

HRA – Visual Resources | Tutorial

Manual for VRA Core 4 Transform tool (DRAFT)

Contents

1. Introduction and project history	3
2. Data preparation	4
2.1. Controlled data	4
2.2. Data sources	5
3. Accessing the tool online	6
4. Uploading a file	7
5. Generating VRA Core 4 XML	8
5.1. Select a mapping	8
5.2. Select a transformation	8
5.3. Define the lines to be processed	8
5.4. Applied XSLs	8
5.5. Generating XML	9
6. Validating the XML	10
6.1. Choosing a schema for validation	10
6.2. Starting the validation	10
6.3. Getting the validation result	10
6.4. Interpreting the validation	11
6.5. Validation errors	11
7. Downloading the generated XML	12
8. Generating RDF XML	13
9. Contact	14
10. Appendix: Elements, displays and repetitions	15
11. Appendix: Columns Full Template	17
12. Appendix: Convert .xlsx file to .csv using OpenOffice Calc	19

1. Introduction and project history

The *VRA Core 4 XML Transform Tool* converts descriptive image metadata from flat tables (.csv) to structured VRA Core 4 XML. It makes use of a template with predefined headers. Users may work with the tool in a simple or advanced mode. The transform tool also offers XML validation and provides feedback in case of errors.

This manual covers whole the workflow of using the *VRA Core 4 XML Transform Tool* from preparation of data via uploading data, adjusting usage settings, data transformation, data validation and handling validation feedback throughout to downloading the XML files. It also covers the (experimental) generation of RDF XML and includes a number of appendices for further reference.

Development of the *VRA Core 4 XML Transform Tool* was based on an initiative by Susan Jane Williams and resulted in a successful application by Matthias Arnold for a [Project Grant](#) funded by the [Visual Resources Association Foundation](#) (VRAF) in Fall 2014. Additional support was given by the [Heidelberg Research Architecture](#) (HRA) at the [Cluster of Excellence "Asia and Europe in a Global Context"](#), Heidelberg University.

"Matthias Arnold, Heidelberg Research Architecture, University of Heidelberg, will use the award to support the development of the VRA Core 4 XML Transform Tool. This tool will enable any user who can supply descriptive image metadata in a standardized CSV form (comma separated values, e.g. via Excel) to create validating VRA Core 4 XML. Useful on its own, this XML also represents an important step for further transformations to other XML schemas, like RDF to support output of data as LOD (Linked Open Data). The tool is being developed in consultation with both the VRA Data Standards and the Core OC Committees, will be shared with the image management community during beta-testing for feedback, and will be open-source and freely available upon completion. Susan Jane Williams (Independent Cataloging and Consulting Services) will serve as coordinator for the mapping templates and will help create documentation, demonstrations (sample records and mappings), and further mapping help in use of the tool. We are confident that this tool will have broad value and impact within the visual resources community, and for the VRA Core 4 standard."

["2014-2015 VRA Foundation Project Grant Winners Named"](#), VRAF News & Events, December 19, 2014

The template csv files and the mappings to XML were developed by Susan Jane Williams, Greg Reser, and Matthias Arnold. Implementation was coordinated by Heidelberg, programming was conducted by HRA's senior developer Matthias Guth.

Please note:

The tool is still in BETA version, currently v.0.0.4. Any feedback on issues, bugs or problems encountered is very much appreciated and should be directed to the developers (see section "Contact").

The VRA Core 4 XML Transform Tool is developed as Open Source, its source code can be found at: <https://github.com/exc-asia-and-europe/csv2xml>

2. Data preparation

You need to prepare your data for transformation. Export records from your local system into spreadsheets or directly into .csv files.

Data transformation is based on templates. The **full template** is available at:

<https://github.com/exc-asia-and-europe/csv2xml/tree/master/doc>

The template uses predefined headers for each column to process data.

The sequence of columns (headers) may be changed and you can even delete columns you will not need. It is, however, essential **not to change the headers** themselves, i.e. not their "text" or "values". Otherwise the transformation will not work.

There is also an Excel (.xlsx) version of the template available. However, only .csv templates can be transformed so make sure you convert spreadsheets (e.g. from Excel or Calc or Google Docs) to .csv text files before transformation. Use UTF-8 as encoding to conserve possible special characters in your data.

For a guide on how to convert Microsoft Excel files into comma separated value (.csv) files see "Appendix: Converting .xlsx file to .csv using OpenOffice Calc" below.

2.1. Controlled data

[VRA Core 4.0](#) in the restricted version uses controlled type lists and date formats. For the **type values** please refer to the [VRA Core 4.0 Restricted Schema Type Values](#) document.

For **date values** there is no pdf file, but the [vra-strict.xsd](#) states for dateValueType:

"Defines a date which follows the ISO 8601 date format, and allows right truncation. [...] In brief, the following formats are allowed:

present	
2006	(2006)
2006-12	(December, 2006)
2006-12-31	(31 December 2006)
-44	(44 BCE)
-44-03	(March, 44 BCE)
-44-03-15	(15 March, 44 BCE)
-10000000	(10 Million Years Ago)
-100000000000	(100 Billion Years Ago)"

To be able to convert your data into valid VRA Core 4 XML these type values must be used and date format rules must be followed.

The transform tool will assist you as far as possible.

For example, in <date> the sub-element <latestDate> must not be empty. But if you only entered <earliestDate> the tool will automatically fill <latestDate> with the <earliestDate> value.

If you accidentally mistyped a controlled type value, the validation will point you to this error in the validation results (see chapter "Validation errors" below).



2.2. Data sources

Any data can be used for transformation, be it an export from your institution's Digital Asset Management System, embedded metadata, or a personal image database. You only need to do two things for preparation:

1. Use the predefined template headers
2. Have the data exported as flat comma separated value (.csv) file

2.2.1. *Using the predefined template headers*

The transform tool uses predefined headers to identify data in the template and write the values to the correct elements or attributes in the XML. The template offers 224 columns for your data (see the complete list in "Appendix: Columns Full Template" below). You do not need to fill each column with data.

