Phase 2 Editing Tasks

## Introduction

Phase 2 editing combines proofreading of Phase 1 editing output with additional editing tasks omitted at the Phase 1 stage. It is possible that the most effective strategy for Phase 2 editing is to use a printout of an HTML rendering of the Phase 1 TEI code. This presents certain technical problems. All features in the code must have an HTML equivalent, and this equivalent must be easy to read. Additionally, getting a browser to printout exactly what is on the screen is a challenge. At present, I am generating HTML versions of TEI files myself. I call these new files the HTML proofs. In order to use them, you must have the Junicode font installed on your system, and the web pages appear to print best in Google Chrome.

Once you have printed out the HTML proof, most Phase 2 tasks can be done by hand-annotating the text on paper. Once the paper version is annotated, the TEI file can be updated. Any ambiguous or questionable forms should be kept in a log file. This should be a plain text file so that you can add code and explanation together without oXygen complaining about validation errors. If the TEI file being edited is called 3aa\_SEL-Brendan.xml, the log file should be called 3aa\_SEL-Brendan\_log.txt.

The TEI file for Phase 2 editing should be kept in a Phase2 folder at the same level as the Phase1 TEI file. Inside the Phase2 TEI file, line 2 should be changed from <!DOCTYPE TEI SYSTEM "**entities.dtd**"> to <!DOCTYPE TEI SYSTEM "**../entities.dtd**">. This will ensure that our Phase 2 TEI file validates.

The task list below groups related tasks in the hope that this creates something of a workflow.

## General Proofreading

1. Go through the text and correct any typographical errors or transcription errors you spot.
2. Ensure all named entities are tagged as such and logged in the xml comments at the top of the TEI file.
3. In addition to normal and long *s*, the text contains what looks like a long, curvy *s* (as opposed to the normal compact *s*) or a long s with a tail stroke. Search for examples of this third type and encode it as <g ref="S">s</g>.
4. Many diacritics have been added algorithmically, and the &combtildevert; and &rabar; abbreviation markers have probably been overused where the form is more likely &isup; or &sup; (see MUFI recommendations for these codes). Likewise, the forms &combmacr; and &combcurl; may not accurately reflect the manuscript forms. All these should be corrected.
5. In general, ensure that diplomatic and critical markup is applied consistently. If long *s* or round *r* appears in the critical markup, change it to normal *s* and *r*.
6. Ensure that text in French or Latin (including forme work and marginalia) has been encoded with <seg xml:lang="fr"> and <seg xml:lang="lat">.
7. Any ambiguous readings should be logged.

## Capitalisation

The Phase 1 editing process ignored verse-initial capitalization. Since this is frequently ambiguous, AEME policy is to make all verse-initial letters capitals. Exceptions may be made, but these must be explained in a <note>.

Additionally, the scribal habit of capitalising the letter following a decorated initial needs to be maintained in the diplomatic markup but converted to lowercase for the critical markup. This has been done inconsistently in Phase 1 editing and needs to be made regular. A common example is the word “SEint”.

Finally, coding of shadow gaps after decorated initials and *litterae notabilores* is inconsistent in Phase 1 editing and needs to be corrected. It should be obvious from the printout when the existing code is not accurately indicating word division.

## Punctuation

Space around pointing has not always been applied consistently in Phase 1 editing. In particular, mid-line *punctus elevatus* has been added algorithmically for the most part, always with space on either side of the punctuation character. If the Phase 1 encoding does not appear to reflect the spacing in the manuscript, it should be corrected.

The *punctus* character is uniformly encoded as &middot; in Phase 1 editing. This may be changed to &period; or &hidot;, if it seems to reflect the manuscript reading more accurately.

Add modern critical punctuation. Modern punctuation marks, including question marks and quotation marks, should be inserted to reflect editorial decisions about the syntactic and rhetorical structure of the language. Recall that the code <pc> is used for diplomatic readings. Critical punctuation should not be tagged. If a medieval punctuation sign occurs in a location where you would like to insert a modern punctuation mark, it should encoded like <pc rend=",">&punctelev;</>. The value of @rend will be used in the critical view in place of the medieval punctuation.

## Metamarks and Transpositions

In the TEI file, do a word search for “metamark” and “transpose”. This will identify whether the file contains metamarks or transpositions. If it does, check that the encoding for accurately reflects the manuscript reading. There are no AEME guidelines for these elements. For usage, see the TEI Guidelines for Metamarks (<http://www.tei-c.org/release/doc/tei-p5-doc/en/html/PH.html#PH-meta>) and Transpositions (<http://www.tei-c.org/release/doc/tei-p5-doc/en/html/PH.html#transpo>).

## Forme Work, Rubrics, and Marginalia

Forme work has been inconsistently coded in Phase 1 editing. Sometimes it has been omitted and sometimes it has been done hastily as a place marker. Each page should have forme work completely encoded following AEME Guidelines. Note that the grid for placement of forme work is probably inadequate. For now, there are just three locations at the top of the page: topLeft, topCenter, and topRight. Items which appear to be in between must be placed in one of these locations. If there are multiple items in the same grid location, that is OK: they may be listed serially.

Pages containing rubrics and marginalia should be logged so that the encoding can be revisited.

## Encoding Notes and XML comments

Editors may have expressed uncertainty about how to encode certain features using xml commentes (between <!-- -->) or inside of <note @type="enc">. If you can determine how the text should be encoded, insert the appropriate markup and remove the comment/note. If you still cannot determine the appropriate coding, remove the comment/note but enter it in your log.

Remove all comments referring to stray marks, blots, and stains, as we are not encoding these.