parent element is removed from EAD 4.0, but there is a transformation route for its child elements and if parent and child elements share attributes, check whether the attributes exist with the child element already; if the child elements include the attributes already, remove the attributes of the parent element along with the element itself; if the child elements do not include the attributes already, move the attributes of the parent element to each of its child elements without them

Element in EAD3	**Element in EAD 4.0**	**Transformation route for element and its content (draft)**	**Transformation route for attributes used with the element (draft)**
physloc	physLoc	Apply camelCasing Transform abbr, expan (if existing) into referringString (for details see rows 4, 71) Transform emph, foreign (if existing) into span (for details see rows 65, 76) Transform ptr, ref (if existing) into reference (for details see rows 128, 136) Remove lb Keep only the content of any sub-elements used in container and include a comment about any data potentially encoded in their attributes; values of attributes also available with referringString, span, and reference will of course be moved over to these three elements accordingly	Remove @encodinganalog and include a comment about the removal; if any parent element in EAD3 indicates a relatedencoding according to MARC21, add the MARC21 namespace and use the attribute @marc21:tag in physLoc to include the value of @encodinganalog (xmlns:marc21="http://www.loc.gov/MARC21/slim" marc21:tag="...") Remove @label and include comment about the removal; an option could be to suggest adding the XHTML namespace and using the attribute @title in physLoc to include the value of @label (xmlns:xhtml="http://www.w3.org/1999/xhtml" xhtml:title="...") Move the value(s) of @parent (if existing) into @target
phystech	physicalOrTechnicalRequirements	Rename to physicalOrTechnicalRequirements If only used with p, keep as is If used with p and any other formatting elements (i.e. blockquote, chronlist, head, list, table), move all into formattingExtension while adapting to XHTML where necessary (see rows 24, 40, 82, 96, 117, 155 for details) If used with nesting, move the sub-elements of the nested physTech into physicalOrTechnicalRequirements/formattingExtension to recreate the complete structure of the element alongside the sub-elements of the unnested (or parent) phystech; in that case, even if the unnested (or parent) phystech is only used with p, those p elements would also be used into physicalOrTechnicalRequirements/formattingExtension	If @localtype is used, apply camelCasing to change to @localType; create a localTypeDeclaration with @id in control and add the default content "List of local types for physical or technical requirements" to its sub-element reference; then add @localTypeDeclarationReference to physicalOrTechnicalRequirements pointing to the @id of said localTypeDeclaration; if a localTypeDeclaration with this content in its sub-element reference already exists, point to this instead of creating another one) Remove @encodinganalog and include a comment about the removal; if any parent element in EAD3 indicates a relatedencoding according to MARC21, add the MARC21 namespace and use the attribute @marc21:tag in physicalOrTechnicalRequirements to include the value of @encodinganalog (xmlns:marc21="http://www.loc.gov/MARC21/slim" marc21:tag="...")
prefercite	preferCite	Apply camelCasing If only used with p, keep as is If used with p and any other formatting elements (i.e. blockquote, chronlist, head, list, table), move all into formattingExtension while adapting to XHTML where necessary (see rows 24, 40, 82, 96, 117, 155 for details) If used with nesting, move the sub-elements of the nested prefercite into preferCite/formattingExtension to recreate the complete structure of the element alongside the sub-elements of the unnested (or parent) preferCite; in that case, even if the unnested (or parent) preferCite is only used with p, those p elements would also be used into preferCite/formattingExtension	If @localtype is used, apply camelCasing to change to @localType; create a localTypeDeclaration with @id in control and add the default content "List of local types for preferred citation" to its sub-element reference; then add @localTypeDeclarationReference to preferCite pointing to the @id of said localTypeDeclaration; if a localTypeDeclaration with this content in its sub-element reference already exists, point to this instead of creating another one) Remove @encodinganalog and include a comment about the removal; if any parent element in EAD3 indicates a relatedencoding according to MARC21, add the MARC21 namespace and use the attribute @marc21:tag in preferCite to include the value of @encodinganalog (xmlns:marc21="http://www.loc.gov/MARC21/slim" marc21:tag="...")

Note: This sheet shows the changes that would need to be applied to each existing element and attribute in order to move it from EAD3 to EAD 4.0, while the sheets "Elements" and "Attributes" show the changes applied to each element and attribute in terms of its definition within the EAD schema. This sheet here therefore only refers to existing elements, but does not cover elements newly added to EAD 4.0.
Furthermore, this sheet only includes the actual changes. Anything that stays the same (e.g. an element or attribute that isn't renamed or replaced or doesn't have camelCasing applied or when an element retains the same attributes in EAD 4.0 as it had in EAD3), won't be mentioned as such.

General attributes' transformation for all elements that have existed in EAD3 already:
If @audience is used in one or more elements, add @audienceEncoding with the default value "EASList" to control
Replace @altrender with @xhtml:style and add the XHTML namespace to the element using @altrender; include a comment to confirm that @altrender was indeed used to indicate alternative formatting
Rename @lang to @languageOfElement; make sure @languageEncoding exists in control; if it doesn't, add the attribute with the default value "otherLanguageEncoding"
Rename @script to @scriptOfElement; make sure @scriptEncoding exists in control; if it doesn't, add the attribute with the default value "otherScriptEncoding"

General transformation for attributes of elements that are removed:
If a parent element is removed from EAD 4.0, but there is a transformation route for its child elements and if parent and child elements share attributes, check whether the attributes exist with the child element already; if the child elements include the attributes already, remove the attributes of the parent element along with the element itself; if the child elements do not include the attributes already, move the attributes of the parent element to each of its child elements without them

Element in EAD3	**Element in EAD 4.0**	**Transformation route for element and its content (draft)**	**Transformation route for attributes used with the element (draft)**
processinfo	processInfo	Apply camelCasing If only used with p, keep as is If used with p and any other formatting elements (i.e. blockquote, chronlist, head, list, table), move all into formattingExtension while adapting to XHTML where necessary (see rows 24, 40, 82, 96, 117, 155 for details) If used with nesting, move the sub-elements of the nested processinfo into processInfo/formattingExtension to recreate the complete structure of the element alongside the sub-elements of the unnested (or parent) processinfo; in that case, even if the unnested (or parent) processinfo is only used with p, those p elements would also be used into processInfo/formattingExtension	If @localtype is used, apply camelCasing to change to @localType; create a localTypeDeclaration with @id in control and add the default content "List of local types for processing information" to its sub-element reference; then add @localTypeDeclarationReference to processInfo pointing to the @id of said localTypeDeclaration; if a localTypeDeclaration with this content in its sub-element reference already exists, point to this instead of creating another one) Remove @encodinganalog and include a comment about the removal; if any parent element in EAD3 indicates a relatedencoding according to MARC21, add the MARC21 namespace and use the attribute @marc21:tag in processInfo to include the value of @encodinganalog (xmlns:marc21="http://www.loc.gov/MARC21/slim" marc21:tag="...")
ptr	(reference)	Transform to reference where XHTML isn't applied Transform to xhtml:a where XHTML is applied	Move values of ptr@href, @linkrole, and @linktitle to reference@href, @linkRole, and @linkTitle If @actuate, @arcole, @show have been used, add the XLink namespace to reference and add @xlink:actuate, @xlink:arcrole, @xlink:show to include the attributes' values (xmlns:xlink="http://www.loc.gov/standards/xlink/xlink.xsd" xlink:show="..." xlink:actuate="..." xlink:arcrole="...") Remove @entityref and @xpointer and include a comment about their removal
ptrgrp	@target	Move the IDs referenced in the @target attribute(s) of ptr and ref grouped within ptrgp into the @target attribute(s) of the new element(s) now including the access term(s) encoded in the parallel single access elements or the parallel namegrp element (see row 106 for details); this could also lead to several IDs referenced in the same @target attribute; include a comment about this transformation (in case ptr and ref were used in other ways or included other attributes or - in the case of ref - other content)	
publicationstatus	@publicationStatus	Remove publicationstatus from control	Add @publicationStatus to control, moving the value of publicationstatus@value into control@publicationStatus Add @publicationStatusEncoding with the default value "EASList" to control
publicationstmt	(findAidDesc)	Remove editionstmt (for the transformation of its child elements publisher, date, address, num and p see rows 9, 52, 63, 117, 132)	