To be covered:

Explanation of colour codes in template

Other data sources: EMWG export-import tool



3. Accessing the tool online

The transform tool can be accessed online for testing at

<http://kjc-ws2.kjc.uni-heidelberg.de:8650/exist/apps/csv2xml/index.xq>

Please note this is a server dedicated to testing software and developments.

The screenshot shows the 'Simple mode' of the CSV2XML tool. At the top left is a 'Reset' button. Below it is the 'Upload CSV' section, which includes a 'Browse...' button, the text 'No file selected.', and an 'Upload' button. To the right of this section is a checkbox labeled 'advanced Mode' which is currently unchecked. Below the upload section, it says '0 lines loaded'. There are two dropdown menus: 'mapping: VRA strict' and 'transform: Unmodified VRA Core 4 XML'. Below these is a 'process lines' section with input fields for '1' and '0' separated by 'to'. At the bottom of this section are two buttons: 'generate XML' and 'download'. The footer of the interface shows the number '11'.

Figure 1: User interface - Simple mode.

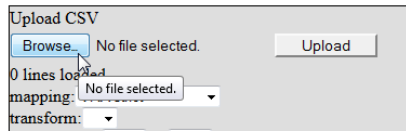
The screenshot shows the 'Advanced mode' of the CSV2XML tool. It has the same top section as the simple mode, but the 'advanced Mode' checkbox is now checked. Below the 'process lines' section, there is a new section titled 'Apply XSLs (you will have to generate the XML again):' followed by a large empty text area. Below this is a 'generate XML' button. The next section is 'Validate against:', followed by a list of three URLs: 'http://www.loc.gov/standards/vracore/vra-strict.xsd', 'http://www.loc.gov/standards/vracore/vra.xsd', and 'http://localhost:8080/exist/apps/csv2xml/mappings/default/xsd/vocab.xsd'. Below the list is an 'add Schema (.xsd .dtd):' section with an empty input field. At the bottom of this section are 'validate' and 'download' buttons. The footer of the interface shows the number '11'.

Figure 2: User interface - Advanced mode.

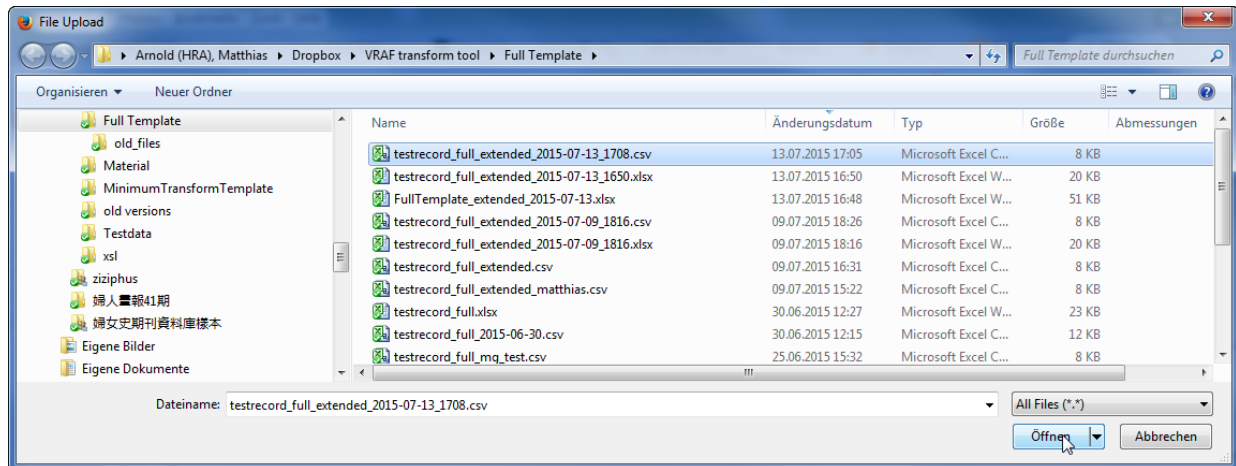


4. Uploading a file

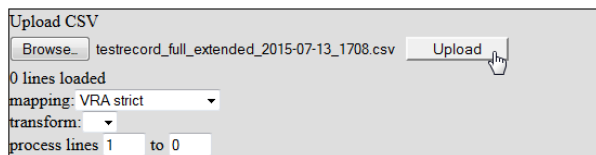
To transform data you need to upload it first. Click the "Browse" button:



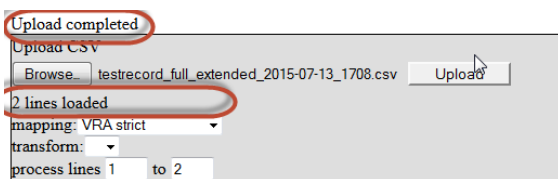
...then select the .csv file on your computer and click "Open".



The name of the .csv file will be displayed. Now click on "Upload".



You will get an "Upload completed" message. Note that the number of lines in your csv will be shown in "x lines loaded" (in this example, the .csv only contains two lines).

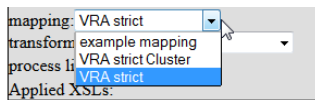




5. Generating VRA Core 4 XML

5.1. Select a mapping

The tool supports generation of XML based on different mappings. For VRA Core, two variants are available: "VRA strict" and "VRA strict Cluster".



5.1.1. *VRA strict*

This is the **default mapping**. It is based on the VRA Core 4 restricted version (<http://www.loc.gov/standards/vracore/vra-strict.xsd>). It uses controlled values for types and date formats.

For more information please refer to <http://www.loc.gov/standards/vracore/schemas.html>.

5.1.2. *4.1.2. VRA strict Cluster*

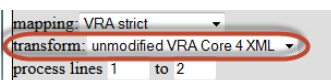
This mapping is an expansion of the restricted VRA Core 4 version. It adds, for example, attributes for multilingual data and an element for geo-coordinates (<http://cluster-schemas.uni-hd.de/vra-strictCluster.xsd>).

For more information please refer to the document "[VRA Core – Extensions](#)".

