EGD LEFTOVERS

please use this file to continue discussion of issues not (yet) covered in the Alpha version of the Guide (https://docs.google.com/document/d/1hjWrrwRZQp4hmEqw4jBhhqoXdwJvRlw3EWboJteOPw0/edit?usp=sharing)

Current discussion: to resolve for 2024-25 EGD V2 revision

Terminology

Rename the guide to Encoding Guide for Documentary Editions?

https://github.com/erc-dharma/project-documentation/issues/300

Arlo, 20240509: I am not necessarily against this proposal but let's take some time to ponder and reach an agreement. I have read Pierazzo's piece, and thank you for sharing it. At the moment, my hesitation is due to the term "Documentary Edition" having no currency (that I am aware of) whether in circles of epigraphists or in South/Southeast Asian studies. None of Pierazzo's examples concern epigraphy, and of course none of them concern our part of the world.

Another option might be to rename the Guide in conformity with what we originally were trying to do, and what is still the core of what we're doing, namely to document how inscriptions should be encoded. (This would require finding a solution for the "diplomatic editions" of manuscripts that we have, which in any case require a solution as @michaelnmmeyer affirms that they do not actually follow the same encoding rules.) And of course we could simply stick to the name that we have while adding a conceptual discussion in the preamble.

Dan, 20240513: I don't insist on the change but find "documentary" better than "diplomatic". I recall that when we changed the title from simply "Encoding Guide" to "Encoding Guide for Diplomatic Editions", the reason for this was that we wanted some explicit distinction from the EGC, and because no better term occurred to us, we settled on Diplomatic. I now feel that Documentary is a better counterpart to Critical, and if the term is not current among epigraphists, I think that's because almost all epigraphic editions are documentary by nature, so there isn't really a need to make this explicit. I'm also open to renaming to Epigraphic editions or suchlike.

Use Logical and Physical Structure instead of Intrinsic and Extrinsic Structure?

I think Arlo has a history of using Logical and Physical in much the same sense as I used Intrinsic and Extrinsic in the EGD. I'm amenable to changing these terms.

Term wanted for "what corresponds in writing to a phoneme"

- originally attached to EGD 6.2.5, "distinguishing correction from deletion and addition"
- original apropos: "the nature of intervention must always be considered on the level of transliterated characters, and not on that of characters or glyph components in

the original script or the transliteration”

  - Arlo (Aug 2020): this is a general rule, could be put at a more prominent place
  - Dan (Aug 2020) but how general do we want to make it?
    - move to section 8 to speak not about intervention but about “any markup that concerns the text not in its original physical manifestation but as a text abstracted from its substrate”?
    - the implications of the basic idea are pointed out separately at various places as relevant there, e.g. 2.1.2 and 2.3.6
    - or if we don’t want to generalise it to a global level, then it could become part of 6.1.3, good practice in editorial intervention
- added problem: we need a term to cover “transliterated character or character combination corresponding to a phoneme of the original language”
  - earlier EGD text used “phoneme”, which is problematic as we’re talking about writing
  - current update has “transliterated character”, but then digraphs are not covered
  - proposed terms:
    - graphemic unit
    - transliteration unit
    - “script element” is used a similar sense in the ISO15959 guide https://web.archive.org/web/20160418005419/http://homepage.ntlworld.com/stone-catend/trielem.htm
      - but I’m not happy with that, because script suggests glyphs to me
  - Dan October 2023: I now think that the concept we want here is almost identical to our "character component" as defined in the TG and used many times in the EGD, EXCEPT that a character component is defined (among other criteria) as "visually distinct", which is not true of the inherent vocalisation /a/ of Indic characters. However, I don't really see that such a definition of character component serves any purpose, and think it is probably an oversight in the definition. There is also a related term "glyph component" for concrete realisations, to which the criterion of visual distinctness should apply.
  - So at the moment I think we should revise the definition of character component slightly. Perhaps also change the term to "logical component" (and then, perhaps change glyph component to "physical component").
- Arlo October 2023: revise the EGD point (“the nature of intervention must always be considered…” etc.) as follows:
  - "on the level of logical character components in the transliterated original script, and not on that of whole characters or physical glyph components in the original script, and certainly not the level of Latin letters (although the required level of analysis may in some cases correspond to a single letter in transliteration)"
- **Dan October 2024**: regarding terminology, I now feel that it will be best to change the definition of “character component” to explicitly include the *a* inherent in Brahmic scripts, and possibly also change the term, e.g. to “logical component” or even “logical character” (as opposed to “physical character” for what we now call character in the TG). The expression “logical character components” seems a bit of an overkill to me. We may, in the process, need to slightly revise the other script-unit definitions in the TG, and certainly to rearrange their presentation.

Authorship of translations (EGD 9.2.1)

As per https://github.com/erc-dharma/project-documentation/issues/322 the EGD is not clear enough on how to handle cases where a partial translation from another scholar is completed for DHARMA.

Resolve in the GitHub issue and revise EGD accordingly.

Other TEI header details

Revision description: status

- <change> elements in the revision description have a @status attribute (always draft in my files; don't know about any others)
  - the EGD now has the following instructions for this
  - optionally as needed, `@status`, to help keep track of significant milestones in the history of the file, with one of the following values
    - `"draft"`
    - `"candidate"`
    - `"approved"`
    - `"published"`
    - `"withdrawn"`
  - Arlo has suggested that we add some descriptions
  - problem is, I don't know who came up with these labels and what they had in mind
  - what is the workflow model we anticipate? given the above labels, I would assume that in theory it's the following sequence:
    - member A creates an edition and calls it draft
    - member A, when he considers the edition final, changes status to candidate
    - after this point, member B (always the PI for the task force?) checks and revises the file and changes status to approved
    - I'm not sure how published would differ from approved, or what action would be needed to change the latter to the former
    - also not sure if withdrawn is necessary: if we want to demote a previously approved or published document, why not just change it back to draft or candidate?
- further, according to a comment by Axelle on Example 11.2.1.A (March 2023), "draft" status is assumed by default
  - if this is still so in the schema (we'll need Michaël's confirmation), then I think @status="draft" should be removed from all existing files and it should be stated in the EGD that draft is default and only a different status needs to be explicitly encoded

A vague outline of what we could do about the TEI header

daba, 20241030

<titleStmt>

<title>the principal DHARMA title, as before</title>

<title type="alt">a known alternative title</title><!--only when applicable, with as many iterations as applicable-->

<author>the author(s) as defined in the responsibilities proposal</author>

<editor>the curator(s) as defined in the responsibilities proposal (only when

```
applicable)</editor>
   <respStmt><resp>encoder</resp><persName>the encoder(s) as defined in the responsibilities proposal (only when applicable)</persName>
</titleStmt>
<editionStmt>
  <edition n=”DHARMA1”>First digital edition for project DHARMA</edition>
  <!--replace with the following only if an already existing DHARMA edition has been revised by a person other than its original author or editor-->
    <edition n=”DHARMA2”>Revised digital edition for project DHARMA</edition>
     <respStmt><resp>Revision of digital edition</resp><persName>the person doing the revision</persName></respStmt>
</editionStmt>
<publicationStmt>
  <authority>DHARMA</authority>
  <pubPlace>as in current practice</pubPlace>
      <idno>the DHARMA ID</idno><!--NB I’m suggesting that we remove @type=”filename” from the idno element, but we might add @type=”dharma”-->
    <availability>Licensed under the Creative Commons Attribution 4.0 Unported Licence. Copyright … </availability><!--we may want to shorten the licence statement to this from the verbose stuff we now have in the files, but we can keep the verbose stuff. I still don’t know how copyright should be expressed.-->
  <date>2019-2026</date><!--or whatever-->
</publicationStmt>
```

Bibliographic citations (EGD 10.4.5), encoding and display

https://github.com/erc-dharma/project-documentation/issues/310

also some earlier discussion on plural recognition in https://github.com/erc-dharma/project-documentation/issues/253

The essence: our use of citedRange, with or without unit, is pretty complex and somewhat lax. Add to this the desire to generate meticulously correct and nicely formatted citation displays, and we have a recipe for disaster. Machine recognition of where to use plural forms of units is especially problematic.

Assimilating Pre-existing Corpora into DHARMA

daba opinion 20241030

- for the IDs in the pre-existing corpora, we should not use altIdentifier in the msDesc; instead I suggest that we add a second <authority> after DHARMA to the <publicationStmt> and a second <idno> e.g. with @type=”siddham”
- I propose this on the basis of the TEI guidelines, “A resource may have (for example) both a publisher and a distributor, or more than one publisher each using different identifiers for the same resource, and so on. For this reason, the sequence of at least one model.publicationStmtPart.agency element followed by zero or more model.publicationStmtPart.detail elements may be repeated as often as necessary.”

daba doubts 20241031

However, proceeding as above would imply that the *same XML file* (rather than *a different XML edition of the same text*) is also available in that other electronic corpus. So I’m getting cold feet and would prefer not to do it that way. Rather, we should not indicate original IDs in the header at all, only figure out a way to refer to the original e-editions bibliographically, and do that in the epigraphic lemma.

older notes on the topic follow

- we have a stub (Appendix G) in the EGD for this, but some details need to be sorted out and finalised there if we want to include this appendix in the next release:
- **how shall we include the original identifier of the text in the header of the DHARMA edition?**
  - back then, Arlo suggested <altIdentifier type="Siddham"><idno type="filename">IN00133</idno></altIdentifier>
  - I have no strong view and can accept anything (can Michaël suggest good practice?), but browsing the TEI guidelines gives me the following ideas
    - <altIdentifier> is in fact meant for manuscripts (and equivalent objects), not for abstract texts, so it does not seem appropriate
      - it is, at any rate, part of the Manuscript Description section of the header
    - on the other hand, I don't know about any TEI header item that would cover this need better, so we could argue that the ingested electronic corpus is also a virtual catalogue of the objects
    - if we stick with altIdentifier, then I think we should use
      - <collection>Siddham</collection> instead of @type="Siddham", and
      - plain <idno> instead of <idno type="filename">
      - also, in this case and specifically for Siddham, we might want to switch to (or include in addition) the Siddham OB (object) identifiers, but this needs manual input, since the OB number is not identical to the IN number in most cases
- **How shall we refer to the original digital editions in the external corpus?**
  - we'll need such references in the bibliography (at least the epigraphic lemma, and probably also the primary bibliography, should mention the earlier digital edition)
  - and we might want to refer to the original digital edition in the apparatus too, so they'll need to have sigla
  - this essentially means that we'll want to be able to use <bibl> elements to refer to these earlier editions, but what shall the <ptr> point to?
    - a URI: but do the earlier editions have long-term stable URIs? I don't really trust Siddham on that count
      - we could perhaps upload all the ingested earlier digital editions to Zenodo or HumaNum and get DOIs for them - but that's a lot of work unless Michaël or someone can do it in a batch for us
    - an entry in our Zotero database? again, in that case we'll need to create lots of separate entries
    - no <ptr>, only a non-rigorous <bibl> element containing the old ID as text content? sloppy…
  - discussion in https://github.com/erc-dharma/project-documentation/issues/137

Semantic markup, e.g. place and person names

- the DHARMA doc at https://docs.google.com/document/d/1-hikCxsSqnhCXo5Bq71s1YZp2fft-bDQ5qnCA9rXrXg/edit#heading=h.2t2vycmqan1d is very much in a draft state
- what we currently have in the EGD (§7.4) is also pretty much a draft
- what I'd like to do (Dan, 2023 October) is to reduce the EGD section to practical considerations (the current §7.4.5) and little to nothing else, and instead refer to the

  semantic markup doc in its current state for all the details
  - otherwise, we'll not be able to publish an EGD v2 before working out some pukka details for semantic markup, and that would need a lot of active input from the people who are actually using semantic markup or want to use it in the very near future
- my suggestion for the immediate future is that I cut and paste these parts (and the attached comments as text) from the EGD into the relevant section of the Semantic Markup doc, where we can pick them up later

Marking words for lexicographic harvesting

https://github.com/erc-dharma/project-documentation/issues/131
- may be worth at least a stub

Stop using <ab>?

I feel that our use of <ab> instead of <p> in particular circumstances (incomplete sentences) does not serve a useful distinction. We could simply replace all existing instances of <ab> to <p> and allow only <p> in the future, for any non-verse passage that is semantically separate from its neighbours.

Bibliography details and good practice

- the bibliography could use some improvement and has some long unresolved comments in the GDOC, but we want to avoid major changes that would entail a revision of all already encoded texts

The secondary bibliography

- should the secondary bibliography include publications which are cited in the XML file (e.g. in the commentary), but are not directly relevant to the text? e.g. dictionary, grammar, dharmaśāstra, etc.

Auto-generating structured bibliographies from the metadata spreadsheets

- the present EGD says the structured bib lists will be auto-filled from the metadata
- has this been implemented? is it in the pipeline? if yes to either, would anyone actually take advantage of this, i.e.
  - first create an XML file for the edition leaving the structured bibliography blank
  - then fill out the metadata
  - then wait (how long?) until the bibliography gets automatically filled
  - then remember to come back to the XML and tidy the autofill manually?
- sounds awkward to me
- can we just get rid of this in the EGD?
- can we instead auto-populate the bibliography in the metadata from the structured bibliography in the XML?

Sorting items in the structured bibliographies

- we sort of take for granted that these should be in alphabetical order, but only say this explicitly in the point about tidying the bibliography auto-filled from the metadata, where arrangement into alphabetical order is listed
- Arlo, commenting in Sept 2021: alphabetical arrangement is problematic because we have entries without an author name
  - Arlo suggests chronological arrangement

- Dan: chronological order is also problematic, because some publications were printed long after they were written
  - alphabetical ordering could be based on the Zotero Short Title when no author names are available (or always)
- whatever the decision, Example 9.4.5.A needs revision; ATM it has a primary bibliography in alphabetical order, and a secondary in chronological
- **but anyway**, do we even need manual ordering in the structured bibliography? Couldn't we just allow encoders to put the items in any order convenient to them?
  - they could then be sorted for display by an algorithm, either chronologically or alphabetically

Shortening references in the epigraphic lemma

- Manu and Arlo (September 2020) agree that we don't need full bibliographic detail in the epigraphic lemma (e.g. `First edited by Cohen Stuart <bibl rend="omitname"><ptr target="bib:CohenStuart1875_01"/><citedRange unit="page">23</citedRange><citedRange unit="item">XIII</citedRange></bibl>`)
  - and could do with just the publication reference instead (e.g. `First edited by Cohen Stuart <bibl rend="omitname"><ptr target="bib:CohenStuart1875_01"/></bibl>`)
  - because the full detail is redundantly present in the structured bibliography
- question: do we want to make this mandatory or optional? I prefer optional
  - mandatory would require manual revision of lots of already encoded bibliographies
  - redundant detail is present in the lemma even in the form of reduced references, so redundancy itself cannot be eliminated
  - having the full detail there may be helpful to the reader (no need to consult the structured bibliography)

Unstructured references with citedRange

- UPDATE January 2024: this is now being discussed in https://github.com/erc-dharma/project-documentation/issues/253
- based on Arlo's suggestion, §10.4.5 now allows using freetext references in <citedRange> without @unit, to handle messy references e.g. "part 1, pp. 216–217; part 2, pl. XI"
- the problem associated with this is that so far, citedRange without @unit has been taken by default to a page number, so if the above is implemented, then all previous unitless citedRanges need to be changed to @unit="page"
- I (Dan, EGD comment, June 2021) proposed that instead, we use @unit="other" for freetext references and keep page as the default unit for attributeless citedRange
- there was no reaction to this, but perhaps Axelle did implement the display for unitless citedRange; at least, "page" or "pages" appears in display when @unit="page" is present, and nothing appears when no unit is present
- need to decide what to do; my strong preference is @unit="other" for unstructured references
  - that way, only those people who have used unstructured references (has anyone except Arlo done that?) would have to manually recheck their references and add the unit
  - while otherwise, either
    - unit=page needs to added globally throughout the corpus and *then* people who have used unstructured references must manually check and revert those to no unit; or

  - everyone must manually change their references to the correct encoding

10.4.5, "s.vv" and potentially other lists in bibliographic references

- UPDATE January 2024: this is now being discussed in https://github.com/erc-dharma/project-documentation/issues/253
- Arlo's used unstructured references to encode <bibl rend="omitname"><ptr target="bib:Zoetmulder1982_01"/><citedRange><foreign>s.vv.</foreign> <foreign>adhikāra</foreign> I.2 and <foreign>tan adhikara</foreign></citedRange></bibl>
  - Dan's suggestion (June 2021) was to use <citedRange unit="entry"><foreign>adhikāra</foreign>, <foreign>tan adhikara</foreign></citedRange> instead, which is after all the proper rigorous way of encoding s.v.; and then
    - either live with having just "s.v." in the display instead of "s.vv"; or
    - change the display algorithm analogously to that already used for pages: if the contents of <citedRange> are a comma-separated list or a hyphen-separated range, then display "s.vv.", otherwise display "s.v." (or whatever the algorithm used for pages is)
- If we choose the latter of the above (which I think we should), then it may be a good idea to enhance the display conversion to deal with comma-separated lists and ranges for any kind of citedRange where this may be applicable, e.g.
    - Appendix/Appendices
    - Note/Notes (or n. / nn., depending on whatever we have now)
- depending on the decision reached here, the issue https://github.com/erc-dharma/project-documentation/issues/135 needs to be picked up (about implementing correct display for citedRange)

Conversion of Roman to Arabic numbers in (various parts of) bibliographic references

- need to harmonise EGD (§10.4.5) with Zotero Guide (§4.4 and §4.6)
- we're pretty much clear on the following:
  - Roman journal volume numbers must always be converted to Arabic (ZG4.4)
  - Book volume numbers, if these are part of the title as recorded in Zotero, must always be preserved as in the original (Roman, spelled out in English words, whatever)
  - non-European page numbers (e.g. Devanagari) must always be converted to Arabic
    - *unless* a book uses both non-European and Arabic numbering in different sections, in which case the non-European numbers should be recorded in the original script
  - Roman page numbers (afaik only used in combination with Arabic page numbers, e.g. in front matter) should be left as they are
- we are *not clear* on references to other kinds of citedRange units, e.g. Table, Plate, Figure, Appendix, Item, Note, etc.
  - Dan (EGD comment, December 2021) would prefer always to convert to Arabic unless this results in ambiguity
    - this is what the EGD now says and what the ZG used to say at some earlier time
  - Arlo (same comment thread) prefers preserving Roman in these other units, for these reasons:
    - it's more difficult for the reader to look up the item if we refer to it in Arabic and

it’s printed in Roman (with which I disagree)

  - it’s easier for encoders to just record whatever is printed, so we can expect better consistency this way (which may be true, but I doubt it’s very important; also, it’s easier to make mistakes in recording Roman numerals)
- ultimately, Dan would prefer to make this optional: preserving Roman and conversion to Arabic could both be OK
  - since it’s unlikely we’ll ever do any machine action on the contents of citedRange, this should not cause any problems

Dates in inscription titling

- in Appendix E, titling conventions, apropos of a title with “year 210”
  - Arlo (Aug 2020) comments:
    - somewhere in this appendix, I feel we need to give guidance on where we prefer non mention of era (as in this example) or mention of era, and guidance on how non CE eras (Śaka, Vikrama, Gupta, etc.) are to be represented.
    - I quote here discussion about the file BengalCharters00054:
      - <!-- argr2daba: shall we make this "Nandapur Plate of GE 169" (or "Nandapur Plate of 169 GE" — I myself would place the era after the date)? --><!--daba: I don't mind. For background: I use "200 CE" and so on, but "GE 200" for some reason. I've just checked to see how consistent that is, and all the Siddham titles that have a GE date in them have GE before the year. I also prefer "ME 200" (for Mālava Era). I have no conscious explanation why, but it I'm pretty sure that's what many scholars do, at least in the older literature. After some checking, I think the reason my be that for those people, abbreviations like GE were analogous to AD (consistently written before the date in older convention), but since the abbreviation CE (consistently after the date) has become widespread, GE etc. are more analogous to that. So anyway, I'm OK with the switch. →
    - I vote for
      - in titles, prescribing the notation YEAR Śaka Era/Gupta Era avoiding any abbreviation except CE
      - allow abbreviation Ś, GE and others in other contexts, but always after the YEAR and separated by a space
  - Dan (Aug 2020) replies:
    - That would go under 4, supplementary details. But, as before, I must point out that this is not an encoding matter at all, and the EG(D) is not the proper place for it. I'm OK with putting the bit on titles here, since we now have this appendix on titles. But this opens up yet another Pandora's box. Do we want rules on how to include regnal years in a title? Years of an uncertainly determined calendar? Preferred names for known calendars? If we do want guidance (and homogeneity in titling) here, how about prescribing simply "year ####" for all dates whenever possible (and anything goes when the encoder deems that unwise)? All the examples with dates that we already have here give just "year ####" anyway.
    - And if you want to regulate/advise how people use dates in discussion, that is quite a different matter and may be put in a separate style (or whatever) guide.
- Dan opinion October 2023: I definitely think we should shelve this
  - pragmatically, it’s an extremely minor detail:

    - the basic titling instruction is to follow the conventions in use in any particular subcorpus, and new titles are only recommended for brand new editions and for subcorpora that don’t have clear conventions - so that would be like what, 10% of the inscriptions in the DHARMA Base?
    - if new titles are needed, then adding a date is only recommended when it is necessary for disambiguation (because there are, e.g., more than one “X plates of A”), so that would be perhaps 5% of the above 10%?
    - and then all this is only feasible when a date is in fact present, so perhaps 20% of the above 0.5%?
  - but giving rigorous rules would, as I noted back then, open a Pandora’s box
- in the EGD, we could, as I note above, prescribe simply "year ####" for all dates whenever possible (and anything goes when the encoder deems that unwise)? All the examples with dates that we already have here give just "year ####" anyway

Discarding Guide sections

Scrapping 1.3.3-4 on XML concepts and terminology

daba November 2024

- I think there is no point in keeping these sections in the guide, since we expect basic XML literacy from all readers and, should anyone be just starting from zero, we do refer to introductory literature
- some of the terms here, e.g. phrase-level elements, would then be moved to 1.3.2 (definitions of terminology), and the rest of these two chapters would be deleted

Scrapping Appendix F. Normalisation Suggestions

- daba comment, May 2021:
  - I now think that we should scrap this entire list as it is. It is hopeless to expect a large number of encoders with various backgrounds to keep in mind a long list of rules pertaining to such minutiae, and anyway, most of our now existing items allow the option of ignoring or flagging. I know that I myself have been inconsistent now and then. When your inscription has e.g. ujvala, the only thing that really matters (as I see things) is that the received reading ujvala is recorded accurately. Whether anything else is encoded, and how, could depend on the encoder's impression of the degree to which any given inscription is correct/standard/meticulous. Any of the following are equally OK as far as encoding is concerned:
    - ujvala
    - u<orig>j</orig>vala
    - uj<supplied reason="omitted">j</supplied>vala
    - u<choice><orig>j</orig><reg>jj</reg></choice>vala
  - Each of these serves the basic purpose of recording the received reading, and the last two also serve the function of simultaneously encoding a normalised text.
- *if we are not scrapping it, then*
  - argr (Sept 2020) suggests: generalise the rule for “use of tv where ttv is expected” to cases of “use of jv where jjv is expected”
  - daba (Sept 2020) responds: yes, and perhaps also
    - add analogous “use of tr where ttr is expected”
    - change the rule “ignore or flag” to “flag or normalise” for all of these

What about Appendix H.Style Suggestions?

- this is completely blank atm (October 2023, since the creation of the stub in Nov 2020), and should I think be scrapped
- it is not an encoding matter and we don’t have the time to write up style guide
- [however, if we ever want a sort of DHARMA style guide with recommendations for print publications by members, I’ve been keeping notes on the style decisions Annette and I have made for the volume we’re editing with the representation panel at the DOT 2022, and we could use that as a draft/skeleton for a style guide]

Minor issues

3.4.1 Boxlike partitions

- Add case study Sangguran?
  - See Github https://github.com/erc-dharma/project-documentation/issues/193
  - The correct Sangguran edition link is now https://erc-dharma.github.io/tfc-nusantara-epigraphy/workflow-output/html/DHARMA_INSIDENKSangguran.html
  - here, we have a stela inscribed on two faces (pagelike partition), but with an extra line at the bottom of face A and two extra lines on the top of face B
    - these are more sloppily engraved than the rest
    - the text continues from the last regular line in A to the first regular line on B
    - it is not clear where the extra lines are meant to go
    - the chosen solution was to encode a textpart for the regular text on AB, and another textpart after it for the extra lines on AB
  - Dan 2023 October: I think there’s no need to write this up as a case study
    - under 3.4.1, it now says that beyond the specific cases that always call for boxlike partitions, “boxlike partitions are only warranted when there is no obvious order in which the zones of text ought to be read, but there is nevertheless good reason for treating them as a single document” - which covers this case
    - also, under 4.4.2 (premodern insertion) we explicitly give boxlike partitions as an option for encoding insertions when it is not clear where they are meant to go, especially if the insertion consists of more than one line

9.2.5 Structural markup in translation

- Arlo wants to add subheadings(?) in translation; <label> proposed
- daba 20241030: I don’t know how old this note is and whether it’s still relevant

Authorship and responsibility

As of 31 October 2024, discussion of roles, responsibilities and copyright is to continue in a separate google doc. The notes, thoughts and stubs below are retained for reference.

20241030 thoughts (daba)

Most of the October 2023 proposal (below) has been endorsed by Manu and Arlo, but I continue to be troubled by the (potential) complexity of this. In addition to the base roles of author, curator and encoder, we may need to add roles for someone who revises an already existing DHARMA edition, and assigning copyright is a nightmare, especially with already separate authorship (i.e. copyright) for translations and potentially separate copyright for commentaries. The necessity of iterating the names of certain contributors in the copyright line of the licence also troubles me, and so does the fact that there is a certain overlap between the epigraphic lemma in the bibliography (which mentions notable previous editions and often indicates how the present digital edition is different and better) and the responsibility / copyright statement. I'd love to have something simpler… No idea what, though.

Perhaps the copyright line in the file itself could simply say "Copyright the editor(s), translator(s) and commentator(s)"...

Also, we could make use of the pre-defined TEI fields for <author> and <editor> (which are specialised forms of the generic responsibility statement) to encode the roles I call author and curator, and use respStmt only for other roles.

One possible way to simplify (daba 20241031)

Could we agree that both the TEI responsibility statement and the copyright line are largely *technical* matters, since intellectual credit is duly assigned in other parts of the file (the epigraphic lemma, the translation attribution, and possibly the commentary)? I feel that the respStmt and the copyright pertain primarily to "this particular electronic document" and not to its text as an abstract entity. Note also that epiDoc files encoded in other projects seem to contain even less information along these lines (see my old notes on "What do others do" below)

If this is an acceptable standpoint, then instead of all the complicated roles, we could simply add an <author> instead of the pre-existing respStmt fields. The author would *always* be the person or persons who have contributed directly to this particular DHARMA edition, whether by merely encoding a published edition, by creating a born-digital edition from scratch, by collating several published editions, by revising a published edition with a facsimile, etc. It would of course be perfectly all right to list multiple people as author, and this could apply not only to direct collaboration on a digital edition but also the use of unpublished draft editions provided by others (e.g. Manu encoding an edition hand-written by Vijayavenugopal or me encoding an EIAD draft that had been made by Vincent and Arlo but never reached publication stage). In such cases, several authors would be listed; the order of appearance could be used to indicate degree of contribution.