Note: This sheet shows the changes that would need to be applied to each existing element and attribute in order to move it from EAD3 to EAD 4.0, while the sheets "Elements" and "Attributes" show the changes applied to each element and attribute in terms of its definition within the EAD schema. This sheet here therefore only refers to existing elements, but does not cover elements newly added to EAD 4.0. Furthermore, this sheet only includes the actual changes. Anything that stays the same (e.g. an element or attribute that isn't renamed or replaced or doesn't have camelCasing applied or when an element retains the same attributes in EAD 4.0 as it had in EAD3), won't be mentioned as such.		**General attributes' transformation for all elements that have existed in EAD3 already:** If @audience is used in one or more elements, add @audienceEncoding with the default value "EASList" to control Replace @altrender with @xhtml:style and add the XHTML namespace to the element using @altrender; include a comment to confirm that @altrender was indeed used to indicate alternative formatting Rename @lang to @languageOfElement; make sure @languageEncoding exists in control; if it doesn't, add the attribute with the default value "otherLanguageEncoding" Rename @script to @scriptOfElement; make sure @scriptEncoding exists in control; if it doesn't, add the attribute with the default value "otherScriptEncoding" **General transformation for attributes of elements that are removed:** If a parent element is removed from EAD 4.0, but there is a transformation route for its child elements and if parent and child elements share attributes, check whether the attributes exist with the child element already; if the child elements include the attributes already, remove the attributes of the parent element along with the element itself; if the child elements do not include the attributes already, move the attributes of the parent element to each of its child elements without them	
Element in EAD3	**Element in EAD 4.0**	**Transformation route for element and its content (draft)**	**Transformation route for attributes used with the element (draft)**
publisher	(agent)	Move the content of each publisher into a separate findAidDesc/agent/agentName and add a parallel findAidDesc/agent/agentRole with the text "Publisher" Keep only the content of any sub-elements (m.mixed.basic) used in publisher and include a comment about any data potentially encoded in their attributes; e. g. <publisher><abbr>ABC</abbr> (<expan>Alphabet Company</expan>) <foreign>à la mode</foreign> as part of their <ref>Grants programme</ref></publisher> becomes <agent><agentName>ABC (Alphabet) à la mode as part of their Grants programme</agentName><agentRole>publisher</agentRole></agent>	Move publisher@localtype to agent@localType; create a localTypeDeclaration with @id in control and add the default content "List of local types for agents" to its sub-element reference; then add @localTypeDeclarationReference to agent pointing to the @id of said localTypeDeclaration; if a localTypeDeclaration with this content in its sub-element reference already exists, point to this instead of creating another one Remove @encodinganalog and include a comment about the removal; if any parent element in EAD3 indicates a relatedencoding according to MARC21, add the MARC21 namespace and use the attribute @marc21:tag in agent to include the value of @encodinganalog (xmlns:marc21="http://www.loc.gov/MARC21/slim" marc21:tag="...")
quantity	quantity		Remove @encodinganalog and include a comment about the removal; if any parent element in EAD3 indicates a relatedencoding according to MARC21, add the MARC21 namespace and use the attribute @marc21:tag in reference to include the value of @encodinganalog (xmlns:marc21="http://www.loc.gov/MARC21/slim" marc21:tag="...")
quote	(span)	Transform to span where XHTML isn't applied; keep only the content of any sub-elements used in quote and include a comment about any data potentially encoded in their attributes Transform to xhtml:q where XHTML is applied	If @localtype is used, apply camelCasing to change to @localType; create a localTypeDeclaration with @id in control and add the default content "List of local types for quotes" to its sub-element reference; then add @localTypeDeclarationReference to referringString pointing to the @id of said localTypeDeclaration; if a localTypeDeclaration with this content in its sub-element reference already exists, point to this instead of creating another one Replace @render with @style in EAD 4.0 or with @xhtml:style in an XHTML transformation context
recordid	recordId	Apply camelCasing If recordid was empty, add default content "0" to recordId and include comment that recordId can't be left empty	Remove @encodinganalog and include a comment about the removal; if control or ead indicates a relatedencoding according to MARC21, add the MARC21 namespace and use the attribute @marc21:tag in recordId to include the value of @encodinganalog (xmlns:marc21="http://www.loc.gov/MARC21/slim" marc21:tag="...") Remove @instanceurl and create an element <findAidDesc href="[value of instanceurl]"> with sub-element title/part including the content of the first control/titlestmt/titleproper

Note: This sheet shows the changes that would need to be applied to each existing element and attribute in order to move it from EAD3 to EAD 4.0, while the sheets "Elements" and "Attributes" show the changes applied to each element and attribute in terms of its definition within the EAD schema. This sheet here therefore only refers to existing elements, but does not cover elements newly added to EAD 4.0. Furthermore, this sheet only includes the actual changes. Anything that stays the same (e.g. an element or attribute that isn't renamed or replaced or doesn't have camelCasing applied or when an element retains the same attributes in EAD 4.0 as it had in EAD3), won't be mentioned as such.		**General attributes' transformation for all elements that have existed in EAD3 already:** If @audience is used in one or more elements, add @audienceEncoding with the default value "EASList" to control Replace @altrender with @xhtml:style and add the XHTML namespace to the element using @altrender; include a comment to confirm that @altrender was indeed used to indicate alternative formatting Rename @lang to @languageOfElement; make sure @languageEncoding exists in control; if it doesn't, add the attribute with the default value "otherLanguageEncoding" Rename @script to @scriptOfElement; make sure @scriptEncoding exists in control; if it doesn't, add the attribute with the default value "otherScriptEncoding" **General transformation for attributes of elements that are removed:** If a parent element is removed from EAD 4.0, but there is a transformation route for its child elements and if parent and child elements share attributes, check whether the attributes exist with the child element already; if the child elements include the attributes already, remove the attributes of the parent element along with the element itself; if the child elements do not include the attributes already, move the attributes of the parent element to each of its child elements without them	
Element in EAD3	**Element in EAD 4.0**	**Transformation route for element and its content (draft)**	**Transformation route for attributes used with the element (draft)**
ref	reference	Rename to reference where XHTML isn't applied Transform abbr, expan, persname, corpname, famname, geogname, name, occupation, subject, genreform, function, title, date, footnote, num (if existing) into reference/referringString (for details see rows 4, 48, 52, 71, 72, 75, 78, 79, 80, 105, 108, 110, 119, 153, 160) Transform emph, foreign, quote (if existing) into reference/span (for details see rows 65, 76, 134) If ref itself does not include @href, @linkrole, or @linktitle, move ptr@href, @linkrole, and @linktitle into reference@href, @linkRole, and @linkTitle; if ptr uses @actuate, @arcole, @show, add the XLink namespace to reference and add @xlink:actuate, @xlink:arcrole, @xlink:show to include the attributes' values (xmlns:xlink="http://www.loc.gov/standards/xlink/xlink.xsd" xlink:show="..." xlink:actuate="..." xlink:arcrole="...") If ref itself already uses linking attributes, move ptr out of ref and create a sibling reference (for transformation of ptr itself see row 128); add an @id to the original (parent) reference and point to this with the @target attribute of the new sibling reference Transform to xhtml:a where XHTML is applied	Move values of ref@href, @linkrole, and @linktitle to reference@href, @linkRole, and @linkTitle If @actuate, @arcole, @show have been used, add the XLink namespace to reference and add @xlink:actuate, @xlink:arcrole, @xlink:show to include the attributes' values (xmlns:xlink="http://www.loc.gov/standards/xlink/xlink.xsd" xlink:show="..." xlink:actuate="..." xlink:arcrole="...") Remove @entityref and @xpointer and include a comment about their removal
relatedmaterial	relatedMaterial	Apply camelCasing If only used with p, keep as is If used with p and any other formatting elements (i.e. blockquote, chronlist, head, list, table), move all into formattingExtension while adapting to XHTML where necessary (see rows 24, 40, 82, 96, 117, 155 for details) If used with archref and/or bibref, replace these elements with p/reference when in a pure EAD 4.0 encoding or with the appropriate XHTML encoding when used together with formattingExtension (see rows 18 and 22 for details); include a comment about reviewing this and considering an encoding using relations instead If used with nesting, move the sub-elements of the nested relatedmaterial into relatedMaterial/formattingExtension to recreate the complete structure of the element alongside the sub-elements of the unnested (or parent) relatedmaterial; in that case, even if the unnested (or parent) relatedmaterial is only used with p, those p elements would also be used into relatedMaterial/formattingExtension	If @localtype is used, apply camelCasing to change to @localType; create a localTypeDeclaration with @id in control and add the default content "List of local types for related material" to its sub-element reference; then add @localTypeDeclarationReference to relatedMaterial pointing to the @id of said localTypeDeclaration; if a localTypeDeclaration with this content in its sub-element reference already exists, point to this instead of creating another one Remove @encodinganalog and include a comment about the removal; if any parent element in EAD3 indicates a relatedencoding according to MARC21, add the MARC21 namespace and use the attribute @marc21:tag in relatedMaterial to include the value of @encodinganalog (xmlns:marc21="http://www.loc.gov/MARC21/slim" marc21:tag="...")