5.2. Select a transformation

The tool supports different transformations. Additional transformations need to be installed. Currently, two transformations are implemented: "unmodified VRA Core 4 XML" and "transform into RDF" (for RDF transformations see below).

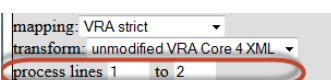
To transform the .csv data into VRA Core 4 XML choose "unmodified VRA Core 4 XML".



5.3. Define the lines to be processed

In the "process lines" fields you can enter the number of the lines that will be processed. This feature allows to process smaller number of records or even individual ones and reduces processing time.

Per default, all lines will be processed.



5.4. Applied XSLs

For the default transformation to VRA Core 4 XML this field will remain empty.

(See also "Generating RDF" below.)

5.5. Generating XML

If all settings are made, click the "generate XML" button.

Upload CSV
 testrecord_full_extended_2015-07-13_1708.csv
2 lines loaded
mapping: VRA strict
transform: unmodified VRA Core 4 XML
process lines 1 to 2
Applied XSLs:

Validate against:
<http://www.loc.gov/standards/vracore/vra-strict.xsd>
<http://www.loc.gov/standards/vracore/vra.xsd>
<http://kjc-sv016.kjc.uni-heidelberg.de:8080/exist/apps/tamboti/resources/schemas/vra-strictCluster.xsd>
<http://kjc-sv016.kjc.uni-heidelberg.de:8080/exist/apps/tamboti/resources/schemas/vraCluster.xsd>
<http://kjc-ws111.kjc.uni-heidelberg.de:8080/exist/apps/csv2xml/mappings/default/xsd/vocab.xsd>
add Schema (.xsd .dtd):

Result:

Note: Processing a large number of records may take some time, please be patient.

Once the XML is generated, a preview will be displayed in the "Result" box.

Result:

```

1. <?xml version="1.0" encoding="UTF-8" standalone="yes" ?>
2. <!-- changelog
3. mapping templates: 2015-02-19 ma; 2015-02-23 ma; 2015-02-25 ma; 2015-02-26 ma; 2015-02-27 ma; 2015-05-19 ma; 2015-05-20 ma; 2015-05-21 ma; 2015-05-22 ma; 2015-05-29 ma; 20
4. csv-map: 2015-06-02 ma; 2015-06-05 ma; 2015-06-26 ma; 2015-06-30 ma; 2015-07-09 ma
5. -->
6. <work id="w_5a24c552-3358-40b5-8121-1458973bb978">
7. <agentSet>
8. <display>attributed to Chiang, Kai-shek (literati (painters)); my-attribution Cranach, Lucas (patrons); WORK_Agent3Attribution WORK_Agent3Name (WORK_Agent3Role)</display>
9. <agent>
10. <name type="personal" vocab="vial" refid="99276871">Chiang, Kai-shek</name>
11. <role>literati (painters)</role>
12. <attribution>attributed to</attribution>
13. </agent>
14. <agent>
15. <name type="personal" vocab="vial" refid="49268177">Cranach, Lucas</name>
16. <role>patrons</role>
17. <attribution>my-attribution</attribution>
18. </agent>
19. <agent>
20. <name type="other" vocab="WORK_Agent3NameVocab" refid="WORK_Agent3NameRefid">WORK_Agent3Name</name>
21. <role>WORK_Agent3Role</role>
22. <attribution>WORK_Agent3Attribution</attribution>
23. </agent>
24. </agentSet>
25. <contextSet>
26. <display>Chinese (culture or style); German Renaissance-Baroque styles</display>
27. <context vocab="aat" refid="300018922">Chinese (culture or style)</context>
28. <context vocab="aat" refid="300021096">German Renaissance-Baroque styles</context>
29. </contextSet>
30. <dateSet>
31. <display>commission 1550 - 1550; creation 1552 - 1553</display>
32. <date type="commission">
33. <earliestDate>1550</earliestDate>
34. <latestDate>1550</latestDate>
35. </date>
36. <date type="creation">
37. <earliestDate>1552</earliestDate>
38. <latestDate>1553</latestDate>
39. </date>
40. </dateSet>
41. <descriptionSet>
42. <display></display>
43. <description source="my-description-source"></description>
44. </descriptionSet>
45. <inscriptionSet>
46. <display>my-inscription-display</display>
47. </inscriptionSet>

```



6. Validating the XML

Just by applying a transformation the tool cannot be sure the resulting xml files are valid. The restricted VRA Core schema defines a number of values. If the values in the .csv document do not match the values expected by the schema, validation will fail. We therefore strongly recommend validating the generated data.

6.1. Choosing a schema for validation

In the "Validate against" box you can choose a schema for validation. Depending on the selected mapping (see above) the respective default validation schema is automatically set.

Validate against:

- <http://www.loc.gov/standards/vracore/vra-strict.xsd>
- <http://www.loc.gov/standards/vracore/vra.xsd>
- <http://kjc-sv016.kjc.uni-heidelberg.de:8080/exist/apps/tamboti/resources/schemas/vra-strictCluster.xsd>
- <http://kjc-sv016.kjc.uni-heidelberg.de:8080/exist/apps/tamboti/resources/schemas/vraCluster.xsd>
- <http://kjc-ws111.kjc.uni-heidelberg.de:8080/exist/apps/csv2xml/mappings/default/xsd/vocab.xsd>

add Schema (.xsd .dtd):

For example, if you selected "VRA strict" as mapping, the corresponding schema <http://www.loc.gov/standards/vracore/vra-strict.xsd> is automatically set.

It is possible to choose different validation schemas. Users may also upload their own schema (ALPHA!).

6.2. Starting the validation

To start the validation process, click "validate".