The most radical difference between this system and the Proposal below is that it completely removes the author of a previous published edition from the rollcall, whereas such an author would get credit in the Proposal system *if and only if* the DHARMA edition is just a mechanical encoding of the published edition. Since we're talking about authorship and copyright *of the electronic document* and not of the *intellectual content*, this should not be a problem, and due credit is given to the pre-published edition in the epigraphic lemma. There's also the argument that even if an edition is encoded without any revision, the encoding involves expertise and effort and adds value and is therefore grounds for claiming authorship.

Now, if we only have <author> then assigning copyright (NB, of the XML file, not of the abstract intellectual content) is simple: it goes to the author(s) and that's that. We could add to the copyright line that translation copyright may be indicated separately and that previous editions are listed in the epigraphic lemma.

October 2023 summary and proposal (daba)

there's also some subsequent discussion in https://github.com/erc-dharma/project-documentation/issues/242

This is my draft for how I think roles, responsibilities and copyright should be handled in the responsibility statement. It addresses most of the problems we have identified earlier, but needs approval in general and specific discussion especially for the parts in blue. See also below for my earlier notes on the issue, and the discussion at https://github.com/erc-dharma/project-documentation/issues/227

- **rationale**:
  - we prioritise first-hand editing on the basis of visual examination
    - therefore, any DHARMA edition whose creation has involved visual examination shall be recognised as the authorial product of the person who performs such examination
      - even if the visual material is incomplete or low-quality
      - and even if the examination does not result in substantial revision of a previously edited text
    - the authors of any previous published editions are of course duly credited in the epigraphic lemma and the primary bibliography (#reference)
  - furthermore, we feel that the scholarly work of collating editions and judging alternative readings is analogous to the creation of a critical edition by collating manuscript witnesses
    - therefore, in the (very rare) cases where visual examination is entirely impossible, but more than one published edition is available, the collation of editions shall be recognised as equivalent to authorship
  - simultaneously, we wish to appreciate the intellectual effort that goes into the creation of digital editions and the added value created in the process, even when visual examination is not possible
    - therefore, this kind of work shall be recognised as curation, distinguished from authorship in the staking out of responsibilities, but equally entitled to intellectual property rights
  - conversely, we wish to avoid equating the intellectually demanding, but largely mechanical task of encoding to the creation of new knowledge
    - therefore, we shall recognise encoding as distinct from both authorship and curation, in both the delineation of responsibilities and in property rights
    - most of the time, the DHARMA contributor in charge of the encoding will also curate and/or re-edit the text, but recording responsibility for encoding alone may be warranted in cases such as
      - encoding a senior DHARMA member's or external DHARMA contributor's non-XML edition
      - encoding an external editor's edition without access to visual material
      - such cases should be considered individually, attempting to judge honestly and fairly whether the person carrying out the encoding has substantially enhanced the work of the author, in which case he/she shall be recognised as a curator rather than an encoder
- **definitions**: in the scope of this section,
  - a **DHARMA contributor** is typically a DHARMA member, but may also be a person who is not formally a member of the project, but has contributed intellectually to a DHARMA edition through channels other than published work
  - an **external editor** is the author of a published edition of an inscription who is not (or was not at the time of publishing the edition) formally or informally associated with DHARMA
  - **editing** means the creation of a complete or partial edition of a previously unedited inscription on the basis of visual examination

  - **re-editing** means the verification or revision of a non-negligible part or the whole of a previously edited inscription on the basis of visual examination, regardless of whether the text thus established differs substantially from the text of a previous edition
  - **visual examination** means studying a non-negligible part or the whole of an original inscription, or of any surrogate of the original, or of any kind of visual representation of the original or a surrogate
  - **collation** means studying two or more previously published editions of an inscription, thereby establishing a consolidated text and a critical apparatus showing alternative readings, regardless of whether the text thus established differs substantially from the text of any previously published edition
  - **enhancement** means the improvement or augmentation of someone else’s edition of an inscription by means other than re-editing and collation, such as
    - correcting (or even just pointing out) presumable typographic errors in the previous edition
    - proposing (through conjecture or on the basis of additional textual evidence) restorations or emendations not present in the previous edition
    - adding scholarly commentary not present in the previous edition, including commentary gleaned from other publications as well as original commentary
- **roles**:
  - **author**: the responsibility statement must always name the person responsible for “intellectual authorship of edition”
    - the author may be any one of the following:
      - a DHARMA contributor who has edited a previously unpublished inscription
      - a DHARMA contributor who has re-edited a previously published inscription
      - a DHARMA contributor who has collated two or more previously published editions without visual examination
      - an external editor, if and only if the creation of the DHARMA edition from that person’s edition involved neither re-editing nor collation
    - more than one person may be named as responsible for intellectual authorship, provided that the people named have worked as a team, specifically:
      - multiple DHARMA contributors who have collaborated on editing, re-editing or collating the editions of an inscription
      - multiple external editors whose names together hallmark a single published edition
  - **curator**: if the work of the author has been enhanced by a DHARMA member other than the author, then and only then this person shall be listed as responsible for “curation of edition”
    - more than one person may be named as responsible for curation
      - multiple curators may have worked as a team, or independently on separate occasions
    - if multiple authors and/or multiple curators are named, then a partial overlap between these responsibilities is acceptable (e.g. two people collaborating as authors, but only one of them curating the edition)
  - **encoder**: if an edition using other markup (such as printed brackets, annotations, or an implementation of EpiDoc other than the DHARMA standard) has been encoded in DHARMA EpiDoc by a person other than the author (and the curator, if there is one), then and only then this person shall be listed as responsible for “encoding”

    - more than one person may be named as responsible for encoding
    - a partial overlap between the responsibility of encoding and those of authorship and curation is acceptable (e.g. two people collaborating as authors, but only one of them encoding the edition)
- **additional roles**:
  - responsibilities other than those named above may be added on a case by case basis as applicable: discuss
    - anticipated future roles, e.g. revision (harmonise with the revisionDesc)
      - EF: this role should indeed be anticipated and integrated.
    - special case roles, e.g. in ingestion of other corpora (we already have pretty detailed guidance about that in Appendix G, which however will need some revision after the above is finalised; I think that if my suggestion of distinguishing “curation” from both authorship and encoding is accepted, then the special responsibilities in App. G. can be reduced, several of them being replaced simply by curation)
      - EF: I endorse the distinction between curation and authorship and think like Daniel about App. G.
  - global: do we want to add further roles to the respStmt?
    - see “What do others do” below
    - discuss, e.g. we might want to list in all editions
      - the PI in charge of the task force (or all PIs)
        - EF: not necessary in my view; the mention of the DHARMA project elsewhere in the edition XMLs themselves seems enough. + will be acknowledged elsewhere (on a dedicated webpage “about” or “acknowledgement” to be displayed on dharmalekha).
      - the TEI manager
        - EF: does not seem necessary to me; successive TEI manager contributions will be acknowledged elsewhere (on a dedicated webpage “about” or “acknowledgement” to be displayed on dharmalekha).
      - the funder (already credited in the projectDesc)
        - EF: not necessary in my view, as redundant. + will be acknowledged elsewhere (on a dedicated webpage “about” or “acknowledgement” to be displayed on dharmalekha).
      - DB to EF’s notes above: the less stuff to include, the happier I am, but we may also need to think of our XML files stored in a data repository and accessed by someone who does not use the dharma website, or at a time after that website is defunct
      - EF to DB’s note above: right. So maybe include in <respStat>: the role of “TEI” or “data manager”. As for funder, do you mean to add this as a role in <respStat>? If so, I guess the ERC should continue to be mentioned in this role even after October 2026. After all, the “infrastructure” has been set up thanks to its funding.
      - DB: yes, I mean that it could be added as a role, either in the <respStmt> or in the element <funder> which exists explicitly in TEI; see “What others do” below.
- **copyright** is to be assigned in the publication statement as follows:
  - the author(s) hold copyright in all cases
  - the curator(s) hold copyright together with the authors, in all cases when curation is involved

- a person whose only role is that of encoder does not hold copyright
- other roles: may claim share of copyright on a case-by-case basis
- copyright for specific parts of the document: needs discussion
  - the authors of translations are already indicated in @resp for the translation division
    - I strongly prefer not to list these authors redundantly in the header
      - EF: I concur
    - if necessary, we could instead add to the licence section that translation copyright belongs to the people indicated in the translation division
      - EF: I concur
- do we want special copyright for any other sections, e.g. commentary?
  EF: It complicates things and I would prefer to keep a single commentary div.
  My first thoughts are as follows:
  I see indeed issues when author and curator are different persons and when the author's comment cannot be indicated with reference to a <bibl>.
  I would say that, the commentary, when there is a curator, is, by default, by the curator. And if it is by the author, it should be made explicit by mentioning his name in plain text.
  if yes, I think this should be handled analogously to translation
  EF: if yes, I concur.
  - it may not be a bad idea in general to add a @resp to the commentary div
    - (mandatorily? or only when different from the resp author? perhaps best to make it mandatory, otherwise how do we know if the commentary is by the author or by the curator?)
    - we could then permit multiple commentaries by different people within a single document
  - and a disclaimer in the licence saying that commentary copyright belongs to the person indicated there
- likewise for bibliography? I think not
  EF: I concur. No need for this for bibliography.

EF: I add here examples to clarify things first for myself

Example 1: a SII edition encoded by DHARMA, without notable enhancement, that is, only conversion to digital edition and DHARMA transliteration scheme (that is, without access to visual documentation)
Author = SII editor
Curator = DHARMA member
Encoder = DHARMA member
Corresponding epigraphical lemma = This edition by [SII editor], curated by/This digital edition by [DHARMA member]

Example 2: a SII edition encoded by DHARMA, with notable enhancement based on visual documentation, autopsy.
Author = DHARMA member
Encoder = DHARMA member
Corresponding epigraphical lemma = This revised edition by [DHARMA member] OR, if two different persons involved, This revised edition by [DHARMA member 1], encoded by [DHARMA member 2]

Example 3: an edition prepared by Vijayavenugopal, reviewed and encoded by EF, with notable revision based on visual documentation (EFEO photos).
Author = Vijayavenugopal
Curator = EF
Encoder = EF
Corresponding epigraphical lemma = This edition by G. Vijayavenugopal, curated by EF

Example 4: a first edition by EF, based on visual documentation, autopsy.
Author = EF
Curator = EF
Encoder = EF
Corresponding epigraphical lemma = This edition by EF

EF on encoding

The <respStat> should thus contain 4 types of responsibilities, each to be optionally, that is, when necessary, filled in (and displayed, if filled in)
Author
Curator
Encoder
Reviewer

Note:

For names of DHARMA contributors, we currently have this in our XML:

```
<persName ref="part:emfr">
        <surname>Francis</surname>
        <forename>Emmanuel</forename>
</persName>
```

As indicated by Michaël, in such a case, this is just enough:

```
<persName ref="part:emfr">
```

As full names are recorded in https://github.com/erc-dharma/project-documentation/blob/master/DHARMA_idListMembers_v01.xml

EF on display

It might be useful to display besides the name, the affiliation, in case of a DHARMA contributor, especially those not formally members of the project, who like their institution of affiliation be explicitly mentioned.

This would mean fill the tag <affiliation> in https://github.com/erc-dharma/project-documentation/blob/master/DHARMA_idListMembers_v01.xml

Old discussion of authorship and responsibility

My (Daniel's) earlier (2021) summary of the authorship problem

- we should establish rules/guidelines for "intellectual authorship" and "copyright" in the header
- locations:
  - name of copyright holder must be stated in the <licence> within <publicationStmt>
  - "intellectual authorship of edition" needs a name/names in the <respStmt>
  - credit for translations (own and other) already dealt with in EGD §9.2
- is "copyright holder" always and fully identical to "intellectual author of edition"?
  - I suppose yes, but what do the PIs say about that?
- **Dan in email, 20200803** or before, summarising earlier discussion:
  - we are going to consider such an edition the current editor's intellectual product, because even if you have not improved the previous edition substantially, you did encode it and added value to it by recording e.g. a distinction of halanta letters and lost/illegible characters, etc. One can also assume that you would have been able to edit the text from scratch and produce something much like this edition even if you had never seen Majumdar's before. So I think there is justification for us to call such editions our own.
- **Arlo in email, 20200803**:
  - I think it will be good, in a next version of the EGD, to make explicit our hesitations on what is intellectually honest and our arguments in favor of claiming intellectual authorship even when our editions don't change much.
- **Arlo, sometime in 2020**, TFC instructions
  - 1. indicate only the name of the original editor if no verification against the stone/estampage has been possible at any stage between the original printed edition and your digital edition
  - 2. indicate only your name if you are responsible both for encoding and for a thoroughly revised reading of the text: if you can honestly claim to have done a more thorough job than your predecessor, even if the difference in terms of meaning of the text is minimal
  - 3. indicate both the original editor's name and your name in case of doubt
  - 4. indicate Dominic's, Dominique's or my name if you are using a txt file that records a revised edition by one of us, and leave away the name of the original editor if the same condition applies to the relevant person among us as the one applying to you in (2) above
  - 5. leave away your own ONLY if you are really doing no more than applying xml tags and you are not verifying against an estampage (but in principle, this should only happen when no reproduction can be obtained)

What do others do (from the 2021 summary)

- here's what I see in the respStmt of other projects' EpiDoc encoded inscriptions
- **EAGLE Europeana**
  - no respStmt at all; publicationStmt contains only <authority> with the name of the content provider

  - this seems to be the case throughout EAGLE; I have checked files from the following content providers
    - Epigraphische Datenbank Clauss - Slaby
    - Epigraphic Database Roma
    - Epigraphische Datenbank Heidelberg
- **Inscriptions of Israel/Palestine at Brown University**
  - <resp>Prinicipal Investigator</resp> and no other responsibilities
  - publication statement points to an URL with a blank publication statement
- **Inscriptions of Sicily - University of Oxford**
  - primary respStmt includes
    - editor
    - principal
    - funder
  - separate respStmt elements for:
    - original data collection and editing
    - conversion to EpiDoc
    - site construction and encoding
    - editing of geodata
    - museum data collection
    - standardisation of template and tidying up encoding
    - automated or batch processes
  - no copyright
- I have not found any epidoc projects who include copyright information in their headers (CC licences are mentioned in most under availability, but no copyright holder)

How to proceed? (from the 2021 summary)

- **daba musings 20210601**
  - the above guidelines by Arlo seem essentially fine
  - I’ll try to break this down into factors
  - 1. people who may be involved in the creation of an XML edition
    - original author of published edition
    - creator of non-XML etext without re-editing
    - author of previous unpublished digital re-edition (XML or other)
    - encoder (without facsimile verification) of previous edition (print or digital, published or unpublished)
    - author of XML edition (encoding and facsimile verification)
    - thus, “people” can be broken down into two sets:
      - “you”, the encoder, who may also be a re-editor
      - “previous editor”, someone whose edition (print or digital, published or unpublished) you use as a basis for your encoding
      - (etext creator should be mentioned in the guide, but only to make it clear that if you worked from an etext that was essentially someone else’s edition typed into a computer by someone, then the creator of the etext gets no credit)
  - 2. degree of original work done by you
    - if you have looked closely at the inscription or a surrogate while preparing your digital edition, then you deserve credit
      - if you have improved upon a previously existing edition, then you alone get the credit, even if your improvement is minor

            - even better to put this in Arlo’s words: “if you can honestly claim to have done a more thorough job than your predecessor”
            - moral justification: we can assume that wherever the previous editor was correct, you could have read the same text correctly (though perhaps with a greater investment of time), so even if you improved his readings only slightly, you deserve credit for an edition
        - if you have encoded a previous edition without looking closely at the inscription or a surrogate, then only the previous editor deserves credit
- **daba difficulties 20210601**
    - there are still some shady areas
    - first, my “favourite” - when you work from a poor (and/or incomplete) facsimile
        - I guess that in this case the question “how much did you improve the previous edition” (“did you do a more thorough job”) can still help decide
    - second, a sort of extreme case of the above: I have a number of published inscriptions that I have encoded without seeing any facsimile at all, yet I feel that I have improved the previous edition, because that contains typos and obvious misreadings that I can correct (confidently in the edition, or tentatively in an apparatus note) without seeing the original
        - so I think I do deserve credit in such cases, but I’m happy to share that credit with the original editor
    - third, the occasional “critical edition of editions”, where multiple previous editions exist, and you’ve collated all of them as well as the original/surrogate
        - should it always be you alone who gets credit at such a time?
        - or should all previous editors share the credit?
        - or should the authors of “good” editions share your credit and the authors of “not so good” editions be left out?
- **daba 20210601** trying to formulate this simply and fairly:
- define “you” = creator of digital edition in XML
- define “predecessor” = author of an earlier edition (print or digital, published or unpublished) that you utilised for your digital edition
- define “the inscription” = the original or any sort of reasonably readable surrogate
- authorship of the digital edition belongs to
    - you alone
        - if there is no predecessor
        - if you checked your predecessor’s work by looking closely at the inscription, and you can honestly claim to have done a more thorough job than your predecessor(s), even if you have improved only marginally on a previous edition
    - you and the predecessor(s)
        - if you checked your predecessor’s work by looking closely at the inscription, but do not feel that you have improved on
    - the predecessor(s) alone
        - if you only encoded a previous edition without adding value to it

Archived discussion

Paris Discussion 2020-03-17

- *italics* = done in EG

- *enforce unique line numbers; make them complex when restarted*
  - *use just that in @loc of app*
  - *Axelle will find a way to refer to fw*
- *TEXTPARTS - reduce use of textparts to essential minimum, but keep requirement of reproducing textparts in translation, apparatus and other divisions*
- *SPACE*
  - *forget about unit="page"; just put the pb, and then the next pb*
  - *also forget about layout space*
  - *keep binding-hole*
  - *discard ascender and descender; if the encoder deems any of these important, it should be mentioned in an apparatus note*
- *ATTRIBUTE VALUES: we keep things as they are for the time being*
- ORDER OF MULTIPLE ATTRIBUTES: Manu will comment in the guide and I'll finalise it
- *"INFORM US" - change to inform the authors and the TEI engineer*
- *location of pb and lb: stick to what we have*
- *Bibliography:*
  - *we'll have 3 bibliographies:*
    - *1. p with free text epigraphic lemma*
    - *2. list with primary bibliography*
    - *3. list with secondary bibliography*
    - *2 and 3 will be auto-generated from spreadsheets*
      - *1 will be manually created copy-pasting from 2, keeping all the redundant detail, and adding text*
- PALAEOGRAPHICAL FEATURES
  - *Arlo: let's include handDesc in the template and put a short, but not very short description of the script there (multiple sentences OK)*
    - *I, DB, shall make a stub in the EG for that and assign Arlo to say there what should be recorded at that point*
    - *nothing palaeographic goes in the commentary; everything about palaeography that is specific to a particular locus goes in the apparatus; everything else goes in the handDesc*
      - *e.g. shapes of final consonants, punctuation marks, glyph shapes not expected given the script classification; alternative glyphs for the same character*
- *TAGGING ISSUE 2, no reading (not even a gap) in an edition where you read something and/or a gap: <rdg source="bib:VenkatasubbaAyyar1943_01"/> encode their reading as an empty element - ADD THIS TO THE EG*
- AUTO-CONVERSION LISTS
  - *create a github issue for them*
- INSCRIPTION TITLES
  - Manu and I (DB) should finalise and close that
- TG and EG: put the offline Word versions on GitHub

Responsibility Statement

- agree on small number of responsibilities
- check EAGLE and other exemplars
- ask MARKUP list for advice
- basic scheme: either

  - 1. Author of encoded edition + encoder
    - multiple persons under 1 role: in order of importance/share of work done
    - copyright goes to author; for published editions, copyright year is the year of publication; for published editions also add a ? mark to flag possibly problematic copyright
  - 2. Author of digital edition
    - if you do even a little bit of actual editing, looking at the original or a facsimile, then you are the author of the digital edition (and Fleet appears only in the bibliography, not in the respStmt)
    - copyright goes to author
- translation: we stick to what we have; no separate indication in the header
- er: no separate mention; some of this is implicit in our workflow, and major reviews will be entered in the revisionDesc

Addition by emfr 2020/08/02

Please note that any inscription can have, besides its main identifier (<idno type="filename"/> in the teiHeader of the EpiDoc file, e.g. INSPallava00236), several identifiers based on a corpus designation (<altIdentifier> in the teiHeader of the EpiDoc file), meaningful to a human reader. Furthermore, a unique system identifier (<msIdentifier> in the teiHeader of the EpiDoc file) will be automatically created by the IT team for disambiguation purposes.

Dan’s comment: The EGD sections on the TEI header were written up rudimentarily last summer, and all of that has been moved to the Leftovers doc (https://docs.google.com/document/d/1ZXM8qL67DjPMScXUMV2g9syLyPraA2ZKmRzcybJ637Q/edit?usp=sharing), under the heading "Describing the Original Document". Within this, <idno> and related matters are under "Identifying the support". This can go back, revised as needed, into the EGD if encoders will need to encode these things manually in an XML file. The corresponding section in the EGD v1 is 11.2. "Describing the Original Document", which at the moment has only one subsection; if a subsection is to be added on idno etc, then it will need to go ahead of that subsection.

Revisiting Punctuation Encoding (09-03-2020)

Background

- ways in which it is possible to deal with punctuation marks in the original text:
  - A. dedicate a transliteration character to each class of punctuation marks
  - B. instead of transliteration characters, use <g type=”symbol”> to represent a punctuation mark, with @subtype describing its graphic appearance
  - C. wrap a transliteration character or a <g> in <pc> to mark it up as punctuation
- of the methods above,
  - A and B can be used alone or in combination
  - C can be used in addition to A or B or a combination, but not on its own

  - the present TG and EG prescribe a combination: A for common signs and B for less common ones (and never C)
- **we need to consider what we want to achieve with encoding punctuation marks**
  - 1. the mere fact that a punctuation mark is **present** at a given spot in the text
    - A or B alone are sufficient for this purpose
    - we definitely want this much
  - 2. a classification of punctuation marks according to their **function** (do they separate items below sentence/stanza level, at sentence/stanza level or above that level?)
    - this can be encoded implicitly using A or B or both, if we agree that certain signs stand for a certain level of punctuation
    - it can be encoded explicitly using @unit on <pc>
    - but the encoder will sometimes face a hard decision, especially when an inscription uses a sign at a certain level, but the same glyph is used - in the same inscription or a related one - at a different level of punctuation
    - we may or may not want to encode this detail
  - 3. a machine-actionable classification of signs by **graphic appearance**
    - of the methods above, B is best suited for this; A only allows it if we dedicate a large number of Unicode characters to punctuation; C could allow this by co-opting @type and @subtype for a graphic classification, the the @type of a <pc> is not intended in TEI to describe the glyph (but rather the type of punctuation it stands for, e.g. interrogative)

Ways to go ahead

- aims to consider:
  - reduce encoding load and code clutter by keeping things simple
  - preclude fuzzy and subjective decisions on the encoders’ part
  - yet make the system versatile and useful for research
- current issue: do we want to prioritise **functional** classification, i.e. dedicate a limited number of transliteration characters to two or three **levels** of punctuation, regardless of glyph shape?
  - the practical advantage is that a lot of inscriptions use either one punctuation sign (sentence and stanza level) or two (the former and a lower-level one), and these could then be simply transliterated (say, as | and ,) instead of requiring code when they are not the exact shape described in the TG
  - but I (DB) now think functional classification must never *overrule* a descriptive classification: e.g. if an inscription uses a daṇḍa at most stanza and sentence ends and a dash at most half-stanzas, but some half-stanzas get daṇḍas and some full stanzas get dashes, this inconsistency would be lost in the encoding if we only encoded the level of punctuation
- **ideally**, all punctuation marks should be encoded as <g type=”symbol”>, with @subtype indicating their graphic appearance, and with the view that in the long run, we’d create a canonical list of such shapes, and use @ref instead of @subtype to point to that canonical list
- but **practically**, having to encode a <g> instead of typing a simple character at every punctuation mark is bothersome, so we need transliteration shorthand, as in the TG now

- the question then is: how do we balance between complexity of the shorthand (leading to confusion and the need to consult lists frequently) against accuracy of encoding
- what would be ideal is to create a way to automatically replace a custom shorthand with proper <g> elements:
  - an encoder would use a limited set of basic punctuation characters (e.g. the present TG's | / ,) for the most common marks in that particular inscription, AND add an equivalence list somewhere in the XML file, i.e. that | in fact stands for <g type=”symbol” subtype=”verticalbar_plain”> while , stands for <g type="symbol” subtype=”dash_convex”>
  - and an algorithm would pick up that list and replace the shorthand characters to the elements
  - but I don't know if this is feasible or practicable, and if it's really more economical than obliging the encoder to manually run a search and replace on the file before finalising it
- the matter is complicated by the fact that sometimes punctuation marks are too unclear to describe properly, and sometimes we'll only have a printed edition to work from
- another complication: what about editorial punctuation?
  - suppose I want to supply a punctuation mark at the end of a semantic paragraph (or reproduce an edition that does so)
  - but if we use a descriptive classification of marks, which shape do I supply?
- the TEI method of “annotating characters” in a <charDecl> in the header may be relevant here: https://www.tei-c.org/release/doc/tei-p5-doc/en/html/WD.html#D25-30
  - but to use this, we still need to use <g> elements which can then be annotated in the header
  - to produce these, we could
    - auto-convert the shorthand signs, e.g. | to <g ref=”#danda”>|</g>
    - and in the header, include e.g. <charDecl><glyph xml:id="danda"><desc>a plain vertical bar from baseline to headline or slightly above</desc></glyph></charDecl>
      - this could be included in the project templates, with instructions to fill out the <desc> of punctuation characters
  - BUT:
    - 1. this still leaves us with the problem of supplied punctuation and punctuation encoded from a printed edition. These would have to escape the auto-tagging and be left without a <g> tag
    - 2. while this would give us a fine way of recording punctuation details for every inscription, I do not see how this could integrate with a canonical list of symbols and machine-actionable symbol classification

Possibly... we could go ahead like this

- get rid of diversity in the transliteration of punctuation
- reserve the character | (without any tag) for punctuation about which we don't encode any glyph information
  - because it is supplied punctuation; or
  - because it is encoded from an edition
- and apart from the above, i.e. for ALL punctuation marks whose appearance we

want to encode, we can opt for either of the following

- 1. forget altogether about shorthand (such as / and ,) and just use <g> in every case
  - keeping in mind that the generic shorthand in TG §4.2.3 will still be available, i.e. if you write $circle in your file, that will (once the auto-conversion scripts are ready) be converted into <g type=”symbol” subtype=”circle”/>
- 2. reintroduce a limited set of shorthand symbols, e.g. consensually declare that / in the text will be auto-converted to <g type=”symbol” subtype=”hookedVerticalBar”/> or whatever

- I’m beginning to like this, with option 1 preferably
- also, the <g> could *enclose* rather than *stand in place of* the generic | character, which would here mean “any punctuation character”, for which the <g> element gives specific details

Misc related

- incidentally, space fillers should probably be tagged as <pc type=”filler”> instead of <g type=”filler”> because:
  - 1. “filler” is what it does (function), and not what it is (glyph); a hyphen is <pc> in TEI, so fillers can be <pc> too
  - 2. this would let us enclose any <g> within those <pc> tags, so the graphic appearance of fillers could be encoded right there, instead of being described in the commentary as the EG now says
- in case we need it, our earlier “fuzzy definition” of punctuation was “any symbol which you deem to be a punctuation mark because it recurs with some frequency in a text and seems to serve a function similar to any Western punctuation characters (including less frequent ones similar in function to a colon or semicolon)”