Note: This sheet shows the changes that would need to be applied to each existing element and attribute in order to move it from EAD3 to EAD 4.0, while the sheets "Elements" and "Attributes" show the changes applied to each element and attribute in terms of its definition within the EAD schema. This sheet here therefore only refers to existing elements, but does not cover elements newly added to EAD 4.0.
Furthermore, this sheet only includes the actual changes. Anything that stays the same (e.g. an element or attribute that isn't renamed or replaced or doesn't have camelCasing applied or when an element retains the same attributes in EAD 4.0 as it had in EAD3), won't be mentioned as such.

General attributes' transformation for all elements that have existed in EAD3 already:
If @audience is used in one or more elements, add @audienceEncoding with the default value "EASList" to control
Replace @altrender with @xhtml:style and add the XHTML namespace to the element using @altrender; include a comment to confirm that @altrender was indeed used to indicate alternative formatting
Rename @lang to @languageOfElement; make sure @languageEncoding exists in control; if it doesn't, add the attribute with the default value "otherLanguageEncoding"
Rename @script to @scriptOfElement; make sure @scriptEncoding exists in control; if it doesn't, add the attribute with the default value "otherScriptEncoding"

General transformation for attributes of elements that are removed:
If a parent element is removed from EAD 4.0, but there is a transformation route for its child elements and if parent and child elements share attributes, check whether the attributes exist with the child element already; if the child elements include the attributes already, remove the attributes of the parent element along with the element itself; if the child elements do not include the attributes already, move the attributes of the parent element to each of its child elements without them

Element in EAD3	Element in EAD 4.0	Transformation route for element and its content (draft)	Transformation route for attributes used with the element (draft)
relation	relation	Depending on the value in @relationtype, relation is transformed in different ways: @relationtype="resourcerelation" as a whole is moved into a new relations parent element within relatedMaterial; in case relatedMaterial/relations already exists, any further relation elements are added after the last existing one in relatedMaterial/relations; include a comment that this should be reviewed as there might be other types of related resources encoded as well (see rows 139, 109, 53, 54, 55, 80, 57 for details on the transformation of relation's sub-elements in this case) From a relation@relationtype="cpfrelation", the relation/relationentry is transformed into an agent/agentName, relation/datesingle, daterange, or dateset is transformed into agent/date, dateRange, or dateSet, relation/geogname is transformed into agent/placeName, relation/descriptivenote is transformed into agent/descriptiveNote; relation/objectxmlwrap is removed and a comment is included about the removal From a relation@relationtype="functionrelation", the relation/relationentry is transformed into an function/term, relation/datesingle, daterange, or dateset is transformed into function/date, dateRange, or dateSet, relation/geogname is transformed into function/placeName, relation/descriptivenote is transformed into function/descriptiveNote; relation/objectxmlwrap is removed and a comment is included about the removal relation@relationtype="otherrelationtype" as a whole is removed and a comment is included about the removal	If relation as such is kept (i.e. only in the first case described in column C), remove @relationtype="resourcerelation" In all other cases, remove @relationtype and (if existing) @otherrelationtype If relation@href has been used, add its value to relatedMaterial/relations/relation/targetEntity@valueURI, agent@valueURI, or function@valueURI depending on the transformation context If @linkrole, @linktitle, @actuate, @show, or @arcrole have been used with relation, add the XLink namespace to relatedMaterial/relations/relation/targetEntity, agent, or function respectively and add @xlink:role, @xlink:title, @xlink:actuate, @xlink:arcrole, @xlink:show to include the attributes' values (xmlns:xlink="http://www.loc.gov/standards/xlink/xlink.xsd" xlink:show="..." xlink:actuate="..." xlink:arcrole="...") Remove @encodinganalog and include a comment about the removal; if any parent element in EAD3 indicates a relatedencoding according to MARC21, add the MARC21 namespace and use the attribute @marc21:tag in relatedMaterial/relations/relation, agent, or function depending on the transformation context to include the value of @encodinganalog (xmlns:marc21="http://www.loc.gov/MARC21/slim" marc21:tag="...") Remove @lastdatetimeverified and include a comment about the removal
relationentry	targetEntity	Move the content of relationentry into relatedMaterial/relations/relation/targetEntity/part (in case of relation@relationtype="resourcerelation"), into agentName (in case of relation @relationtype="cpfrelation"), into term (in case of relation @relationtype=" functionrelation) In case of a transformation to targetEntity/part, create several part elements within the same targetEntity in case of repeated relationentry elements in the same relation, where all have the same - or no - language attribution; in case of repeated relationentry elements in the same relation having different language attribution, create a relatedMaterial/relations/relation for each relationentry and then add the content of that relationentry only in targetEntity/part	Remove @encodinganalog and include a comment about the removal; if any parent element in EAD3 indicates a relatedencoding according to MARC21, add the MARC21 namespace and use the attribute @marc21:tag in targetEntity, agentName, or function/term depending on the transformation context to include the value of @encodinganalog (xmlns:marc21="http://www.loc.gov/MARC21/slim" marc21:tag="...") Move the value of @transliteration into a new conventionDeclaration (in its sub-element reference) with @id in control; add @conventionDeclarationReference to targetEntity, agentName, or function/term depending on the transformation context and point to the new conventionDeclaration; should a conventionDeclaration with this content in its sub-element reference exist alreday, point to that instead Remove @localtype and include a comment about the removal

Note: This sheet shows the changes that would need to be applied to each existing element and attribute in order to move it from EAD3 to EAD 4.0, while the sheets "Elements" and "Attributes" show the changes applied to each element and attribute in terms of its definition within the EAD schema. This sheet here therefore only refers to existing elements, but does not cover elements newly added to EAD 4.0. Furthermore, this sheet only includes the actual changes. Anything that stays the same (e.g. an element or attribute that isn't renamed or replaced or doesn't have camelCasing applied or when an element retains the same attributes in EAD 4.0 as it had in EAD3), won't be mentioned as such.		**General attributes' transformation for all elements that have existed in EAD3 already:** If @audience is used in one or more elements, add @audienceEncoding with the default value "EASList" to control Replace @altrender with @xhtml:style and add the XHTML namespace to the element using @altrender; include a comment to confirm that @altrender was indeed used to indicate alternative formatting Rename @lang to @languageOfElement; make sure @languageEncoding exists in control; if it doesn't, add the attribute with the default value "otherLanguageEncoding" Rename @script to @scriptOfElement; make sure @scriptEncoding exists in control; if it doesn't, add the attribute with the default value "otherScriptEncoding" **General transformation for attributes of elements that are removed:** If a parent element is removed from EAD 4.0, but there is a transformation route for its child elements and if parent and child elements share attributes, check whether the attributes exist with the child element already; if the child elements include the attributes already, remove the attributes of the parent element along with the element itself; if the child elements do not include the attributes already, move the attributes of the parent element to each of its child elements without them	
Element in EAD3	**Element in EAD 4.0**	**Transformation route for element and its content (draft)**	**Transformation route for attributes used with the element (draft)**
relations	relations	**Note:** relations is moved from being under archdesc respectively the numbered and unnumbered c elements to being used directly as sub-element of otherFindAid, publicationNote, relatedMaterial, seperatedMaterial, and the new formAvailable; furthermore some of the relations sub-elements are being re-used or have equivalents in other contexts such as agent, function, and place Therefore, the element relations itself isn't really transformed, but its sub-elements find a new place (see rows 138, 139, 109, 53, 54, 55, 80, 57 for details): relation@relationtype="resourcerelation" as a whole is moved into a new relations parent element within relatedMaterial; in case relatedMaterial/relations already exists, any further relation elements are added after the last existing one in relatedMaterial/relations; include a comment that this should be reviewed as there might be other types of related resources encoded as well From a relation@relationtype="cpfrelation", the relation/relationentry is transformed into an agent/agentName, relation/datesingle, daterange, or dateset is transformed into agent/date, dateRange, or dateSet, relation/geogname is transformed into agent/placeName, relation/descriptivenote is transformed into agent/descriptiveNote; relation/objectxmlwrap is removed and a comment is included about the removal From a relation@relationtype="functionrelation", the relation/relationentry is transformed into an function/term, relation/datesingle, daterange, or dateset is transformed into function/date, dateRange, or dateSet, relation/geogname is transformed into function/placeName, relation/descriptivenote is transformed into function/descriptiveNote; relation/objectxmlwrap is removed and a comment is included about the removal relation@relationtype="otherrelationtype" as a whole is removed and a comment is included about the removal	Remove @localtype and include a comment about the removal Remove @encodinganalog and include a comment about the removal; if any parent element in EAD3 indicates a relatedencoding according to MARC21, add the MARC21 namespace and use the attribute @marc21:tag in the element to which the content(s) of relations' sub-elements have been moved to include the value of @encodinganalog (xmlns:marc21="http://www.loc.gov/MARC21/slim" marc21:tag="..."); this only applies, in case the respective sub-element didn't use @encodinganalog itself

Note: This sheet shows the changes that would need to be applied to each existing element and attribute in order to move it from EAD3 to EAD 4.0, while the sheets "Elements" and "Attributes" show the changes applied to each element and attribute in terms of its definition within the EAD schema. This sheet here therefore only refers to existing elements, but does not cover elements newly added to EAD 4.0.
Furthermore, this sheet only includes the actual changes. Anything that stays the same (e.g. an element or attribute that isn't renamed or replaced or doesn't have camelCasing applied or when an element retains the same attributes in EAD 4.0 as it had in EAD3), won't be mentioned as such.