Validate against:

- <http://www.loc.gov/standards/vracore/vra-strict.xsd>
- <http://www.loc.gov/standards/vracore/vra.xsd>
- <http://kjc-sv016.kjc.uni-heidelberg.de:8080/exist/apps/tamboti/resources/schemas/vra-strictCluster.xsd>
- <http://kjc-sv016.kjc.uni-heidelberg.de:8080/exist/apps/tamboti/resources/schemas/vraCluster.xsd>
- <http://kjc-ws111.kjc.uni-heidelberg.de:8080/exist/apps/csv2xml/mappings/default/xsd/vocab.xsd>

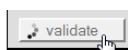
add Schema (.xsd .dtd):

validate download

Result:

```
1. <vra xmlns="http://www.vrweb.org/vracore4.htm" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="http://www.vrweb.org/vracore4.htm http://www.loc.gov/standards/vracore/vra-strict.xsd">
2.   <!-- changelog
3.   mapping templates: 2015-02-19 ma; 2015-02-23 ma; 2015-02-25 ma; 2015-02-26 ma; 2015-02-27 ma; 2015-05-19 ma; 2015-05-20 ma; 2015-05-21 ma; 2015-05-22 ma; 2015-05-29 ma; 20
4.   csv-map: 2015-06-02 ma; 2015-06-09 ma; 2015-06-26 ma; 2015-06-30 ma; 2015-07-09 ma
5.   -->
6.   <work id="w_e190f1c9-03f1-4493-b2bf-f332e57df276">
7.     <agentSet>
8.       <display>WORK_AgentDisplay</display>
9.       <agent>
10.        <name type="WORK_AgentNameType" vocab="WORK_AgentNameVocab" refid="WORK_AgentNameRefid">WORK_AgentName</name>
11.        <role>WORK_AgentRole</role>
12.        <attribution>WORK_AgentAttribution</attribution>
13.      </agent>
14.    </agentSet>
15.  </work>
16. </vra>
```

Note: Validating a large XML file may take some time, please be patient.



6.3. Getting the validation result

Once the validation is finished, a new browser tab will be opened with the validation result. In addition, the button's color will change:



validate

Result NOT VALID

validate

Result VALID

6.4. Interpreting the validation

A valid result will look like this:

```
1. <report-result>
2.   <xsd:http://www.loc.gov/standards/vracore/vra-strict.xsd/xsd>
3.   <report>
4.     <status>valid</status>
5.     <namespace>http://www.vrweb.org/vracore4.htm</namespace>
6.     <duration unit="sec">17</duration>
7.   </report>
8. </report-result>
```

6.5. Validation errors

In case of problems with the validation, the error messages will be displayed in a separate tab. Errors indicate both line and column of the respective generated XML document. This is displayed in the "Result" box.

Example 1:

```
<report-result>
<xsd:http://kjc-uv016.kjc.uni-heidelberg.de:8080/exist/apps/vamboti/resources/schemas/vra-strictCluster.xsd/xsd>
<report>
  <status>invalid</status>
  <namespace>http://www.vrweb.org/vracore4.htm</namespace>
  <duration unit="sec">26</duration>
  <message level="Error" line="587" column="24">cvc-pattern-valid: Value 'early1' is not facet-valid with respect to pattern 'present|(-)*[0-9]{1,12}(-[0-9]{2})(-[0-9]{2})*)*' for type 'dateValueType'.</message>
  <message level="Error" line="587" column="24">cvc-complex-type.2.2: Element 'date' must have no element [children], and the value must be valid.</message>
</report>
```

The error message has two lines, each pointing to the same line/column in the generated XML file (here line="587" column="24"):

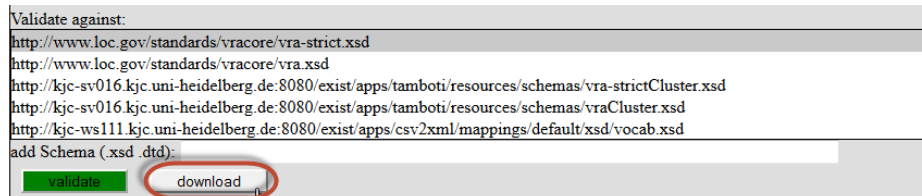
```
<message level="Error" line="587" column="24">cvc-pattern-valid: Value 'early1' is not facet-valid with respect to pattern 'present|(-)*[0-9]{1,12}(-[0-9]{2})(-[0-9]{2})*)*' for type 'dateValueType'.</message>
<message level="Error" line="587" column="24">cvc-complex-type.2.2: Element 'date' must have no element [children], and the value must be valid.</message>
```

In this case the column "WORK_EarliestDate1" contained the invalid "early1" value, therefore the subelement <earliestDate> did not validate. The second line indicates that because of the in-valid element <earliestDate> its parent element <date> also does not validate. The message contains the error message of element <date>.



7. Downloading the generated XML

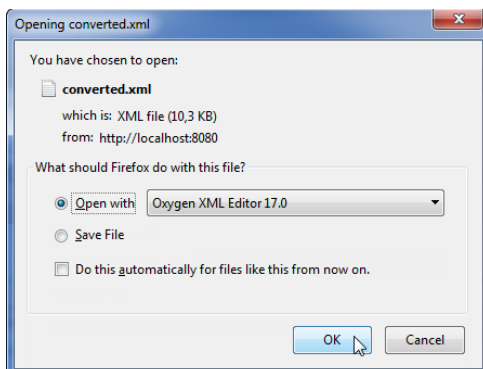
After successful validation click the "download" button to copy the valid XML to your computer.



Result:

```
1. <vra xmlns="http://www.vraweb.org/vracore4.htm" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="http://www.vra
2. <!-- changelog
3. mapping templates: 2015-02-19 ma; 2015-02-23 ma; 2015-02-25 ma; 2015-02-26 ma; 2015-02-27 ma; 2015-05-19 ma; 2015-05-20 ma; 201
4. csv-map: 2015-06-02 ma; 2015-06-05 ma; 2015-06-26 ma; 2015-06-30 ma; 2015-07-09 ma
5. -->
6. <work id="w_608941cb-317e-4743-818d-5a1458a72238">
7. <agentSet>
8. <display>attributed to Chiang, Kai-shek (literati (painters)); my-attribution Cranach, Lucas (patrons); WORK_Agent3Attribution
9. <agent>
10. <name type="personal" vocab="viaf" refid="99276871">Chiang, Kai-shek</name>
11. <role>literati (painters)</role>
12. <attribution>attributed to</attribution>
13. </agent>
```

The default file name will be "converted.xml".