Textparts and Languages

- **13 Jan 2019: Daniel’s summary of options**
- the way I see things is, language in a multilingual inscription is yet another “hierarchy” [in the XML tree-model sense], independent of those I call intrinsic and extrinsic structure
  - it’s also in many ways parallel to what I have grouped under “visual features” (EG 7.5) such as script and alignment
- I definitely want to avoid using <div type=”textpart”> *for the sake of* language, because this creates a confusion:
  - textparts are used for extrinsic structure
    - note: “textpart” is not a TEI thing; it’s a special kind of <div> (which *is* a TEI thing), used in EpiDoc to divide an edition into sections on any basis the editor likes (but foremost on the basis of extrinsic structure, both in my understanding of the EpiDoc guidelines and in my preference expressed in the EG)
    - in fact, the EpiDoc guidelines’ illustration of language-based textparts at https://www.stoa.org/epidoc/gl/latest/trans-foreigntext.html is from the inscription http://inslib.kcl.ac.uk/irt2009/IRT481.html , which, as it appears now on that repository, does NOT have language-based textparts: it is a trilingual inscription with three textparts, but the textparts represent blocks (extrinsic structure), each of which bears text in two or more languages (and language is not encoded at all in the XML shown on that website)

  - textparts created for the sake of language may be confused with “proper” textparts and, if both are present, may lead to the problem of overlapping hierarchies
- TEI Guidelines do not seem to discuss multilingual texts as a topic, and the general TEI approach is that xml:lang may be added to any element already present; or, if there is none such, <foreign> may be used to tag strings for language
- what I have in mind follows this, and should perhaps be stated in even clearer/simpler terms in the EG: if your XML document already includes an element that covers the stretch of text in a different language, then encode xml:lang for that element
  - that is to say, if your inscription has textparts as per 3.4 and one of these is in a different language, then add xml:lang to that textpart
    - but do not create a new textpart just because a part of the text is in a different language
    - and do not create a new textpart just because the different-language section has a different topic
      - e.g. if a grant has an invocation, a praśasti section, an executive section and a colophon, you would not create textparts for any of these in an all-Sanskrit inscription, so you should likewise not create them in a Khmer inscription with a Sanskrit praśasti
  - if your inscription has one or more paragraphs or stanzas in a different language, then add xml:lang to that (or each of those) elements
  - note the above is essentially the same as what the EG does say explicitly under §7.5/The scope of visual features - but not so explicitly said for language in the EG
- the only problem (that I can see) with the above is the need to tag each of twenty-odd stanzas as Sanskrit in inscriptions like K0266
  - I do not see this as a real problem
    - it’s certainly legit TEI
    - it is straightforward and requires neither a complex or subjective decision, nor lengthy instructions
    - it is not that much work to encode
  - and therefore I would like to carry on recommending the above
  - but if it is a problem, then we could try the following
- OPTION 1: as you (Arlo) have suggested, we could introduce extra <ab> elements to wrap stretches in a different language
  - by TEI, <ab> may include <lg> so it’s possible to wrap a series of stanzas in a single <ab>
  - however, <ab> cannot wrap <p>, nor can it wrap other <ab>s, so a series of prose paragraphs cannot be wrapped in an <ab> element
  - note: the only reason the EpiDoc guidelines keep mentioning <ab> in association with language is that they wrap all non-verse text in <ab>
    - or rather, the Guidelines are unclear about whether they also wrap <lg> elements in <ab>, or don’t, or do not care if they do or don’t
  - note2: while my pet idea is to use <p> for most prose, and to break up most texts into multiple <p>s, this is not mentioned anywhere in EpiDoc
  - but once we permit using <ab> in ways other than what the EG now says (i.e. for bits of prose smaller than a complete sentence), we’ll have to settle a number of issues, and encoders will have to keep making decisions about these
    - 1. to start using <ab> this way, we’d probably have to forget about using <p>

altogether

      - this may actually be welcomed by some of our encoders who have previous EpiDoc experience but are not used to <p>
      - and it may also bring our files closer to EpiDoc encoded by others
    - 2. we should then also downplay semantic segmentation, and only use multiple <ab> elements for major sections (e.g. invocation, prasasti, executive, colophon) instead of for every semantic “paragraph”
    - 3. we’d need to decide how <ab> and <lg> work together:
      - shall it be mandatory to wrap all <lg>s in <ab> (and if yes, on what basis do we decide how many <ab>s to create for a stretch of stanzas where the topic drifts from one thing to another)
      - or shall it be optional (and if yes, on what basis do we decide when a stanza shall be outside an <ab> and when inside)
- OPTION 2: we could forget about textparts for extrinsic structure and instead do the following:
  - retain textparts for unconnected fragments (which would then be viewed as a special kind of intrinsic structure instead of a kind of extrinsic structure)
  - retain textparts for unconnected short inscriptions on a single object (e.g. graffiti or multiple labels), where these are not encoded as separate XML files
  - allow optional textparts for semantic sectioning of inscriptions, including monolingual ones, e.g. into invocation, prasasti, executive, colophon
  - when one of such sections is in a different language, add @xml:lang to that
  - use only milestones (probably introduce a <milestone type=”boxlike”> in addition to the “pagelike” type already in the EG) for extrinsic structure
  - problems:
    - requires a major rethinking and revision of affected Guide sections
    - I feel this will make the encoder’s decision even harder because it blurs the borders between my concepts of extrinsic and intrinsic structure (so maybe they are not good concepts? anyway, they are the best I could come up with so far)
      - e.g. how would copperplates and seals be handled in this scenario? Create both a textpart (semantic) and a milestone (physical) for the seal, and likewise for the plates?
- **the bottom line**
  - option 1 has some appeal, but I don’t think the gain is worth the amount of trouble needed to reconsider and rewrite our guidelines
  - option 2 is even less attractive
  - so, having considered the alternatives, I’d still go for my original proposal
- bonus question: can you think of any gain from grouping together foreign-language elements (e.g. 24 Sanskrit stanzas) in a single element?
  - if there is a real research purpose that this can serve, then I shall consider it further
  - but if the only point is to be able to add a heading, we could still easily allow an optional editorial <head> at any point in the edition, as I mentioned in a comment

Guide Items that Need Axelle’s Expertise

- devise method for **encoding common name** under alternative names (mainly of deities)
  - I suggest @key with a free value; values used here could be harvested later on to

start work on an authority file, after which @key can be replaced with @ref or @nymRef or whatever is most suitable for the purpose

- suggest method for adding a credit paragraph to the beginning of translations
  - see the comments there for details
  - Arlo says <note> should not be used for this purpose
  - I have misgivings about using <p> in case the translation has textpart divs
- give opinion on the comment attached to “words **left untranslated** shall be tagged as `<foreign>`” in the guide
- perhaps discuss directly with Arlo the comment attached to “when importing previously assembled metadata into our TEI files, we will automatically gather all citations used in the metadata” in the Guide and decide if it will be feasible to harvest citations from metadata table as a starting point for the biblio div
- give suggestions on how best to use (or avoid?) <citedRange> in complex references (e.g. Volume III page 25; Figure 72 on page 163) for the sake of generating tidy display
- references to inscriptions in the DHARMABase: is it possible to simplify this any further? E.g.
  - could the extension .xml be omitted from filenames and supplied algorithmically since it’s ubiquitous?
  - is it not possible to omit reference to the repository and rely just on the file naming conventions, which should be adequate to make sure all filenames are unique?
- finalising the language code appendix
- suggestions, refinement, expansion anywhere else in the Header section
- suggestions anywhere else in the text, especially the parts concerned with IT and good practice, viz. Terms and Definitions *and* General Guidance for Encoding

DB Jottings for Revision of Symbol Encoding (spring 2020)

Summary of what we had before

- we distinguished three categories of symbols: punctuation, space filler and miscellaneous
- for punctuation, we had 3 dedicated transliteration characters (|/,) to be selected depending on the shape of the glyph
  - marks that roughly matched one of the basic shapes were to be transliterated with one of these characters and described in handDesc
  - the transliteration character was to be used without XML markup
  - any punctuation mark that did not match any of the three basic shapes had to be treated as a generic symbol
- for space fillers, we had 1 dedicated character (§)
  - this was to be used regardless of glyph shape
  - the transliteration character was to be wrapped in `<g type="filler">` (redundantly)
  - the shape of glyphs was to be described in handDesc
- for generic symbols (including punctuation marks not matching any of the three basic categories), the empty element `<g type="symbol"/>` was to be used with a @subtype

Problems with the above

- it doesn’t feel right to treat some punctuation marks differently from others
- a particular glyph may be used as a generic symbol or as a space filler or also as a

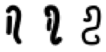
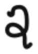
punctuation mark

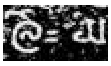
Summary of proposed revision

- for space fillers, instead of `<g type="filler">` we use `<g type="symbol">` to wrap the dedicated character §, with @subtype available in exactly the same way as for generic symbols
- for punctuation characters, we introduce the new dedicated character . to represent punctuation marks universally, and mandatorily wrap it in `<g type="symbol">` with @subtype
  - the previous dedicated characters become “shorthand”, to be auto-converted to the above markup
  - possibly add/change some details to increase flexibility and convenience; see details below

Details of proposed revision

- @emphasise that the same sign may be punctuation or something else, e.g. opening - best not to use a . in the latter case, but it’s OK since we can’t expect to be able to pay attention to that distinction all the time
- use type, not subtype
- **changes to TG** §4.2.1
  - auto conversion is out, the shorthand remains as suggestions, but people will have to do their own conversion
    - suggest not using the full stop there
  - add the dedicated transliteration character . (full stop, period) for generic punctuation without any specifics of appearance or punctuation level
    - introducing . in this function has added advantages: in some cases we will want to use punctuation in our editions without implying anything about its shape, namely
      - when supplying punctuation (for syntactic analysis / segmentation)
      - when encoding a text from a printed edition without consulting a facsimile
      - in these cases, . would be used in our XML files without the <g> wrapper
  - keep the three basic characters listed in the TG, but redefine them as placeholders, to be auto-converted into . wrapped in <g> with a specific set of attributes (see below on <g> wrappers)
  - perhaps add further characters as shorthand for some further common shapes of punctuation glyphs:
    - preferably introduce ~ (tilde) for dash-shaped punctuation marks
      - (Annette and Daniel have punctuation of this shape quite often)
    - possibly introduce ¤ (currency sign) or some other character for dots and small circles used as punctuation
      - (if Arlo or anyone else needs a shorthand for these as distinct from the SE-Asian half-daṇḍa; see below)
  - keep the shorthand involving $ for generic symbols
- **changes to EGD** §4.2
  - for generic symbol markup, keep empty `<g type="symbol">` with @subtype
  - for punctuation marks, require the use of . as outlined above, i.e.
    - without a <g> wrapper for “abstract” punctuation (supplied or encoded from a print edition)
    - with a <g> wrapper for punctuation whose shape is encoded at some level

    - the shorthand characters of the TG remain available and will be auto-converted to . in a <g> wrapper
  - for space fillers, require wrapping § in `<g type="symbol">` with optional @subtype
    - the § will still make it clear that this is a space filler
    - the wrapper with the @subtype allows us to encode its shape conveniently where this is desired
    - deprecate `<g type="filler">`, to be auto-replaced in already existing files to `<g type="symbol">` without @subtype (which can be added manually where desired)
  - **@type and @subtype in the <g> wrapper**: needs thinking and decision
    - it seems that some of us want to encode more detail about symbol glyphs, while others are ready to make do with less
    - our rudimentary table of symbol subtypes has started paving the way to allow this with the use of complex tokens for @subtype in which a “class name” is followed by a more specific name, e.g. circleSmall, circleHorned, circleTarget, etc.; and spiralL, spiralR
    - to make this system more complete, we can go one of two ways:
      - 1. instead of @type=”symbol” we introduce a limited number of new types such as “danda”, “dash”, “dot” etc.
      - 2. we can make the system of hierarchically composed complex tokens more complete, e.g. by using “danda” “dandaHook” “dandaHookCross” as increasingly fine classification
      - I think method 1 is the long-term way to go, because it is better structured
      - but I think for the time being we should stick with method 2, which we have already begun using, and continue to allow some inconsistency in the tokens we use (i.e. not enforce any strict rules), but rather to have a look at all the actually used tokens a year or two from now, and then decide on the best way to re-organise into a hierarchical controlled vocabulary of type and subtype
      - that is to say, @subtype=”danda” could be used for describing any mark that is foremost and basically a vertical bar, and @subtype=”circle” could be used for any that is foremost and basically a circle
        - and any specifications should be tagged on to these basic words, e.g. “dandaPlain” to make it explicit that it is a bar without ornamentation; “dandaHook” to say it has a hook; “circleSmall” to say it is small, etc.
        - some of the specifications will be non-hierarchical e.g. “circleHighSmall” and “circleSmallHigh” are both plausible
        - but we don’t really have to worry about that anyway, because the decision on what order these should come in can be made at a later stage, when we create the controlled vocabulary and assign specific display to (some of) the types or subtypes
- **iterated and complex punctuation marks**: needs thinking and decision
  - this is mainly about double daṇḍas and double dashes, which are normally not just iterations of their single counterpart
    - e.g. in the Cambodian examples above,  are definitely not two iterations of , though some other double daṇḍas could conceivably be seen as two single daṇḍas

    - likewise, the sign in the middle of  is not two iterations of a dash
  - that means that we should probably distinguish double daṇḍas and double dashes (any other doublings that are not simply iterative?) at the very top of our hierarchy, i.e.
    - if we go by method 1, then with e.g. @type=”doubleDanda”
    - or if we go by method 2, then with a dedicated first word in the sequence, e.g. @subtype=”dDanda” or “danda2”
      - preferably not @subtype=”doubleDanda” because that would imply that “danda” is a subcategory of double
      - and also preferably not @subtype=”dandaDouble” because that would imply that the less specific ”danda” incorporates the more specific ”dandaDouble” just as it incorporates the more specific “dandaOrnate”
  - if you agree with distinguishing double dandas and dashes as a separate top-level category, then we need to make sure that the shorthand || gets auto-converted into <g type=”symbol” subtype=”dDanda”>.</g> and not into <g type=”symbol” subtype=”Danda”>.</g><g type=”symbol” subtype=”Danda”>.</g> (exact details of the markup variable as above)
  - the most straightforward way to do so would be to require a space between iterations of glyph, so that
    - if we have e.g. a series of three vertical bars, we could simply type | | | and get that auto-converted into three iterations of whatever the exact markup for a single danda is
    - but if we have a double daṇḍa as a single complex punctuation mark, we could still type || and get that converted into a single instance of the markup for a double daṇḍa
    - this could even be generalised so that we could type ~~ as shorthand for the markup for double dashes
- **dealing with half-daṇḍas**:
  - since I don't really know the diversity of the signs Arlo and his colleagues prefer to transliterate as a comma, and to what extent they are interested in encoding the details of that diversity, I can’t propose a solution out of hand
  - if it includes dots and circles, then we need to think further and possibly limit the use of the comma to exclude them (and, as mentioned above, perhaps introduce one more character as shorthand for dots and small circles)
  - if it only includes "half-sized daṇḍas and the raised comma-like sign that is the basic punctuation sign on Java and Bali" (TG), then we could call it a HDanda or dandaH or suchlike at the top level and optionally allow further subclassification (and we can still introduce a character for dots and circles if we want to)

Editorial normalisation and correction

- Arlo, we should now decide if we want to use all of orig, reg, sic and corr in our practice
- they are all in the EG now, with some fairly acceptable suggestions on when to use each - but I seem to recall you agreeing that the distinction between orig and sic may bring more pain than happiness
- I’m still undecided, thoughts are welcome
- one point that has not occurred to me before, but which may be in favour of

discarding this dichotomy, is that in other methods of editorial correction (e.g., <surplus> and <supplied reason="omitted">) there is no way to encode whether we consider the received text merely non-standard or plain erroneous
- so it may be best to agree that any intervention of ours is just regularisation, and use only <orig> with or without <reg>, but never <sic> or <corr>
- however, that leaves us unable to flag basic typos like *candragutpa* or *śuṇa* (for *guṇa*) as such, and they get in the same category as any other linguistic irregularity
- another point possibly in favour of using <sic> is that it can be used to highlight clearly legible but unintelligible text, though we could decide to use <orig> for that purpose, or to retain <sic> only for that purpose and not to use <corr> ever
- THOUGHTS 20191205
  - Tricky. In my previous practice I've only ever used sic and corr, and only supplied/omitted and surplus for editorial addition/deletion. I have briefly considered that IF we want both sic/corr and orig/reg, then we should likewise distinguish addition and deletion by cases of correction and standardisation. But I fear it gets too complicated for little gain - this is in fact part of why I'm a little in favour of not making this distinction at all.
  - **also, something I've just realised**: using <reg> on its own is not meant to be "addition for regularisation" (so our plan to use <reg>'</reg> for supplied avagraha should also be discarded)
    - <reg> without an <orig> simply means that "this text has been regularised and I'm not telling you what the original was", and while technically this could include cases where the original was nil, that is not how most people will understand it
      - the out-of-the-box EpiDoc transformation simply displays text tagged as <reg> (showing it in both diplomatic and logical edition), and that is also how someone taking our files for a different purpose will understand it
    - we could, perhaps, use <supplied reason="subaudible"> instead of <reg> for this purpose; see https://www.stoa.org/epidoc/gl/latest/trans-subaudible.html
    - or we could, nonetheless, declare that plain <reg> is always an editorial addition in our files
  - at any rate, the whole thing is complicated: there's no clear way to mark up text as suppressed or added for normalisation (as distinct from correction)
  - there's also the question: **what do you want to do with it?**
    - do we want to display correction and normalisation differently? be able to hide normalisation but show correction in some displays? instruct a search engine to treat these two in a different way?
- SCHEME:

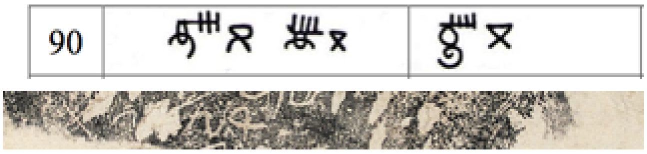
	non-standard	erroneous
flag	<orig>	<sic>
correct	<choice><orig></orig><reg></reg></choice>	<choice><sic></sic><corr></corr></choice>
suppress	???	<surplus>
add	???	<supplied reason="omitted">

- WHAT TO DO?
- 1. reduce diversity
  - use only sic+corr, surplus and supplied/omitted for correction/standardisation, making no distinction between the two
  - use sic on its own for flagging non-standard (instead of <orig>) as well as erroneous
  - editorial avagrahas would thus have to be supplied/omitted
  - pro: simple, standards-compliant, objective
  - con: not sufficient for our purposes?
- 2. retain diversity and co-opt markup for suppression and addition for normalisation
  - use sic, sic+corr, surplus and supplied/omitted only for correction
  - use orig, orig+reg, <surplus reason=”non-standard”> and <supplied reason=”subaudible”> for the same interventions done as standardisation
    - note: <surplus> can take @reason and its values are not constrained by the EpiDoc schema, so we can use any @reason we agree on, but this is not in any standard; the values of @reason permitted with <supplied> *are* constrained by the EpiDoc schema, so “subaudible” is our only option, but at least that is standard and pretty close to what we have in mind
  - editorial avagrahas would then be supplied/subaudible
  - pro: allows a fine distinction
  - con: complex and partly non-standard markup, subjective in many cases
- 3. retain diversity, but restrict options in normalisation
  - use sic, sic+corr, surplus and supplied/omitted only for correction
  - for standardisation, allow only <orig> to flag, or orig with reg to correct
    - addition and suppression for standardisation could then be done in either of two ways (we should decide which and stick to that consistently):
      - A. always require tagging whole words (if sandhi etc permits), e.g.
        - <choice><orig>makana</orig><reg>maṁkana</reg></choice>
        - <choice><orig>evaṁm</orig><reg>evam</reg></choice> eva
      - B. allow either <orig> or <reg> to be empty, e.g.
        - ma<choice><orig></orig><reg>ṁ</reg></choice>kana
        - eva<choice><orig>ṁ</orig><reg></reg></choice>m eva
      - no straightforward way to automatically encode editorial avagrahas in option A, but B can work for them
        - Arlo: we could still use A and treat avagrahas as special and do them by B or supplied subaudible
    - pro: allows a full distinction while remaining standard-compliant
    - con: subjective; requires more complex markup for normalisation by addition/suppression
- **Dan’s bottom line**: 20191205
  - having thought this through, I would vote definitely for option 1 and discard this distinction, unless you have a specific concept for which it is necessary (i.e. back to the question of “what do we want to do with it?”)
  - in the latter case, I believe we option 3B is slightly preferable to the others
  - **bottom bottom line: 3a and separate handling for avagraha**

Cambodian composite numerals

- DB: all resolved, TG and EG edited accordingly on 4 November

- Arlo, this came up in discussion with Kunthea
- composite numerals such as 90 expressed as 80+10



- what is your preference in marking up and displaying these? shall we go for
  - A: display as 80 10, encoded as either of the following
    - <num value="90">80+10+</num> (Transliteration Guide 4.1)
    - OR <num value="90"><g type="numeral">80</g><g type="numeral">10</g></num> (Encoding Guide §Numeral signs transliterated as anything other than a single Arabic digit)
    - this is what we should do for the sake of consistency and on the basis of what the Guides now say
    - but I guess the convention may be to print ‘90’ in editions instead - is it?
  - if you prefer, we could opt to treat such numerals analogously to multiples of 100, i.e. disregard the fact that the symbol is complex and simply display 90, encoded as either of the following
    - <num value="90">90+</num>
    - <num value="90"><g type=”numeral”>90</g></num>

Add some shorthand for V and C to Transliteration Guide

- this too came up in discussion with Kunthea: she’s been using V in her transliteration
  - I’m trying to explain to her that we don’t have that notation and the phenomenon must be handled in markup
  - I hope she’ll understand, but if she (and/or anyone else) will be encoding a lot of editions that use V and C, then maybe we should add some notation (perhaps even V and C, since special final forms of the consonants *v* and *c* are not likely to occur anywhere) that we’ll auto-convert to the rather complex markup required for these cases
  - UPDATE 1 November: I think she understands now, but she did say she had no access to a reproduction. (This was also about the inscription K.66.)
    - what I suggested to her was to use <unclear cert=”low”>a</unclear> for vowels about which she has absolutely no idea, and the same markup with a different vowel if a particular vowel seems more likely
    - this would help avoid the very complex sub-aksara markup involving <seg> and <gap> with a unit of “component”
      - but unfortunately there is no similarly simple way to deal with C
    - I was thinking that if she’ll be encoding a lot of editions prepared by others, which do use C and V, then we _could_ add C and V (or something else) to the Encoding Guide, as shorthand to be converted to the full markup with seg and gap - but this may not be easy to do (since the auto-conversion would also have to add markup to the rest of the akṣara containing the C or V)
    - so, given that you did not have an overwhelmingly positive reaction to the idea of introducing new shorthand, and that I now think auto-conversion will not, after all, be so simple - shall we stick to what we have now?
      - if yes, is it OK with you if I explicitly mention <unclear cert=”low”>a</unclear>

in the EG as a simpler way to encode “absolutely illegible vowel component”, or should I discourage that?

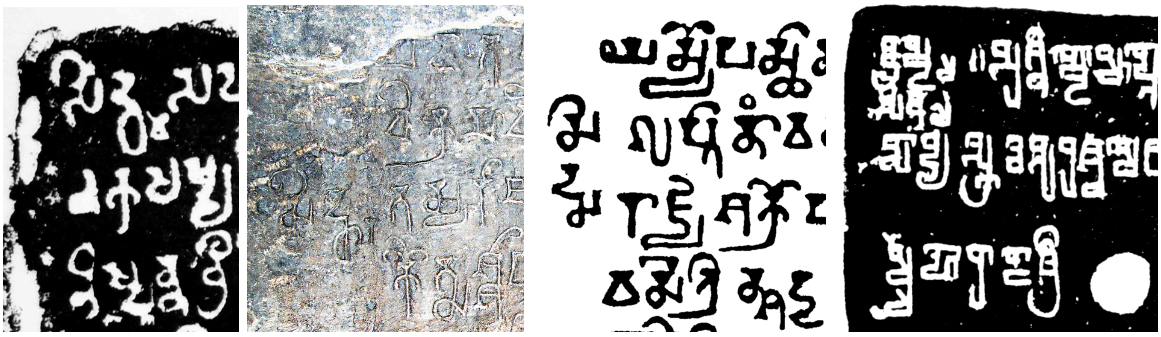
- if no, we could simplify auto-conversion by not requiring `<seg type="aksara">` around lacunose aksaras, but this may not be necessary
- the shorthand could be C= and =V, with the = indicating that these items belong to the same aksara as the transliterated characters on the other side of the = sign

- as for display, I have not given it much thought, but what I had in mind for Siddham was to display “illegible consonant” as a # sign [which was also my transliteration of an illegible consonant], and “illegible vowel” as a prosodic symbol, i.e. anceps ⏓ when the prosodic quantity is unknown, and ⏑ or − as applicable when it is known. Given that the EG already recommends the use of @met in this case, this could easily be implemented.

Simplifying Partition Encoding, DB 20191031

- here are some ideas for reducing the complexity of partition markup, or rather the number of cases where complex markup is required
- **1. limit textpart divisions to major separations**
  - I would require the use of textpart divs only for physically distinct inscriptions
    - copperplate sets with seals
    - fragments with non-contiguous text
    - several inscriptions on a single natural rock or sculpture, when these are not edited as separate items
  - in all other cases where a partition could be described as boxlike, I would encourage people to mark it up as pagelike instead
    - thus, all those four-faced stelae would be encoded as pagelike (i.e. with <cb/> separating one facepair from another), sparing the encoder the need to judge whether the content of one facepair is semantically distinct from that of the second
    - we may leave encoders the option to encode textparts in such cases when they feel the texts are sufficiently different, e.g. one language on one facepair and another language on another
      - the touchstone of when textparts are called for in chunks of texts written on a single object could be the question: is there an evident order in which these chunks should be read?
        - if not, then better encode as textparts
        - if yes, then better encode as pagelike transitions (even if the field boundaries coincide with some semantic boundary)
- **2. limit gridlike divisions to separate surfaces**
  - encoding gridlike divisions would not be mandatory in any case (this is already so in the guide)
  - I would strongly recommend them for physically separate objects, i.e.
    - fragments with contiguous text
    - inscriptions across several architectural blocks
  - I would keep them optional for subsurfaces of a complex surface, e.g.
    - individual faces of a four-faced stela (where lines run from A to b and from C to d)
    - facets of an object with a polygonal cross-section

  - I would discourage (or, to reduce subjective decisions, expressly prohibit) the use of milestones for text divided into columnlike areas by spaces (such as metrically conditioned columns)
  - in all cases where an encoder opts not to use milestones, they would encode space as applicable (“typographic space”)
- **3. get rid of forme work except for pagination**
  - forme work is a complication as I mentioned above
  - it could be simplified if we agreed that we would not use it for anything except pagination/foliation in copper plates (which is, after all, what this TEI element is intended for)
  - if we agree on this, then I could (once again) discard the requirement of using <ab> within <fw>, and of numbering lines within <fw>
  - but it would leave us with the problem of what to do with about incipits arranged in various ways



  - I propose that we forget about encoding information about the position of such opening formulae and instead, simply encode them as regular text, just as old printed editions tend to do, namely:
    - if the text is horizontal and more or level with the first line (regardless of whether or not it is flush with the left margin) -> simply encode it as the first word(s) of the first line
    - if it is above or below the first line, or if it is vertical, then encode it as an extra line before the first
      - the markup under §Directionality and orientation would be applicable to encode these features
      - to accommodate the extra line at the beginning,
        - EITHER the “real” first line of the body text would have to become <lb n=”2”> (or even “3”, for the last example above)
        - OR a special line number, could be used for these extra lines, e.g. <lb n=”0”> (or “01” and “02” in the last example below)
      - information about where on the page these extra lines are located would not be encoded in the edition; the place to describe this is the layout description in the metadata
- **the application of pagelike partitions would not change**
  - everything we now have on <pb> and <cb> remains unchanged
  - <cb> plays the role of a jack-of-all-trades, applicable to fields in any spatial arrangement

Revising symbol encoding

- DB 8 October: I now think we should do as follows. My basic idea is to minimise the number of special signs we use in transliteration and to make the system as clear and objective as possible.

