For TXSTEP version March 2021, 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.txt   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.txt   as *atsf*, but plain text file (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 containing 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.txt   as *ibamr*, but system data format (plain text file).

ibamt.txt   as *ibamr.txt*, but without metrical patterns.

kolinfo   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> ...   "   "   "   "   "

kolinfo.txt as *kolinfo*, 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.txt   Bibliography (1982): "Computers in scholarly editing":
             plain text file, 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.txt as *bibl.txt*, 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.

biographies.xml
                additional information about the persons
                mentioned 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*.

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

Files in subdirectory txstep/ueb

txstep.tu   file for the TUSTEP program generated from the
                txstep script.

Gedicht.dtd as in /uebq, for the xml file generated.

schellkac.xml
                copy of ../uebq/schellkac.xml .

schell.diffi sample of a source file for precorr.xml

Files in subdirectory txstep/ueb/schell

fra3.html   frame for html version of the editon
                        generated by schell-html_build.xml .

texta-u.html   heading line for version a (above the
                        text field in the left column).

textb-u.html   heading line for version b (above the
                        text field in the right column).

textc-u.html   heading line for version c (above the
                        text field in the right column).

variantena-u.html   heading line for the apparatus to
                        version a (below the text field in the
                        left column).

variantenb-u.html   heading line for the apparatus to
                        version b (below the text field in the
                        right column).

variantenc-u.html heading line for the apparatus to
                        version c (below the text field in the
                        right column).

TXSTEP scripts (directory txstep/scripts)

1. Transformation of files; text analysis

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

  inh_f2xml.xml   ditto for *inh.txt* .

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

  inh_f2xml_2.xml   ditto for *inh.txt* .

  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 *inh.xml* into a tustep file
                        in the same format as file *inh* (see above).

  inh_length.xml   calculate the length in pages of each paper
                        listed in *inh.txt*, 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_2.xml as inh_length.xml, but (<insert-at-end>...).

  inh_length_3.xml as inh_lengh_2.xml, but <brackets><opening-bracket>
                        instead oft <read-values-from>.

  inh_length_3b.xml as inh_length_3.xml, but using also <closing-bracket>.

  inh_length_4.xml as inh_length.xml, but only for lines containing
                        "Autor" or "information"; replace "e" in front of
                        "information" by "E".

  inh_length_mrk.xml
                        as *inh_length.xml*; in addition, mark the
                        longest and the shortest paper.

inh_length_proz.xml
                calculate the length in pages of each paper
                 listed in *inh.txt*, add this info plus
                 the percentage of the total number of pages
                 to each entry; add a line showing the total
                 number of pages

inh_title_length.xml
                count the words occurring in the titles of the papers
                 listed in *inh.xml*, add the number pf words as
                 as an attribute to tag <t words="n">; at end of file,
                 add a record: "<summary>There are n words in n titles
                 = n.n words/title</summary>".

names_pages.xml   from file inh, copy only the names of the authors
                 and the page numbers to an xml-file:
                 <name>Hindle, Anthony</name> <name>Raper, Diane</name>
                 <pages>27-54</pages>.

names_pages2.xml as names_pages.xml, different technical solution
                  (AKn/EKn instead of XX).

names_pages3.xml as names_pages2.xml, different technical solution
                  ((Kn/)Kn instead of AKn/EKn).

names_pages3b.xml as names_pages3.xml ((Kn/)Kn and eliminate instead
                  of select).

names_pages4.xml as names_pages3.xml, but one entry per author:
                 <name>Hindle, Anthony</name><pages>27-54</pages>
                 <name>Raper, Diane</name><pages>27-54</pages>

name_pages.xml   transform tustep file *inh* into a xml file
                 containing only the names of the authors and the
                 page numbers, one element for each author:
                 <item><name><familyName>Holm</familyName>,
                 <givenName>Bart E.</givenName></name>
                 <pages>5-26</pages>
                 </item>

name_pages_2.xml as name_pages.xml, different technical solution
                  (ETE, instead of XX in a loop for each author).

vor_famnam.xml   make a copy of tustep file *inh*, 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

aut_ref_2.xml   as aut_ref.xml, different technical solution (KEN, KSP).

aut_ref_3.xml   as aut_ref.xml, different technical solution
                 (ZFZ/ZSP instead of ZF+ in a loop).

snr_rest.xml   make a copy of file *inh*, placing the page numbers
                 to the beginning of each record.

snr_rest_2.xml   as snr_rest.xml, different technical solution
                       (subunits instead of text-parts);
                       only author names and page numbers to be copied.

word_search.xml   copy the contents of file *inh* 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 *word_search.xml*, containing syntactic
                       errors (to show highlighting by oXygen).

words_search.xml   copy the contents of file *inh* into two
                       different files, the first file containing
                       entries whose title contains all the wprds named
                       in the script (here: information and library),
                       the second file containing the remaining entries.

words_search2.xml   from file inh, copy the entries containing both
                       "information" or "automation" and "library" into
                       the first of two destination files, the remaining
                       entries to the second file.

title_length.xml   count the number of words of the titles in file inh,
                       write a respective comment plus the title to the
                       output file.

title_length_2.xml
                       as title_length.xml, different technical solution.