Display of the generated file in Oxygen.





8. Generating RDF XML

In addition to generating VRA Core 4 XML it is possible to transform the data to RDF. At the moment, this feature is still experimental.

The tool makes use of the XSLT stylesheet as provided by the VRA-RDF-Project <https://github.com/mixerj/VRA-RDF-Project>.

To transform your data to RDF first make sure the .csv data will validate in VRA Core.

Then go back and select "transform into RDF" in the "transform" line. Note that the entry "xsl/VRA2RDF.xsl" appears in the "Applied XSL's" box.

Upload CSV
Browse... testrecord_full_extended_2015-07-13_1708.csv Upload
2 lines loaded
mapping: VRA strict
transform: transform into RDF
process lines 2 to 2
Applied XSLs:
xsl/VRA2RDF.xsl
generate XML
Validate against:

Now click "generate XML".

Once generated, the RDF XML will be displayed in the "Result" box and can be downloaded.

Result:

```
1. <rdf:RDF xmlns:library="http://purl.org/library/" xmlns:void="http://rdfs.org/ns/void#" xmlns:vra="http://purl.org/vra/" xmlns:dcterms="http://purl.org/dc/terms/" xmlns:rdfs="http://www.w3.org/1999/02/22-rdf-syntax-ns#" xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#" >
2.   <!--Skipping over-->
3.   <!-->
4.   <rdf:Description rdf:about="#">
5.     <rdf:type rdf:resource="http://purl.org/vra/CreativeWork"/></rdf:type><!--Skipping over-->
6.     <!--Skipping over agentSet--><!--Skipping over-->
7.     <!--Skipping over display--><!--Skipping over attributed to Chiang, Kai-shek (literati (painters)); my-attribution Cranach, Lucas (patrons); WORK_Agent3Attribution WORK_Agent3Name-->
8.     <!-->
9.     <vra:creator>
10.       <rdf:Description>
11.         <rdf:type rdf:resource="http://purl.org/vra/Agent"/></rdf:type>
12.         <rdf:type rdf:resource="http://purl.org/vra/Person"/></rdf:type>
13.         <vra:name>Chiang, Kai-shek</vra:name>
14.       </rdf:Description>
15.     </vra:creator><!--Skipping over-->
16.     <!-->
17.     <vra:creator>
18.       <rdf:Description>
19.         <rdf:type rdf:resource="http://purl.org/vra/Agent"/></rdf:type>
20.         <rdf:type rdf:resource="http://purl.org/vra/Person"/></rdf:type>
21.         <vra:name>Cranach, Lucas</vra:name>
22.       </rdf:Description>
23.     </vra:creator><!--Skipping over-->
24.     <!-->
25.     <vra:creator>
26.       <rdf:Description>
27.         <rdf:type rdf:resource="http://purl.org/vra/Agent"/></rdf:type>
28.         <vra:name>WORK_Agent3Name</vra:name>
29.       </rdf:Description>
30.     </vra:creator><!--Skipping over-->
31.     <!--Skipping over-->
32.     <!--Skipping over culturalContextSet--><!--Skipping over-->
33.     <!--Skipping over display--><!--Skipping over Chinese (culture or style); German Renaissance-Baroque styles--><!--Skipping over-->
34.     <!-->
35.     <vra:culturalContext>
36.       <rdf:Description>
37.         <vra:name>Chinese (culture or style)</vra:name>
38.       </rdf:Description>
39.     </vra:culturalContext>
40.   </rdf:Description>
41. </rdf:RDF>
```



9. Contact

If you have problems or questions please contact Matthias Arnold at marnold@asia-europe.uni-heidelberg.de

Contact details

arnold@asia-europe.uni-heidelberg.de

Phone: +49 (0) 6221 - 54 4094

Fax: +49 (0) 6221 - 54 4012

Skype: matz-skype

Office address

Karl Jaspers Centre, Room 005b

MediaLab

Karl Jaspers Centre, Room 005c

Further links

HRA Portal

<http://hra.uni-hd.de/>

MediaLab Tutorials in Sharepoint

<https://sharepoint.urz.uni-heidelberg.de/vjc/kjc/hra/medialab/>

10. Appendix: Elements, displays and repetitions

WORK

[workID]

Agent display

3x nameType – name – nameVocab – nameRefid – role

CulturalContext display

2x culturalContext – vocab – refid

Date display

2x type – earliestDate – latestDate

Description display [display = description]

descriptionSource

Inscription display [display = text]

Location display

Location notes

2x type – Name – NameType – NameVocab – NameRefID – ObjType – ObjRefID – Geo –
GeoVocab – GeoRefid

Material display

4x material – vocab – refid

Measurements display

2x extent – unit – value1-4 – type1-4

Relation display

2x relatedWork – type

[transform adds work-image link]

Rights display [display = text]

Source display

StateEdition display

Style/Period display

4x stylePeriod – vocab – refid

Subject display

8x subject – vocab – refid – type

Technique display

4x technique – vocab – refid

TextRef display

Title display

titlePref, titlePrefType, titleAlt, titleAltType

Worktype display

3x worktype – vocab – refid

IMAGE

[Filename (-> href); accessionID (-> refid)]

Agent display

1x nameType – name – nameVocab – nameRefid – role

CulturalContext display

Date display

Description display [display = description]

Inscription display [display = text]

Location display

[localRepoName, accession-id]

Material display

Measurements display

Relation display

[transform adds image-work link]

Rights display

Rights notes

1x type – holder – text

Source display

[VendorName, VendorID]

1x sourceType, sourceValue

StateEdition display

Style/Period display

Subject display

3x subject – vocab – refid – type

Technique display

[Technique: <technique vocab="AAT" refid="300237903">digital imaging</technique>]

TextRef display

Title display

1x type, title

WorkType display

[Worktype: <worktype vocab="AAT" refid="300215302">digital images</worktype>]