General attributes' transformation for all elements that have existed in EAD3 already:
If @audience is used in one or more elements, add @audienceEncoding with the default value "EASList" to control
Replace @altrender with @xhtml:style and add the XHTML namespace to the element using @altrender; include a comment to confirm that @altrender was indeed used to indicate alternative formatting
Rename @lang to @languageOfElement; make sure @languageEncoding exists in control; if it doesn't, add the attribute with the default value "otherLanguageEncoding"
Rename @script to @scriptOfElement; make sure @scriptEncoding exists in control; if it doesn't, add the attribute with the default value "otherScriptEncoding"

General transformation for attributes of elements that are removed:
If a parent element is removed from EAD 4.0, but there is a transformation route for its child elements and if parent and child elements share attributes, check whether the attributes exist with the child element already; if the child elements include the attributes already, remove the attributes of the parent element along with the element itself; if the child elements do not include the attributes already, move the attributes of the parent element to each of its child elements without them

Element in EAD3	**Element in EAD 4.0**	**Transformation route for element and its content (draft)**	**Transformation route for attributes used with the element (draft)**
repository	(agent)	Transform repository into agent within a new agents plural element added directly to archDesc respectively the relevant numbered or unnumbered c element (if agents already exists on this level of description, add the new agent to it after any already existing agent elements) Create a separate agent element for each corpname, famname, name, and/or persname element in the same repository element Create an agentName sub-element in each agent and move the content of the part sub-element(s) of repository's corpname, famname, name, and/or persname sub-elements into agentName; if a corpname, famname, name, and/or persname element includes several part elements, concatenate these into the same agentName element and include a comment about this concatenation Create an agentType sub-element in each agent and add the default content "Corporate Body" (if repository was used with corpname), "Family" (if repository was used with famname), or "Person" (if repository was used with persname); in case repository was used with name, do NOT create an agentType sub-element In all cases, create an agentRole sub-element in each agent and add the default content "Repository" Move the content of each repository/address/addressline into a separate agent/placeName; if addressline was used with a @localtype, move the value of this attribute to placeName@localType in this context When transforming several repository elements from the same level of description, keep the agent elements in the same sequence as the transformed repository elements	Move repository/localtype to agent@localType; create a localTypeDeclaration with @id in control and add the default content "List of local types for agents" to its sub-element reference; then add @localTypeDeclarationReference to agent pointing to the @id of said localTypeDeclaration; if a localTypeDeclaration with this content in its sub-element reference already exists, point to this instead of creating another one Remove @encodinganalog and include a comment about the removal; if any parent element in EAD3 indicates a relatedencoding according to MARC21, add the MARC21 namespace and use the attribute @marc21:tag in agent to include the value of @encodinganalog (xmlns:marc21="http://www.loc.gov/MARC21/slim" marc21:tag="...") Remove @label and include comment about the removal; an option could be to suggest adding the XHTML namespace and using the attribute @title in abstract to include the value of @label (xmlns:xhtml="http://www.w3.org/1999/xhtml" xhtml:title="...")
representation	(findAidDesc)	Create one findAidDesc for each representation Add formattingExtension as sub-element and move content of representation into a p in the XHTML namespace as encoded within formattingExtension (as it's not foreseeable what the content in representation might represent, formattingExtension is the most generic option)	Move @audience, @id, @languageOfElement, @scriptOfElement to findAidDesc Remove @encodinganalog and include a comment about the removal; if control or ead indicates a relatedencoding according to MARC21, add the MARC21 namespace and use the attribute @marc21:tag in findAidDesc to include the value of @encodinganalog (xmlns:marc21="http://www.loc.gov/MARC21/slim" marc21:tag="...") Remove @localtype and include a comment about the removal Move @href, @linkRole, @linkTitle to <findAidDesc> (apply camelCasing to @linkrole and @linktitle in this context) If @actuate, @arcole, @show have been used, add the XLink namespace to findAidDesc and add @xlink:actuate, @xlink:arcrole, @xlink:show to include the attributes' values (xmlns:xlink="http://www.loc.gov/standards/xlink/xlink.xsd" xlink:show="..." xlink:actuate="..." xlink:arcrole="...")

Note: This sheet shows the changes that would need to be applied to each existing element and attribute in order to move it from EAD3 to EAD 4.0, while the sheets "Elements" and "Attributes" show the changes applied to each element and attribute in terms of its definition within the EAD schema. This sheet here therefore only refers to existing elements, but does not cover elements newly added to EAD 4.0. Furthermore, this sheet only includes the actual changes. Anything that stays the same (e.g. an element or attribute that isn't renamed or replaced or doesn't have camelCasing applied or when an element retains the same attributes in EAD 4.0 as it had in EAD3), won't be mentioned as such.		**General attributes' transformation for all elements that have existed in EAD3 already:** If @audience is used in one or more elements, add @audienceEncoding with the default value "EASList" to control Replace @altrender with @xhtml:style and add the XHTML namespace to the element using @altrender; include a comment to confirm that @altrender was indeed used to indicate alternative formatting Rename @lang to @languageOfElement; make sure @languageEncoding exists in control; if it doesn't, add the attribute with the default value "otherLanguageEncoding" Rename @script to @scriptOfElement; make sure @scriptEncoding exists in control; if it doesn't, add the attribute with the default value "otherScriptEncoding" **General transformation for attributes of elements that are removed:** If a parent element is removed from EAD 4.0, but there is a transformation route for its child elements and if parent and child elements share attributes, check whether the attributes exist with the child element already; if the child elements include the attributes already, remove the attributes of the parent element along with the element itself; if the child elements do not include the attributes already, move the attributes of the parent element to each of its child elements without them	
Element in EAD3	**Element in EAD 4.0**	**Transformation route for element and its content (draft)**	**Transformation route for attributes used with the element (draft)**
rightsdeclaration	rightsDeclaration	Apply camelCasing Rename sub-elements abbr to shortCode and citation to reference Apply camelCasing to sub-element descriptiveNote Make sure sub-element reference comes first, followed by shortCode (if existing) and descriptiveNote (if existing)	Remove @encodinganalog and include a comment about the removal; if control or ead indicates a relatedencoding according to MARC21, add the MARC21 namespace and use the attribute @marc21:tag in rightsDeclaration to include the value of @encodinganalog (xmlns:marc21="http://www.loc.gov/MARC21/slim" marc21:tag="...") Remove @localtype and include a comment about the removal
row	(formattingExtension)	transform to xhtml:table/tr Depending on whether row is a sub-element of tbody or thead, transform its own sub-element entry into either xhtml:table/tr/td or xhtml:table/tr/th	Transform @audience="internal" into e.g. @xhtml:hidden (while removing the attribute if the value "external" is used) Transform @id into @xhtml:id Transform @altrender into @xhtml:style Transform @lang into @xhtml:lang Integrate @script with the value of @lang in @xhtml:lang by using ietf-bcp-47 values (or remove the attribute) Integrate @rowsep and @valign in @xhtml:style
scopecontent	scopeContent	Apply camelCasing If only used with p, keep as is If used with p and any other formatting elements (i.e. blockquote, chronlist, head, list, table), move all into formattingExtension while adapting to XHTML where necessary (see rows 24, 40, 82, 96, 117, 155 for details) If used with nesting, move the sub-elements of the nested scopecontent into scopeContent/formattingExtension to recreate the complete structure of the element alongside the sub-elements of the unnested (or parent) scopecontent; in that case, even if the unnested (or parent) bioghist is only used with p, those p elements would also be used into scopeContentt/formattingExtension	If @localtype is used, apply camelCasing to change to @localType; create a localTypeDeclaration with @id in control and add the default content "List of local types for scope and content" to its sub-element reference; then add @localTypeDeclarationReference to scopeContent pointing to the @id of said localTypeDeclaration; if a localTypeDeclaration with this content in its sub-element reference already exists, point to this instead of creating another one) Remove @encodinganalog and include a comment about the removal; if any parent element in EAD3 indicates a relatedencoding according to MARC21, add the MARC21 namespace and use the attribute @marc21:tag in scopeContent to include the value of @encodinganalog (xmlns:marc21="http://www.loc.gov/MARC21/slim" marc21:tag="...")
script	writingSystem	Rename to writingSystem	Apply camelCasing to @scriptCode Remove @encodinganalog and include a comment about the removal; if any parent element in EAD3 indicates a relatedencoding according to MARC21, add the MARC21 namespace and use the attribute @marc21:tag in writingSystem to include the value of @encodinganalog (xmlns:marc21="http://www.loc.gov/MARC21/slim" marc21:tag="...") Remove @label and include comment about the removal; an option could be to suggest adding the XHTML namespace and using the attribute @title in abstract to include the value of @label (xmlns:xhtml="http://www.w3.org/1999/xhtml" xhtml:title="...")