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 occuring twice"
                       – "unknown category" – "wrong sequence of categories".

atsf_pruf_3.xml   as atsf_pruf_2.xml, but use "text-contains-ordered"
                       (ZF, ZFM, ZFH) instead of "text-contains" (ZF+).

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.

inh_von.xml   convert records from file inh to "title (von: authors)",
                       using <text-part-start> and text-part-end>.

inh_von2.xml   as inh_von.xml, but using <brackets>.

ibam_silben2.xml   in ibam.txt, count the words consisting of more than two
                       syllables, add the number of those words at line end.

ibam_silben2_mrk.xml
                       as *ibam_silben2.xml*, but also mark the respective
                       words.

ibam_silben.xml   in ibamr.txt, count the words, syllables and elided
                       syllables per verse, number of words consisting of
                       more than two syllables; average values.

ibam_speech.xml   in ibamr.txt, extract the direct speech passages
                       and store them with poem- and line-number
                       (using <prepare-index> and <generate-index>)

ibam_speech_t.xml as ibam_speech.xml, but using <transform>

ibam_speech_tp.xml
                       as ibam_speech_t.xml; in addition, calculate the ratio
                       (number of words) of direct speech passages
                       to the whole poem

bibl_aut1.xml   make a copy of *bibl.txt*, replacing the name of
                       the author by "(id.)" when an entry shows the same
                       author as the previous one.

recker_do1.xml   tag each word by <w>...</w> (in the form which Recker
                       specified in her above-mentioned mail).

recker_do.xml   as *recker_do1.xml*, but regard punctuation marks
                       not as part of the word.

abs_lines.xml   add an attribute lines="n" to the tag <abs> where n
                       gives the number of lines (records) of the respective
                       paragraph (illustrating the use of mode=from-end in
                       <transform>).

2. Compare, further processing the found differences

2.1. TUSTEP files

ve.xml   Compare the files *schella.x* and *schellb.x* word by
                       word, produce a synoptic listing showing the differences;:
                       write the differences into a file in the syntax of
                       correcting instrictions to be used by the tustep program
                       KAUSFUEHRE (= correct files by prepared correcting
                       instructions contained in a correction file).

ka.xml   make a copy *schellk.x* of the file *schella.x* by
                       correcting it with the help of the correcting
                       instructions generated by *ve.xml*. This copy
                       is expected to be identical – except for line breaks –
                       to the contents of the file used as versionB in *ve.xml*;
                       check this by comparing *schellk.x* to this file.

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 version-a
                       shows the correct version are marked (manually) by a "-"

in front of the first character of the respective line.
Where both transcriptions have been mistaken, "-+" could
be inserted (manually); as an additional entry, a
revised correcting instruction should be added manually.
Then, before running the (automatic) correction (e.g. by
a script as ka.xml), the entries will be written to two
different files, containing the unmarked (file 1) and the
marked (file 2) entries. Only the file containing the
unmarked ones will be used for running the automatic
correction script. This small additional effort (marking
instead of deleting the faulty correcting instructions
and writing them to an additional ile) allows an exact
documentation of the steps undertaken for this purpose.

(For correction files in xml-format, generated by
scripts like ve.xml or vexx.xml, the same effect of
a controlled and documented workflow may be achieved
by marking the faulty correction lines by <!-- and
--> as xml comment instead of deleting them; those
lines will be ignored by the module <correct> as
used in the script kax.xml, see below.)

var_lev.xml   compute the Levenshtein distance for replacement variants
                       found by ve.xml and stored in schell-diff,
                       add it to each entry and list them in the original
                       order and after sorting them in ascending order
                       of their Levenshtein distance

sep_diff.xml   Classify the variant readings found by *ve.xml* and
                       stored in the file schell.diff as required for the
                       preparation of a critcal apparus and store them in
                       four different files:
                       differences in writing of initial upper case umlauts
                         only to file *schellkv*,
                       differences in punctuation marks only to *file schellkp*,
                       orthographic only differences (th - t, c - k, y - i)
                         to file *schellko*
                       other readings to file *schellka* (= possible
                         apparatus entries).

ve_tuscr.xml   as *ve.xml*, for demonstrating the possibility to
                       insert original tustep and tuscript code into a
                       TXSTEP script (for demo only, useless for a workflow
                       starting with *ve.xml*).

