

For TXSTEP version July 2014, the following files and test scripts may serve to get better acquainted with this powerful tool. Most of the materials are based on the respective materials used for the TUSTEP introductory courses held at Tübingen University by Wilhelm Ott.

Files (directory `txstep/uebq`)

`inh` Table of contents of a (German) anthology, tustep file format, one record per contribution, e.g.:
Autor, Adam: Titel (1-11)

`inh.fdf` as `inh`, but plain text file (system data format)

`jnh` as `inh`, but containing errors:
missing blank after punctuation mark,
wrong sequence of contributions

`inh.xml` as `inh`, but well-formed xml format, one record per element
Tags: `<inhalt>`
`<beitrag>`
`<a>...`
`<t>...</t>`
`<s>...</s>`
`</beitrag>`
...
`</inhalt>`

`inhb5.xml` as `inh.xml`, but `<a>...` tags for each author

`atsf` as `inh`, tustep file format, one record per element:
author(s), title, pages, starting with
`=a= =t= =s=` respectively.
The file contains many errors (same markup more than once per contribution; missing markup; wrong sequence of markup; unforeseen markup like `=x=`)

`atsf.sdf` as `atsf`, but system data format

`dedale` tustep file with french words. Words on page 1 are shown in the correct alphabetic order for french. (for sorting exercises: these words should remain in the same order after sorting the whole file)

`span` tustep file with spanish words, two pages with identical entries, each showing the entries in the required traditional (i.e. before 1994) spanish order. (For sorting exercises: these words should remain in the same alphabetic order after sorting the whole file)

`sans` tustep file containing two identical sets of records (two pages) containing the initial letters of sanscrit words in the alphabetical order required for sanscrit word lists. (For sorting exercises: these records should remain in the same alphabetic order after sorting the whole file.)

- ibam Horatius, sermones I,9; latin hexameter poem, showing after each verse its metrical pattern: 0 for spondeus, 1 for dactylus; tustep file format, page number = number of poem, line number = number of verse
- ibamr as *ibam*, but poem and vers number in front of each verse in square brackets, e.g. "[9.2] nescio quid meditans..."
- ibamr.fdf as *ibamr*, but system data format (plain text file)
- ibamt.fdf as *ibamr.fdf*, but without metrical patterns
- kolinfor list of contributions to the "Tübingen Colloquia for electronic data processing in the humanities":
tustep file format, structure:
<ko>1. Kolloquium: 17. 11.1973</ko>
<au> given name <fn>surname</fn> (institute);
given name <fn>surname</fn>:
<ti> title of paper
<au> ... (if more than one contribution)
<ti> ... " " " " "
- kolinfor.fdf as *kolinfor*, but system data format (plain text file)
- lucas first lines of the greek text of the Gospel of Luke (containing a wrong accent - grave instead of acute - in the first word), tustep file format and encoding
- lucas.xml as *lucas*, but xml-file, utf-8 encoding
- schella.x 3 versions of about 90 lines from a text of Schelling,
schellb.x tustep file format with xml tags. File *schella.x*
schellc.x contains the text as shown in the 1988 editon,
schellb.x and *schellc.x* contain freely invented
"witnesses" of the same text, containing freely
invented variant readings as a basis for
demonstrating a computer-aided workflow for
editorial work
- schella.xml as *schella.x*
schellb.xml *schellb.x*
schellc.xml *schellx.c*, but utf-8 encoded xml-files
- schellkac.xml (see below under *schellsatz* and in the
insert_app.xml script)
- bibl.fdf Bibliography (1982): "Computers in scholarly editing":
system data format, ansi encoding, "preface",
"part 1: critical editing", "part 2: "typesetting",
markup: &.u1...&.1u heading level 1
&.u2...&.2u heading level 2
\$ start of paragraph (in preface)
@a Author
@t title
@u ("unselbständig): article in an
anthology or periodical
@s ("selbständig"): book
- bibl_see.fdf as *bibl.fdf*, but with added current number of
contribution (e.g. "@n 1: @a Author ...") and,

after each entry, pointers to other publications
 from the same publishing house or from the same
 periodical in the form
 "(same publiser / periodical: @m 7 @m 50)"

persons.xml index of persons containing references to
 biographies.xml additional information about the same
 persons contained in *biographies.xml*

recker.xml sample xml file from a mail by Ute Recker-Hamm
 in the TEI mailing list of 26.04.2011
 problem: tag each word with <w>...</w>
 despite of other tags overlapping word
 boundaries

gedicht.xml small collection of poems, xml file

gedicht.dtd DTD for *gedicht.xml*

ref-input.xml source file for script *ref-script.xml* and
 ref.xsl

bau1 the 9th article of the "12 Artikel der Bauern" (1525),
 bau2 12 different versions, proprietary encoding of
 bau3 non-standard letters
 bau5
 bau7
 bau33
 bau_bau
 bau_germ
 bau_heli
 bau_mem
 bau_och
 bau_parv