11. Appendix: Columns Full Template

IMAGE_Filename	WORK_Location1GeoVocab
IMAGE_AccessionID	WORK_Location1GeoRefid
IMAGE_VendorID	WORK_Location2Type
IMAGE_VendorName	WORK_Location2Name
LOCAL-REPO-NAME	WORK_Location2NameType
WORK_ID	WORK_Location2NameVocab
WORK_TitleDisplay	WORK_Location2NameRefID
WORK_TitlePref	WORK_Location2ObjType
WORK_TitlePrefType	WORK_Location2ObjRefID
WORK_TitleAlt	WORK_Location2Geo
WORK_TitleAltType	WORK_Location2GeoVocab
WORK_AgentDisplay	WORK_Location2GeoRefid
WORK_Agent1NameType	WORK_MaterialDisplay
WORK_Agent1Name	WORK_Material1
WORK_Agent1NameVocab	WORK_MaterialVocab1
WORK_Agent1NameRefid	WORK_MaterialRefid1
WORK_Agent1Role	WORK_Material2
WORK_Agent1Attribution	WORK_MaterialVocab2
WORK_Agent2NameType	WORK_MaterialRefid2
WORK_Agent2Name	WORK_Material3
WORK_Agent2NameVocab	WORK_MaterialVocab3
WORK_Agent2NameRefid	WORK_MaterialRefid3
WORK_Agent2Role	WORK_Material4
WORK_Agent2Attribution	WORK_MaterialVocab4
WORK_Agent3NameType	WORK_MaterialRefid4
WORK_Agent3Name	WORK_MeasurementsDisplay
WORK_Agent3NameVocab	WORK_Measurements1Extent
WORK_Agent3NameRefid	WORK_Measurements1Unit
WORK_Agent3Role	WORK_Measurements1Value1
WORK_Agent3Attribution	WORK_Measurements1Type1
WORK_CulturalContextDisplay	WORK_Measurements1Value2
WORK_CulturalContext1	WORK_Measurements1Type2
WORK_CulturalContext1Vocab	WORK_Measurements1Value3
WORK_CulturalContext1Refid	WORK_Measurements1Type3
WORK_CulturalContext2	WORK_Measurements1Value4
WORK_CulturalContext2Vocab	WORK_Measurements1Type4
WORK_CulturalContext2Refid	WORK_Measurements2Extent
WORK_DateDisplay	WORK_Measurements2Unit
WORK_Date1Type	WORK_Measurements2Value1
WORK_EarliestDate1	WORK_Measurements2Type1
WORK_LatestDate1	WORK_Measurements2Value2
WORK_Date2Type	WORK_Measurements2Type2
WORK_EarliestDate2	WORK_Measurements2Value3
WORK_LatestDate2	WORK_Measurements2Type3
WORK_DescriptionDisplay	WORK_Measurements2Value4
WORK_DescriptionSource	WORK_Measurements2Type4
WORK_InscriptionDisplay	WORK_RelationDisplay
WORK_LocationDisplay	WORK_RelatedWork1
WORK_LocationNotes	WORK_RelationType1
WORK_Location1Type	WORK_RelatedWork2
WORK_Location1Name	WORK_RelationType2
WORK_Location1NameType	WORK_RightsDisplay
WORK_Location1NameVocab	WORK_SourceDisplay
WORK_Location1NameRefID	WORK_StateEditionDisplay
WORK_Location1ObjType	WORK_StylePeriodDisplay
WORK_Location1ObjRefID	WORK_StylePeriod1
WORK_Location1Geo	WORK_StylePeriodVocab1



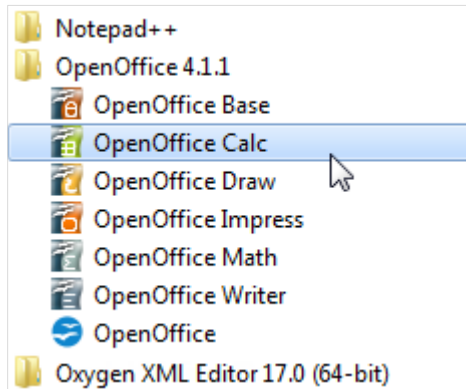
WORK_StylePeriodRefid1	WORK_TechniqueRefid4
WORK_StylePeriod2	WORK_TextrefDisplay
WORK_StylePeriodVocab2	WORK_WorktypeDisplay
WORK_StylePeriodRefid2	WORK_Worktype1
WORK_StylePeriod3	WORK_WorktypeVocab1
WORK_StylePeriodVocab3	WORK_WorktypeRefid1
WORK_StylePeriodRefid3	WORK_Worktype2
WORK_StylePeriod4	WORK_WorktypeVocab2
WORK_StylePeriodVocab4	WORK_WorktypeRefid2
WORK_StylePeriodRefid4	WORK_Worktype3
WORK_SubjectDisplay	WORK_WorktypeVocab3
WORK_Subject1	WORK_WorktypeRefid3
WORK_SubjectType1	IMAGE_AgentDisplay
WORK_SubjectVocab1	IMAGE_Agent1NameType
WORK_SubjectRefid1	IMAGE_Agent1Name
WORK_Subject2	IMAGE_Agent1NameVocab
WORK_SubjectType2	IMAGE_Agent1NameRefid
WORK_SubjectVocab2	IMAGE_Agent1Role
WORK_SubjectRefid2	IMAGE_CulturalContextDisplay
WORK_Subject3	IMAGE_DateDisplay
WORK_SubjectType3	IMAGE_DescriptionDisplay
WORK_SubjectVocab3	IMAGE_InscriptionDisplay
WORK_SubjectRefid3	IMAGE_LocationDisplay
WORK_Subject4	IMAGE_MaterialDisplay
WORK_SubjectType4	IMAGE_MeasurementsDisplay
WORK_SubjectVocab4	IMAGE_RelationDisplay
WORK_SubjectRefid4	IMAGE_RightsDisplay
WORK_Subject5	IMAGE_RightsNotes
WORK_SubjectType5	IMAGE_RightsType
WORK_SubjectVocab5	IMAGE_RightsHolder
WORK_SubjectRefid5	IMAGE_RightsText
WORK_Subject6	IMAGE_SourceDisplay
WORK_SubjectType6	IMAGE_SourceValue
WORK_SubjectVocab6	IMAGE_SourceType
WORK_SubjectRefid6	IMAGE_StateEditionDisplay
WORK_Subject7	IMAGE_StylePeriodDisplay
WORK_SubjectType7	IMAGE_SubjectDisplay
WORK_SubjectVocab7	IMAGE_Subject1
WORK_SubjectRefid7	IMAGE_SubjectRefid1
WORK_Subject8	IMAGE_SubjectVocab1
WORK_SubjectType8	IMAGE_SubjectType1
WORK_SubjectVocab8	IMAGE_Subject2
WORK_SubjectRefid8	IMAGE_SubjectRefid2
WORK_TechniqueDisplay	IMAGE_SubjectVocab2
WORK_Technique1	IMAGE_SubjectType2
WORK_TechniqueVocab1	IMAGE_Subject3
WORK_TechniqueRefid1	IMAGE_SubjectRefid3
WORK_Technique2	IMAGE_SubjectVocab3
WORK_TechniqueVocab2	IMAGE_SubjectType3
WORK_TechniqueRefid2	IMAGE_TechniqueDisplay
WORK_Technique3	IMAGE_TextrefDisplay
WORK_TechniqueVocab3	IMAGE_TitleDisplay
WORK_TechniqueRefid3	IMAGE_Title
WORK_Technique4	IMAGE_TitleType
WORK_TechniqueVocab4	IMAGE_WorktypeDisplay