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 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 breaks – 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 for semi-automatic corrections 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 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
fille *schellb.xml* used as versionB in *cmp.xml*.

vexx_2.xml   compare file *schellb.xml* to file *schellk.xml* generated by
kax.xml. The file schellk-*diff.xml* used as the "variants"
file shoud contain only differences concerning initial
upper-case umlauts and "als" vs. "wie".

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 uper 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
                ditto, 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 */uebq*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
                        appartuses at page end, one for variants of type
                        "orthographic only" and one for variants of type
                        "punctuation only".

insert_app_a22.xml
                        ditto., 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 differenes.

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 differces
                        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.

 – *output as html file*

 schell-html_cmp.xml
                     Compare version a to version b and version c, and also
                     version b to version a and version c to version a
                     in order to also allow links from variants in version
                     b and c to the respective locations in version a.
                     Analyse the variants as in sep_variants.xml (see above),
                     sort the remaining variants according to location,
                     type of variant, variant reading, and witness code.

 schell-html_build.xml
                     generate a html version of the edition:
                     left half of the screen showing version A, highlighting
                     the text parts for which variant readings have been found
                     in version B or version C; the highlighted parts are links
                     to the critical apparatus shown in the lower frame. In the
                     apparatus frame, the witness codes are links to the right
                     half of the screen where the text of the respective
                     witness will be shown with highlighted text parts where
                     it differs from version A; also here, highlighted parts
                     are links to the apparatus in the lower frame.
                     (The files defining the frames and containing the
                     headings for text and apparatus are provided in
                     subdirectory /ueb/schell.)

3. Generating indexes; sorting:

 autorenliste.xml   alphabetical list (xml-file) of author names from
                    file *inh.xml*; page numbers as references.

 autorenliste_rel.xml
                     as *autorenliste.xml*, but relative paths for defining
                     file names.

 autreg_inh.xml   alphabetical list (tustep file) of author names and
                    page numbers from tustep file *inh*, with xml-tags
                    <eintrag> ... </eintrag> 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 *inh.xml* 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.

 inh_alpha2.xml   as inh_alpha.xml, different technical solution
                    (<brackets> instead of <text-part-start...>).

 inhx_stichw.xml   alphabetical index of title words starting with the
                    letter t; source file: *inh.xml*.

 rv_greek_s.xml   alphabetical list of wordforms from tustep file *lucas*
                    (greek text, tustep-encoding).

rv_greek_x.xml   ditto, but source *lucas.xml* (xml-file, utf-8); entries
                    containing the letter x (= greek chi) are output in
                    in upper case letters.

rv_greek_xs.xml   alphabetical list of word forms from file *lucas.xml*,
                    typesetting the result via built-in tustep-command.

rv_greek_xr.xml   reverse index of word forms from file *lucas.xml*.

rv_greek_xrl.xml   ditto, output as listing.

ibamr_metren.xml   index of metrical patterns of latin hexameter poem
                    (source file: ibamr).

ibamr_metren_rf.xml
                    as ibamr_metren.xml, but with relative frequency plus a
                    list of the metrical patterns in descending frequency.

ibamr_metren2.xml   as ibamr_metren.xml, sorting the patterns by the number
                    of dactylic feet ("slow" verses first).

ibamr_freq.xml   alphabetical list of word forms from file *ibamr*,
                    with references, and the same list, sorted by descending
                    frequency, without references.

ibams_freq.xml   ditto, but from file *ibamr.txt*; better commented
                    script than *ibamr_freq.xml*.

ibamrs_freq.xml   as ibamr_freq.xml, but two files for the alphabetic list:
                    one for the hapax legomena, one for the more frequent
                    word forms.

ibamr_kwic.xml   KWIC concordance for hexameter poem (file *ibamr*,
                    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 *bibl.txt* by
                    year of publication.

ed_bibl3_ext.xml   as *ed_bibl3.xml*, but using the procedure defined
                    there as external procedure.

ed_bibl3see.xml   sort the two parts contained in file *bibl-see.txt*
                    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: *kolinfo*); Reference:
                    current number of colloquium + date.

import_rtf.xml   word index for a short rtf file (generated from .docx);
                    references = from text (two numbers, enclosed in
                    ((...)) and separated by ".").

import_rtf-rec.xml
                    as import_rtf.xml, but references = current line number.

   import_rtf-std.xml
                      as import_rtf-rec.xml, but explicit declaration of
                      start and length of sort key in <sort>.

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 *procdef.xml*).

   cmp3pars.xml   as *cmp3.xml* (without listing), but using a parameter
                      controlled procedure (see above in 2.2.2).

   cmp3def.xml   as *cmp3.xml* (without listing), but using a parameter
                      controlled procedure with defined file names (see above
                      im 2.2.2).

5. Frame for a TXSTEP script

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

More information: http://www.tustep.uni-tuebingen.de/txstep.html
                   http://www.txstep.de