TXSTEP scripts (directory txstep/scripts)

1. Transformation of files

inh2xml.xml transform tustep file *inh* into a tagged xml file,
 tags as in *inh.xml* (see above)

inh_f2xml.xml ditto for *inh.fdf*

inh.xsl dito as xslt script

inh2xml_2.xml as *inh2xml.xml*, different solution

inh_f2xml_2.xml ditto for *inh.fdf*

inh2xml_2n.xml as *inh2xml_2.xml*, but number the authors

inh2xml_a.xml as *inh2xml.xml*, but replace
 "Autor, Adam" by "Adam AUTOR"
 (i.e. given name(s), followed by surname
 in capital letters)

inh_lnr.xml insert 3-digit current number in front of
 each author in file *inh.xml*

xml2inh.xml	re-convert file <i>inh.xml</i> into a tustep file in the same format as file <i>inh</i> (see above)
inh_length.xml	calculate the length in pages of each paper listed in <i>inh.fdf</i> , add this info to each entry; at end of file, add a record showing the length of the shortest and of the longest paper
inh_length_mrk.xml	as <i>inh_length.xml</i> ; additionally mark the longest and the shortest paper
inh_title_length.xml	count the words occurring in the titles of the papers listed in <i>inh.xml</i> , add the number of words as attribute to tag <code><t words="n"></code> ; at end of file, add a record: <code><summary>There are n words in n titles = n.n words/title</summary></code>
name_pages.xml	transform tustep file <i>inh</i> into a xml file containing only the names of the authors and the page numbers, one element for each author: <code><item><name><familyName>Holm</familyName>, <givenName>Bart E.</givenName></name> <pages>5-26</pages> </item></code>
vor_famnam.xml	make a copy of tustep file <i>inh</i> , converting the names of the authors from "surname, given name(s)" to "given name(s) SURNAME"
aut_ref.xml	generate pointers from the second and further authors of a paper to the first author; example: Verfasser, Zeno see Autor, Adam
snr_rest.xml	make a copy of file <i>inh</i> , placing the page numbers to the beginning of each record
word_search.xml	copy the contents of file <i>inh</i> into two different files, the first file containing entries whose title contains one of a list of words (here: the words "in" and "of"), the second file containing the remaining entries
word_search_f.xml	as <i>word_search.xml</i> , containing syntactic errors (to show highlighting by oXygen)
words_search.xml	copy the contents of file <i>inh</i> into two different files, the first file containing entries whose title contains all the words named in the script (here: information and library), the second file containing the remaining entries
satzz_pruf.xml	check the punctuation marks: copy entries not showing a blank after punctuation

- marks into the first file, the remaining (correct) entries into the second file
- atsf_pruf.xml* make a copy of file *atsf* where the faulty entries are marked
- atsf_pruf_2.xml* make a copy of file *atsf* where the faulty entries are marked by a detailed error message, specifying one of "missing category" - "same category occurring twice" - "unknown category" - "wrong sequence of categories"
- atsf2xml.xml* as *atsf_pruf_2.xml*, but copy the correct entries into a tagged xml-file, the faulty entries provided with the respective error messages into a text file
- atsf2xml_b.xml* as *atsf2xml.xml*; take also empty lines and lines beginning with a blank space as the beginning of an entry which should contain author, title and pages
- ibam_silben2.xml* count the words consisting of more than two syllabls, add the number of those words at line end
- ibam_silben2_m.xml* as *ibam_silben2.xml*, but also mark the respective words
- bibl_aut1.xml* make a copy of *bibl.fdf*, replacing the name of the author by "(id.)" when an entry shows the same author as the previous one
- recker_dol.xml* tag each word by `<w>...</w>` (in the form which Recker specified in her above-mentioned mail)
- recker_do.xml* as *recker_dol.xml*, but regard punctuation marks not as part of the word
- ged_xsl.xml* Demo for showing the integration of xslt scripts into TXSTEP scripts: part 1 of the TXSTEP script converts the title of the poems into capital letters, then an xslt script converts the file into an html file (rearranging author and year of publication), then, in the second part of the TXSTEP script, extract the titles of the poems from the html file
- abs_lines.xml* add an attribute `lines="n"` to the tag `<abs>` where n give the number of lines (records) of the respective paragraph (illustrating the use of `mode=from-end` in `<transform>`)
- procdef.xml* example for first defining and then (repeatedly) calling procedures