Note: This sheet shows the changes that would need to be applied to each existing element and attribute in order to move it from EAD3 to EAD 4.0, while the sheets "Elements" and "Attributes" show the changes applied to each element and attribute in terms of its definition within the EAD schema. This sheet here therefore only refers to existing elements, but does not cover elements newly added to EAD 4.0.
Furthermore, this sheet only includes the actual changes. Anything that stays the same (e.g. an element or attribute that isn't renamed or replaced or doesn't have camelCasing applied or when an element retains the same attributes in EAD 4.0 as it had in EAD3), won't be mentioned as such.

General attributes' transformation for all elements that have existed in EAD3 already:
If @audience is used in one or more elements, add @audienceEncoding with the default value "EASList" to control
Replace @altrender with @xhtml:style and add the XHTML namespace to the element using @altrender; include a comment to confirm that @altrender was indeed used to indicate alternative formatting
Rename @lang to @languageOfElement; make sure @languageEncoding exists in control; if it doesn't, add the attribute with the default value "otherLanguageEncoding"
Rename @script to @scriptOfElement; make sure @scriptEncoding exists in control; if it doesn't, add the attribute with the default value "otherScriptEncoding"

General transformation for attributes of elements that are removed:
If a parent element is removed from EAD 4.0, but there is a transformation route for its child elements and if parent and child elements share attributes, check whether the attributes exist with the child element already; if the child elements include the attributes already, remove the attributes of the parent element along with the element itself; if the child elements do not include the attributes already, move the attributes of the parent element to each of its child elements without them

Element in EAD3	**Element in EAD 4.0**	**Transformation route for element and its content (draft)**	**Transformation route for attributes used with the element (draft)**
separatedmaterial	separatedMaterial	Apply camelCasing If only used with p, keep as is If used with p and any other formatting elements (i.e. blockquote, chronlist, head, list, table), move all into formattingExtension while adapting to XHTML where necessary (see rows 24, 40, 82, 96, 117, 155 for details) If used with archref and/or bibref, replace these elements with p/reference when in a pure EAD 4.0 encoding or with the appropriate XHTML encoding when used together with formattingExtension (see rows 18 and 22 for details); include a comment about reviewing this and considering an encoding using relations instead If used with nesting, move the sub-elements of the nested separatedmaterial into separatedMaterial/formattingExtension to recreate the complete structure of the element alongside the sub-elements of the unnested (or parent) separatedmaterial; in that case, even if the unnested (or parent) separatedmaterial is only used with p, those p elements would also be used into separatedMaterial/formattingExtension	If @localtype is used, apply camelCasing to change to @localType; create a localTypeDeclaration with @id in control and add the default content "List of local types for separated material" to its sub-element reference; then add @localTypeDeclarationReference to separatedMaterial pointing to the @id of said localTypeDeclaration; if a localTypeDeclaration with this content in its sub-element reference already exists, point to this instead of creating another one Remove @encodinganalog and include a comment about the removal; if any parent element in EAD3 indicates a relatedencoding according to MARC21, add the MARC21 namespace and use the attribute @marc21:tag in separatedMaterial to include the value of @encodinganalog (xmlns:marc21=" http://www.loc.gov/MARC21/slim" marc21:tag="...")
seriesstmt	(findAidDesc)	Remove titlestmt (for the transformation of its child elements titleproper, num, and p see rows 108, 117, 161)	
source	source	Replace sourceentry with reference; if no sourceentry exists, add an empty <reference/> element for validation and include a comment that this should be used to provide e.g. a title of the source Apply camelCasing to sub-elements descriptiveNote and objectXMLWrap (if existing) and change their order (descriptiveNote first, objectXMLWrap second)	Apply camelCasing to @linkRole and @linkTitle (if existing) If @actuate, @arcole, @show have been used, add the XLink namespace to source and add @xlink:actuate, @xlink:arcrole, @xlink:show to include the attributes' values (xmlns:xlink="http://www.loc.gov/standards/xlink/xlink.xsd" xlink:show="..." xlink:actuate="..." xlink:arcrole="...") Remove @encodinganalog and include a comment about the removal; if any parent element in EAD3 indicates a relatedencoding according to MARC21, add the MARC21 namespace and use the attribute @marc21:tag in source to include the value of @encodinganalog (xmlns:marc21="http://www.loc.gov/MARC21/slim" marc21:tag="...") Remove @lastdatetimeverified and include a comment about the removal
sourceentry	(reference)	Move the content of each source/sourceentry into a separate source/reference	Remove @encodinganalog and include a comment about the removal; if any parent element in EAD3 indicates a relatedencoding according to MARC21, add the MARC21 namespace and use the attribute @marc21:tag in reference to include the value of @encodinganalog (xmlns:marc21="http://www.loc.gov/MARC21/slim" marc21:tag="...") Remove @transliteration and include a comment about the removal; move the value of source/sourceentry@transliteration into a conventionDeclaration/reference, add an @id to conventionDeclaration and a @conventionDeclarationReference to source/reference pointing to that @id

Note: This sheet shows the changes that would need to be applied to each existing element and attribute in order to move it from EAD3 to EAD 4.0, while the sheets "Elements" and "Attributes" show the changes applied to each element and attribute in terms of its definition within the EAD schema. This sheet here therefore only refers to existing elements, but does not cover elements newly added to EAD 4.0. Furthermore, this sheet only includes the actual changes. Anything that stays the same (e.g. an element or attribute that isn't renamed or replaced or doesn't have camelCasing applied or when an element retains the same attributes in EAD 4.0 as it had in EAD3), won't be mentioned as such.	**General attributes' transformation for all elements that have existed in EAD3 already:** If @audience is used in one or more elements, add @audienceEncoding with the default value "EASList" to control Replace @altrender with @xhtml:style and add the XHTML namespace to the element using @altrender; include a comment to confirm that @altrender was indeed used to indicate alternative formatting Rename @lang to @languageOfElement; make sure @languageEncoding exists in control; if it doesn't, add the attribute with the default value "otherLanguageEncoding" Rename @script to @scriptOfElement; make sure @scriptEncoding exists in control; if it doesn't, add the attribute with the default value "otherScriptEncoding" **General transformation for attributes of elements that are removed:** If a parent element is removed from EAD 4.0, but there is a transformation route for its child elements and if parent and child elements share attributes, check whether the attributes exist with the child element already; if the child elements include the attributes already, remove the attributes of the parent element along with the element itself; if the child elements do not include the attributes already, move the attributes of the parent element to each of its child elements without them

Element in EAD3	Element in EAD 4.0	Transformation route for element and its content (draft)	Transformation route for attributes used with the element (draft)
sources	sources		Remove @encodinganalog and @localtype and include a comment about the removal If control or ead indicates a relatedencoding according to MARC21, add the MARC21 namespace and use the attribute @marc21:tag in sources to include the value of @encodinganalog (xmlns:marc21="http://www.loc.gov/MARC21/slim" marc21:tag="...")
sponsor	(agent)	Move the content of each sponsor into a separate findAidDesc/agent/agentName and add a parallel findAidDesc/agent/agentRole with the text "Sponsor" Keep only the content of any sub-elements (m.mixed.basic) used in sponsor and include a comment about any data potentially encoded in their attributes; e.g. <sponsor><abbr>ABC</abbr> (<expan>Alphabet Company</expan>) <foreign>à la mode</foreign> as part of their <ref>Grants programme</ref></sponsor> becomes <agent><agentName>ABC (Alphabet) à la mode as part of their Grants programme</agentName><agentRole>sponsor</agentRole></agent>	Move sponsor@localtype to agent@localType; create a localTypeDeclaration with @id in control and add the default content "List of local types for agents" to its sub-element reference; then add @localTypeDeclarationReference to agent pointing to the @id of said localTypeDeclaration; if a localTypeDeclaration with this content in its sub-element reference already exists, point to this instead of creating another one Remove @encodinganalog and include a comment about the removal; if any parent element in EAD3 indicates a relatedencoding according to MARC21, add the MARC21 namespace and use the attribute @marc21:tag in agent to include the value of @encodinganalog (xmlns:marc21="http://www.loc.gov/MARC21/slim" marc21:tag="...")