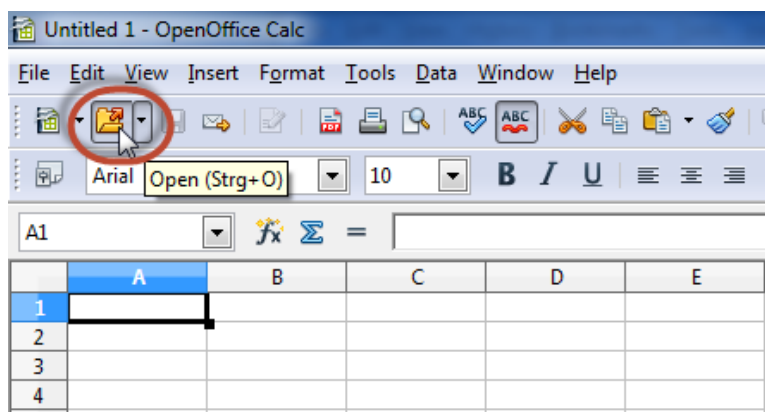
12. Appendix: Converting .xlsx file to .csv using OpenOffice Calc

Save your Excel spreadsheet as usual.

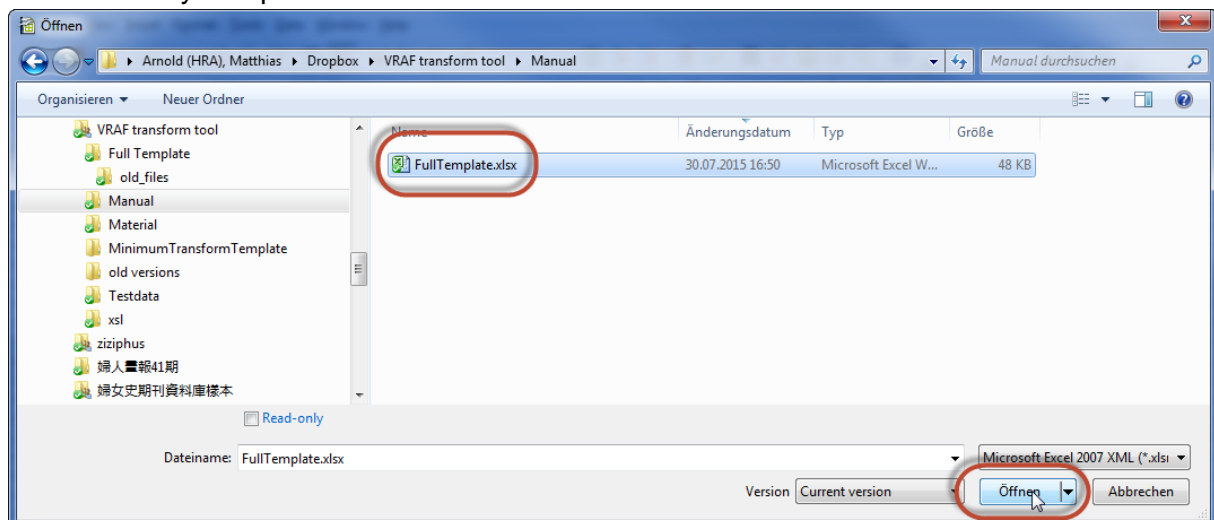
Start OpenOffice Calc

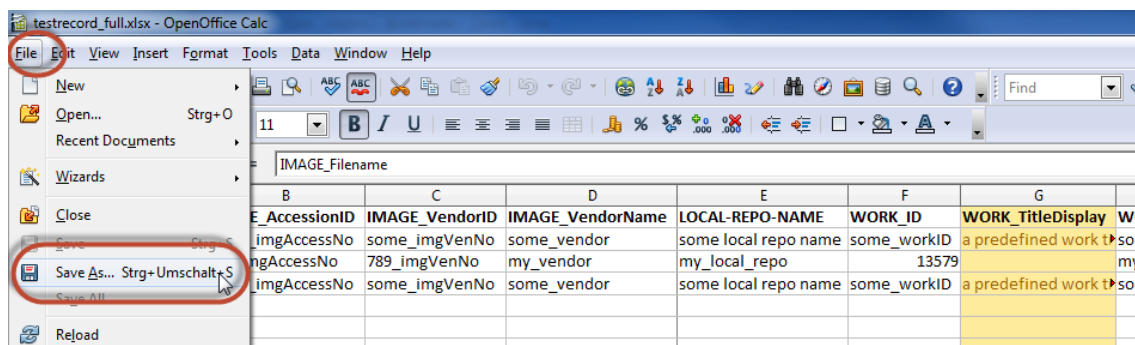


Click "Open"...