- **space fillers**
  - we can retain <g type="filler"> with a <note> as presently in the Encoding Guide
  - but I'm more and more in favour of not encoding fillers as **characters**, but rather as **space**
  - so instead, I suggest <space type="filled"> (or some other applicable value of @type), with @unit="character" and @quantity giving the number of space filler symbols used
  - if the filler symbols need to be described, this belongs in the commentary, not in a localised note
- **punctuation marks**
  - common punctuation: (i.e. the signs explicitly listed in the Transliteration guide)
    - transliterated with individual signs as per the Transliteration guide, e.g. /|=
    - marked up with <pc> but NOT with <g>
      - markup to be added automatically to the permitted characters
    - displayed as the transliterated characters
  - uncommon punctuation: (i.e. punctuation marks other than those explicitly listed in the transliteration guide)
    - NOT transliterated
      - in other words, get rid of the option in the Transliteration guide to use any Unicode character in conjunction with | and get it automatically encoded as a punctuation mark
      - people who only type texts but do no encoding will not be able to deal with "uncommon punctuation"
        - if it is essential that they should be able to do so, then use e.g. the pilcrow sign suggested below for display [but we'd still need a description of the sign]
    - encoded as <pc><g type="punctuation"/><note>DESCRIPTION</note></pc>
      - this <g> is an empty element, rather than one containing an arbitrary Unicode character
      - a <note> element next to the <g> describes the symbol in English
      - contents of these notes may be harvested later on for the creation of a standard vocabulary, which could then be used for @subtype with this @type of <g>
    - displayed in all instances as some unique character, e.g. ¶ `U+00B6` PILCROW SIGN; the adjacent <note> contents may be displayed as a tooltip or footnote
- **"milestone" symbols**: e.g. mangala sign at beginning, fleurons between sections or at the end, etc
  - I'm more and more certain these should **not** be subsumed in punctuation; "punctuation mark" should only mean marks used repeatedly for the segmentation of smaller chunks such as list items, sentences, stanzas and sub-stanza units of verse
  - these marks could, however, be subsumed in "other" symbols, because distinguishing between "other" and "milestone" symbols can be very subjective
  - so for the time being I suggest treating these in exactly the same way as "other" symbols (below); if that is voted down, then I still suggest the same basic method, only with a different @type and a different display
- **all other symbols**: anything deliberately engraved that is not an alphabetic character, not a numeral, not an editorial mark [as covered under premodern

deletion and premodern correction] and not covered in the points above

  - NOT transliterated
    - in other words, get rid of the option in the Transliteration guide to use any Unicode character in conjunction with $ and get it automatically encoded as a miscellaneous symbol
    - people who only type texts but do no encoding will not be able to deal with “other symbols”
      - if it is essential that they should be able to do so, then use e.g. the section sign suggested below for display [but we’d still need a description of the sign]
  - encoded as <g type=”symbol”><note>DESCRIPTION</note>
    - or some other value of @type, e.g. “miscellaneous”
    - a <note> element next to the <g> describes the symbol in English
    - contents of these notes may be harvested later on for the creation of a standard vocabulary, which could then be used for @subtype with this @type of <g>
- DB Responses 20191101
  - 1. since you don’t mention it anywhere, I gather that you are willing to discard the idea of using a wide variety of symbol characters such as ◊ 卐 and ✤ in the text and circulating a list of such symbols from which encoders could copy and paste
    - if so, I’m glad, because our lives will be a lot simpler this way
    - we can, at any point, decide to display symbols with such characters based on @type (or whatever)
  - 2. I think we still have too many types of symbols, with no objective way to classify actual symbols into these types
    - we seem to agree that we don’t need a special class for “milestones”, and I’m now convinced that “punctuation” should also not be part of the classification of **glyphs** themselves
    - I therefore suggest that we forget about <g type=”punctuation”> and instead allow only the following types of @g:
      - numeral
      - filler (if we discard my suggestion of using <space> for those)
      - symbol (including everything else)
    - for @type=”symbol” we could then require @subtype, containing an unconstrained short description which we can later harvest to create a constrained list of permitted values and associate those values with display characters
      - e.g. quatrefoil, double_dash, floret, etc.; no spaces permitted, so underscore doubles as space
  - 3. we still need to make a decision on what punctuation characters the TG should permit
    - at present we have | / − and , in addition to the problematic = sign
      - please state which of these we MUST retain in your opinion
      - I’m happy to retain just | and nothing else, but I’ll go with whatever you say
  - 4. do you think we absolutely must mark up punctuation characters explicitly?
    - I would rather not, since the special characters permitted in 3 above would be seen as punctuation marks by default, and for most other symbols the decision whether or not they are punctuation would be subjective
    - but if you say we need this, I would prefer <pc> instead of the @type of <g>, handled as follows:
      - the transliterated punctuation marks (under 3) would either NOT be tagged as

<pc> since they are implicitly punctuation marks (like a ? and ! in a Western text); OR they could be auto-tagged with <pc> to make this explicit without putting an extra burden on the encoders

  - any other symbols could get a <pc> tag around the <g> element representing these symbols if the encoder feels certain that they are for punctuation
    - if you want this, could you produce a reasonably objective definition of *when* a symbol should be tagged as a punctuation mark?
- 5. there remains the question of shorthand
  - filler and numeral shorthand is good as we have it now
  - symbol shorthand could be #siddham#, #quatrefoil# etc. as you suggest, which would be auto-converted to <g type=”symbol” subtype=”siddham etc.”/>
    - [though perhaps better to use just #siddham followed by a mandatory space, to avoid accidents whereby half of somebody’s edition gets sucked into the @subtype of a <g> because he inadvertently deleted the closing hashmark]
  - <pc> tagging could remain purely in markup, i.e. I propose that we use no special shorthand for miscellaneous symbols interpreted as punctuation marks

Segmentation in verse inscribed in spaced columns

Precedents

- **Arlo, email, 20210212** As I (vaguely) recall, we have toyed with the idea using <column> or <milestone> but in the end decided to use <space>.
- **Dan, email, 20210112** We did more than toy with it. The EGD (§3.6.5 and Example 3.6.6.A) in fact explicitly recommends milestones - optionally - for cases when the spaces between metrical units create a neat grid, while recommending <space> (§4.3.2) in general for the encoding of spaces used for segmentation.
  - Where you do have a neat grid, I would prefer milestones, but this is optional (though recommended, in the present wording of the EGD). Under §3.6.5, there is already an explicit statement saying you can choose not to do this; to this I've now added an instruction saying you should then use spaces where applicable. I shall not remove the bit on milestones now, but if you prefer to use space even for neat grids, that is OK to me. The only advantage to milestones that I can think of now is that you can number milestones and, e.g. refer to (pseudo-)columns a and b (etc.) in your commentary or metadata. I think I was under the impression, back in the early days, that this was desirable for you. Another reason to prefer milestones, at least for neat grids, is that we have a general policy of not encoding "space for layout" (§4.3.4).
- **Arlo, email, 20210212** I’d like to ask you what defines a ‘neat grid’. In visual terms both cases furnish relatively clear columns, right? Does the fact that the <milestone>s would have to fall quite often inside word pose any problem?
- **Dan, email, 20210112** I do not consider the interference of breaks with word boundaries a problem when deciding what counts as a neat grid. But this is not a rigorously definable term and I can't express it any better than that.
  - This is a neat grid:

```
XXXX XXXX
XXXX XXXX
XXXX XXXX
```

  - This is also a pretty neat grid:

```
XXX     XXXXX
```

XXXX   XXXX
XXX   XXXXX
XXXXX XXX

  - And this is not a neat grid:

XXX XXXXX
XXXX XXXX
XXX XXXXX
XXXXX XXX

  - I guess if you could draw a vertical line between two areas, then I would call it a neat grid, and if you could not draw a vertical line without cutting off a bit of one area or the other, then I would not call it neat.
- **Arlo, EGD comment, 20210319**: It seems we need to sort out this business of how to encode SEA-style verse layout once and for all!
- **Arlo, EGD comment (§4.3.2), 20210319**: reading this I am again under the impression that we recommend <space/> for representing SEA-style verse layout, whereas I thought you'd recently explained to me that <milestone> is to be used.
- **Dan, EGD comment (§4.3.2), 20210517**: To my mind, the use of spaces between stanzas or parts thereof is a phenomenon completely different from the use of space for visual layout. The two coincide in the case of your stanza columns, which is what gives rise to ambiguity. What I have in mind when recommending space is spacing after stanzas (or lines or whatever) that DO NOT line up vertically, e.g.

blah-blah-blah-blah-blah SPACE blah-
blah-blah-blah-blah SPACE blah-blah-
blah-blah-blah SPACE blah

  - In this case, the spaces serve for "highlighting some aspect of semantic structure", in the same way as spaces may be used in the other situations listed here, e.g. ends of prose sentences, or major topic changes. They are thus covered under space, since we do not use milestones for encoding such situations.
  - This is radically different from spaces that do line up, as in

blah-blah-blah SPACE blah-blah-blah
blah-blah-blah SPACE blah-blah-blah
blah-blah-blah SPACE blah-blah-blah

  - where the spaces are used to create "visually demarcated areas on a simplex surface", and are thus encoded as per §3.6. For stanza-based columns, the boundaries of these areas coincide with semantic boundaries, but in principle this need not always be the case; an inscription could be prose (as in Example 3.6.6A) and still be divided into such columns. Arguably, you could interpret the Guide to mean that for stanzas laid out in such columns, you should use BOTH space and milestone, but I see no advantage to doing that and feel that using milestones overrides the need to use spaces. So my recommendation is (and has always been) milestones for that purpose.
  - Conversely, when for an inscription laid out in stanza-based columns you choose the encoding with <space>, what you are in fact doing is to disregard the visual aspect (which the EGD permits under 3.6), and default to encoding the semantic spaces. This is not wrong, but it will not tell your reader that the stanza quarters line up in columns, since you would use the exact same encoding for spaces that do not line up.

The bottom line, 20210531

– Arlo, since you and your team have the largest number of texts with this sort of layout, I need you to decide how you want to encode them. Having thought the matter through, my preference is still definitely for milestones when there is a “neat grid” in place. See my EGD comment of 20210517 (quoted above in full) for the details. But if changing the already encoded inscriptions to that system would be too much trouble, we can leave the optionality in place or, if that is your definite preference, we can even opt to endorse only the <space> method and never use milestones in this case.

Notes for schema

Appendix #: Attribute Values

incomplete, needs more work after guide is finalised

with element	attribute	value	note
milestone	type	yati	
ab	type	colophon	
p	type	colophon	
lg	type	colophon	
fw	type	pageNum	
fw	place	top-left top top-right right bot-right bottom bot-left left inset-top-left inset-top inset-top-right inset-right inset-bot-right inset-bottom inset-bot-left inset-left	
div type="textpart"	subtype	fragment surface zone field seal	

		etc.	
milestone	unit	fragment block surface zone etc.	
seg	type	aksara component	
g	type	numeral punctuation filler symbol	
space	type	vacat defect binding-hole descender ascender	
add	place	inline below above top bottom left right overstrike	
add	rend	mark	for kākapada
del	rend	mark corrected	
lg	n		
l	n		
lg	met		
l	real		
l	enjamb		
l	met		
div type="textpart"	rend	grantha ornate large	

		tall	
fw	rend	grantha ornate large tall	
ab	rend	grantha ornate large tall	
p	rend	grantha ornate large tall	
lg	rend	grantha ornate large tall	
lb	rend	grantha ornate large tall	
hi	rend	grantha ornate large tall	
div type="textpart"	style	text-align: right text-align: center text-align: left text-align: justify	
fw	style	text-align: right text-align: center text-align: left text-align: justify	
lb	style	text-align: right text-align: center text-align: left text-align: justify	
div type="textpart"	rend	bt-rotated tb-rotated bt-upright	

		tb-upright	
fw	rend	bt-rotated tb-rotated bt-upright tb-upright	
lb	rend	bt-rotated tb-rotated bt-upright tb-upright	
supplied	evidence	parallel previouseditor lost	
milestone	type	yati	
milestone	break	yes no	
gap	reason	lost illegible omitted	
gap	quantity	{number}	
gap	unit	character line component	
space	unit	character line page	
space	quantity	{number}	
seg	cert	low	only in translations
supplied	reason	subaudible	only in translations
gap	reason	ellipsis	only in apparatus

Notes for auto-conversion

- DB 2020-03-19: the notes below are now deprecated; up-to-date and more extensive notes on auto-conversion may be found in https://docs.google.com/document/d/1fb0A7sD1MoCYF8UMDmtw28uD-RugJCG0m-Kx7iZ4fNI/edit?usp=sharing

- any number of Arabic numerals or Latin l letters followed by a plus sign to be wrapped in <g type="numeral">; the sequence of numerals or l-s is to be retained, while the + sign must disappear, and a space must be added after the closing </g>
  - regex replace ([0-9l]+)\+ to <g type="numeral">$1</g>
- any character that follows a | with no intervening white space is to be tagged with <pc>, deleting the |
  - e.g. |◊ must become <pc>◊</pc>
  - this must be done before step 2, since | on its own (followed by white space) is also a punctuation character
  - if it is easy to implement, it may be a good idea to exclude alphabetic characters as a failsafe: if | is followed by an alphabetic character, then it is to be ignored by this algorithm and picked up only by 2. below
- <pc> element to wrap one or more iterations of any of the following characters: | \ , – (en dash) =
  - iterations have no other intervening characters and no combinations of the above characters
    - so e.g. \\ becomes <pc>\\</pc> but \\– becomes <pc>\\</pc><pc>–</pc>
    - so I guess this needs five separate grep actions, not one grep with the above characters as a class
  - any character that follows a $ with no intervening white space is to be tagged with <g type="symbol">, deleting the $
    - e.g. $¤ must become <g type="symbol">¤</g>
    - if it is easy to implement, it may be a good idea to exclude alphabetic characters as a failsafe: if $ is followed by an alphabetic character, then it is to be ignored by this algorithm and the locus could be flagged for human attention or an empty <g type="symbol"/> could be inserted

Display Issues

Display in Tamil Script

Email Exchange with Manu, October 2019

(2) Another desideratum was to Markup conjunct letters in Grantha. Do you confirm that in fact no Markup is necessary? As by default, e.g. Grantha “nma” will be treated as a conjunct, while “n·ma” will not (because of *virāma*).

- DB: This may be a problem, but if I understand you right, then it's a problem for later. At the moment our transliteration guide assumes that there are no conjuncts whatsoever in Tamil inscriptions, and so the transliteration nma corresponds to na+supplied puḷḷi+ma. (See Guide 3.3.2, " for the time being our default assumption is that any consonant cluster in transliterated Tamil involves an implicit *puḷḷi* ") This works at the transliteration level, but you seem to be thinking of display in Taml/Grantha script. I'm not sure when we'll get to the stage of creating text in any Indic font from our XML editions. If we really need that, there will be many associated problems, and I believe displaying Tamil text tagged as Grantha with conjuncts and Tamil text not tagged as Grantha with separate characters will be a minor one among those. I mean, it can be solved, but let's not think about it just yet.
- MANU: OK too! For the moment we will just highlight (<hi>) the Grantha letters in our Tamil inscriptions.

  - Let us assume there are no conjuncts in Tamil inscriptions (which is untrue, for the bits in Grantha, and even for some Tamil letters, see point (3)).
  - But let us keep it in mind as I think it would be very useful to have our Tamil inscriptions also in Tamil script in our XML editions and displays, considering our possible Tamil readers (who would probably not make the effort to be acquainted with our transliteration scheme).
  - And sure, we will have problems with fonts. I will, in due time, discuss these problems with Vinodh (see the email on Grantha Unicode I just transferred to you).

(3) Interestingly we have also conjunct letters in Tamil: e.g. tt, kk, kku, nt

Markup seems necessary here as the puḷḷi is not consistently used in Tamil inscriptions.

- DB: This is definitely a problem because it conflicts with what we have in the current Transliteration Guide. Again, it is not really problematic so long as we're using texts only in transliteration, and we don't mind the lack of such a distinction. Let's discuss this a little more sometime soon, so the Transliteration Guide can be changed if necessary.
  - MANU: Yes, no emergency
- DB: What would you normally do in a printed edition? Do you consider the distinction of e.g. 1) tta written as ta+implicit puḷḷi+ta from 2) tta written as a ligature to be important? How would you distinguish them in a transliterated edition?
  - MANU: In some volumes of SII they have created and used special fonts for these letters
- DB: The idea that comes readily to mind is to use <seg type="aksara">, which we are already using for a number of purposes. This way, we'd get the following:
  - 1) tta - keeping the default, i.e. a consonant cluster means an implicit puḷḷi.
  - 2) <seg type="aksara">tta</seg> would assert that the transliterated characters t-t-a belong to a single akṣara of the original, i.e. a conjunct.
  - But the question is: is this uncommon enough and important enough to bother with? If it is important but common, then you'll have to be prepared for a lot of "bother". And we should make a decision early on, and then stick to it consistently.
- MANU: Maybe we can make it optional.

Handling partitions

DB 20191111: jottings to self about head and related stuff

- textpart: require <head> or <label>
  - numbering for internal reference only
- pagelike partitions (other than actual pages, which continue to be treated as before): display heading in diplomatic and inline label in logical editions
  - 1. require <label> and distinguish from <label> of textparts with @type
    - bad: clutter; requires attention to always adding @type
    - good: flexible

    - problem: still require @n for internal reference?
  - 2. require <label> and distinguish from textparts by requiring <head> there
    - bad: a bit of clutter; greatest deviation from standard epidoc and from tei intent
    - good: flexible
    - problem: still require @n for internal reference?
    - question: nonetheless, change over from <cb> to <milestone>?
  - 3. use <cb> with @type and @n to generate label
    - bad: less flexible than <label> but not really a problem
    - bad: cb doesn’t take @type in any TEI examples (though it could)
    - good: least clutter
  - 4. use <milestone> with @unit and @n to generate label
    - bad: need to distinguish from inline (gridlike) milestones
    - good: most TEI-compliant
    - good: relatively simple
  - how to eliminate the bad of 4 above?
    - A. use @type for pagelike milestones, no @type for gridlike (and also, require @type for any other milestones we introduce later, such as @type=”yati” already in the Guide)
      - bad: adds a bit of clutter, but not much if we only require @type for pagelike milestones
      - what could be a good value for that @type? “pagelike”?
    - B. try to rely on XSL/XPATH to check if our <milestone> precedes a <lb>?
      - but the <milestone> may be outside <p> etc elements, or it may be in the same <p> etc as the <lb>
        - seems to be a very thorny problem
- 20191112: perhaps allow some extra complexity in exchange for greatest flexibility?
- **5. we could do this**:
  - **for textparts**:
    - require @n, but allow any form (1, 2, 3; a, b, c; Ab, Cd; A, B, C; I, II, III)
    - use @subtype for general-nature textparts, so an auto-generated header is sufficient for these
    - but no @subtype for specific-identity textparts
    - in addition, optionally allow <head> for further specification
    - headings/labels would ALWAYS be auto-generated from attributes, but if a <head> is present, that would be **added to** the heading (instead of being the only thing used for heading generation, as previously planned), so e.g.
      - <div type=”textpart” subtype=”fragment” n=”1”> GETS “Fragment 1”
      - <div ... subtype=”fragment” n=”1”><head xml:lang=”En”>Upper left corner</head> GETS “Fragment 1, Upper left corner” OR “Fragment 1 (Upper left corner)”
      - <div ... subtype=”face” n=”A”><head ...>Frontal</head> GETS “Face A, Frontal” OR “Face A (Frontal)”
      - <div ... subtype=”faces” n=”Ab”> GETS “Faces Ab”
      - <div ... n=”I”><head ...>Seal</head> GETS “I, Seal” OR “I (Seal)”
      - [or, to reduce complexity, use <label> instead of head; in this scenario we don’t need a hard-coded distinction between textpart and column labels]
  - **for pagelike partitions** (other than pages): use <milestone type=”pagelike”> instead of <cb>
    - require @unit and @n in all cases and always auto-generate label/heading from

those

    - but also allow an optional <label> for further clarification, handled in the same way as for textparts above (except that in a logical edition, this gets displayed as an inline label, not a heading line)

-

DB 20191031: alternative method for textpart divs, using <head>

- see my original proposal below, without <head>
- now that Gabby says it's acceptable to add editorial <head> elements to an edition, shall we revert to that solution? It may allow us to simplify textpart subtypes and numbers while allowing editorial freedom in headings
- below, I'll try to write up a scheme for that, to see if it is really workable and efficient
- the starting point: **allow <head> elements at the top of textpart divs**
  - COROLLARY: if we want these heads to improve our life rather than make it more difficult, then we need to simplify other aspects of the markup, which we can achieve by drastically reducing the complexity of textpart @subtype and numbering
    - this means that automatically generated headings are out, so it will have to be made clear to encoders that, while adding such <head> elements is not mandatory, the displayed editions will not include a heading unless they explicitly add one
- in that case, **how shall we encode textpart divs?**
  - it's probably best if we completely discard @subtype, because we don't need it for labelling, and we (I believe) don't want computers to be able to know what kinds of textparts we have – it's enough if our readers know that
  - for the sake of machine-actionable referencing, I would then recommend that we only allow one type of numbering in textparts
    - numbering would be mandatory and always use simple Arabic numerals, since it's only for referencing and not for display; thus e.g.
      - in a copperplate set with a seal, the seal would be textpart number 1, and the plates would be textpart number 2
      - in an object consisting of three non-contiguous fragments, those would be 1, 2 and 3
      - in a stela with two facepairs in a boxlike (not pagelike) partition, faces Ab would be textpart 1, and faces Cd would be textpart 2
        - (however, I'm now leaning toward restricting the use of textpart divisions to physically separate objects, e.g. cp seals and non-contiguous fragments, and nothing else that I know of; I'll write about this separately later)
- **how shall we encode <cb/> elements?**
  - <cb/> will remain necessary for
    - stela facepairs when the transition from one facepair to the next is pagelike
    - faces of 3D objects when inscribed as text columns
    - potentially, columnlike patches of text arranged side by side or otherwise on a large surface such as a rock
  - this variety, paired with the fact that we don't want to introduce @type to distinguish the above cases, means that we'll want to allow editorial headings after each <cb/> and never use auto-generated headings for these
    - therefore, here too we can (and imo should) stick to a simple internal numbering scheme, which could be either of the following

        - nothing but lowercase letters, a b c d
            - note that it would still be perfectly OK to use things like “Faces Ab” or “Fragment B2” in the headings and in human-readable references in the commentary
        - or we could allow any alphabetic sequence to be used as numbers
        - note: if we use numbers for textpart numbering and prohibit numbers in cb numbering, we can simplify internal references in @loc, where e.g. 1:1 would mean line 1 of textpart 1, while Ab:1 would mean line 1 after <cb> Ab, without the need for the reference string to specify textpart or column
- **do we need any change in encoding <pb/> elements?**
    - I think not; pages should continue to get auto-generated headings and use the numbering style we have agreed on
    - i.e. <head> elements in pages had best be prohibited; if we want to allow them, then we’ll need extra wizardry in transformations so that no auto-generated heading is displayed when an explicit editorial heading is present
- **what about milestones in gridlike partitions?**
    - where do we need such milestones?
        - definitely:
            - contiguous fragments
            - adjacent architectural blocks
        - not so definitely:
            - stela facepairs with lines running across them
            - polygonal column facets with lines running across them
            - text divided by spaces into visual columns (often corresponding to metrical units)
    - we certainly shouldn’t use <head> after each and every <milestone> element, given that we have two or more of these in each line of the text
        - so in this case we would still have to display automated labels generated from @unit and @n
        - I suggest that we agree not to use these milestones in machine-actionable referencing
        - in that case, values of both @unit and @n can be left to the encoder’s preference, with a short list of recommendations in the guide
- **what about forme work?**
    - forme work is something of a black sheep: we need it for some special cases, but need pretty complex rules for it when we use it
    - I have not yet managed to think of a good way of displaying anything marked up as forme work
    - but I think we should not complicate it with editorial headings and just generate automated headings in some way, using @type and @location for this purpose
    - the @n of forme work items is probably not necessary for headings but will need to be used in internal referencing
        - we’ll probably have to use some explicit marker for forme work in @loc references, if we use any numeral for textpart references and any alphabetic value for <cb> references
- **costs and benefits: is it worth?**
    - what this definitely gains us is:
        - facility to use free editorial headings for textparts and cb elements
        - simplification of textpart div attributes

- most importantly, it lets us forget about textparts with “general nature” and “specific identity” and all the associated jiggery-pokery
- what it can’t give us is:
  - editorial headings for gridlike milestones, but these should be small and rigorous anyway because they will have to be displayed inline, not as heading lines
- what it costs us:
  - a little extra work to add head elements to all textparts and columns, because there will be no auto-generated headings for these [though I guess we could make the transformation smarter and generate a very basic auto-heading such as “Part 1” or “Column A” if there is no editorial <head>]
  - slight reduction of projectwide consistency, which may not be a bad thing
- bottom line: I’m beginning to like this solution, especially if we can further simplify partition encoding, see below

DB 20191029: new proposal for textpart numbering - DISCARD

- this is about using @n and @subtype in textpart divisions in a way that allows us both flexible heading display and consistent referencing, while avoiding the introduction of <head> elements for editorial headings
- I now think the best way to keep the cake and eat it would be to go as follows
- **1. general case: textparts based on “general nature”**
  - @subtype for description
    - recommended values as before, e.g. fragment, face, facet
    - free choice of other values (single word or at most two words separated by an underscore)
  - @n for identification
    - recommended values: uppercase Latin letters
    - free choice of other simple tokens (uppercase or alternating letters; double letters such as Ab and Cd; Arabic or Roman numbers)
  - in display, both of these would appear as an editorial heading, e.g. “Fragment A”
  - in internal referencing (@loc), we could use something like subtype:number.linenumber to identify lines even when the line counter is reset in new textparts, e.g. @loc=”facet:A.1” to refer to line 1 on facet A
- **2. special case: textparts based on “specific identity”**
  - @subtype mandatorily a special term, e.g. “unit” or “module” or “item” (or whatever we agree on; only one option)
  - @n for description and, if necessary, identification
    - recommended values include
      - “seal” and “plates” for copperplate sets
      - “seal_A” or “first_seal” or “larger_seal” to specify one seal in a cp with multiple seals
      - “head”, “halo”, “pedestal”, “left_thigh” and suchlike for sculptures
    - free choice of other values (single word or at most two words separated by an underscore)
  - the display transformation will have to be made aware of this special term, so that if @subtype is the special term, then only @n will be displayed, e.g. “Seal”, “Plates”, “First seal”, “Left thigh” etc.
  - in internal referencing we’d still include both of these, e.g. @loc=”unit:seal_A.1” to refer to line 1 on Seal A

DB 20191029: second thoughts about complex line numbering

- having seen one of Kunthea’s files where she used line numbers like A1, A2 … B1, B2 instinctively, I’m having some second thoughts on requiring simple line tokens
- should we reintroduce complex line numbers, such as 1r1, 1r2 … 1v1 etc in copper plates and A1, A2 … B1 in stelae
- OPTION 1: complex line numbers are mandatory in editions where line numbering is restarted from 1 at any point
  - pro:
    - perhaps intuitively the way to go for some people like Kunthea?
    - if the @n attributes of all <lb>-s are made unique in this way, then we don’t need the rigorously complex method of referencing described above (e.g. @loc=”facet:A.1”) and can instead simply use the line number (e.g. @loc=”A1”)
  - con:
    - the iteration of the higher-level counter in each line number puts extra work on encoders and opens doors to human error
    - no simple way to produce line numbers for “specific identity” textparts:
      - do we use @n=”seal.1” (as I did in Siddham) and @n=”plates.1” (as I did *not* since machine-actionable referencing was not an objective)
      - do we use @n=”left_thigh.1” and suchlike in weird layouts?
- OPTION 2: complex line numbers are expressly forbidden
  - pro:
    - simpler and less error-prone encoding
    - simpler rules to understand and follow
    - projectwide consistency is good
  - con:
    - requires complex @loc values in referencing
- OPTION 3: complex line numbers are permitted but not mandatory
  - pro:
    - flexibility
    - could give us the best of both worlds if set up like this:
      - complex line numbers are recommended in straightforward cases such as A1 and 1r1, but counter-recommended in complex and presumably rare cases such as left_thigh.1 (not sure which to prefer in complex and common cases of seal and plates)
      - whenever an edition contains only unique line numbers (thanks to the use of complex line numbers), @loc referencing can happen simply with the line number
      - but when an edition includes any non-unique line numbers (because the editor has chosen not to use complex line numbering), @loc referencing must happen in the complex way described above
    - con:
      - the guide must be pretty complex, describing both ways and advising on when to use either
      - encoders must read, understand and remember that complex guidance
      - may give rise to inconsistency

Partial resolution, 25 October

Navigational numbering: rewrite guide as follows:

- the only hard-and-fast rule is that page, column, **textpart** and fw numbers must be unique.