Note: This sheet shows the changes that would need to be applied to each existing element and attribute in order to move it from EAD3 to EAD 4.0, while the sheets "Elements" and "Attributes" show the changes applied to each element and attribute in terms of its definition within the EAD schema. This sheet here therefore only refers to existing elements, but does not cover elements newly added to EAD 4.0. Furthermore, this sheet only includes the actual changes. Anything that stays the same (e.g. an element or attribute that isn't renamed or replaced or doesn't have camelCasing applied or when an element retains the same attributes in EAD 4.0 as it had in EAD3), won't be mentioned as such.		**General attributes' transformation for all elements that have existed in EAD3 already:** If @audience is used in one or more elements, add @audienceEncoding with the default value "EASList" to control Replace @altrender with @xhtml:style and add the XHTML namespace to the element using @altrender; include a comment to confirm that @altrender was indeed used to indicate alternative formatting Rename @lang to @languageOfElement; make sure @languageEncoding exists in control; if it doesn't, add the attribute with the default value "otherLanguageEncoding" Rename @script to @scriptOfElement; make sure @scriptEncoding exists in control; if it doesn't, add the attribute with the default value "otherScriptEncoding" **General transformation for attributes of elements that are removed:** If a parent element is removed from EAD 4.0, but there is a transformation route for its child elements and if parent and child elements share attributes, check whether the attributes exist with the child element already; if the child elements include the attributes already, remove the attributes of the parent element along with the element itself; if the child elements do not include the attributes already, move the attributes of the parent element to each of its child elements without them	
Element in EAD3	**Element in EAD 4.0**	**Transformation route for element and its content (draft)**	**Transformation route for attributes used with the element (draft)**
subject	subject (referringString)	When subject is used as sub-element of controlaccess, indexentry, or namegrp, transform it to subject within subjectHeadings Move the content of the part sub-element(s) of subject into term; if subject includes several part elements, concatenate these into the same term element and include a comment about this concatenation Create a targetRole sub-element and add the value of the attribute @relator (if that was used with subject); otherwise include a comment about targetRole being available for further definition When subject is used as a mixed content element in ref, archref, bibref, abstract, unittitle, physfacet, or p, transform it to referringString in reference, abstract, unitTitle, physFacet and p respectively Move the content of the part sub-element(s) of subject into referringString; if subject includes several part elements, concatenate these into the same referringString element and include a comment about this concatenation When subject is used as a mixed content element in event, item, or entry, find appropriate XHTML encoding to capture its information, e.g. the XHTML element <span> within the XHTML elements <li>, <td>, or <dt> together with the XHTML @title attribute including the default value "Name of a subject"	Move the value of subject@identifier to subject@valueURI or referringString@valueURI Move the value of subject@source to subject@vocabularySource or referringString@vocabularySource Move the value of subject@rules to a newly created conventionDeclaration with @id in control and add its value to the conventionDeclaration sub-element reference; then add @conventionDeclarationReference to subject or referringString pointing to the @id of said conventionDeclaration; if a conventionDeclaration with this content in its sub-element reference already exists, point to this instead of creating another one Remove @normal and include a comment about its removal If @relator is used in the context of controlaccess, indexentry, or namegrp, move its value into a targetRole element within the newly created subject element; if @relator is used in any other context, remove the attribute and include a comment about its removal If @localtype is used, apply camelCasing to change to subject@localType or referringString@localType; create a localTypeDeclaration with @id in control and add the default content "List of local types for subjects" to its sub-element reference; then add @localTypeDeclarationReference to subject or referringString pointing to the @id of said localTypeDeclaration; if a localTypeDeclaration with this content in its sub-element reference already exists, point to this instead of creating another one) Remove @encodinganalog and include a comment about the removal; if any parent element in EAD3 indicates a relatedencoding according to MARC21, add the MARC21 namespace and use the attribute @marc21:tag in subject or referringString to include the value of @encodinganalog (xmlns:marc21="http://www.loc.gov/MARC21/slim" marc21:tag="...")
subtitle	(title)	Move the content of each subtitle into a separate title (i.e. its sub-element part) Keep only the content of any sub-elements (m.mixed.basic) used in subtitle and include a comment about any data potentially encoded in their attributes; e.g. <subtitle><abbr>ABC</abbr> (<expan>Alphabet</expan>) <foreign>à la mode</foreign> as seen at <ref>Le Louvre, Paris</ref></subtitle> becomes <title><part>ABC (Alphabet) à la mode as seen as Le Louvre, Paris</part></title>	If subtitle doesn't use @localtype already, add a @localType with value "subtitle"; if @localtype is used already, apply camelCasing to change to @localType (in either case, create a localTypeDeclaration with @id in control and add the default content "List of local types for titles" to its sub-element reference; then add @localTypeDeclarationReference to the new title element pointing to the @id of said localTypeDeclaration; if a localTypeDeclaration with this content in its sub-element reference already exists, point to this instead of creating another one) Remove @encodinganalog and include a comment about the removal; if any parent element in EAD3 indicates a relatedencoding according to MARC21, add the MARC21 namespace and use the attribute @marc21:tag in title to include the value of @encodinganalog (xmlns:marc21="http://www.loc.gov/MARC21/slim" marc21:tag="...")

Note: This sheet shows the changes that would need to be applied to each existing element and attribute in order to move it from EAD3 to EAD 4.0, while the sheets "Elements" and "Attributes" show the changes applied to each element and attribute in terms of its definition within the EAD schema. This sheet here therefore only refers to existing elements, but does not cover elements newly added to EAD 4.0.
Furthermore, this sheet only includes the actual changes. Anything that stays the same (e.g. an element or attribute that isn't renamed or replaced or doesn't have camelCasing applied or when an element retains the same attributes in EAD 4.0 as it had in EAD3), won't be mentioned as such.

General attributes' transformation for all elements that have existed in EAD3 already:
If @audience is used in one or more elements, add @audienceEncoding with the default value "EASList" to control
Replace @altrender with @xhtml:style and add the XHTML namespace to the element using @altrender; include a comment to confirm that @altrender was indeed used to indicate alternative formatting
Rename @lang to @languageOfElement; make sure @languageEncoding exists in control; if it doesn't, add the attribute with the default value "otherLanguageEncoding"
Rename @script to @scriptOfElement; make sure @scriptEncoding exists in control; if it doesn't, add the attribute with the default value "otherScriptEncoding"

General transformation for attributes of elements that are removed:
If a parent element is removed from EAD 4.0, but there is a transformation route for its child elements and if parent and child elements share attributes, check whether the attributes exist with the child element already; if the child elements include the attributes already, remove the attributes of the parent element along with the element itself; if the child elements do not include the attributes already, move the attributes of the parent element to each of its child elements without them

Element in EAD3	**Element in EAD 4.0**	**Transformation route for element and its content (draft)**	**Transformation route for attributes used with the element (draft)**
table	(formattingExtension)	transform to xhtml:table Transform sub-element head into e.g. xhtml:h3 Transform each t**head**/row/entry of table/tgroup into an xhtml:table/tr/t**h** and each t**body**/row/entry of table/tgroup into an xhtml:table/tr/**td**	Transform @audience="internal" into e.g. @xhtml:hidden (while removing the attribute if the value "external" is used) Transform @id into @xhtml:id Transform @altrender into @xhtml:style Transform @lang into @xhtml:lang Integrate @script with the value of @lang in @xhtml:lang by using ietf-bcp-47 values (or remove the attribute) Integrate @frame, @colsep, @rowsep, @pgwide in @xhtml:style
tbody	(formattingExtension)	Transform each tbody/row to an xhtml:table/tr	Remove all other attributes of tbody (given that the element is mainly used to indicate wether row/entry elements end up in xhtml:tr/td or xhtml:tr/th, but is skipped in itself, while only its sub-elements are transformed)
term	term	**Note:** the element term is only used as a sub-element of localcontrol in EAD3. As localcontrol is removed without a replacement, there is **NO** need for a transformation route for term neither. The term that will be available in EAD 4.0 as sub-element of legalStatus, function, and subject is newly introduced, so to speak, and follows the element's definition in EAC-CPF 2.0.	
tgroup	(formattingExtension)	Transform each t**head**/row/entry sub-element of table/tgroup into an xhtml: table/tr/t**h** and each t**body**/row/entry sub-element of table/tgroup into an xhtml: table/tr/**td** For the transformation of sub-element colspec see row 42)	Move @align into @xhtml:style with the parent xhtml:table Remove all other attributes of tgroup (given that the element in itself is skipped and only its sub-elements are transformed)
thead	(formattingExtension)	Transform each thead/row to an xhtml:table/tr	Remove all other attributes of thead (given that the element is mainly used to indicate wether row/entry elements end up in xhtml:tr/td or xhtml:tr/th, but is skipped in itself, while only its sub-elements are transformed)