procext.xml	example for calling a (external) procedure defined in <i>procdef.xml</i> (see also <i>cmp3.xml</i> with <i>cmp3pars.xml</i> and <i>cmp3def.xml</i>)
ref_script.xml	"pattern matching across elements": convert parentheses in the source file to <code><ref></code> and <code></ref></code> : <code><p>This theory has been refuted (Jones <i><i>et al.</i></i> 2003).</p></code> should be converted to <code><p>This theory has been refuted <ref>Jones <i><i>et al.</i></i> 2003</ref>.</p></code>
ref.xsl	xslt script for same problem

2. Compare, further processing the found differences

2.1. TUSTEP files

ve.xml	Compare the files <i>schella.x</i> and <i>schellb.x</i> word by word, produce a synoptic listing showing the differences; write the differences into a file in the syntax of correcting instructions to be used by the tustep programm KAUSFUEHRE (= correct files by prepared correcting instructions contained in a correction file)
ka.xml	make a copy <i>schellk.x</i> of the file <i>schella.x</i> correcting it with the help of the correcting instructions generated by <i>ve.xml</i> . This copy is expected to be identical - except for line divisions - to the contents of the file used as versionB in <i>ve.xml</i> ; check this by comparing <i>schellk.x</i> to this file
sep_diff.xml	Classify the variant readings found by <i>ve.xml</i> and stored in the file <i>schell.diff</i> as required for the preparation of a critical apparatus and store them in four different files: differences in writing of initial upper case umlauts only to file <i>schellkv</i> , differences in punctuation marks only to file <i>schellkp</i> , orthographic only differences (th - t, c - k, y - i) to file <i>schellko</i> other readings to file <i>schellka</i> (= possible apparatus entries)
ve_tuscr.xml	as <i>ve.xml</i> , for demonstrating the possibility to insert original tustep and tuscript code into a TXSTEP script (for demo only, useless for a workflow starting with <i>ve.xml</i>)
bau_cmp.xml	compare 5 versions of the "12 Artikel der Bauern", show the differences in a synoptic listing

bau_bau_cmp.xml as *bau_cmp.xml*, but 12 versions

2.2. XML files

2.2.1. General

cmp.xml Compare the files *schella.xml* and *schellb.xml* word by word, produce a synoptic listing showing the differences;
write the differences into a file with TEI inspired tags, still containing all the information necessary for being used as correcting instructions by the *tustep* program KAUSFUEHRE (see above under *ve.xml*)

cmp_s.xml as *cmp.xml*; from VersionB only two selected passages will be compared to the respective passages in VersionA

cmp_corr.xml make a copy (*schellk.xml*) of the file *schella.xml* corrected with the help of the variants file generated by *cmp.xml*. This copy is expected to be identical - except for line divisions - to the contents of the file *schellb.xml* which has been used as versionB in *cmp.xml*

cmp_n.xml as *cmp.xml*, but ignore the differences consisting in the writing of upper case Umlauts (Ae = Ä, Oe = Ö, Ue = Ü) when comparing the words

vex.xml compare the files *schella.xml* and *schellb.xml* word by word, ignoring the differences in the writing of initial upper case umlauts (Ae = Ä etc.). In addition, regard the words "als" and "wie" as identical (for demo purposes only, not useful for the preparation of a critical edition or semi-automatic correction of double-keyed texts). Generate a synoptic listing of the differences, store the differences in the syntax of *tustep* correcting instructions with minimal xml markup including the lemma and its context (= wording of versionA) and the position of the variant reading in versionB

A normalization like this of the texts to be compared may be useful above all for controlling the results of automatic or manual text manipulation

vexx.xml as *vex.xml*, but store the differences as entries showing TEI compatible tags but still containing all the information necessary for being used as *tustep* correcting instructions

kax.xml make a copy (*schellk.xml*) of the file *schella.xml* corrected with the help of the variants file generated by *vex.xml* or *vexx.xml*. This copy is expected to be identical - except for line divisions and the differences regarding the initial umlauts and the replacement of the words "als" by "wie"