- everything else is recommended; the recommendations will follow what I now have in the guide as hard rules, but will include the option to reset line numbering on a new page or column, and to alternate uppercase and lowercase in the cases you described in your comment.

Textpart subtype and number: continue thinking

DB summary: My problem is that ideally, each textpart div should have a unique number so we can use just its @n (rather than a combination of @n and @subtype) to refer to any textpart. But then, in the rare cases where a text includes A) both specific and general textparts (e.g. set of plates with two seals) or B) two or more kinds of general textparts (this probably won't happen, but who knows) - in these cases we could not give unique @n attributes to textparts. I'm considering using only @subtype for display purposes, with complex values such as @subtype="Fragment_A", "Seal_A" or "Faces_Cd", and requiring unique @n attributes which would not be displayed to the reader but intended only for internal reference.

ARLO IN EMAIL 20191027:

My question here is: can we resort to use of <head> (possibly with types corresponding to textpart subtypes) to give encoders control of what they want to see as headings in their edition, and then handle <div type=‘edition’> in the way you propose? I sense that we may be able to propose a closed set of values for @subtype, but may not thus be able to cover all the varities of display that our team members might desire.

DB IN EMAIL 20191028:

I'm not sure I understand exactly what you are proposing. By " handle <div type=‘edition’> in the way you propose? " do you mean that we should then simply give unique numbers to textparts and not display those numbers to the reader?

As for adding <head>, I have thought of myself of doing so at the beginning of each <div type="textpart">. If this is what you mean, then we don't need to complicate the matter with @type; the fact that the <head> comes after an opening <div> element is indication enough that it belongs to that particular div. However, I have a deep aversion to this solution (see also my comment in the gdoc to the text " or *specific identity* where the divisions are different in nature " under " Textpart subtypes and numbers "), mainly because <head> in TEI is definitely intended to encode a heading present in the original, and not an editorial addition to a text. While we could use @resp to indicate that these headings are editorial, I still feel this would be faithless to the spirit of TEI and thus bad practice.

If you feel I could throw aside that concern, adopting <head> could be convenient for us, but I do have a couple of smaller objections to it.

One is that if we introduce numbering (when necessary) to these head elements, then those textparts will have two (different) numbers, and whenever such a textpart is added or their structure is changed, the encoder will have to remember to change both of them correctly.

The other is that I think too much editorial freedom is undesirable. It would be much better if there was a projectwide consistency (and simplicity) in how text structure is displayed. More complex details can be added to headings in print editions, and to the metadata or commentary in XML files.

We can let this matter lie for a few days and return to it later; no need for an urgent reply.

Summary 21 October

what the Guide says at the moment

item	XML element	current	note
epigraphic lines	`<lb/>`	1 2 3	
stanzas	`<lg>`	1 2 3	OR Roman numbers?
verse lines	`<l>`	a b c	OR 1 2 3?
textparts	`<div type="textpart">`	A B C	when textparts are of the same subtype, e.g. seal no distinction between “major” and “minor” textparts
copperplate pages	`<pb/>`	1r 1v 2r	
columns	`<cb/>`	A B C	no distinction between “major” and “minor” columns
gridlike partitions	`<milestone @unit="block"/>` etc.	a b c	
forme work	`<fw>`	a b c	

- the numbering of all items is always consecutive throughout an inscription EXCEPT the following:
  - verse lines restart from a in each stanza
  - <lb/> numbering restarts from 1 in each textpart div and <fw>
  - <pb/> numbering restarts from 1 in each textpart div in the unlikely case where this is applicable
- the numbering tokens always consist of a single item (no complex tokens, e.g. <lb n=”A.a.1”>
- I devised the above rules to **maximise** what I see as the **main goals**:
  - GOAL 1. projectwide consistency of numbering schemes
    - achieved by not creating any special rules dependent on language or region
  - GOAL 2. relative simplicity of schemes and the rules describing them
    - achieved by making the numbering scheme dependent on the type of encoding used: a certain type of element always gets the same sort of number
  - GOAL 3. possibility in the long-term to refer to any XML item in an edition by its @n
    - achieved by restarting counters only *within* higher-level containing elements that are themselves numbered, e.g.

    - verse lines a, b, c, d reoccur frequently, but that's OK because they are "line a *in* stanza 2" etc.
    - physical line numbers restart in textpart divs and fw (so it's "line 1 in textpart B" or "line 1 in fw b"), but do not restart on pages and in columns, because pages and columns do not enclose anything
    - NOTE: I'm not entirely sure that non-containing elements are a problem
      - I'm *certain* it's easy and straightforward to use XPATH to select items in this way, e.g. GET ME THE <lb/> with @n="x" *WITHIN* <div type="textpart"> with @n="y"
      - I *think* it's hard, and perhaps impossible, to use XPATH to select items in the way I want to avoid, e.g. GET ME THE FIRST <lb/> with @n="x" *AFTER* <pb/> with @n="y"
      - but if I'm wrong, and the latter can actually be done within a reasonable level of complexity, then there's no technical objection to restarting line (and possibly other) counters for pages and columns

Items to discuss and revise

- Arlo suggests: change **stanza numbers** from Arabic to (uppercase) Roman:
  - DB: I take it, Arlo, that you are now OK with continuing in Arabic numbers?
    - if not, please see the item "changing **stanza numbers** from Arabic to (uppercase) Roman:" in the greyed-out earlier summary below
- Manu suggests: allow **verse line numbering** in (Arabic) numbers to handle Tamil stanzas with a lot of lines (theoretically hundreds possible; stanzas of several dozen lines do occur in actual inscriptions)
  - this is a problem and I only see not-quite-good solutions to it:
    - EITHER allow <l> numbering of multiple kinds
      - but this compromises goals 1 and 2
    - OR decide to ALWAYS use 1 2 3 for verse line numbering (in which case we'd have to stick to those numbers in referencing, but we could opt to display them as a b c in most or all cases)
      - but this goes against what we like to do
    - see "laissez faire" below
    - other suggestions welcome
- Arlo suggests: allow **numbering of stela faces** in mixed A b C d scheme (regardless of whether they are encoded as <cb/> or as <div type="textpart">
  - I see no technical objection to this
  - I'm slightly against it on principle because it compromises goal 2
  - if we decide to adopt this, then I'd like you, Arlo, to formulate the rules (in the Encoding Guide) for exactly when to use lowercase letters
- Arlo suggests: **reset line numbers on each page/face**
  - as I explain in detail above, I think this would be incompatible with Goal 3
  - see complex tokens below
- NEW: Arlo suggests: allow numbering of textparts or <cb> elements encoding stela face-pairs as "faces Ab" and "faces Cd"

Laissez faire?

- we could decide that our numbering schemes are recommendations rather than rules, and people can decide to deviate from them on a case-by-case basis or consistently by subcorpus or task force

- this would allow us to permit numbering verse lines in numerals and stela faces in mixed upper- and lowercase
- it would definitely reduce our chances of making @loc references machine-actionable
- I'm worried about the inconsistencies it would introduce to the project as a whole, though these may be within a tolerable limit

Complex tokens?

- we could decide to **require** complex tokens for physical lines
  - thus, in any document that has more than one instance of <lb n="1">, ALL line numbers would include the number of the non-containing element after which the line counter is reset, e.g. 1recto.1, 1verso.1 etc; or A.1, b.1, C.1, d.1
- this would allow us to restart line numbers on every page/column, yet keep Goal 3
- BUT it would increase complexity by quite a lot (conflicting Goal 2) and open up many ways for human error to creep into our files

Misc issues related to numbering

- **textpart @subtype and milestone @unit**: prioritise functional categorisation or descriptive displayable identification?
  - DB email to Arlo 21-10-19

My recommended values for textpart @subtype and milestone @unit are meant to be short lists of clearly defined and mutually exclusive categories, and I think the key reason why you don't like them is that you want labels that are more flexible and thus more meaningful and context-appropriate when displayed.

If I'm correct to assume this, then maybe I just need to reset my thinking to prioritise that concern. I'm sure it will lead to some inconsistency, but so long as at least each encoder remains relatively consistent within their own practice, we can live with that.

Let's put this aside until after we've come to an agreement about numbering, because it also touches on specific-identity textpart @subtype names like "head", "torso", "seal", etc. If we continue to allow those, then by the same logic "face", "facet" etc. should also be acceptable in both contexts. But if we manage to find a way to encode display headings/tags separately from @subtype classification, then the same could be done for milestones as well.

OLD, IGNORE, RETAINED FOR REFERENCE Summary, 2019-09-23

- here is what our guide prescribes at the moment:

item	XML element	current
epigraphic lines	`<lb/>`	1 2 3
stanzas	`<lg>`	1 2 3
verse lines	`<l>`	a b c
textparts	`<div type="textpart">` of the same subtype, e.g. seal	A B C

copperplate pages	`<pb/>`	1r 1v 2r
columns	`<cb/>`	1 2 3
gridlike divisions	`<milestone @unit="block"/>` etc.	A B C

- *as the guide now stands*, the numbering of items is always consecutive EXCEPT the following:
  - verse lines restart from a in each stanza
  - <lb/> numbering restarts from 1 in each textpart div
  - DB current opinion: we should continue like this, but in principle, I could imagine ALSO restarting line numbers in the following cases:
    - after each <pb> in a copperplate
    - after each <cb> in text set out in columns (text flows from bottom of one column to top of next; see example below)
- *as the guide now stands*, the numbering of items other than verse lines always consists of a single counter, EXCEPT when a document has textpart divs, in which case
  - line in each div have div subtype or div subtype and number prefixed to them with . as separator
  - stanza numbers restart in each div and have div subtype or div subtype and number prefixed to them with . as separator
  - DB current opinion: we should retain the rule about restarting the counter, but we could discard the composite numbers
- **revising the numbering system**:
  - changing **stanza numbers** from Arabic to (uppercase) Roman:
    - arguments for:
      - you (Arlo) like it
      - ease of reference in @loc attributes of lemmata
    - arguments against:
      - cumbersome, especially with inss of many stanzas
      - you (Arlo) can live without it, especially if we have the option of displaying Roman numerals (but do we really want to complicate matters with that?)
      - references in @loc attributes of lemmata will need disambiguation, e.g. l1 and v1 (or st1, if you insist, but I wouldn't; see also the topmost comment at https://www.quora.com/What-is-the-different-between-stanza-and-verse about verse and stanza)
    - DB current opinion: I'd rather keep them Arabic, for no strong objective reason
  - changing **other numbering** sequences: I now think the following would be best
    - 1. **textparts**, when numbered, should continue as A B C
      - this includes minor ones, e.g. multiple seals, but it also includes major ones, e.g. non-contiguous fragments as in the EpiDoc demonstration, or separate areas of an inscription
      - I would rather not allow mixing e.g. A b C d based on an arbitrary decision of which parts are minor
    - 2. **columns** (as in a book column, where text flows from the end of one column to the beginning of the next) should be changed to uppercase lettering A B C
    - 3. **gridlike divisions** (where each line runs across two or more divisions) should

be changed to lowercase lettering

– note on stelae with 4 inscribed sides, running in pairs and flowing from one pair to the next

illustration, with numbering scheme shown below

In a hole in the a hobbit. Not a hole, filled with the an oozy smell, nor	ground there lived nasty, dirty, wet ends of worms and yet a dry, bare,	sandy hole with down on or to eat: hole, and that	nothing in it to sit it was a hobbit- means comfort.
a	b	c	d
A		B	

markup:

```
<p>
<cb n="A"/>
<lb n="1"/><milestone unit="facet" n="a"/>In a hole in the<milestone
unit="facet" n="b"/>ground there lived
<lb n="2"/><milestone unit="facet" n="a"/>a hobbit. Not a<milestone
unit="facet" n="b"/>nasty, dirty, wet
<lb n="3"/><milestone unit="facet" n="a"/>hole, filled with
the<milestone unit="facet" n="b"/>ends of worms, and
<lb n="4"/><milestone unit="facet" n="a"/>an oozy smell,
nor<milestone unit="facet" n="b"/>yet a dry, bare
<cb n="B"/>
<lb n="5"/><milestone unit="facet" n="c"/>sandy hole
with<milestone unit="facet" n="d"/>nothing in it to sit
<lb n="6"/><milestone unit="facet" n="c"/>down on or to
eat:<milestone unit="facet" n="d"/>it was a hobbit-
<lb n="7"/><milestone unit="facet" n="c"/>hole, and that<milestone
unit="facet" n="d"/>means comfort.
</p>
```

Numerals in words

Pasting older comments here after we've decided not to tag numerals expressed in words.

Aug 2019: I'm increasingly convinced that we do not want to mark these up as numerals. Nor do we want to mark up numbers written out in regular words (catvāriṁśat, etc.). Trying to do so would open the can of worms that is word tagging - I think we are not ready for that.

Show less

Axelle Janiak

11:51 21 Aug

I leave the final decision to Arlo and you, you know your material and can juge if it's worthy enough to justify the work.

Arlo Griffiths
14:48 29 Aug
What about Javanese cases like these?

1. An inscription consists only in a chronogram, and may be read/divided in various ways yielding different interpretations of @value:

vvit rūpaja saṅa naruka
vvit rūpa ja[nma] saṅ anaruk[·]

1111?
1119?
1391?

2. Or another:

dhī bhūta bhava

expressing either 1157 or 1158

In such cases, we must at least express uncertainty about the date of the inscription (in the header), and if we do so, we might as well express uncertainty about the value of the numerical expression (in the edition).

I am thus still inclined to favor allowing the possibility of encoding words with numerical value, but I'd suggest we do so with a scale from "encoding highly recommended" in the case of chronograms, to "encoding allowed but not recommend" in the case of normal numeral words outside of chronogram context.

(sometimes, chronograms themselves contain normal numeral words: e.g., śakarāje muni*nava*rasakair for 697.)
Show less

Dániel Balogh
17:05 12 Sep
To me these are the cases where <num> markup is the least applicable. To represent such complex cases properly, the markup would need to be exceedingly complex, and it would serve no purpose, since we could neither develop a way of displaying it, nor a way of querying such things selectively. The problem should be discusses in the commentary.
Show less

Header Metadata Archived

Appendix #: Recording Dates Rigorously

This machine-readable shall be put in the attributes and must be expressed following the ISO 8601/W3C date syntax.

In this standard, a year should be recorded with 4 digits with addition of leading zeros when necessary. A month and a day are to be represented with 2 digits with a leading zero as necessary. A en dash separates the year, the month and the day from each other.

- yyyy (year known)
- yyyy-mm (year and month known)
- yyyy-mm-dd (year, month and day known)
- --mm (month known)
- --mm-dd (month and day known)

The date before Christ/before common era should be recorded as negative numbers, being preceded by an en-dash. (i. e. 56 BC gives -0056) Note that there is no year 0000, the system starts at 0001.

Calendar related notes

Manu on dates in metadata

I do think it's a good idea to provide specific columns in the spreadsheet for the date tag attributes required in the TEI EpiDoc file.

In my view, the comprehensive list is:
@datingMethod | @when-custom | @notBefore-custom | @notAfter-custom

Bear in mind that @when-custom and the pair (@notBefore-custom | @notAfter-custom) are mutually exclusive.

Arlo on dates in metadata

Your answer does raise the new question of what to do about representing the indigenous dating system. In the case I am now working on, for the dates with @datingMethod='Śaka', we have four variables as opposed to three for @datingMethod='Julian'. (The four are year, month, fortnight, day.) I am rather strongly inclined not to separate out the possible values in columns also for such Śaka dates, because this is probably just a tip of the iceberg of all the complexity that we may expect in the entire DHARMA corpus, and we can always apply more structure to the relevant data once they have been migrated to XML.
...
certain events recorded epigraphically typically occur in certains months and on certain quantièmes, and not on others. … My instinctive response is that the way I propose to represent Śaka dates with four variables in a single column of a spreadsheet should allow later processing of such data in some way, even if it isn't through @origDate. And if that doesn't work, we can presumably also tag the relevant words in the texts of the inscriptions themselves, to build indexes etc. on that basis

Dan on dates in metadata

I still suggest a free-text field to record the original date for possible later processing. The editor would distil any date info in the inscription into that free-text field (year | expired or current or unknown | certainty and alternative readings of the year | month or season info | paksa if any | tithi or divasa if any | weekday if any | possibly any other relevant astronomical data if available). And note in this field the line number or other reference to where the inscription text has the date.
Another thing. I have strong qualms about recording exact dates in the Julian calendar. How do we even know that Julian is more accurate for our purposes than Gregorian? How do we know whether the guy who recorded the date used a kārttikādi or a caitrādi year? Expired or current? How do we know what text he used to calculate his tithis? It requires a tremendous amount of

work to make educated guesses on such factors. I would suggest a Julian equivalence no finer than the level of years, and usually a range of 2-3 years in most cases, unless a specific date can be verified e.g. against a weekday or astronomical event.

AJ on rigorous dates

The attribute @calendar defines the calendar used for the content inside the <date> or <origDate> element. On the other hand, the attribute @datingMethod sets the calendar of the dates given inside the attributes. By default, its value equals '#julian'. [Should I expect another calendar as reference? if not, the Julian reference could be added automatically. Check the necessity to display it or "mute" it.]

Both @calendar and @datingMethod work with a pointer. It means the value of the attribute is a reference to a feature defined in the file, i. e. inside the <calendarDesc> of the <teiHeader>. [Possible to point toward an external resource through a URI, but I doubt the use here.] The value itself combines a # and the value of the @xml:id identifying the resource as such calendar="#julian" calendar="#śaka" or datingMethod="#julian". [to rewrite -> could be clearer; is the pointer system explained somewhere else so I could avoid explaining it here?]

E.g.

[Set an example that fits the project, but it is a good example from Epidoc Guidelines.]

```
<calendarDesc>
<calendar xml:id="creationOfWorld">
 <p>"Since the creation of the world"</p>
</calendar>
</calendarDesc>
<!--  -->
<date when="1382-06-28"
 when-custom="6890-06-20" datingMethod="#creationOfWorld"> μηνὶ ’Iουνίου
εἰς <num>κ</num> ἔτους <num>ςωϟ</num>
</date>
```

The attributes containing the dates are the following: @when-custom, @notBefore-custom, @notAfter-custom, @from-custom and @to-custom. We have to go for the class 'att.datable.custom' since the others in the TEI are set on the Gregorian calendar.

@when-custom should be used for a precise date, i.e. <origDate when-custom="0783-05-16" datingMethod="#julian" calendar="#śaka">10 waxing Śuci (= Jyeṣṭha), 705</origDate>

@from-custom and @to-custom can be used as a pair for a precise range date.

@notBefore-custom and @notAfter-custom can't be combined with @when-custom. They mark the beginning and end of the possible span of dates, i.e. <origDate notBefore-custom="1421" notAfter-custom="1422" datingMethod="#julian" calendar="#śaka">1343</origDate>

[Would you want to date an inscription thanks to an @period? what about @evidence?]

Precision and certitude for dates.

Among those dates, some are not given as confidently as others. The following lines set the rules to characterize the certainty and precision of them.

According to Epidoc Guidelines, if the date is associated with a question mark or is expressed as just a possibility, the certitude should be at a low level with an attribute @cert. This one indicates the degree of certainty you have on the encoding.

[Honestly, I am not convinced by the description given in the Epidoc Guidelines for the certitude and precision Guidelines. If we were to follow them, the following examples, taken from EIAD, should have a precision on medium rather than low since it is based on palaeographical reasons.

<origDate calendar="Julian" notBefore="0300" notAfter="0400" precision="low">Undated. Attributable to the 4th century CE on palaeographic grounds.</origDate>

<origDate calendar="Julian" notBefore="0001" notAfter="0100" precision="low">Undated. Attributable to the first century CE on palaeographic grounds.</origDate>

Moreover, they give 3 different encoding methods. Do you feel it necessary for such a range of possibilities and nuances while you encode your material? What have you done so far? I have taken the following example from Campa, how would encode the precision in this case: high?

```
        <origDate>
                    <date calendar="Śaka">14th century, possibly 1365</date>
                      <date calendar="Julian" notBefore="1443" notAfter="1444">15th c., possibly 1443/4</date>
                 </origDate>]
```

[not sure where is the best place for this paragraph? Solution given by the Epidoc Guidelines: are you satisfied with it?]

The element <origDate> can be repeated to record different dates because several texts are on the support or any other situation implying multiple dates.

[We could use @corresp to link <origDate> with texts… the possibility must be checked.]

E.g.

```
<history>
        <origDate datingMethod=”#julian” calendar="#śaka">Text 1: 10<hi rend="superscript">th</hi> century</origDate>
        <origDate notBefore-custom="1233" notAfter-custom="1234" datingMethod=”#julian” calendar="#śaka">Text 2: 1155</origDate>
```

example taken for Epidoc Guidelines: about prose description and combination of dates inside the prose. Necessity?

```
<origDate notBefore-custom="1409" notAfter-custom="1410" datingMethod="#julian">The text records a composition date of <date calendar="#śaka">1331</date>(Śaka)</origDate>
```

DB on Julian calendar

very problematic and I know too little. Some thoughts:

1. Axelle's notes (now moved to the end of this doc) say that @datingMethod is by default Julian, in which case we needn't include it. But it seems to me from everything I read that @datingMethod is by default Gregorian. In that case we do need this attribute if we really want to use Julian.

2. do we actually need and want Julian? is any of us expert enough on calendar studies to decide this? from what I gather, the Julian calendar was used in most parts of Europe from around 709 to the late 16th century, and it is the custom (/rule) of European historians to express all pre-16th-century dates in the Julian calendar.

2A. The first question is: did the scholars who write in detail about Indic calendrical systems use the Julian calendar when they converted Indic dates to the Christian calendar (or whatever they called it), or did they use Gregorian or neither?

2B. The second question is: do we care? Are any of our dates accurate enough for this distinction to matter? Apparently (https://en.wikipedia.org/wiki/Conversion_between_Julian_and_Gregorian_calendars), the difference between Julian and Gregorian dates was never more than 5 days in the first millennium CE, after which it grew gradually to reach 10 days by 1500 CE. Most of our dates are inaccurate in a range of a year and a half plus or minus, and for the dates we consider very accurate (because e.g. verified against astronomical events) we don’t clearly know (or at least I don’t) whether those astronomical events were catalogued in Julian or Gregorian.

So: couldn’t we simply decide to use the Gregorian calendar for all our dates and thus get rid of at least some of the complexity of our markup?

Describing the Original Document

- the final element of the `<fileDesc>` is the source description, `<sourceDesc>`
  - this mandatory element records details of the original from which the digital text is derived
- TEI permits the use of various elements in a source description, but in the case of epigraphic documents its only child element is `<msDesc>`, signifying “manuscript description” and applicable to any text-bearing object besides manuscripts in the strict sense
- the following subsections describe the EpiDoc-sanctioned elements of the manuscript description as used in our project
  - many of these elements are nested within lower-level grouping elements; see the structure overview in § for precise details

Identifying the support

- the description of the support begins with its identification, contained in the element `<msIdentifier>`
- the **primary identification** of the object shall be stated using an applicable combination of the following elements
  - `<settlement>`, containing the name of the settlement where the object is located
    - optionally add `<country>`?

  - <institution>, containing the name of the museum or other institution on whose premises the object is found
    - objects in situ: omit <institution> or name caretaking organisation such as ASI where applicable?
    - @type/xml:id? [Note AJ: I like EAID syntax for @xml:id better than the one used in Campa.]
  - <idno>, where applicable, containing the reference used by the caretaking organisation to identify the object (acquisition number, inventory number, etc.)
    - @type?
  - if any of these elements are not applicable to the object you are recording or the information cannot be obtained, omit the element altogether instead of leaving it blank
- if there are any other relevant identifiers (e.g. number in a thematic inventory compiled by an institution; or a pre-existing online corpus)
  - for each such ID create an <altIdentifier> element within the <msIdentifier> element
  - and within <altIdentifier> , use <settlement>, <institution> and <idno> as applicable to record the additional identifier
- examples by AJ:

```
<msIdentifier>
                <institution>EFEO</institution>
                <idno xml:id="EIAD0001_inv-eiad" type="inv-eiad">EIAD 1</idno>
                <altIdentifier type="museum-inventory">
                        <repository>Nagarjunakonda Museum</repository>
                        <idno xml:id="EIAD0001_museum-inventory">79</idno>
                </altIdentifier>
</msIdentifier>

 <msIdentifier>
        <institution>EFEO</institution>
        <idno xml:id="inv-general">C. 152</idno>
        <altIdentifier type="cancelled">
                 <institution>EFEO</institution>
                  <idno>C. 166</idno>
        </altIdentifier>
         <altIdentifier type="museum-inventory">
                <settlement>Đà Nẵng</settlement>
                <repository>BTC</repository>
                <idno xml:id="inv-btc">BTC 86</idno>
        </altIdentifier>
        <altIdentifier type="museum-inventory-former">
                <settlement>Đà Nẵng</settlement>
                <repository>Musée Parmentier</repository>
                <idno xml:id="inv-musee-parmentier">45,2</idno>
          </altIdentifier>
 </msIdentifier>
```

Classifying the text by topic

Gabby at Berlin Kickoff: several ways of doing that; EpiDoc sanctioned way soon to be decided; probably involving an element called textClass

there are several lists of existing terminologies for this to consult (e.g. EAGLE)

msContents: DISCARD?