Note: This sheet shows the changes that would need to be applied to each existing element and attribute in order to move it from EAD3 to EAD 4.0, while the sheets "Elements" and "Attributes" show the changes applied to each element and attribute in terms of its definition within the EAD schema. This sheet here therefore only refers to existing elements, but does not cover elements newly added to EAD 4.0. Furthermore, this sheet only includes the actual changes. Anything that stays the same (e.g. an element or attribute that isn't renamed or replaced or doesn't have camelCasing applied or when an element retains the same attributes in EAD 4.0 as it had in EAD3), won't be mentioned as such.	**General attributes' transformation for all elements that have existed in EAD3 already:** If @audience is used in one or more elements, add @audienceEncoding with the default value "EASList" to control Replace @altrender with @xhtml:style and add the XHTML namespace to the element using @altrender; include a comment to confirm that @altrender was indeed used to indicate alternative formatting Rename @lang to @languageOfElement; make sure @languageEncoding exists in control; if it doesn't, add the attribute with the default value "otherLanguageEncoding" Rename @script to @scriptOfElement; make sure @scriptEncoding exists in control; if it doesn't, add the attribute with the default value "otherScriptEncoding" **General transformation for attributes of elements that are removed:** If a parent element is removed from EAD 4.0, but there is a transformation route for its child elements and if parent and child elements share attributes, check whether the attributes exist with the child element already; if the child elements include the attributes already, remove the attributes of the parent element along with the element itself; if the child elements do not include the attributes already, move the attributes of the parent element to each of its child elements without them

Element in EAD3	**Element in EAD 4.0**	**Transformation route for element and its content (draft)**	**Transformation route for attributes used with the element (draft)**
title	(publicationNote) (referringString)	When title is used as sub-element of controlaccess, indexentry, or namegrp, move it to a new publicationNote element Move the content of the part sub-element(s) of title into the part elements of publicationNote/relations/relation/targetEntity Create a relation/targetRole sub-element and add the value of the attribute @relator (if that was used with title); otherwise include a comment about targetRole being available for further definition Create a relation/targetType sub-element and add the default text "title of a work" When title is used as a mixed content element in ref, archref, bibref, abstract, unittitle, physfacet, or p, transform it to referringString in reference, abstract, unitTitle, physFacet and p respectively Move the content of the part sub-element(s) of title into referringString; if title includes several part elements, concatenate these into the same referringString element and include a comment about this concatenation When title is used as a mixed content element in event, item, or entry, find appropriate XHTML encoding to capture its information, e.g. the XHTML element <span> within the XHTML elements <li>, <td>, or <dt> together with the XHTML @title attribute including the default value "Title of a work"	Move the value of title@identifier to targetEntity@valueURI or referringString@valueURI Move the value of title@source to targetEntity@vocabularySource or referringString@vocabularySource Move the value of title@rules to a newly created conventionDeclaration with @id in control and add its value to the conventionDeclaration sub-element reference; then add @conventionDeclarationReference to targetEntity or referringString pointing to the @id of said conventionDeclaration; if a conventionDeclaration with this content in its sub-element reference already exists, point to this instead of creating another one Remove @normal and include a comment about its removal If @relator is used in the context of controlaccess, indexentry, or namegrp, move its value into a targetRole element within the newly created publicationNote/relations/relation element; if @relator is used in any other context, remove the attribute and include a comment about its removal Remove @localtype and include a comment about its removal Remove @render and include a comment about its removal Remove @encodinganalog and include a comment about the removal; if any parent element in EAD3 indicates a relatedencoding according to MARC21, add the MARC21 namespace and use the attribute @marc21:tag in targetEntity or referringString to include the value of @encodinganalog (xmlns:marc21="http://www.loc.gov/MARC21/slim" marc21:tag="...")
titleproper	(title)	Move the content of each titleproper (of titlestmt and of seriesstmt) into a separate title (i.e. its sub-element part) Keep only the content of any sub-elements (m.mixed.basic) used in titleproper and include a comment about any data potentially encoded in their attributes; e.g. <titleproper><abbr>ABC</abbr> (<expan>Alphabet</expan>) <foreign>à la mode</foreign> as seen at <ref>Le Louvre, Paris</ref></titleproper> becomes <title><part>ABC (Alphabet) à la mode as seen as Le Louvre, Paris</part></title>	For each seriesstmt/titleproper, check whether @localtype is used already; if it isn't, add a @localType with value "series-title"; if @localtype is used already, apply camelCasing to change to @localType (in either case, create a localTypeDeclaration with @id in control and add the default content "List of local types for titles" to its sub-element reference; then add @localTypeDeclarationReference to the new title element pointing to the @id of said localTypeDeclaration; if a localTypeDeclaration with this content in its sub-element reference already exists, point to this instead of creating another one) Rename @render (if existing) to @style and include a comment that the value of @style should ideally be a standard CSS property (see e.g. https://www.w3schools.com/cssref/index.php) Remove @encodinganalog and include a comment about the removal; if any parent element in EAD3 indicates a relatedencoding according to MARC21, add the MARC21 namespace and use the attribute @marc21:tag in title to include the value of @encodinganalog (xmlns:marc21="http://www.loc.gov/MARC21/slim" marc21:tag="...")
titlestmt	(findAidDesc)	Remove titlestmt (for the transformation of its child elements titleproper, subtitle, author and sponsor see rows 20, 152, 154, 161)	

Note: This sheet shows the changes that would need to be applied to each existing element and attribute in order to move it from EAD3 to EAD 4.0, while the sheets "Elements" and "Attributes" show the changes applied to each element and attribute in terms of its definition within the EAD schema. This sheet here therefore only refers to existing elements, but does not cover elements newly added to EAD 4.0.
Furthermore, this sheet only includes the actual changes. Anything that stays the same (e.g. an element or attribute that isn't renamed or replaced or doesn't have camelCasing applied or when an element retains the same attributes in EAD 4.0 as it had in EAD3), won't be mentioned as such.

General attributes' transformation for all elements that have existed in EAD3 already:
If @audience is used in one or more elements, add @audienceEncoding with the default value "EASList" to control
Replace @altrender with @xhtml:style and add the XHTML namespace to the element using @altrender; include a comment to confirm that @altrender was indeed used to indicate alternative formatting
Rename @lang to @languageOfElement; make sure @languageEncoding exists in control; if it doesn't, add the attribute with the default value "otherLanguageEncoding"
Rename @script to @scriptOfElement; make sure @scriptEncoding exists in control; if it doesn't, add the attribute with the default value "otherScriptEncoding"

General transformation for attributes of elements that are removed:
If a parent element is removed from EAD 4.0, but there is a transformation route for its child elements and if parent and child elements share attributes, check whether the attributes exist with the child element already; if the child elements include the attributes already, remove the attributes of the parent element along with the element itself; if the child elements do not include the attributes already, move the attributes of the parent element to each of its child elements without them

Element in EAD3	**Element in EAD 4.0**	**Transformation route for element and its content (draft)**	**Transformation route for attributes used with the element (draft)**
todate	toDate	Apply camelCasing	Apply camelCasing to @standardDate, @notAfter, @notBefore Apply camelCasing to @localType; create a localTypeDeclaration with @id in control and add the default content "List of local types for single dates" to its sub-element reference; then add @localTypeDeclarationReference to fromDate pointing to the @id of said localTypeDeclaration; if a localTypeDeclaration with this content in its sub-element reference already exists, point to this instead of creating another one
unitdate	unitDate	Apply camelCasing Transform abbr, expan (if existing) into referringString (for details see rows 4, 71) Transform emph, foreign (if existing) into span (for details see rows 65, 76) Transform ptr, ref (if existing) into reference (for details see rows 128, 136) Remove lb Keep only the content of any sub-elements used in container and include a comment about any data potentially encoded in their attributes; values of attributes also available with referringString, span, and reference will of course be moved over to these three elements accordingly	Apply camelCasing to @unitDateType (if used); add @unitDateTypeEncoding with value "EASList" to control (if it doesn't exist already) Apply camelCasing to @dateChar (if used) Move the value of @normal into @standardDate Remove @encodinganalog and include a comment about the removal; if any parent element in EAD3 indicates a relatedencoding according to MARC21, add the MARC21 namespace and use the attribute @marc21:tag in agent to include the value of @encodinganalog (xmlns:marc21="http://www.loc.gov/MARC21/slim" marc21:tag="...") Remove @label and include comment about the removal; an option could be to suggest adding the XHTML namespace and using the attribute @title in abstract to include the value of @label (xmlns:xhtml="http://www.w3.org/1999/xhtml" xhtml:title="...")
unitdatestructured	unitDateStructured	Apply camelCasing Apply camelCasing to the sub-elements dateRange and dateSet Replace sub-element datesingle with date	Apply camelCasing to @unitDateType (if used); add @unitDateTypeEncoding with value "EASList" to control (if it doesn't exist already) Apply camelCasing to @dateChar (if used) Remove @encodinganalog and include a comment about the removal; if any parent element in EAD3 indicates a relatedencoding according to MARC21, add the MARC21 namespace and use the attribute @marc21:tag in agent to include the value of @encodinganalog (xmlns:marc21="http://www.loc.gov/MARC21/slim" marc21:tag="...") Remove @label and include comment about the removal; an option could be to suggest adding the XHTML namespace and using the attribute @title in abstract to include the value of @label (xmlns:xhtml="http://www.w3.org/1999/xhtml" xhtml:title="...")

