

Conversion of EAD 1.0 to EAD 2002

Table of Contents

Introduction

What do I need to do?

Notes on the execution command

Conversion Assumptions

Section A: Introduction

This document describes a process for converting EAD instances written according to Version 1.0 of the EAD DTD into ones that conform to Version 2002. This is achieved through the use of an XSLT stylesheet and utilizes a series of files and applications compressed as ead2002conv.exe. The following description consolidates and amplifies information that is found in the readme.txt file included in the zip file and as comments in the accompanying stylesheets. The zip file includes a copy of the XSLT processor Saxon, associated stylesheets, a copy of the EAD2002 DTD, and a script written in the Perl programming language. When it is uncompressed, a series of folders are created into which the various files are placed.

Transformation is performed using an XSLT processor called Instant Saxon, which is supplied in the conversion package. As used here, Saxon is a command line processor. It may be executed either from a DOS prompt or from the Windows Run command which is accessed via the Start button on the desktop's toolbar. Institutions working in other operating systems will have to use the full version of Saxon, compiled for their OS. Saxon also requires the presence of a Java Virtual Machine. More recent versions of Windows (though not XP) and Internet Explorer contain the necessary software. Otherwise, Sun's JDK may be employed.

One can perform the transformation from version 1.0 to version 2002 either directly in Saxon as described below or by using a Perl script which serves as a "wrapper" program for Saxon and which is also contained in the conversion package. Using Saxon directly will be the more straightforward option for those not familiar with Perl though the latter has the advantage that it can convert the contents of an entire directory in a single operation. Saxon converts only one file at a time and, while more time-consuming, this approach actually may be an advantage if one wishes to verify the results of each conversion as it occurs. No directions for the use of Perl are included here beyond those that are common to both processes. Some additional information on Perl is included in the readme.txt file. The transformation process can also be set to produce a report file that describes exactly what actions were taken in the conversion of each EAD file. This report is structured as an HTML document for viewing in a browser.

Section B: What do I need to do?

1. Download and unzip the file ead2002conv.exe. The conversions assume that the software is installed in the directory c:\ead2002conv. If you install the software in another directory, be sure to alter the path statements in the execution command described in step 3 below. You will also need to change the path statement for the destination of the conversion report file.
2. The software assumes that the directory v1 will be the location of version 1.0 source files and that v2002 will be the location of version 2002 result files.
3. Review the assumptions about the conversion process that are described in the following section to ensure that the changes that are occurring as part of the conversion conform to institutional expectations.
4. Verify the character set encoding that you have used in your source document and the encoding you wish in the resulting file. The stylesheet uses ISO 8859-1 as the default output.
5. Transform the EAD files.

Type a variation of the following command, substituting the proper path statements and file name, on the command line in DOS or within the Run box in Windows. There are five parts to the command.

The directory location of conversion software
The location and name of the output file
The location and name of the input file
The location and name of the stylesheet
Any variable parameters you wish to specify. Options are listed under Notes below.

Generalized, the command follows this syntax:

```
LocationOfTheConversionSoftware  
-o PathToTheOutputFile\OutfileFileName.xml  
PathToTheSourceFile\SourceFileName.xml  
PathToTheStylesheet\StylesheetName.xsl  
parameterName=parameter value
```

A typical command in Run might read:

```
c:\ead2002conv\bin\saxon -o h:\ead\2468.xml h:\ead2002\2468.xml  
isoconvdate=2003-11-19 convdate November 11, 2003 mainagencycode=Mnhi  
reportpath=h:\conversion.html
```

The same command from the DOS prompt would read:

```
c:\ead2002conv>bin\saxon -o h:\ead\2468.xml h:ead2002\2468.xml
isoconvdate=2003-11-19 convdate November 11, 2003 mainagencycode=Mnhi
reportpath=h:\conversion.html
```

6. After executing this command, review the conversion report for any anomalies or actions that do not conform to your desired output
7. Edit the resulting EAD file as necessary in your favorite XML editor. In particular, you will probably have to edit the <eadid> element for reasons that are described in the next section.
8. Parse the resulting EAD instance against the EAD2002 DTD to ensure that the resulting file is valid.