- DB: I don’t think we should use this element; brief reasons:
  - it is not mentioned in the EpiDoc guidelines
  - the reason why it is in TEI seems to be for describing codexes comprised of several, partly or completely independent manuscripts - not relevant to epigraphy unless we want to use it in the case of objects bearing multiple inscriptions [which would bring up a whole lot of new problems]
  - the purpose for which it is used in Axelle’s examples (from the Campa corpus?) below can, as far as I can see, be served perfectly well by using @xml:lang in the <div type="edition">, which is not only endorsed by explicitly required by EpiDoc
    - any additional languages can be taken care of by adding @xml:lang to <div type="textpart"> or <seg> elements within the edition
- retaining original notes by AJ:

  The next section is <msContent>. This tag is one to describe the intellectual content of the text.

  [TBD: would you need of the tag <summary>?]

  It contains the <msItem> inside which you can type the tag <textLang> to describe languages and writing systems of the text. It can be associated with the attribute @mainLang which must be used only for the dominant language, the others are to be registered thanks to the attribute @otherLangs. The values of both attributes should be the codes of each language.

  If you need to record more than on language in the attribute @otherLangs use a blank space between each language code.

  [Those code need to be set. Do you want to give an xml:id here?]

_E.g._

```
 <msContents>
         <msItem>
                  <textLang        xml:lang="eng"        mainLang="pra-Brah">Middle
Indo-Aryan, Southern Brāhmī script</textLang>
         </msItem>
</msContent>
```

_E.g._

```
<msContents>
          <msItem>
                   <textLang mainLang="x-oldcam-Latn-CI">Old Cam in Old Cam
script</textLang>
               </msItem>
</msContent>
```

Describing the support

- the mandatory description of the physical support is wrapped in the element `<support>`, which is the first (and, in our case, only) child of the grouping element `<supportDesc>`, which in turn is the first child element of `<objectDesc>`, which in its own turn is the first child element of `<physDesc>`, which follows

`<msIdentifier>` within `<msDesc>` (see § for an overview of the structure)

- DB: the following is my recommendation based on Siddham practice
- the contents of the `<support>` element consist of two parts: a structured list of mandatory metadata optionally followed by one or more paragraphs of freeform text
- the **structured list** is comprised of all of the following elements, in this order:
  - `<material>` containing the name of the material, e.g. stone, copper, etc.
  - `<objectType>` describing the class of object, e.g. stela, pillar, architectural block, etc.
  - `<dimensions unit="cm">` containing all of the following elements:
    - `<width>` containing the object's largest extent left to right
    - `<height>` containing the object's largest extent top to bottom
    - `<depth>` containing the object's largest extent front to back
    - all of these dimension elements must contain only numbers
      - the unit for all measurements shall be centimetres, as specified in the `<dimensions>`
      - if your data are accurate to the millimetre, record one decimal digit using a decimal point
    - dimensions are to be understood relative to a viewpoint facing the object's inscribed side or, if there are multiple inscribed sides, the side where the inscription commences
- the **freeform text** following the structured list shall be wrapped in the element `<p>` and contain a human-readable description of the object
  - this description should not repeat any of the information encoded in the structured list, but briefly mention any relevant information not already covered there, especially:
    - a description of the shape of the object, including any relevant measurements of its various parts
    - the location of the inscription on the object, if not self-evident
  - the description may contain additional markup as permitted in § and must, in particular, include markup for any bibliographic references (see §)
    - any secondary dimensions (including their units) mentioned in the freeform description should be tagged with the elements `<width>`, `<height>` or `<depth>`, as applicable, e.g. `<p>`The upper section of the pillar is octagonal and only `<width>`thirty centimetres`</width>` wide.`</p>`
- DB: ideally, we should use controlled vocabularies for the structured elements material and objectType. Do we want to adopt a pre-existing one? Or shall we allow these to be freeform, and then harvest data and some point and create controlled vocabularies of our own?
- DB: do we want to indicate the source of our dimensions, i.e. name a publication or specify "Autopsy, X. Y., ####.##.##?"
- DB: do we want to provide separately for approximate dimensions or just mention in the freeform text that certain dimensions are approximate when applicable?
  - [AJ: Would you like to have access to attributes like @atLeast and @atMost?]
- AJ: If the dimensions you need to record are not linear, like an arc, a circumference or diameter, use <dim> combined with an attribute @type. [Do you want this possibility or do you rather use only the linear dimensions]
  - DB: this seems very complicated to me. From my limited experience, I'm happy to record the dimensions of the containing rectangle for e.g. a circular column. The width of the inscribed field (in Layout below) is, however, recorded as a surface

dimension.

- DB: any particular method for handling unknown dimensions (e.g. depth not reported for many slab inscriptions and hardly ever reported for copper plates)
  - AJ: If you don’t have access to the dimensions of the object, please write “dimensions unknown” and toggle the elements in a commentary. If you are able to get the information later, it will be easier for you to add it if you keep the structure. [I recommend this solution to avoid the rendering of EIAD: h. unavailable × w. unavailable × d. unavailable cm. that I find not really satisfactory]
  - DB: I see no problem with “h. unavailable ” etc but think that if some of the elements are commented out, the file may not validate against the EpiDoc schema; I may be wrong about that. In Siddham we used empty elements, e.g. <depth/> when one of the data was not available.

- alternative examples by AJ: only freeform text, but containing the essential metadata with tags

```
    <physDesc>
        <objectDesc>
            <supportDesc>
                <support>
                    <p>Bottom part of an octagonal
                        <objectType>pillar</objectType>.
                        Dimensions:
                        <dimensions unit="cm">
                            <height atLeast="116">116</height>
                            <width>31</width>
                            <depth>22.5</depth>
                        </dimensions>.
                    </p>
                </support>
            </supportDesc>
        [</objectDesc>]
    [</physDesc>]
```

```
<support>
    <p>Back of a sculptural
        <objectType>stela</objectType> of Viṣṇu; apparently of
        <material>sandstone</material>; according to
        <bibl><ptr target="dharma-bibl:cabaton1904"/>: 687</bibl>,
        the sculpture itself is maximally
        <dimensions unit="cm">
        <height>150</height>
        </dimensions>
        cm in height.
        The dimensions we have ourselves recorded (
        <dimensions unit="cm">
        <height>136</height>
        <width>83</width>
        <depth>40</depth>
```

```
            </dimensions>) concern the sculptural stela itself, in its present state of
        insertion in a large altar, which may explain the discrepancy for height.
        </p>
</support>
```

Describing the layout of the inscription

- the mandatory description of the layout is wrapped in the element `<layout>`, which is the first (and, in our case, only) child of the grouping element `<layoutDesc>`, which in turn is the second and last child element of `<objectDesc>`, which in its own turn is the first child element of `<physDesc>`, which follows `<msIdentifier>` within `<msDesc>` (see § for an overview of the structure)
- DB: the following is my recommendation based on Siddham practice
- the contents of the `<layout>` element consist of two parts: a structured list of mandatory metadata optionally followed by one or more paragraphs of freeform text
- the **structured list** is comprised of the following element and child elements:
  - `<dimensions unit="cm">` containing all of the following elements:
    - `<width>` containing the inscribed field's largest extent left to right
    - `<height>` containing the inscribed field's largest extent top to bottom
    - these dimension elements must contain only numbers
      - the unit for all measurements shall be centimetres, as specified in the `<dimensions>`
      - if your data are accurate to the millimetre, record one decimal digit using a decimal point
    - dimensions are to be understood relative to a viewpoint facing the inscription
    - for inscriptions spanning multiple adjacent surfaces of an object, the dimensions of the inscription as a whole are to be recorded here
    - however, for complex inscriptions covering non-adjacent surfaces (e.g. especially copper plates), record the typical dimensions of one surface
- the **freeform text** following the structured list shall be wrapped in the element `<p>` and contain a human-readable description of the layout
  - this description should not repeat any of the information encoded in the structured list, but briefly mention any relevant information not already covered there, especially:
    - the height of lines, measured as the average vertical distance from the baseline of one line to the baseline of the next line (i.e. including character height)
    - the height of characters, measured as the average height of one *akṣara* from baseline to headline (without any descenders and ascenders)
    - a description of the shape of the inscribed field (*campus*) if it is not rectangular
    - for complex inscriptions with multiple boxlike divisions: the relative position and size of each
    - any lines shorter than the width of the inscribed field
    - the location of any extensive areas of weathering or loss
    - the directionality of writing if other than top to bottom left to right
  - the description may contain additional markup as permitted in § and must, in particular, include markup for any bibliographic references (see §)
    - any secondary dimensions (including their units) mentioned in the freeform description should be tagged with the elements `<width>` or `<height>`, as applicable, e.g. `<p>`The last two lines are only `<width>`7.6 cm`</width>` long.`</p>`

Describing the script

- in epigraphic parlance, a “hand” means a particular combination of writing features
- the “hand” or script of an inscription is to be described in the element `<handDesc>`, which is the second (and last) child element `<physDesc>`, which follows `<msIdentifier>` within `<msDesc>` (see § for an overview of the structure)
- for inscriptions written in a single hand throughout, simply create a `<p>` element within `<handDesc>`, containing a freeform description of the script including in particular
  - a conventional palaeographic classification of the script, if applicable
  - any information on palaeographic dating
  - a description of any punctuation marks beyond the conventional forms specifically listed in §4.2 of the Transliteration Guide
  - a description of any other symbols, including space fillers appearing in the text
  - for Indonesian texts, record here whether the “Indian” or the “Indonesian” mode of reading the superscript *r* marker (about which see §) is the default (dominant) for that particular text
    - to do so, use the standardised expressions “r-mode: Indian” or “r-mode: Indonesian”
    - keep in mind any instances of the other method within the same text will need to be marked up as per §
- **if your inscription is written in several**, clearly distinguishable **hands or scripts**, and your edition encodes the hands responsible for various parts of the inscription (as per §), then and only then do the following:
  - to the `<handDesc>` element, add the attribute @hands, whose value shall be the number of different hands in the document, e.g. `<handDesc hands="3">`
  - instead of or after the freeform hand description, create one `<handNote>` element for each hand,
    - with the attribute @xml:id with a value consisting of an Arabic number assigned to the hand, prefixed with a lowercase h, e.g. h1, h2, etc.
      - this identification label is necessary in order to be able to refer to hands from the edition
    - containing a human-readable description of the hand

decoDesc: DISCARD?

- DB: I don’t think we should use this element, primarily because I think decorative and iconographic details have a more natural place under the object description (e.g. sculptural paraphernalia) or the hand description (e.g. calligraphy and ornamental symbols)

The origin of the inscription

- data about the creation of the inscription must be recorded in the element `<origin>`, the first child element of `<history>`, which in turn is the last child element of `<msDesc>`, appearing after the `<physDesc>` element (see § for an overview of the structure)
- DB: the following is my recommendation based on Siddham practice and my current thoughts. there’s a lot to decide about dating.
- the origin shall in our practice be comprised of up to two parts:
  - a rigorous encoding of the date of creation (see § below)
  - an optional paragraph(s) of freeform English text wrapped in `<p>`,
    - describing any available information about the circumstances in which the

inscription was created, including in particular

    - separate information about the creation of the support and the inscription if the inscription was added to the support at a later time
    - information about any premodern relocations
  - this paragraph may contain markup permitted under § as necessary, and should in addition include the element `<origPlace>` tagging the name of the place where the object was created, if known
    - [AJ: I guess we could set a @ref or at @key toward a file containing the geographical information: dharma-geo:something]

The date of the inscription

- the date of the inscription must be recorded in the element `<origDate>`, which shall be a child element of `<origin>` (see § for an overview of the structure)
- the **contents of this element** shall be a human-readable description
  - where available, of the internal date of the inscription followed by the CE equivalent, e.g.
    - Gupta Era 82 āṣāḍha śukla 11, ca. 402 CE
    - Śaka Era 556, ca. 633 CE
  - where no internal date is available, the basis of estimation followed by the estimated date, e.g.
    - palaeographically the 8th or 9th century CE
    - in the reign of Samudragupta, ca. 350-375 CE
    - sixth regnal year of Siri-Vīrapurisadatta, ca. 225-275 CE
- the **attributes of this element** shall constitute a rigorous record of our knowledge of the date and include the following, as applicable
  - see Appendix § on the format of dates used as attribute values
  - @datingMethod="#julian"
  - @when-custom: to record the date CE when expressed as a single year
  - @notBefore-custom: to record the earliest possible date CE when expressed as a range of years
  - @notAfter-custom: to record the latest possible date CE when expressed as a range of years
    - note: it is mandatory to record a date, either as @when-custom or as a combination employing both @notBefore-custom and @notAfter-custom
  - @evidence: to record the basis of dating, with one or more of the following values (separated by a space in case of multiple values)
    - internal-date: if the text has an explicit internal date
    - lettering: if the inscription is dated on a palaeographic basis
    - prosopography: if the basis of dating is the text's reference to dateable persons such as rulers
    - context: if the basis of dating is the archaeological, epigraphic, iconographic or other context of the text support
    - DB: anything else? EpiDoc further suggests "nomenclature" and "titulature", which I think aren't relevant to our field
  - @certainty="low" if the date or date range given in the above attributes is regarded to be uncertain (as distinguished from inaccurate, see @precision below)
  - @precision: if the date or date range given in the above attributes is regarded to be approximate, with the following permitted values

    - medium: for a date deemed to be accurate within up to a decade plus or minus
    - low: for a date deemed to be accurate within up to a century plus or minus

Subsequent history

- significant episodes in the modern history of the object shall be recorded as `<provenance>` elements within `<history>`, after `<origin>` (see § for an overview of the structure)
- for each event you wish to record, create one `<provenance>` item in chronological order
  - the contents of each item shall be a concise English description of the event
  - if parts of the object have separate histories (e.g. were found in different circumstances or are kept at separate institutions), create provenance items for events in each history
  - [AJ suggests wrapping the description in <p> within <provenance>. Does this serve a useful function? Some examples in TEI documentation use <p>, others don't. EpiDoc documentation doesn't mention <p>]
  - provenance elements may be classified using @type and @subtype, for which EpiDoc recommends values
  - [AJ has collected those values below. DB: I retain these for the time being, with a couple of comments. Some of these would be useful for us because we might want to use them as search parameters, e.g. "I want to search only in autopsied inscriptions". But the scheme is very complex and I don't see a lot of advantage to encoding provenance types rigorously; the freeform contents should be sufficient for most purposes. If we decide not to use provenance typing, I'll want to use this list to create some recommendations for the sort of events to record in provenance.]
  - <provenance type="found"> - information about the first appearance, or key re-appearance, of the text-bearing object in modern times
    - @subtype : "discovered", "rediscovered", "first-seen" and "first-recorded".
  - <provenance type="observed"> - information about subsequent modern observations
    - @subtype : "seen", "recorded", "identified", "photographed", "autopsied", "squeeze-taken", "rubbing-taken", "ink-transfer-taken", "reported", "built-into" and "reappeared"
  - <provenance type="not-observed"> - information about a specific, unsuccessful attempt to locate an object in a presumed or previously recorded location
    - @subtype : "lost", "stolen", "destroyed", "drawn", "reported-lost", "reported-stolen" and "reported-destroyed"
  - <provenance type="transferred"> - information about documentable modern relocations of the text-bearing object
    - @subtype : "moved", "sold", "given" and "loaned"
  - [DB: all in all, I suggest we adopt these four types, but none of the subtypes - $rewrite accordingly]
  - in addition to @type, each `<provenance>` element must have a date marked up either as @when-custom or as a combination employing both @notBefore-custom and @notAfter-custom
    - these attributes shall be used as described under § above, except that the dates should be given in the Gregorian calendar
      - DB: shall we prescribe month and date for autopsy by team members, or is a

year sufficient in all cases?

- the text of each provenance item may contain additional markup as permitted under §
- AJ: It is possible to add a bibliographic reference inside your description of the event with the element <bibl>
- DB: it may be a good idea to make this mandatory where references for events are available

Example by AJ:

```
<provenance type="found" when="1928">
        <p>According to the first supplement to Cœdès' inventory, the object bearing this unique inscription was found at Khánh Lễ in Bình Định province (see <bibl><ptr target="dharma-bibl:coedes1937-1966"/>, <biblScope>vol. I [1937], p.273</biblScope></bibl>).[...]
        </p>
</provenance>
```

$Additional settings for <msDesc>

The section following <history> is <additional> which regroups several information of administrative, bibliographic or curatorial kind. The most interesting element of this section is <surrogates> which contains the description of the representations of the object.

[Don't know the material enough to make a decision here. Shall we follow the same description schema than EIAD & Campa? photo; photo-estampage; rti; representation for others digitized stuff.]

Make sure to close the <msDesc>, the <sourceDesc> and the <fileDesc> elements. Before starting the next huge section: <encodingDesc>.

Header Elements Beyond the File Description

Encoding description: <encodingDesc>

The <encodingDesc> documents the relationship between an electronic text and the inscription from which it was derived.It is the second major component of the<teiHeader>.

TBD

Something like that could be of use?

<samplingDecl> (sampling declaration) contains a prose description of the rationale and methods used in sampling texts in the creation of a corpus or collection.

+ <classDecl> +<p> with a ref towards the Epidoc schema and eventually our own guidelines.

Profile Description: <profileDesc>

The third section of the <teiHeader> provides a detailed description of non-bibliographic aspects of a text and its context. The main point of interest here is the <langUsage>.

[start from here]

adding a calendarDesc

The <revisionDesc> (revision description) follows the changes made in the file and by whom.

Tagging script in multi-script inscriptions

with the added complication of the possibility of reading a character in either one script or another

- **initial problem from Manu** in GitHub Issue https://github.com/erc-dharma/project-documentation/issues/110, q.v. for details
  - **Dan's summary of problem (20210531)**: our current system of using e.g. <hi rend="grantha"> cannot be used when there is a <choice> of two <unclear> elements in different scripts (because EpiDoc prohibits <hi> within <unclear>), but such cases can occur (e.g. unsure whether a certain character is a grantha śa or a Tamil ca)
- **Dan's summary of MARKUP list responses (20210531)**:
  - 1. EpiDoc will *not* permit anything but <g> inside <unclear>, because to them everything else is "editorial interpretation", not "character data"
    - we can live with that, though I do not fully agree
  - 2. for dealing with the problem of two different script possibilities for one character, Gabby suggests <app type="alternative">
    - I would prefer not to complicate our markup and file handling with this and suggest we search for a different method
  - 3. nobody in the EpiDoc community agrees with our use of <hi> as a container for segments of text written in a different script
  - 4. for marking up script, everyone who spoke up in the MARKUP discussion thinks we should use @xml:lang
    - in particular, Gabby simply suggests <seg xml:lang="Gran">śa</seg>, arguing that there is no need for "tam-Gran" since we can take "tam" to be inherited from the parent element
      - should we go this way, we should probably use "Qgra" or suchlike to indicate that we are using script tags in a way other than the defined standard
    - hand-in-hand with the above, the EpiDoc community seems to think we are wrong to use the -Latn suffix in our editions, since our markup should record the script of the original (if anything), and not that of the edition
      - it is also implicit in the EpiDoc guidelines that when they use a script subtag, they use it to encode the script of the original, and not that of the edition
        - e.g. at https://epidoc.stoa.org/gl/latest/supp-language.html you find egy-Egyd for "Egyptian in Demotic script"; egy-Egyh for "Egyptian in Hieratic script" and egy-Egyp for "Egyptian Hieroglyphic" - and I don't think scholars publish editions of Egyptian texts in those original scripts
- **Dan's summary of separate problems, 20210531**
  - **problem 1: shall we use @xml:lang for identifying the script of the original?**
    - Gabby on MARKUP: "I'm not sure how I feel about using `@lang="sa-Latn"` to represent the fact that your transcription of this text is in Latin transliteration, rather than representing the script of the original inscription itself, partly for this

very reason—that you might want to encode the script of the source text—and partly because I'm not sure how true that is to the spirit of EpiDoc and TEI use of the attribute. It's potentially ambiguous, of course, and I know it has been discussed here before, but I did want to flag that”

  - I do not know why we use -Latn, except that this seems to have been inherited from an earlier project (Champa? EIAD?), and I guess it was suggested by Tom Elliot back then
  - Axelle said over Zoom that we should stick to -Latn and that using @xml:lang for the script of the original would be contrary to TEI and/or that it would be contrary to RFC
  - I find no explicit instructions, either in the TEI guidelines or in RFC5646, pertaining to cases where an original source is represented in a transliterated digital edition
    - I realise that marking up our text as -Latn may be useful for vague and undefined future purposes where someone not intimately familiar with our project might want to machine-process our corpus, and their machines would be misled into thinking our Sanskrit text was written in Devanagari unless we specified -Latn for it
    - but I’m not sure this is a problem
  - at any rate, there were also strong opinions voiced on MARKUP at an earlier time, for using xml:lang to describe the contents of the element, and not the original
    - Tom Elliot on MARKUP (20191029)
    - Hugh Cayless on MARKUP (20191029) “My inclination would be to describe the languages and scripts of the source in the profileDesc and to use @xml:lang strictly to describe the language/script of the content of the element bearing that attribute”
  - **it would be good to know what other people who edit exotic originals in transliteration do**
    - I could not locate XML files for Gandharan inscriptions, Maya hieroglyphics and for Mesopotamian inscriptions
    - in the Beta maṣāḥəft project they seem not to use script subtags at all, and they happily have e.g. <title **xml:lang="gez"** xml:id="t1">ድርሳን፡ ዘአባ፡ ያዕቆብ፡ ወተመይጠት፡ ማርያም፡ ትግባእ፡ ቤታ፡ እምድኅረ፡ ዝንቱ፡ እስመ፡ በጽሐ፡ ጊዜ፡ ትለድ፡ ወለተ፡ ሌዋዊያን፡</title> <title **xml:lang="gez"** type="normalized" corresp="#t1">Dərsān za-ʾabbā Yāʿqob wa-tamayṭat Māryām təgbāʾ betā ʾəm-dəḫra zəntu ʾəsma baṣḥa gize təlad walatta Lewāwiyān</title>
      - so they tag both the original Ethiopic (or whatever it is) and the transliteration simply as “gez”
  - Dan’s bottom line on problem 1: I have now been brought around and agree that we should continue to use -Latn for our transliterated texts, and not attempt to use @xml:lang to encode the original script
- **problem 2**: if not @xml:lang, then what *should* we use to encode script?
  - Arlo and Axelle have proposed a method involving @corresp, which I have now (20210604) written up in the EGD following their suggestions
  - I still see this as somewhat problematic
  - The TEI definition of @corresp is: “points to elements that correspond to the current element in some way”
    - I think saying that an OpenTheso script token “corresponds in some way” to

let's say a stanza written in a particular script is an extreme stretch and is definitely not what the authors of TEI intended

      - (TEI guidelines give two examples for the use of @corresp: an exact one, where a sentence in one language corresponds to a sentence in the translation of that text to another language; and a fuzzy one, where the city London in a gazetteer corresponds to a personification of that city in a literary work)
  - So why don't we use @rend, which we had in the EGD previously, and just change the former <hi> to <seg> where a phrase-level container is needed?
    - TEI definition of @rend: "indicates how the element in question was rendered or presented in the source text"
    - is it that you want to avoid multiple items in the value of @rend?
      - But we already permit multiple values in our use of rend to date, even if they will occur rarely (e.g. @rend="bt-rotated ornate", example from EGD §7.5.2)
      - and you already call for two values in @corresp (class and maturity)
      - is it really that much of a complication that in perhaps one percent of our texts, a third value will have to be added to those two, and in perhaps one percent of that one percent, a fourth value too?
  - OR, as a new alternative, why don't we use @rendition?
    - This attribute is perhaps better than rend for canonical references to our controlled vocabulary (TEI definition: "points to a description of the rendering or presentation used for this element in the source text")
    - Adopting @rendition would also eliminate possible conflict with values of @rend encoded for other purposes
  - **daba bottom line (20210604) on problem 2**: I've added this proposal as a comment in the revised EGD section and hope to continue the discussion there
- **problem 3**: Manu's original problem involving ambiguity e.g. between Tamil ca and Grantha sa
  - none of the above methods (alone) can solve this problem, because of the blanket EpiDoc ban on any markup within <unclear> aside from <g>
  - could we introduce a new method, avoiding unclear, simply involving two seg elements within a choice?
    - e.g. like this: <choice><seg rend="Grantha">ś</seg><seg rend="Tamil">c</seg></choice>
    - this is not sanctioned by EpiDoc, but nor is it prevented by the EpiDoc or DHARMA schema (except that the DHARMA schema permits only "pun" as a value of @rend on <seg>, but that can be changed)
    - TEI explicitly permits <seg> in <choice>, but probably not exactly for our purposes - nonetheless, we could perhaps co-opt this, only for the cases where there is a script uncertainty regarding a character or string of characters
    - this would also avoid the use of unclear for something that is clearly readable (Gabby: [unclear], in EpiDoc, uniquely records incomplete, damaged, or illegibly executed characters, not per se the editor's uncertainty in interpretation of them)
    - this method would of course work with attributes other than @rend (including @xml:lang), so if we adopt a different method for tagging script, we can still use the seg in choice method to solve Manu's original problem
    - I cannot think of anything else to solve this problem
  - **daba bottom line (20210604) on problem 3**: I've summarised this suggestion

in the relevant GitHub issue and hope to continue the discussion there

Bits of Guide scrapped for now but retained just in case

1.4. OLD Overview: The Structure of an EpiDoc Edition

- this section presents an overview of the elements comprising a digital edition in EpiDoc
  - note that in the present version of the Guide, the code shown here is from the generic EpiDoc template
  - in the near future we will have a template specific to the DHARMA project and once that is finalised, it will be included in future releases of this Guide
  - thus, you will not need to learn and produce this code, only to find your way around it and adapt it to your particular needs
- text printed in black below is not part of the code, but serves for clarification and explanation of the structure
- XML documents normally begin with a declaration specifying what sort of document this is
- you will not have to edit this declaration; simply make sure you make no changes to the one found in the template, so that it remains at the very beginning of each of your edition files

```
<?xml version="1.0" encoding="UTF-8"?>
<?xml-model
href="http://www.stoa.org/epidoc/schema/latest/tei-epidoc.rng"
schematypens="http://relaxng.org/ns/structure/1.0"?>
<?xml-model
href="http://www.stoa.org/epidoc/schema/latest/tei-epidoc.rng"
schematypens="http://purl.oclc.org/dsdl/schematron"?>
```

- the element `<TEI>` must wrap all TEI-compliant content as a root tag, which in our case means everything other than the XML declaration

```
<TEI xmlns="http://www.tei-c.org/ns/1.0">
```