Note: This sheet shows the changes that would need to be applied to each existing element and attribute in order to move it from EAD3 to EAD 4.0, while the sheets "Elements" and "Attributes" show the changes applied to each element and attribute in terms of its definition within the EAD schema. This sheet here therefore only refers to existing elements, but does not cover elements newly added to EAD 4.0.
Furthermore, this sheet only includes the actual changes. Anything that stays the same (e.g. an element or attribute that isn't renamed or replaced or doesn't have camelCasing applied or when an element retains the same attributes in EAD 4.0 as it had in EAD3), won't be mentioned as such.

General attributes' transformation for all elements that have existed in EAD3 already:
If @audience is used in one or more elements, add @audienceEncoding with the default value "EASList" to control
Replace @altrender with @xhtml:style and add the XHTML namespace to the element using @altrender; include a comment to confirm that @altrender was indeed used to indicate alternative formatting
Rename @lang to @languageOfElement; make sure @languageEncoding exists in control; if it doesn't, add the attribute with the default value "otherLanguageEncoding"
Rename @script to @scriptOfElement; make sure @scriptEncoding exists in control; if it doesn't, add the attribute with the default value "otherScriptEncoding"

General transformation for attributes of elements that are removed:
If a parent element is removed from EAD 4.0, but there is a transformation route for its child elements and if parent and child elements share attributes, check whether the attributes exist with the child element already; if the child elements include the attributes already, remove the attributes of the parent element along with the element itself; if the child elements do not include the attributes already, move the attributes of the parent element to each of its child elements without them

Element in EAD3	**Element in EAD 4.0**	**Transformation route for element and its content (draft)**	**Transformation route for attributes used with the element (draft)**
unitid	unitId	Apply camelCasing Transform abbr, expan (if existing) into referringString (for details see rows 4, 71) Transform emph, foreign (if existing) into span (for details see rows 65, 76) Transform ptr, ref (if existing) into reference (for details see rows 128, 136) Remove lb Keep only the content of any sub-elements used in container and include a comment about any data potentially encoded in their attributes; values of attributes also available with referringString, span, and reference will of course be moved over to these three elements accordingly	Apply camelCasing to @countryCode and @repositoryCode (if existing) Move the value of @identifier into @valueURI If @localtype is used, apply camelCasing to change to @localType; create a localTypeDeclaration with @id in control and add the default content "List of local types for unit identifiers" to its sub-element reference; then add @localTypeDeclarationReference to archDesc pointing to the @id of said localTypeDeclaration; if a localTypeDeclaration with this content in its sub-element reference already exists, point to this instead of creating another one Remove @encodinganalog and include a comment about the removal; if any parent element in EAD3 indicates a relatedencoding according to MARC21, add the MARC21 namespace and use the attribute @marc21:tag in unitId to include the value of @encodinganalog (xmlns:marc21="http://www.loc.gov/MARC21/slim" marc21:tag="...") Remove @label and include comment about the removal; an option could be to suggest adding the XHTML namespace and using the attribute @title in abstract to include the value of @label (xmlns:xhtml="http://www.w3.org/1999/xhtml" xhtml:title="...")
unittitle	unitTitle	Apply camelCasing Transform abbr, expan, persname, corpname, famname, geogname, name, occupation, subject, genreform, function, title, date, footnote, num (if existing) into referringString (for details see rows 4, 48, 52, 71, 72, 75, 78, 79, 80, 105, 108, 110, 119, 153, 160) Transform emph, foreign, quote (if existing) into span (for details see rows 65, 76, 134) Transform ptr, ref (if existing) into reference (for details see rows 128, 136) Remove lb Keep only the content of any sub-elements used in abstract and include a comment about any data potentially encoded in their attributes; values of attributes also available with referringString, span, and reference will of course be moved over to these three elements accordingly	If @localtype is used, apply camelCasing to change to @localType; create a localTypeDeclaration with @id in control and add the default content "List of local types for unit titles" to its sub-element reference; then add @localTypeDeclarationReference to archDesc pointing to the @id of said localTypeDeclaration; if a localTypeDeclaration with this content in its sub-element reference already exists, point to this instead of creating another one Move the value of @normal into @valueURI and check that this doesn't violate the data type of anyURI (if it does, remove @normal and include a comment about the removal) Remove @encodinganalog and include a comment about the removal; if any parent element in EAD3 indicates a relatedencoding according to MARC21, add the MARC21 namespace and use the attribute @marc21:tag in unitTitle to include the value of @encodinganalog (xmlns:marc21="http://www.loc.gov/MARC21/slim" marc21:tag="...") Remove @label and include comment about the removal; an option could be to suggest adding the XHTML namespace and using the attribute @title in abstract to include the value of @label (xmlns:xhtml="http://www.w3.org/1999/xhtml" xhtml:title="...")

Note: This sheet shows the changes that would need to be applied to each existing element and attribute in order to move it from EAD3 to EAD 4.0, while the sheets "Elements" and "Attributes" show the changes applied to each element and attribute in terms of its definition within the EAD schema. This sheet here therefore only refers to existing elements, but does not cover elements newly added to EAD 4.0. Furthermore, this sheet only includes the actual changes. Anything that stays the same (e.g. an element or attribute that isn't renamed or replaced or doesn't have camelCasing applied or when an element retains the same attributes in EAD 4.0 as it had in EAD3), won't be mentioned as such.		**General attributes' transformation for all elements that have existed in EAD3 already:** If @audience is used in one or more elements, add @audienceEncoding with the default value "EASList" to control Replace @altrender with @xhtml:style and add the XHTML namespace to the element using @altrender; include a comment to confirm that @altrender was indeed used to indicate alternative formatting Rename @lang to @languageOfElement; make sure @languageEncoding exists in control; if it doesn't, add the attribute with the default value "otherLanguageEncoding" Rename @script to @scriptOfElement; make sure @scriptEncoding exists in control; if it doesn't, add the attribute with the default value "otherScriptEncoding" **General transformation for attributes of elements that are removed:** If a parent element is removed from EAD 4.0, but there is a transformation route for its child elements and if parent and child elements share attributes, check whether the attributes exist with the child element already; if the child elements include the attributes already, remove the attributes of the parent element along with the element itself; if the child elements do not include the attributes already, move the attributes of the parent element to each of its child elements without them	
Element in EAD3	**Element in EAD 4.0**	**Transformation route for element and its content (draft)**	**Transformation route for attributes used with the element (draft)**
unittype	unitType	Apply camelCasing	Move the value of @identifier into @valueURI and confirm it is valid against the data type anyURI Move the value of @source into @vocabularySource Move the value of @rules into a new conventionDeclaration/reference element in control; add @id to that conventionDeclaration element and @conventionDeclarationReference to unitType pointint to it Remove @encodinganalog and include a comment about the removal; if any parent element in EAD3 indicates a relatedencoding according to MARC21, add the MARC21 namespace and use the attribute @marc21:tag in reference to include the value of @encodinganalog (xmlns:marc21="http://www.loc.gov/MARC21/slim" marc21:tag="...")
userestrict	useConditions	Rename to useConditions If only used with p, keep as is If used with p and any other formatting elements (i.e. blockquote, chronlist, head, list, table), move all into formattingExtension while adapting to XHTML where necessary (see rows 24, 40, 82, 96, 117, 155 for details) If used with nesting, move the sub-elements of the nested userestrict into accessConditions/formattingExtension to recreate the complete structure of the element alongside the sub-elements of the unnested (or parent) userestrict; in that case, even if the unnested (or parent) userestrict is only used with p, those p elements would also be used into useConditions/formattingExtension	If @localtype is used, apply camelCasing to change to @localType; create a localTypeDeclaration with @id in control and add the default content "List of local types for conditions governing use" to its sub-element reference; then add @localTypeDeclarationReference to useConditions pointing to the @id of said localTypeDeclaration; if a localTypeDeclaration with this content in its sub-element reference already exists, point to this instead of creating another one) Remove @encodinganalog and include a comment about the removal; if any parent element in EAD3 indicates a relatedencoding according to MARC21, add the MARC21 namespace and use the attribute @marc21:tag in useConditions to include the value of @encodinganalog (xmlns:marc21="http://www.loc.gov/MARC21/slim" marc21:tag="...")