Section C: Notes on the Execution Command:

1. The example of an execution command in step 3 uses the Saxon application in the directory c:\ead2002conv\bin to convert the file 2468.xml in the folder h:\ead into a new file with the same name, 2468.xml, and places it in the directory h:\ead2002. In the resulting file, the maintenance agency code in <eadid> is set to Mnhi for the Minnesota Historical Society and the date of the conversion is reported as text in the YYYY element and as an ISO code in its normal attribute. The expression "-o" is a Saxon parameter statement that indicates that the following value is the location and/or name of the output file. Follow the syntax of the example above. There is an extra \ in the examples in the readme.txt file.
2. The unpacking of the file ead2002.exe automatically creates default directories for version 1.0 source files and version 2002 result files. The default folders are named "v1" and "v2002". Other source and result locations may be employed provided the appropriate path statements are spelled out in the execution command. During the transformation process, Saxon will attempt to locate the copy of the EAD version 1.0 DTD at the address given in the system id portion (that which falls within the last set of quotation marks) of the DOCTYPE declaration of your version 1.0 files. If it fails to find the DTD, the transformation will fail immediately. If you are placing your version 1.0 files in the folder "v1", you may need to adjust the system id statement in your EAD files accordingly or include a copy of ead.dtd (and eadbase.ent) there, whether the path to ead.dtd is relative or absolute.
3. The XSLT stylesheet specifies that the location of the conversion report is c:\ead2002conv\doc. The zip file automatically creates this directory. If you wish to direct the report elsewhere, you will either have to change the path in the XSLT stylesheet (hint: it is on line 86) or specify the location in a parameter as described in the section on

parameters below. Of course, you can also use a parameter to disable the generation of the report completely.

4. Certain parameter statements in the stylesheet determine various aspects of the transformation process. These are documented in the stylesheet itself and are summarized below. One can override the default values either by editing the stylesheet or including overriding statements at the end of the execution command, as shown in the examples under step 3 above. The parameters are:

countrycode

The stylesheet default added to <eadid> as @countrycode is "us". Use ISO 3166-1 values.

mainagencycode

The stylesheet default added to <eadid> as @mainagencycode is "ctY". (Guess where Stephen Yearl works.) Use ISO 15511 values when replacing this value which you will want to do unless you too work at Yale.

convdate

The stylesheet default value for the date of the transformation, as added to <revisondesc><change><date>, is "July 17, 2003"

isoconvdate

The stylesheet default value for date, expressed as an ISO code, as added to @normal in <revisondesc> <change><date>, is "2003-07-16".

docname

This is the file name for the conversion report. The stylesheet default value is "conversion". The path to this file is set in the stylesheet as directoryname:\ead2002conv\doc. Either the file name may be overridden from the command line here. The rest of the path statement may be overridden from reportpath.

dtdpath

This parameter sets the value that is included in the system identifier within the DOCTYPE declaration of new EAD 2002 file as the location for the file ead.dtd. The stylesheet default is /dtds/ead.dtd which assumes that the ead.dtd file is located in a directory called "dtds" that resides beneath the directory where the EAD files is kept. If this does not match your file structure, you will need to change the statement.

report

This parameter determines whether or not a report of the conversion process is produced. The default value in the stylesheet is "y".

reportpath

This parameter defines the location of the conversion report. The stylesheet default has several parts: the name of the drive where this software was installed, a path statement

"\lead2002conv\doc\", the value of the docname parameter, with the extension ".report.html".

If the software was installed in the directory H: and the default value for docname is retained, the default reportpath would be

h:\lead2002conv\doc\conversion.report.html

bundle

The wrapper elements <admininfo> and <add> have been deprecated in EAD2002. To use them, one needs to adjust the DTD file itself. This parameter determines what will be done with these elements during conversion. There are two possibilities: replace <admininfo> and <add> with the generic wrapper <descgrp> or remove the wrapper elements completely. This parameter indicates whether to bundle them into <descgrp> or not. The stylesheet default value is "n".

You must understand, however, that even if this parameter value is set to "n", these elements may be bundled as described in the discussion of <add> and <admininfo> in the Conversion Assumptions section below.

langlang

This parameter determines what happens as a result of the presence of the langmaterial attribute in <archdesc> in version 1.0 documents. The conversion stylesheet creates a new element in the EAD 20002 document, thusly.

```
<langmaterial>  
  <language langcode="value">value</language>  
</langmaterial>
```

The coded value of @langmaterial is written out in the <language> element according to a conversion table in the file iso639-2.xml and the value of @langmaterial is copied into its langcode attribute.