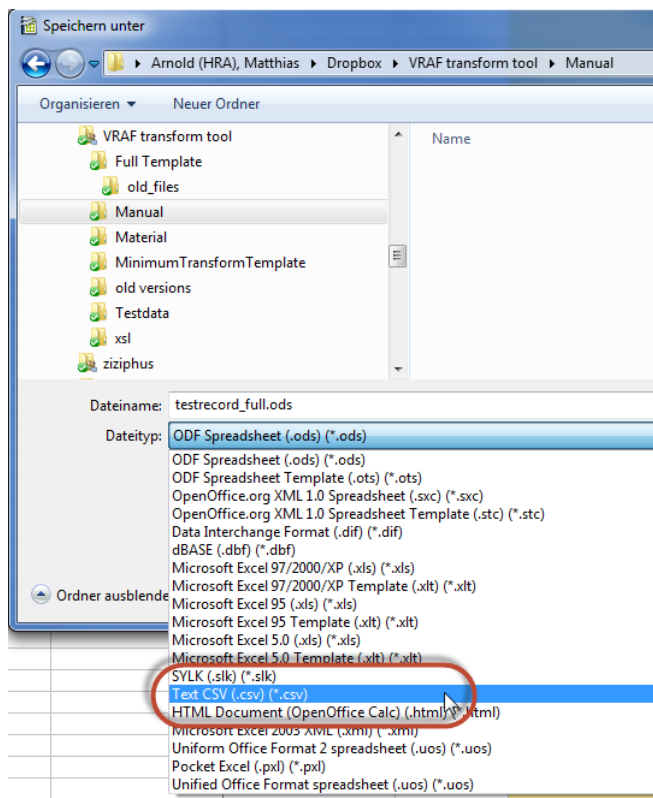


...and select your spreadsheet.



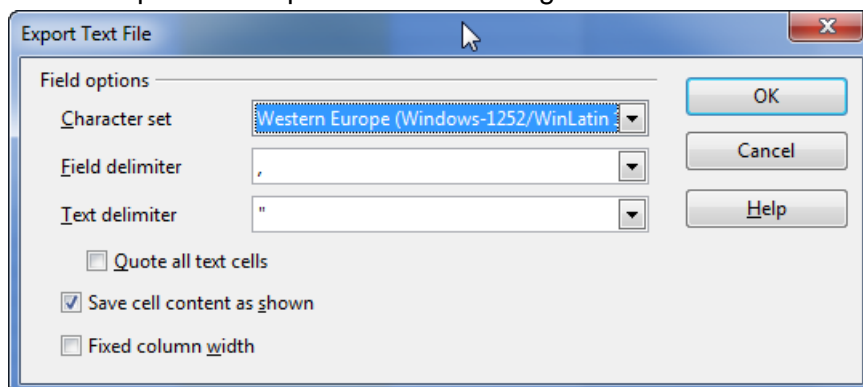


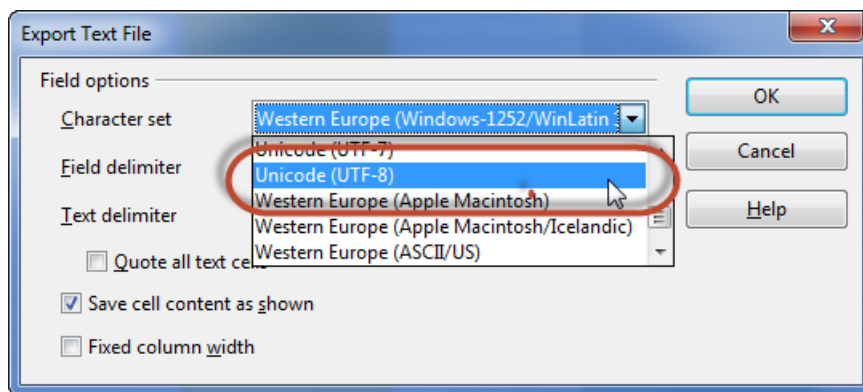
Once the document is opened go to the "File menu" and select "Save As..."



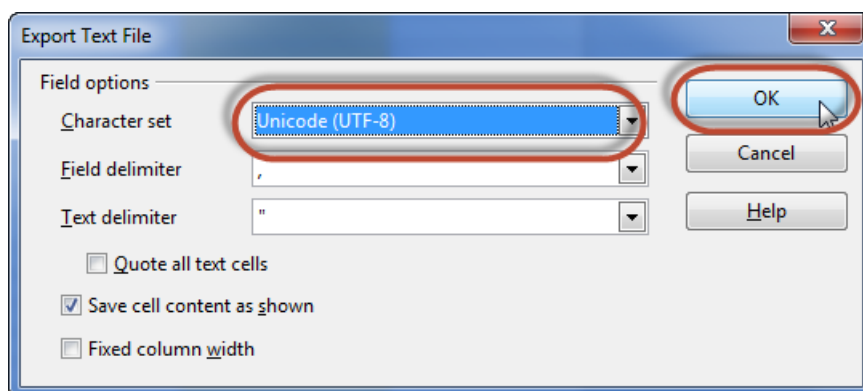
Choose "Text CSV (.csv) (*.csv)" and hit the "Save" button.

This will open the "Export Text File" dialog.

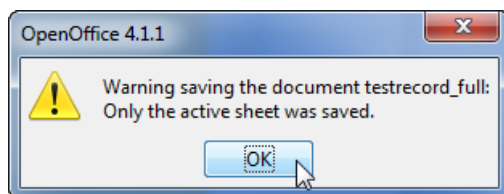




Change the "Character set" to "Unicode (UTF-8)".



Then click "OK".



A message will be displayed, informing you that only the active sheet was saved to .csv format. Click "OK" to confirm.

You can now close Calc.

To double-check if your data is correctly encoded, open the .csv file in an editor like NotePad++. If it displays correctly, you can proceed to XML transformation.