- a header section identifying and containing additional descriptive information (metadata) about the digital document and the encoded text is a mandatory component of every TEI document
- the header must always include a file description and may include various other items; see §11 about the composition of the header as used in our documents
- the contents of the header are grouped into sections called statements and descriptions

```
<teiHeader xml:lang="eng">
<fileDesc>
  <titleStmt>
    <title>title of document</title>
  </titleStmt>
  <publicationStmt>
    <authority></authority>
    <idno type="filename"></idno>
  </publicationStmt>
  <sourceDesc>
    <msDesc>
```

- text and support metadata in TEI format
  - for the present, record your metadata in spreadsheets and do not worry about their representation in TEI

```
    </msDesc>
  </sourceDesc>
</fileDesc>
</teiHeader>
```

- all text-related content must be wrapped in the element `<text>`
  - in TEI the text container can include elements other than `<body>`, but EpiDoc convention does not use any of these elements, so `<text>` and `<body>` are always opened and closed at the same point

```
<text xml:space="preserve">
<body>
```

- the body container includes a mandatory subdivision containing the edition, and may include further optional subdivisions as follows

```
<div type="edition" xml:lang="san-Latn">
edition encoded as per §§2–7; see also §7.2 about the attribute
@xml:lang
</div>
<div type="apparatus">
apparatus encoded as per §9.1
</div>
<div type="translation">
translation encoded as per §9.2
</div>
<div type="commentary">
commentary encoded as per section §9.3
</div>
<div type="bibliography">
bibliography encoded as per section §9.4
</div>
```

- closing tags for the text containers

```
</body>
</text>
```

- closing tag for the TEI container

```
</TEI>
```

Correction and Normalisation BEFORE REWRITE OF 20191215

- the editorial rectification of a phenomenon deemed to be a scribal mistake is here referred to as **correction**
- the editorial alteration of a phenomenon deemed to be non-standard usage into something that fits the standard more closely is here referred to as **normalisation**
- distinguishing scribal error from non-standard usage may be problematic and will often involve a subjective decision
  - deviations that involve the exchange of a character to a graphically similar one are likely to be scribal errors
  - deviations from expected forms that occur repeatedly in inscriptions are more likely to be non-standard usage
  - deviations that seem to be governed by the immediate phonemic context are more likely to be non-standard usage
  - deviations that involve the exchange of a character to a phonetically similar one are likely to be non-standard usage
  - grammatical solecisms are to be considered non-standard usage, not scribal error
  - when in doubt, prefer `<orig>` and use `<sic>` only in clear cases of scribal error

Correction

Flagging and correcting scribal errors

- this subsection concerns cases where one or more characters of the received text are deemed erroneous by the editor and substituted by one or more characters of corrected text
  - the editorial restitution of scribal omission and the editorial suppression of superfluous inscribed text are special cases of correction dealt with in the following subsections
  - non-standard language (as opposed to scribal error) is dealt with in §XXX below; see also §XXX above for some guidance on the distinction between the two
- depending on how trivial or significant you deem a scribal error to be and how much it might affect the understanding of the text, you may choose to
  - ignore scribal errors,
  - flag them as erroneous without correction,
  - or flag and correct them
- **to flag** scribal errors, wrap the relevant characters in the element `<sic>`
  - e.g. `mahār<sic>a</sic>ja`
- **to correct** scribal errors,
  - flag the locus with `<sic>` as above,
  - add the corrected equivalent of the flagged text, wrapped in a `<corr>` element, right after this,
  - and wrap both these elements in the element `<choice>`
  - e.g. `mahār<choice><sic>a</sic><corr>ā</corr></choice>ja`
- the size of segments flagged as scribal errors or corrected should normally be kept to a minimum (i.e. restricted to the affected transliteration characters), but see also §XXX on avoiding non-essential complexity
  - scribal errors of omission should be encoded up as such (§XXX) in preference to merely flagging the error

    - flagging a scribal omission without correction, if this is deemed essential, can only take place by including in the scope of `<sic>` some of the surrounding characters
    - scribal omissions should never be corrected with a `<sic>`/`<corr>` pair
  - scribal errors involving superfluous characters or components should be marked up as editorial deletion (§XXX) when correction is desired
    - it is acceptable merely to flag the superfluous segment with `<sic>`
    - but correction should never be implemented with a `<sic>`/`<corr>` pair
  - flagging a scribal error consisting in the omission of a segment of text is only possible by including some of the surrounding characters in the scope of `<sic>`
    - it is therefore recommended that you encode a scribal omission (§XXX) in preference to merely flagging such an error
    - do not correct such errors with a sic/corr pair

Editorial deletion

- this subsection concerns cases where you deem a segment of text to be superfluous, and correct the text by suppressing the segment concerned (without substituting anything else for it)
  - see §XXX for pre-modern deletion in the original
- to mark up any original characters as superfluous, enclose them in the element `<surplus>`
- editorial deletion should always be used to highlight instances of dittography, e.g.
  - `naika-samara-śata<surplus>ta</surplus>vijayinā`
  - `veda-vyāsena vyāsena <surplus>vyāsena</surplus>`
- other superfluous characters or components may, at your discretion, be deemed erroneous and corrected in this way, or be considered non-standard usage and treated as such (§XXX)

Correction of scribal omission

- this subsection concerns cases where you find that a segment of text was erroneously omitted by the scribe, and correct this omission by restoring the expected segment
- keep in mind that omission must be considered at the level of character components corresponding to phonemes: the scribe’s failure to engrave a stroke is not necessarily an omission, e.g.
  - if *mahāraja* was engraved instead of *mahārāja*, this needs to be rectified as a correction (of *a* to *ā*) even though in the process of writing the error consisted of the omission of the *ā* marker
  - if *lora* was engraved instead of *loka*, this is again a correction (of *r* to *k*) even if what in fact happened was that the engraver neglected the cross-stroke that would have turned a *ra* into a *ka*
- to correct definite scribal omissions, wrap your restoration in the element `<supplied reason="omitted">`, e.g.
  - `dhanada-varuṇendrānta<supplied reason="omitted">ka</supplied>-samasya`
- omissions of a single phoneme may, at your discretion, be deemed erroneous and corrected in this way, or be considered non-standard usage and treated as such (§XXX)
- small components (such as a superscript *r* or an *anusvāra*), which are expected to

be present but cannot be made out in the original or a facsimile, might better be marked up as lost and restored (as per §) unless you are certain that the cause is scribal omission, not damage to the support
- when encoding editorial corrections, keep in mind that it must always be possible to produce the received text by ignoring the segment tagged with `<corr>` and, vice versa, to produce the corrected text by ignoring the segment tagged with `<sic>`

Good practice in editorial correction

- the number one rule for editorial alterations of the received text is that they must **never be silent**
  - your digital edition must always include the text as found on its support, and any changes you make to create an abstract text must be shown in markup, as detailed below
  - apparent exceptions to this rule (such as editorial hyphenation, *avagraha*s, etc.) are only apparent, as our system will know that they are editorial and will be able to strip them away to obtain a purely diplomatic edition
- the number two rule is that editorial correction, with the appropriate markup, is optional; in many cases less is better
  - in particular, do not supply punctuation marks or verse numbers where such are not present in the original (but do restore them as per § if you are certain that such things *were* present and have been lost to damage)
- in the orthography of your editorial corrections, attempt to
  - respect the orthography and, if applicable, the language usage of the rest of the document, e.g. correct
    - *karppa* to *karmma* (rather than fully standard *karma*) if the inscription normally doubles nasals after *r*
  - keep your corrections to a plausible minimum, e.g. correct
    - *karpa* to *karma* (rather than an expected *karmma*), assuming the engraver made the simple mistake of inscribing *p* for *m* (rather than the complex mistake of inscribing *p* for *mm*)
    - *viṅgati* to *viṅśati* (rather than fully standard *viṁśati*)
    - continue: minimum deletion, minimum addition etc

Normalisation

Flagging and normalising non-standard usage

- this subsection concerns cases where you deem an unexpected form to be due to non-standard usage rather than scribal error (see §XXX above for some guidance on this somewhat subjective decision)
- depending on how trivial or significant you deem a usage to be and how much it might affect the understanding of the text, you may choose to
  - ignore non-standard usage,
  - flag it as non-standard without normalisation,
  - or flag and normalise it
- **to flag** any feature as non-standard, wrap the relevant characters in the element `<orig>`, e.g.
  - `dine <orig>Āśvoja</orig>-śuklasya` (*Āśvayuja* or *'śvayuja* expected)
  - `sahasrā<orig>n</orig>i` (*ṇ* expected)
- **to normalise** non-standard spelling or grammar,

  - flag the locus with `<orig>` as above,
  - add the normalised equivalent of the flagged text, wrapped in a `<reg>` element, right after this,
  - and wrap both these elements in the element `<choice>`
  - e.g. `e<choice><orig>ś</orig><reg>ṣ</reg></choice>a`
- it is recommended that your normalisations still conform to the orthography of the rest of the document in details that you would not normalise elsewhere, e.g. normalise *varnna* to *varṇṇa* rather than *varṇa* (unless the inscription happens not to double nasals after *r* elsewhere)
- the size of segments flagged as non-standard or normalised should
  - be kept to a minimum for non-standard phonology, but including the immediate phonemic context of the problematic characters is acceptable
  - extend to whole words (as far as sandhi permits) for non-standard morphology
  - extend to whole phrases (as far as sandhi permits) for non-standard syntax
- unlike the correction of scribal errors, where EpiDoc furnishes a separate method for the suppression of superfluous text and the restoration of scribal omission, normalisation consisting of the editorial addition or deletion of characters must be handled in the same way as one consisting of substituting text for received characters
  - therefore, when normalising in such a situation, your markup should extend to whole words (as far as sandhi permits) in order to avoid ending up with an empty `<orig>` or `<reg>` element (which is not in itself erroneous, but which may be difficult to interpret and even to display)

Good practice in normalisation

- near-universal features of inscriptional orthography, such as the following, should be ignored unless you consider a particular instance important for some reason
  - the doubling of plosives, nasals and glides after an *r*
  - the use of an *anusvāra* instead of the class nasal or vice versa
- common orthographic features, such as the following, may be ignored or flagged depending on how widespread they are in your corpus, but should not as a rule be normalised:
  - the doubling of sibilants after an *r*
  - the doubling of consonants before an *r*
  - the use of *tv* where *ttv* is expected
  - infidelity to the correct length of vowels in words borrowed from Sanskrit, in languages where inconsistency in spelling of vowel-length is rampant (see also TG §3.3.7 and EG §XXXAutomated normalisation below)
  - the use of an *anusvāra* in place of a final *m·* or *M* or vice versa
  - the use of a nasal in place of an *anusvāra* before a sibilant or *h*
  - the exchange of one phonetically similar character for another, such as the use of one nasal instead of another or one sibilant instead of another, provided that the substitution occurs with some frequency in the text or corpus
  - in Old Javanese, the spelling of long *pepet* with *ə* plus length mark (to be represented as *ə:*, as per TG §3.3.6)
  - in Old Javanese, the use of the signs *R̥* or *L̥* (and potential long versions thereof) in words whose dictionary spelling has *rə* or *lə* (or the variants with macron on schwa)
  - in Old Khmer, the non-spelling of *virāma*, or the representation of final consonant

C by the spelling *CCa*, including cases where final /h/ is represented as *ḥha*

- non-orthographic deviations from standard language, such as the following, should normally be at least flagged
  - non-standard or sub-standard grammar, e.g. *rājasya* for *rājñaḥ*; *kr̥tedam* for *kr̥tam idam*; *sā gataḥ* for *sā gatā*
  - presumable non-standard sandhi, e.g. *anugrahāyam udaka-pūrvveṇa* (superfluous *m*); *paṁcas-triṁśottaratame* (superfluous *s*)
  - presumable hypercorrection, e.g. *dattvā* instead of *dattā*; *rakṣya* instead of *rakṣa*; *prārk-kriyamāṇaka* instead of *prāk-kriyamāṇaka*
- flagging should be generally preferred to normalisation whenever you are dealing with a language where a less rigorous understanding exists about what is wrong and right than in the case of Sanskrit

Automated normalisation

- some specific cases of normalisation will be automated in our workflow, so certain characters in your transliteration will be converted to markup
- **editorial long vowels in Dravidian languages** where the script does not distinguish short and long *e* and *o*
  - as per TG §3.2, the transliterated characters *ē* and *ō* will be automatically marked up as normalised, i.e. that *e* or *o* were originally inscribed, but these represent long vowels, e.g.
    - <choice><orig>e</orig><reg>ē</reg></choice>
- **explicit short vowels in Sanskrit loanwords** where a long vowel is expected
  - as per TG §3.3.7, the transliterated characters *ă*, *ĭ* or *ŭ* will be automatically marked up as short in the original and normalised to their long equivalents, e.g.
  - <choice><orig>a</orig><reg>ā</reg></choice>
- **editorial *avagraha*s**
  - as per TG §2.6.3, any *avagraha* (i.e. ’ [right single quote] or ' [plain apostrophe] followed by an alphabetic character) found within the <div type=”edition”> will be assumed by default to be non-original and automatically marked up as <supplied reason="subaudible">’</supplied>[1]
  - original *avagraha*s transliterated as ’! will not be auto-tagged in this way, but the exclamation mark will be removed automatically

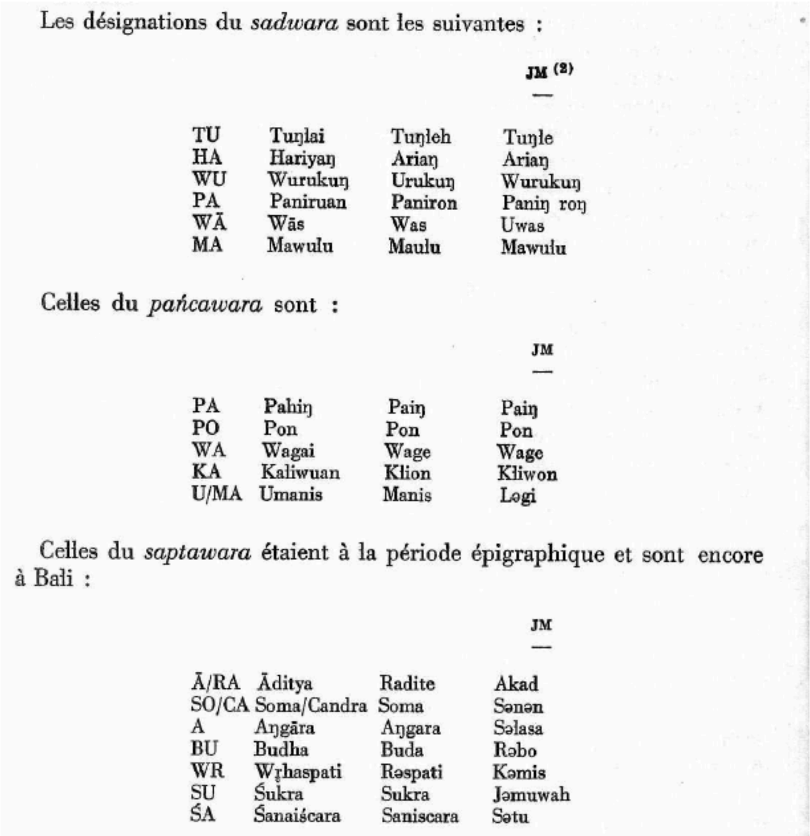
Abbreviations

Extracting from comments by Arlo

Javanese: ka for kaliṅanya, ṅa for ṅaranya, bhre for bhaṭāra i, samgat for saṅ pamgat — and many cases for monetary/weight values, such as su for suvarṇa, mā for māṣa, etc.

if there is a difference between expected actual spelling and normalized spelling of the expanded part, use normalized spelling. e.g., in Old Javanese, pa is a very common abbreviation of a name of a day in the 6-day week whose dictionary lemma would be Pahiṅ. This name could be spelled pahiṁ (with anusvāra) or pahiṅ· (with ṅ and virāma). if it were spelled in full. Inside the <expan> tag, please put normalized hiṅ.

[1] We adopt this markup in order to eliminate the drudgery of encoding an <orig>/<reg> pair with either an empty <orig> element or a whole redundant word.



Les désignations du *sadwara* sont les suivantes :

			JM[2]
TU	Tuŋlai	Tuŋleh	Tuŋle
HA	Hariyaŋ	Ariaŋ	Ariaŋ
WU	Wurukuŋ	Urukuŋ	Wurukuŋ
PA	Paniruan	Paniron	Paniŋ roŋ
WĀ	Wās	Was	Uwas
MA	Mawulu	Maulu	Mawulu

Celles du *pańcawara* sont :

			JM
PA	Pahiŋ	Paiŋ	Paiŋ
PO	Pon	Pon	Pon
WA	Wagai	Wage	Wage
KA	Kaliwuan	Klion	Kliwon
U/MA	Umanis	Manis	Ləgi

Celles du *saptawara* étaient à la période épigraphique et sont encore à Bali :

			JM
Ā/RA	Āditya	Radite	Akad
SO/CA	Soma/Candra	Soma	Sənən
A	Aŋgāra	Aŋgara	Səlasa
BU	Budha	Buda	Rəbo
WR	Wr̥haspati	Rəspati	Kəmis
SU	Śukra	Sukra	Jəmuwah
ŚA	Śanaiścara	Saniscara	Sətu

<abbr>tu</abbr><ex>ṅlai</ex>
<abbr>ha</abbr><ex>riyaṅ</ex>
<abbr>vu</abbr><ex>rukuṅ</ex>
<abbr>pa</abbr><ex>nirvan</ex>
<abbr>vā</abbr><ex>s</ex>
<abbr>ma</abbr><ex>vulu</ex>

<abbr>pa</abbr><ex>hiṅ</ex>
<abbr>po</abbr><ex>n</ex>
<abbr>va</abbr><ex>gai</ex>
<abbr>ka</abbr><ex>livvan</ex>
<abbr>U</abbr><ex>manis</ex>
OR <abbr>ma</abbr><ex>manis</ex>

<abbr>Ā</abbr><ex>ditya</ex>
OR <abbr>ra</abbr><ex>vi</ex>
<abbr>so</abbr><ex>ma</ex>
OR <abbr>ca</abbr><ex>ndra</ex>
<abbr>A</abbr><ex>ṅgāra</ex>
<abbr>bu</abbr><ex>dha</ex>
<abbr>vr̥</abbr><ex>haspati</ex>
<abbr>śu</abbr><ex>kra</ex>
<abbr>śa</abbr><ex>naiścara</ex>

ARCHIVE NEWER Massive lacunae

- larger lacunae will in many cases obscure the boundaries between elements of intrinsic or extrinsic structure
- even where no actual text can be restored, it is desirable
  - to reconstruct as much of the (extrinsic and intrinsic) structure of the lost chunk of text as is feasible, e.g.
    - by extrapolation from the structure of the extant text (e.g. inferring the existence of initial or final lines from a partially extant stanza)
    - by inference from the shape and size of the lacuna as far as it can be determined
    - by inference from the shape, size and nature of the extant fragment of the support
  - and to populate the reconstructed structure with multiple inline `<gap/>` elements (§) instead of a single multiline `<gap>`
- reconstructed structural elements must not be marked up as supplied
- reconstructed structure may be sketchy and approximate
  - any doubt concerning your reconstruction may be described in your commentary
- to **reconstruct structural elements** at any point where a lacuna meets extant text
  - for any **stanzas** interrupted by the lacuna
    - extrapolate the full stanza structure (i.e. an `<lg>` element with the expected number of `<l>` elements)
    - fill up each `<l>` element with a `<gap>`, encoding its length and metre as applicable
  - but where a **prose block** (`<p>` and `<ab>`) is interrupted by a massive lacuna that is presumed to have included further structural blocks (one or more stanzas or prose blocks), there is no way of predicting the length to which such a block would have continued within a lacuna; therefore
    - close the prose block before the lacuna starts (rather than partway into the lacuna) if the prose block precedes the lacuna
    - open the prose block after the lacuna ends (rather than partway into the lacuna) if the prose block follows the lacuna
  - add an `<ab>` element to wrap the section of the lacuna not already wrapped in a reconstructed stanza structure (i.e. all of the lacuna if it interrupts a prose block on both sides)
  - create `<lb/>` elements for each lost line (and, if applicable, `<pb/>` elements for each lost page) whose existence can be inferred with confidence
    - if you are also reconstructing the structure of a partially lost stanza, estimate the position of such a break relative to the stanza structure and do not worry about the accuracy of the estimate
  - if applicable, encode an unknown number of lost lines (§XXX) beyond or instead of reconstructed `<lb/>` elements
- in a text with a massive lacuna, **reconstructed elements that require numbering** (i.e. `<lb/>`, `<pb/>` and `<lg>`) must be numbered in the same way as, and contiguously with, extant elements of the same kind; thus,
  - for an **initial lacuna**, start the numbering with the first reconstructed element and continue it in the extant elements
  - for a **final lacuna**, continue numbering in the reconstructed elements
  - for a **medial lacuna**,

    - if the **structure** of the lost medial chunk can be **reconstructed with reasonable accuracy** (i.e. you can confidently estimate the number of lost lines, stanzas if applicable, and pages if applicable), then simply continue numbering from the last extant element through the reconstructed one(s) and on into the next extant ones
        - in copper plates with a lost medial plate, it is sufficient to have accurate knowledge of the number of lost pages (i.e. it is not a problem if the number of lost lines per plate is unknown), so long as you restart line numbering on every page (see also §XXX)
    - if the **structure** of the lost medial chunk **cannot be reconstructed** with sufficient confidence (e.g. because all you have is two unconnected fragments of the support, or if the number of lost lines and/or stanzas is too dubious), encode the extant sections as textpart divisions (§XXX)
        - the first of these will thus have a final lacuna, and the second an initial lacuna, for each of which structure can be reconstructed as above
        - the numbering of numbered elements must then be restarted from 1 in the second textpart division
        - here, you are essentially treating your inscription as consisting of unconnected fragments (§XXX), even if the massive lacuna is not the result of the fragmentation of the support and the loss of one or more fragments

ARCHIVE OLDER Massive lacunae

- larger lacunae will in many cases obscure the boundaries between elements of intrinsic or extrinsic structure
- the way to handle massive lacunae depends on the extent to which you can reconstruct the lost structure, e.g.
    - by extrapolation from the structure of the extant text (e.g. inferring the existence of initial or final lines from a partially extant stanza)
    - by inference from the shape and size of the lacuna as far as it can be determined
    - by inference from the shape, size and nature of the extant fragment of the support
- the **preferred method** of dealing with massive lacunae is to reconstruct as many elements of intrinsic and extrinsic structure as possible, and to populate these elements with multiple inline `<gap/>` elements (§)
    - reconstructed structure may be sketchy and approximate; any doubt concerning your reconstruction may be described in your commentary
    - reconstructed elements do not need to be marked up as supplied
    - reconstructed elements that require numbering must be numbered in the same way as extant elements of the same kind
    - **reconstructing intrinsic structure**
        - if any **stanzas** are interrupted by the lacuna on either side (or both),
            - create the full stanza structure (i.e. an `<lg>` element and the expected number of `<l>` elements)
            - number the partially lost stanzas consecutively with the extant ones
            - if there is good reason to assume that the lacuna included one or more full stanzas, create the structure for these as well
            - if the lacuna is larger than the stanzas whose structure is possible to reconstruct, wrap the remaining chunk in an `<ab>` element even if there is a chance that it may have been in verse (with one or more stanzas)
            - after the lacuna, continue stanza numbering from the next higher number (i.e.

do not skip numbers for possibly lost stanzas)

    - **prose blocks** (`<p>` and `<ab>`) interrupted by a massive lacuna on either side should be closed before the lacuna and opened after it, because there is no way of predicting the length to which such a block would have continued within a lacuna
      - in this case, wrap the entire lacuna in an `<ab>` element even if there is a chance that (some of) it may have been in verse
  - **reconstructing extrinsic structure**
    - create numbered `<lb/>` elements for each lost line
      - unless you are dealing with lost copper plates, in which case it is sufficient to create numbered `<pb/>` elements, provided that the number of lost lines per plate is unknown and you restart line numbering on every page (see §XXX for details)
    - if you are also reconstructing some elements of intrinsic structure (stanzas), estimate the position of the line beginnings relative to those elements
      - do not worry about the accuracy of your estimate
    - after the lacuna, continue numbering lines from the next higher number
- if the **structure** of the lost text **cannot be reconstructed** with sufficient confidence (e.g. because all you have is two unconnected fragments of the support, or if the number of lost lines and/or stanzas is too dubious), encode the extant sections as textpart divisions (§XXX)
  - in this case, any block-level containers interrupted by the beginning of the lacuna (including verse containers with partially extant text) should be closed before the closing tag of the former division, and new block-level containers must be opened as applicable after the opening tag of the latter division
  - however, it is still advisable to reconstruct as much structure as feasible (to the end or from the beginning of an interrupted stanza, or if the text is prose, to the end or from the beginning of a partially extant line), and to populate this reconstruction with one or more inline `<gap/>` elements
  - the numbering of lines and stanzas must be reset to 1 in each division

Textpart subtypes and numbers (pre-November 2019)

- $to be rewritten after decision on this matter
- the attribute @subtype serves to give a brief identification of each textpart
  - the value of @subtype will be employed as a heading when your digital edition is displayed
  - any value is permitted for the time being; at a subsequent stage we intend to harvest the words used for this purpose by encoders and create a controlled vocabulary
  - however, if you introduce values not explicitly recommended below, please keep the following in mind:
    - the value should be in lowercase throughout to avoid inconsistencies; display can easily be rendered with a capital initial
    - the value should not include spaces; if you absolutely need a multi-word value, use an underscore (_) instead of a space, which can be rendered as a space in display
    - having introduced a custom value, try to use it consistently and send us the value and a short definition/description of the case where you have used it, so it can be included in later versions of this guide

  - **recommended values for @subtype** may describe
    - *general nature* where several divisions of the same kind exist, e.g.
      - “fragment” for fragments with non-contiguous text
      - “face”
      - “facet”
      - “surface” for physically distinct surfaces of a three-dimensional object
      - “zone” for visually distinct zones of a two-dimensional surface
      - “field” for topographically complex fields subdivided into pagelike or gridlike partitions but bearing a single unit of text (see §Case study 1 for an example)
    - or *specific identity* where the divisions are different in nature, e.g.
      - “seal” and “plates” for a copperplate charter with an inscribed seal
      - “head”, “halo”, “back” and “pedestal” (or other part names) on a statue
- textparts must be numbered if and only if their @subtype is not unique within the document, i.e.
  - textparts whose @subtype describes *general nature* must always be numbered, e.g. `<div type="textpart" subtype="fragment" n="A">`
  - textparts whose @subtype describes *specific identity* shall by default not be numbered, e.g. `<div type="textpart" subtype="seal">`
    - **except** when a document includes more than one textpart of the same specific subtype, e.g.
      - `<div type="textpart" subtype="seal" n="A">` and `<div type="textpart" subtype="seal" n="B">` for a set of plates with two inscribed seals

The term Incipit

- in European codices, *incipit* (Latin for ‘it begins’) refers to the first few words of a text, which were often enlarged and decorated; in this guide we co-opt the term for words or symbols used at the beginning of inscriptions for any purpose[2]
- incipits do not require markup as such, so if an inscription begins with a conventional auspicious word that is an integral part of the first line, treat it simply as the first word of the text

Precise location of line beginnings within a lacuna

- if there is lost text both before and after a line break, then the simplest thing to do is to encode a lacuna of unknown or uncertain length at the end of the former and the beginning of the latter line (see § about lacunae)
- however, if you supply the lost text (for which see §), you may wish to indicate in your markup that the line break was not necessarily at the precise spot of the supplied text where you show it
  - for instance, in deva-ni`<supplied reason="lost">`keta`</supplied><lb break="no" n="7"/><supplied reason="lost">`naṁ`</supplied>`, the line end may have been after *ke* or after *naṁ* instead of after *ta*, as shown in the markup
  - to do so, you will first need to give the line beginning tag an identifier, e.g. `<lb break="no" n="2" xml:id="xxxx"/>`

[2] The original purpose of such “incipits” in Indic inscriptions is uncertain. In some cases they may simply serve as a conventional indicator of the starting point. More commonly, they were probably used as auspicious words or symbols, and in many cases they may (also) have had the function of documenting the execution of the act described in the text of the inscription.