The stylesheet default value is "eng" which produces the following output.

```
<langmaterial>  
  <language langcode="eng">English</language>  
</langmaterial>
```

converter

This parameter defines the name of the conversion stylesheet. The default value is "v1to02.xsl"

Section D: Conversion Assumptions

The Stylesheet makes various assumptions during the conversion process based on changes made in the 2002 version of EAD. These are spelled out below.

OBSOLETE:

Elements and attributes NOT available in EAD 2002.

elements

<spanspec>

v1to02.xsl eliminates this element from both <tspec> and <tgroup>

<tfoot>

v1to02.xsl writes text in a <tfoot> to <row altrender="tfoot">

attributes

behavior, content-role, content-title, extent, inline, orient, pubstatus, rotate, shortentry, spanname, tabstyle, tgroupstyle, tocentry, xlink:form

v1to02.xsl eliminates these attribute values

numbered

converts to altrender="numbered" or altrender="unnumbered"

othersource

converts to source="value_of_othersource"

systemid

converts to <eadid>System ID=value_of_systemid</eadid>

targettype (not actually in EAD v1.0)

DEPRECATED:

Elements and attributes strongly recommended not to be used, and are not permitted by default. Such elements may only be made allowable by modifying the EAD 2002 DTD.

elements

<add>

<admininfo>

If the "bundle" parameter is set to "y", as described above, the two elements are not removed or changed.

If the "bundle" parameter is set to "n", which is the default value as described above, any of several actions may occur depending on the structure of the element as described below. The conversion process for both <add> and <admininfo> follows the same pattern in the scenarios described below.

A review of the content model of these elements follows as it is necessary to understand the relationship between their data structure and the conversion process.

Both Administrative Information <admininfo> and Adjunct Descriptive Data <add> have been deprecated in EAD 2002. While the DTD may be modified to revalidate them, the EAD Working Group discourages this and makes no promise that they will be available in future versions of the DTD. Now is the time to make the transition. Where it is absolutely necessary to bundle groups of elements together, the DTD now provides a generic wrapper element, Description Group <descgrp>. Before adopting this element wholesale, a repository should seriously consider the practical value of adding this additional element.

In EAD version 1.0, both <add> and <admininfo> may contain three types of child elements, i.e. those nested within it. In addition to the parent element (add or admininfo), there may be a head, the generic block elements (address, blockquote, chronlist, list, note, paragraph, and table), and certain descriptive elements.

For <add>, the descriptive elements are add, bibliography, fileplan, index, otherfindaid, relatedmaterial, and separated material.

For <admininfo>, the descriptive elements are accessrestrict, accruals, acqinfo, admininfo, altformavailable, appraisal, custodhist, prefercite, processinfo, and userrestrict.

The conversion algorithm may produce any of several outputs depending on the structure of the parent element.

Output 1: If the parent element contains block elements but no descriptive elements, the parent element is converted to <odd> and the name of the parent element is added as an attribute, e.g. <odd type="add">.

Output 2: If the parent element contains a single descriptive element and no block elements, the parent element and its head element are removed. Any head element associated with the descriptive element remains.

Output 3: If the parent element contains a head element and more than one descriptive element, the parent element is converted to <descgrp> and the name of the parent element is added as an attribute, e.g. <descgrp type="add">.

Output 4: If the parent element lacks a head element and contains more than one descriptive element, the parent element is removed.

Output 5: If the parent element is used recursively as a descriptive element and the parent has a head, the parent element is removed and descriptive element is converted to <odd> and the name of the descriptive element is added as an attribute, e.g. <odd type="add">. NOTE: The stylesheet does not distinguish between the situation where <add> or <adminifo> is used as a parent element and when it appears recursively as its own child. This is generally not a problem except for the next and somewhat obscure scenario.

Output 6: If the parent element is used recursively as a descriptive element but lacks a head, the stylesheet removes the parent element completely resulting in an invalid document.

<dentry>
 content written straight to the <did> or other component child element
<drow>
 eliminated.
<organization>
 converted to <arrangement>
<tspec>
 eliminated

attributes

langmaterial
 converts to <langmaterial><language langcode="ISO639-2b_code">language_name (in French or English)</language>

legalstatus
 converts to, e.g. <legalstatus type="public">Public</legalstatus>

otherlegalstatus
 converts to <legalstatus type="my_other_legalstatus">my other legalstatus</legalstatus>