- or vice versa - to the contents of the file *schellb.xml* used as versionB in *cmp.xml*
- vexx_2.xml compare file *schellb.xml* to file *schellk.xml* generated by *kax.xml*. In file *schellk-diff.xml* used as the "variants" file, only differences concerning initial upper-case umlauts and concerning "als" vs. "wie" should be contained
- precorr.xml example for a controlled and well documented workflow:
for semi-automatic correction after double keying, by comparing the two transcriptions with *ve.xml*, a diff file will be generated. Those entries where verisonA has the correct version are marked (manually) by a "-" character preceding the <corr> tag. Where both transcriptions have been mistaken, a double "--" will be inserted (manually) and, as an additional entry, a respective correcting instruction will be added manually.
Then, before running the (automatic) correction (not shown in the script), the entries will be written to two different files, containing the marked (file 1) and the unmarked (file 2) entries. Only the latter ones will be used for running the automatic correction script. This small additional effort (marking instead of deleting the wrong correcting instructions and writing them to a separate file) allows an exact documentation of the steps undertaken for this purpose.

2.2.2 Preparing a critical edition

- cmp.xml (see above); continue with *sep_variants.xml* or *sep_variants_a2.xml*
- sep_variants.xml a critical apparatus will normally not show all differences shown in the witnesses. Frequent differences (e.g. in orthography) which may be characteristic to a certain source or group of sources may be treated in the preface, leaving only the more substantial variants for the apparatus.
This script gives an example how one could proceed in such cases; it distinguishes four types of variants:
- different writing of initial upper case umlaut (written to file *schellkv.xml*)
 - differences in punctuation only (written to file *schellks.xml*)
 - other differences in orthography, here: th vs. t, c vs. k, y vs. i (written to file *schellko.xml*)
 - there remain more substantial variants for the critical apparatus (written to file *schellka.xml*).
- After inspecting these files (and revising

- the entries in *schellka.xml*), continue with *insert_app.xml*
- sep_variants_2.xml*
dito, different technical solution:
instead of defining search tables with identical content each time for more than one search, add the attribute `name="table-name"` when defining them for the first search. This allows to apply (instead of re-defining it) the same table for further searches using the attribute `apply="tablename"` for the tag `<search-table>`
- cmp-list.xml* produce a listing of the variants selected by *sep_variants* (and enhanced by later revision) for the critical apparatus, showing them in parallel below the respective lines of versionA
- insert_app.xml* transform the entries contained in *schellka.xml* into apparatus entries, insert them by means of the `<correct>` module into the text of versionA (which will serve as the edition text). The resulting file *schella_app.xml* needs further transformation for being usable as the source file for typesetting by an external tustep procedure (procedure `$typeset = file schellsatz`). By this procedure, a postscript file will be generated showing the layout of a traditional printed edition
- The entries in file *schellka.xml* need additional philological / editorial revision. Examples for three instances are given in file */uebg*schellkac.xml* where e.g. the inversion of the order of words (as "Lob oder Nahrung" vs. "Nahrung oder Lob"), recorded by *cmp.xml* as two replacements, is replaced by a single entry for a replacement (here: lemma "Lob oder Nahrung", reading "Nahrung oder Lob"). (In *insert_app.xml*, this file is already present as a comment entry in the `<variables>` element.)
- schellsatz* previously prepared procedure for typesetting a critical editon; used in *insert_app.xml* and other similar scripts
- check_var.xml* for checking the result of the manual revisions performed in file *schellka.xml*, generate a synoptic listing of versionA-vs. versionB based on the revised *schellka.xml* file
- sep_variants_a2.xml*
as *sep_variants.xml*, but add to the tag `<rdg>` an attribute `typ="umlaut"|"punctuation"|"orthographic"` for the first three types of readings mentioned above and write all types except "umlaut" to file *schellka.xml*+. When continuing with *insert_app_a2.xml* or *insert_app_a22.xml*, a printed edition showing more than one apparatus at page end will be generated

insert_app_a2.xml

as *insert_app.xml*, but - starting from the results of *sep_variants_a2.xml* - providing two additional apparatuses at page end, one for variants of type "orthographic only" and one for variants of type "punctuation only"

insert_app_a22.xml

dito., different technical solution using <reassemble> instead of <modify> in the passes "check-lemm" and "more-word-lemma"

insert_app_a2_numerg.xml

as *insert_app_a2.xml*, but starting from abbreviated position information in the diff file (as may be the case when using the default parameter values of the original tustep #VERGLEICHE command instead of txstep's <compare>