    - where xxxx stands for an identifier that will have to be unique throughout the DHARMA corpus
        - to create a unique ID, use the name of your file (given as per the file naming conventions/rules to be issued separately) followed by an arbitrary number that you must ensure will not be used in any other IDs within the same file
    - and next to the `<lb/>` element, add the following code: `<certainty target="#xxxx" locus="location"/>`[3]
        - where xxxx stands for the ID you have given to the line beginning tag (note that the ID must be preceded by a hash mark # here)

Arbitrary segments

- it is sometimes necessary to apply markup to arbitrary segments of text, i.e. segments that are neither spatially nor semantically separated from the surrounding text, yet have a distinguishing intrinsic feature that you wish to encode
    - in general, such segments differ from the surrounding text in either language (§) or script (§)
    - additionally, there are some special purposes for which we use arbitrary segments:
        - to wrap individual *akṣara*s where necessary to indicate which transliterated characters belong to a single original character (§)
        - to wrap lacunae for which we know the prosodic pattern (§)
        - to wrap lacunae affecting only part of an *akṣara* (§)
        - to mark bits of translation as tentative (§)
    - to mark up an arbitrary segment, use the element `<seg>` with attributes as called for in the sections referred to above
    - where necessary, `<seg>` elements may be contained within other `<seg>` elements

Hand changes

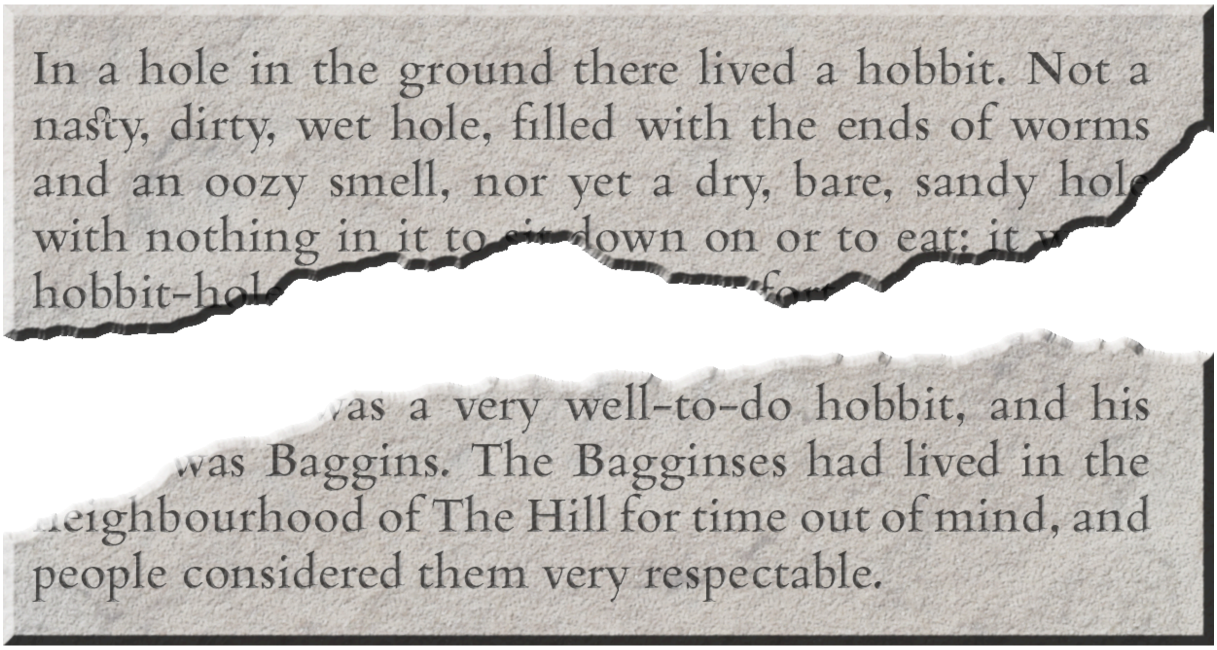
- in epigraphic parlance, a “hand” means a particular combination of writing features, often indicative of one scribe taking over the work of another
- if certain **textparts** (as per §) of the inscription are **in a different hand or script**
    - add the attribute @hand to the `<div type="textpart">` element corresponding to the unit written in a different hand
    - the value of @hand shall be the xml:id of the hand responsible for that unit of text, e.g. h1, h2, etc. (see § about rigorously encoding multiple hands in the TEI header)
- if certain **arbitrary segments** of the inscription are **in a different hand or script** (i.e. if the scope of the alternative script does not coincide with a textpart as per §),
    - wrap the relevant segment in the element `<seg>` with the attribute @hand
    - the value of @hand shall be the xml:id of the hand responsible for that unit of text, e.g. h1, h2, etc. (see § about rigorously encoding multiple hands in the TEI header)
- if **the hand changes at an arbitrary point** and carries on to the end or to another arbitrary point,

[3] This solution was suggested on the MARKUP mailing list by Gabriel Bodard on 17 March 2017. TEI allows for the possibility of quantifying the editor’s certainty and listing additional possibilities, each with an assigned certainty. We do not need that level of detail here and use the `<certainty>` element simply to flag less-than-absolute certainty.

- add the empty element <handShift/> without any attributes at the point of the change
- do likewise if the hand changes back to the original hand, or changes to a third
- in the <handDesc> section of your header (§), describe all the hands involved in the inscription
  - if this is the only type of hand change in your text, then there is no need to include a rigorous description of hands (§) in the header

Fractured Inscriptions Example B: non-contiguous fragments

- SCRAPPED; JUST USE TEXTPARTS IN THIS CASE
- if the text connecting two fragments cannot be restored, but its **physical structure can be inferred** to some degree, then textpart divisions are not necessary; instead, the fragments may be encoded using gridlike partitions as in Example B



In a hole in the ground there lived a hobbit. Not a
nasty, dirty, wet hole, filled with the ends of worms
and an oozy smell, nor yet a dry, bare, sandy hole
with nothing in it to sit down on or to eat: it w
hobbit-hol for

as a very well-to-do hobbit, and his
was Baggins. The Bagginses had lived in the
eighbourhood of The Hill for time out of mind, and
people considered them very respectable.

- this example involves the same imaginary scenario as Example A above
- this time, supposing you infer, on the basis of a parallel text or conjecture, that exactly one line was lost between these fragments,[4] it is then possible to reconstruct the structure of the lacuna, allowing you to number lines contiguously
- this scenario would not require textpart divisions, and should be encoded as follows

```
<p>
  <!--The two fragments have been encoded here as milestones, as
per §. This is optional. -->
  <lb n="1"/><milestone unit="fragment" n="a"/>In a hole in the
ground there lived a hobbit. Not a
  <lb n="2"/>nasty, dirty, wet hole, filled with the ends of worms
  <lb n="3"/>and an oozy smell, nor yet a dry, bare, sandy
hol<unclear>e</unclear>
```

[4] Incidentally, you would be wrong. But this does not matter for the illustration.

```
  <lb n="4"/>with nothing in it to <supplied
reason="lost">s</supplied><unclear>i</unclear><supplied
reason="lost">t</supplied> down on or to eat: it
<unclear>w</unclear><supplied reason="lost">as a</supplied>
  <lb n="5"/>hobbit-hol<supplied reason="lost">e</supplied> <gap
reason="lost" quantity="18" unit="character"
precision="low"/><unclear>f</unclear><unclear
cert="low">o</unclear><gap reason="lost" quantity="12"
unit="character" precision="low"/>
  <lb n="6"/><gap reason="lost" extent="unknown" unit="character"/>
  <lb n="7"/><milestone unit="fragment" n="b"/><gap reason="lost"
quantity="12" unit="character"
precision="low"/><unclear>w</unclear>as a very well-to-do hobbit,
and his
  <lb n="8"/><gap reason="lost" quantity="5" unit="character"
precision="low"/><unclear>w</unclear>as Baggins. The Bagginses had
lived in the
  <lb n="9"/><unclear>ne</unclear>ighbourhood of The Hill for time
out of mind, and
  <lb n="10"/>people considered them very respectable
</p>
```

A couple of illustrations for normalisation/corretion

- other superfluous characters or components may, at your discretion, be deemed erroneous and corrected in this way, or be considered non-standard usage and treated as such (§XXX), e.g.
  - duplication of phonemes:
    - `para-dattāṁ v<surplus>v</surplus>ā` OR `para-dattāṁ <orig>vv</orig>ā`
  - extra phonemes inserted as likely hypercorrection:
    - `iyaṁ rāja-śatair datt<surplus>v</surplus>ā` OR `iyaṁ rāja-śatair <orig>dattvā</orig>`
    - `yatnād rakṣ<surplus>y</surplus>a yudhiṣṭhira<surplus>ḥ</surplus>` OR `yatnād <orig>rakṣya</orig> yudhiṣṭhira<orig>ḥ</orig>`
    - `prā<surplus>r</surplus>k-kriyamāṇaka` OR `prā<orig>rk</orig>-kriyamāṇaka`
  - extra phonemes inserted as likely non-standard sandhi
    - `mātā-pittror anugrahāya<surplus>m</surplus> udaka-pūrvveṇa pratipāditam` OR `mātā-pittror <orig>anugrahāyam udaka</orig>-pūrvveṇa pratipāditam`
    - `paṁca<surplus>s</surplus>-triṁśottaratame` OR `<orig>paṁcas-triṁśottaratame</orig>`
 –
 – be normalised by supplying the missing character(s) or simply flagged (often with

a slightly wider context) as non-standard, e.g.

  - `brāhmaṇasā<supplied reason="omitted">d</supplied> gatāḥ` OR `<orig>brāhmaṇasā</orig> gatāḥ`
  - `eta<supplied reason="omitted">t</supplied> kṣetraṁ` OR `<orig>eta</orig> kṣetraṁ`

Appendix #: Linking Digital Images to an inscription

<facsimile> contains the representations of the written source as a set of images rather than as transcribed or encoded text. This block has to be encoded between the <teiHeader> and the <text>, e.g. <teiHeader>[...]</teiHeader><facsimile>[...]</facsimile><text>[...]</text>.

Mostly, it will be used to provide images of the text and give their location inside the Dharma files. One tag <graphic> represents one picture. Repeat it as much as necessary.

The pictures related to the project will all be stored in the Dharma Sharedocs, see the Guide written by Adeline Levivier on this matter. To connect those pictures with the XML file, use the attribute @url. Use the attribute @xml:id to give a unique identifier to any picture. You can reuse the naming convention set for pictures. Finally, add an attribute @n to number them following the order you want them to be displayed. E.g. <facsimile><graphic xml:id="DHARMA_TNut0007_image01.jpg" url="https://sharedocs.huma-num.fr/wl/?id=l4Z0533EYkzGmJP16wQb0RHxCBOIiQdB" n="1" /></facsimile>

Note that the tag <graphic/> can be used as an empty tag. However, if you want to associate a title, creator, date and/or copyright, you can record it in the tag <desc>. This description of the image can be written in prose, e.g. <desc>Photograph of EFEO estampage n. 2129.</desc>.

Apparatus guide scrapped on 20191203

Location reference: @loc

- the value of the attribute @loc must be a human-readable indication of a locus within your text, in one of the following forms:
  - a numeral and nothing else (e.g. “3”) to indicate the number of the line containing the locus
    - for a locus that begins in one line and ends in the next, include both line numbers in the reference (see below for the method)
  - the letter v followed by a numeral and nothing else (e.g. “v12”) to indicate the number of the stanza containing the locus
  - separate <app> entries within your <div type="apparatus"> may refer to either verse or line numbers; there is no need to use just a single referencing system throughout an apparatus
  - in general, localised apparatus entries should be referenced by line number, and references to stanza numbers should preferably be used only for apparatus notes pertaining to stanzas as a whole
- where necessary, it is possible to reference **multiple lines or multiple stanzas** in one of the following ways
  - use a hyphen to define a range, e.g. <app loc="18-20"> for an entry that concerns lines 18, 19 and 20
  - use a comma and a space to separate items in a list, e.g. <app loc="v5, v8, v11">

for an entry that concerns stanzas 5, 8 and 11

- $CONSIDER AND WRITE UP: LINE NUMBERS IN NUMBERED SUBDIVISIONS
  - if we retain subdivision numbers within the line numbers themselves, then this should be obvious but still mentioned here just in case; if we decide to forget about complex line numbering, then we'll have to explicitly include division numbers in loc
  - $also add way of referring to content of fw elements

Precise location referencing: <lem>

- $RETAINED COMMENT:*
- the lemma should appear **exactly as it appears in your digital edition**, including any markup that encodes information about reading difficulties and editorial intervention, but if you copy and paste the marked-up text of a lemma, pay attention to the following:
  - do not include structural markup (e.g. line beginnings and verse wrappers) in your lemmas
  - pay attention to start-tags and end-tags:
    - add the start-tag for retained markup commencing before and ending inside your lemma
    - add the end-tag for retained markup commencing inside your lemma and ending after it
    - add start and end-tags for a lemma snipped from within a longer stretch of phrase-level markup
- there are no strict rules for the **extent of your lemmas**; as with any critical apparatus, lemmas should be large enough to make them unambiguous within the line referred to in the @loc attribute and small enough to remain concise
  - lemmas should preferably be whole words (which may be compound members)
  - lemmas should not include any characters not present in the edited text, such as hyphens or ° signs at the beginning or end
  - avoid very long lemmas, if at all possible, by breaking them up into several smaller ones
- if a previous editor deserves credit for a particular reading, restoration or emendation that you have adopted in your edition, use the attribute @source in <lem> to give credit; see § for details

Alternative readings, restorations and emendations: <rdg>

- alternatives to your text should be given **as they appear in the edition you cite them from**
  - the alternative text should be marked up with XML tags to clearly indicate what the cited editor deemed unclear, emended or supplied
    - do not retain any editorial markup (such as brackets or asterisks), but convert it to XML
- the **extent of an alternative text segment** should always correspond exactly to the extent of its lemma
  - alternative text segments should not include any characters not present in the edited text, such as hyphens or ° signs at the beginning or end
- alternatives **must always be credited** to the editor(s) who proposed or endorsed them, using the attribute @source in <rdg>; see § for details

Encoding additional information about readings

- in order to avoid overcomplication, we recommend that if you wish to record any further details about your base reading or a variant, you should use a <note> within the relevant <app> entry
- we may later on decide to use additional attributes (with <lem> and/or <rdg>) as follows:
  - @wit to specify a text witness (as in manuscript studies, but applicable to epigraphy where multiple copies of an inscribed text exist)
  - @resp to specify the person encoding a reading
- do not, for the time being, use these attributes without first consulting us

Freeform apparatus notes: <note>

- **$apparatus notes**
  - optionally, add the attribute @loc to the <app> element to roughly indicate the locus (physical line, $verse-line or stanza) to which your entry pertains
  - optionally, use <lem> as the first element within <app> to specify a string of characters to which your note pertains
  - mandatorily, use the element <note> within <app>, containing a human-readable note in freeform text
-
-
-
- notes should preferably be complete sentences in English, starting with a capital letter and ending with a period (full-stop)
- they can be generic notes pertaining to the edition or the apparatus as a whole, or to a part of the text that cannot be conveniently limited to a specific locus with a line number and a lemma
  - the distinction between the use of apparatus notes and commentary entries is not clear-cut
  - we generally recommend using the commentary for anything that is not clearly an issue of choice of reading
- notes localised only with @loc may pertain to:
  - lines as a whole, e.g. to describe damage or the omission or different numbering of a particular line in a previous edition
  - stanzas as a whole, e.g. to describe their metre, or to document occurrences of an identical or closely related stanza in other inscriptions
- notes localised with @loc and <lem> pertain to a delimited locus, e.g. to specify the location of a partial restoration in an extended lacuna, or to add a comment on the locus
- notes will by default be assumed to be your own
  - where applicable, use the attribute @source in any <note> note element to credit a note cited verbatim from a previous editor; see § for details
  - or, if you paraphrase or summarise a previous editor's note, add a bibliographic citation (§) to the contents of your note

Apparatus example

- a snippet from the edition <div> (Allahabad *praśasti* of Samudragupta):

```
<div type="edition">
```

```
[...]
<lg n="2" met="śārdūlavikrīḍita">
<l><lb n="7"/><supplied reason="lost"
cert="low">ā</supplied><unclear>ry</unclear>y<supplied
reason="lost" cert="low">ai</supplied><unclear>h<unclear>īty
upaguhya bhāva-piśunair utkarṇṇitai romabhiḥ</l>
[...]
</div>
```

- the corresponding apparatus:
<div type="apparatus">

```
[...]
<app loc="7">
<lem source="#Bhandarkar_1981">
<supplied reason="lost"
cert="low">ā</supplied><unclear>ry</unclear>y<supplied
reason="lost" cert="low">ai</supplied><unclear>h<unclear>īty
</lem>
<rdg source="#Fleet_1888">
<supplied reason="lost">ā</supplied><unclear>ry</unclear>y<supplied
reason="lost">o</supplied> <unclear>h<unclear>īty
</rdg>
<rdg source="#Goyal_1967">
<supplied reason="lost">a</supplied><unclear>rh</unclear>y<supplied
reason="lost">o</supplied> <unclear>h<unclear>īty
</rdg>
<rdg source="#Agrawala_1983">
<supplied reason="lost">e</supplied><unclear>h</unclear>y <supplied
reason="lost">e</supplied><unclear>h<unclear>īty
</rdg>
</app>
[...]
</div>
```

Vipulā anustubh

removed from the main Guide

- in addition to the regular or *pathyā* form shown in the table above, *anuṣṭubh* permits certain alternative cadences known as *vipulā* in the odd quarters (while even quarters must always conform to the standard pattern)
  - *vipulā* cadences also place further restrictions on the first four syllables of the quarter in which they occur
  - *vipulā* lines may be optionally tagged as such, as per §
  - see the table below for the prosodic templates of recognised *vipulā*s and the label

to use as the value of @met

Permitted patterns in *pathyā anuṣṭubh*

	1	2–4	5	6	7	8
odd	⏓	– – ⏓ ⏑ – ⏓ – ⏑ ⏓	⏑	–	–	⏓
even	⏓	– – ⏓ ⏑ – ⏓ – ⏑ ⏑	⏑	–	⏑	⏓

Permitted patterns in *vipulā anuṣṭubh* (even lines only)

na-vipulā	⏓	– – – ⏑ – – – ⏑ –	⏑	⏑	⏑	⏓
bha-vipulā	⏓	– ⏑ –	–	⏑	⏑	⏓
ma-vipulā	⏓	– ⏑ –	–|	–	–	⏓
ra-vipulā	⏓	– – – ⏑ – – – ⏑ –	|–	⏑	–	⏓

Split-off aksara components

pasting my verbose summary (with some errors) here; the master copy of the guide contains only terse guidelines with fewer details

Scenario A: simple interrupted akṣaras

When a line break or a space (e.g. a binding hole) interrupts an akṣara by separating a prescript or postscript vowel marker from the other components of the akṣara, handle it as follows:

1. put ALL of the transliterated characters pertaining to the akṣara concerned on ONE side of the interruption, namely on the side where the main body is located

2. if an initial component is separated, add the character ⌜ at the beginning of this sequence; if a final component is separated, add the character ⌝ at the end of this sequence, ALSO on the same side of the interruption

THUS:

க<lb/>◌ா / ◌ெ<lb/>க / கெ<lb/>◌ா / ◌ெ<lb/>கா (where <lb/> may instead be <space/>) would be transliterated as

kā⌝<> / <>⌜ke / ko⌝<> / <>⌜ko

Scenario B: simple ambiguous kāl

When the context doesn't clearly tell how a kāl should be understood, the standard markup for ambiguities must be used:.

THUS:

கா would be k<choice><unclear>ā</unclear><unclear>ara</unclear><unclear>ar</unclear><unclear>ra</unclear></choice>

and கொ would be k<choice><unclear>o</unclear><unclear>era</unclear><unclear>er</unclear><unclear>re</unclear></choice>

I expect there will be very few cases where all of the alternatives are indeed possible, and I'm not sure கொ could ever be read as "kre" as in the last alternative I gave above. (Manu?)

In most cases, it will probably be acceptable (and much simpler) to encode these as

k<unclear>ā</unclear> and k<unclear>o</unclear> OR ka<unclear>ra</unclear> and ke<unclear>ra</unclear> (whichever seems more likely); readers who care will I assume be able to figure out the alternatives

Scenario C: interrupted akṣara with ambiguous kāl

Combine the above approaches.

THUS:

க<lb/>◌ா would be k<choice><unclear>ā</unclear><unclear>ara</unclear><unclear>ar</unclear><unclear>ra</unclear></choice><lb/>⌉

கெ<lb/>◌ா would be k<choice><unclear>o</unclear><unclear>era</unclear><unclear>er</unclear><unclear>re</unclear></choice><lb/>⌉

and ெ◌<lb/>கா would be ⌈<lb/>k<choice><unclear>o</unclear><unclear>era</unclear><unclear>er</unclear><unclear>re</unclear></choice>

Notice that in the above cases, the sign ⌉ may appear after an interruption preceded by *ra* - hence the clause "or sequence" in my definition above. This would then mean that "if the interruption is preceded by *ra* in transliteration, this means that the whole of *ra* is in fact engraved before the interruption; but since we think it could also be an *ā*, we have had to put its transliteration before the interruption".

Scenario D: interrupted akṣara followed by a lacuna with supplied kāl

Again, combine the above approaches. If for example you have க or கெ at the end of the line and the beginning of the next line is lost, but you know from the context that the correct reading must be kā or ko, then mark up

k<supplied reason="lost">ā</supplied><lb/>⌉

and k<unclear>o</unclear><lb/>⌉

Notice that in the latter case the o is not tagged as supplied but as unclear, since கெ (with part of what you read as *o*) is visible at the end of the former line.

We might also want to mark up ⌉ in these cases as <supplied>, since there is nothing actually legible in the latter line; but I think this is unnecessary trouble, since our ⌉ is not the transliteration of a character but a metamark conveying information about grapheme distribution.

Scenario E: interrupted akṣara followed by a lacuna with possibly lost kāl

I don't think anything needs to be done in the markup to indicate a "possibly lost kāl" in a lacuna where you can't make a restoration from context. Thus, if you have க or கெ followed by a lacuna (directly or after an intervening line break or binding hole), then I guess you would simply transliterate ka/ke followed by <gap/> (possibly adding <unclear> to the vowel in those transliterations), and again leave it to readers to know that a ka or ke before a lacuna may actually be a partially preserved kā or ko, even if the lacuna comes after an interruption.

Problematic numerals: OLDER GUIDE TEXT

Difficulties in reading numbers

- problems with reading numeral signs (e.g. lacunae, unclear and ambiguous readings) can be marked up in the same way as other reading difficulties (see §XXX)
- tags for reading difficulties go **outside** any <g> elements applied to numeral characters, but they go **inside** the <num> element that wraps numbers as a whole
- numbers whose reading is problematic will usually not have a definite @value attribute for the <num> element, so that in addition to marking up the reading difficulty, the problem of value needs to be handled in one of the following ways, depending on the nature of the problem
  - **1. number wholly lost**: use the <num> element without @value
    - e.g. for one lost/illegible numeral character: `<num><gap @></num>`
  - **2. number partly lost, with a range of possible values**: instead of the attribute @value, use @atLeast and @atMost in the <num> element to record the lowest and highest possible value of the number as a whole
    - e.g. for three digits in place value notation, where the first two digits are 1 and 0, and the last digit is illegible: `<num atLeast="101" atMost="109">10<@gap></num>`
  - **3. number partly lost, with two or more non-contiguous possible values**: record the most likely (or, if no such exists, the most straightforward) value of the number as a whole in the @value attribute, and use one or more <certainty> elements before the closing </num> tag to record the possible alternatives
    - the syntax for these elements is: `<certainty match="../@value" locus="value" assertedValue="#"/>` where # represents the alternative value (and all other details of the code remain unchanged
    - e.g. for three digits in place value notation, where the first and last digits are 1 and 2, and the middle digit is illegible:

```
<num value="102">1<@gap>2<certainty match="../@value" locus="value"
assertedValue="112"/><certainty match="../@value" locus="value"
assertedValue="122"/><certainty match="../@value" locus="value"
assertedValue="132"/><certainty match="../@value" locus="value"
assertedValue="142"/><certainty match="../@value" locus="value"
assertedValue="152"/><certainty match="../@value" locus="value"
assertedValue="162"/><certainty match="../@value" locus="value"
assertedValue="172"/><certainty match="../@value" locus="value"
assertedValue="182"/><certainty match="../@value" locus="value"
assertedValue="192"/></num>
```

    - theoretically, it should be possible to assign a degree of confidence to each of these alternatives, using the attribute @degree in the <certainty> element

    - @degree takes values between 0 and 1, so e.g. @degree="0.3" in a single <certainty> element would mean that the editor assigns a 30% probability to the value asserted in that element, and implicitly mean that the value listed for the parent <num> element is deemed to be 70% likely
    - in fact, the EpiDoc guidelines do not use @degree at all, and using this attribute would be cumbersome and error-prone in long lists of alternative values, provided that the total value of certainty degrees for any given <num> element is supposed to add up to 1 (which is nowhere said explicitly in the TEI guidelines, but which I take to be understood)

- IN ADDITION TO OR INSTEAD OF THE ABOVE COMPLEX METHODS, WE COULD USE THE FOLLOWING METHOD FOR ALL READING PROBLEMS ASSOCIATED WITH NUMERALS. This method involves the following steps, all of which must be done for all difficulties involving numbers
    - a. record the most likely (or, if no such exists, the most straightforward) value of the number as a whole in the @value attribute
    - b. before the closing tag </num>, add the element `<certainty match="../@value" degree="#"/>`
        - where # represents the degree of certainty you assign to the recorded most likely value
        - this degree of certainty is to be expressed by a number above 0 and below 1, e.g. 0.6 meaning “a sixty percent chance of being correct”
    - c. also before the closing tag </num>, add a note in the following format: <note type="certainty">TEXT</note>
        - where TEXT represents a human-readable note in which you describe the problem and the possible alternative values of the number as a whole

Editorial corrections to numbers

- occasionally, an editor may be able to restore a lost number, or even emend an incorrectly inscribed one, e.g. on the basis of the number being also written out in words
- as with reading difficulties, tags for editorial correction go **outside** any <g> elements applied to numeral characters, but they go **inside** the <num> element that wraps numbers as a whole
- the @value attribute of the <num> element should in this case reflect only the corrected/restored value
- $optionally add <certainty> or <precision> to restored numerals?

Potentially useful in future

Tagging names that may signify one person or another

- see discussion on MARKUP for suggestions, https://lsv.uky.edu/scripts/wa.exe?A2=ind2002&L=MARKUP&P=16155