- more than two witnesses:

cmp3.xml

compare more than 2 witnesses (here: compare *schella.xml* to *schellb.xml* and to *schellc.xml*), producing - in addition to the files containing the variant readings - a single cumulated synoptic listing of the found differences

sep_variants_3.xml

as *sep_variants_a2*, but for more than two witnesses. Compared to *sep_variants_a2*, additional steps are required for cumulating and sorting the variant readings

insert3_app.xml

as *insert_app_a2.xml*, for more than two witnesses. Compared to *insert2_app.xml*, additional steps are necessary. For details, see the comments contained in the script

cmp3_b.xml

as *cmp3.xml*, with context-for-parcelling="1": instead of relating one another the found differences on a word-by-word basis, in some cases better results may be obtained by requiring that up to n consecutive words should be identical between the correlated text parts to form respective lemma-reading pairs

sep_variants_3b.xml

as *sep_variants_3.xml*, but readings found by *cmp3_b.xml*

cmp3pars.xml

as *cmp3.xml* (without listing), but using a parameter-controlled procedure

cmp3def.xml

as *cmp3.xml* (without listing), but using a parameter-controlled procedure with defined file names

3. Generating indexes; sorting:

autorenliste.xml	alphabetical list (xml-file) of author names from file <i>inh.xml</i> ; page numbers as references
autorenliste_rel.xml	as <i>autorenliste.xml</i> , but relative paths for defining file names
autreg_inh.xml	alphabetical list (tustep file) of author names + page numbers from tustep file <i>inh</i> , with xml-tags <code><eintrag> ... </eintrag></code> surrounding each entry. Instead of erasing a non-empty destination file beforehand, append the results to its content
inh_alpha.xml	sort the entries in file <i>inh.xml</i> alphabetically by the name of the first author. For entries showing more than one author, generate entries, pointing from the other authors to the first one, and insert them at the proper alphabetical location
inhx_stichw.xml	alphabetical index of title words starting with the letter t; source file: <i>inh.xml</i>
rv_greek_s.xml	alphabetical list of wordforms from tustep file <i>lucas</i> (greek text, tustep-encoding)
rv_greek_x.xml	dito, but source <i>lucas.xml</i> (xml-file, utf-8); entries containing the letter x (= greek chi) are output in upper case letters
rv_greek_xs.xml	alphabetical list of word forms from file <i>lucas.xml</i> , typesetting the result via built-in tustep command
rv_greek_xr.xml	reverse index of word forms from file <i>lucas.xml</i>
rv_greek_xr1.xml	dito, output as listing
ibamr_metren.xml	index of metrical patterns of latin hexameter poem (source file: <i>ibamr</i>)
ibamr_metren2.xml	as <i>ibamr_metren.xml</i> , sorting the patterns by the number of dactylic feet ("slow" verses first)
ibamr_freq.xml	alphabetical list of word forms from file <i>ibamr</i> , with references, and the same list, sorted by descending frequency, without references
ibams_freq.xml	dito, but from file <i>ibamr.fdf</i> ; better commented script than <i>ibamr_freq.xml</i>
ibamr_kwic.xml	KWIC concordance for hexameter poem (file <i>ibamr</i> , tustep format); context: 1 verse
ins.xml	insert bibliographic information provided in a separate file into an index of persons

ed_bibl3.xml	sort the two parts contained in file <i>bibl.fdf</i> by year of publication
ed_bibl3_ext.xml	as <i>ed_bibl3.xml</i> , but using the procedure defined there as external procedure
ed_bibl3see.xml	sort the two parts contained in file <i>bibl-see.fdf</i> by year of publication, re-number the entries, update the respective pointers
kol_aut.xml	prepare index of authors for the papers given at the Tübingen Colloquia (file: <i>kolinfo</i>); Reference: current number of colloquium + date

4. Define and execute procedures

procdef.xml	example for defining a procedure and executing it in the same script
procext.xml	example for using a pre-defined external procedure (here: the procedure defined in <i>procdef.xml</i>)
cmp3pars.xml	as <i>cmp3.xml</i> (without listing), but using a parameter controlled procedure (see above under 2.2.2)
cmp3def.xml	as <i>cmp3.xml</i> (without listing), but using a parameter controlled procedure with defined file names (see above under 2.2.2)

5. Frame for a TXSTEP script

txstep_rahmen.xml	file which (after saving it under a new name) may be used as a frame for a TXSTEP script
-------------------	--