

The Menota handbook

Guidelines for the electronic encoding of Medieval Nordic primary sources

Version 2.0
TEI P5 conformant

Odd Einar Haugen
(editor)

Tone Merete Bruvik, Matthew Driscoll, Odd Einar Haugen,
Karl G. Johansson, Rune Kyrkjebø and Tarrin Wills
(contributors)

The Medieval Nordic Text Archive
www.menota.org

Bergen 16 May 2008

ISBN 978-82-8088-400-8

Preface

The first version of the Menota handbook, v. 1.0, was published on 20 May 2003, and a minor revision, v. 1.1., on 5 May 2004. Both versions are TEI P4 conformant. It was soon agreed among the editors that a new version should be published as soon as TEI P5 was finalised. This took place in November 2007, and the present v. 2.0 of the handbook is now TEI P5 conformant. While the previous versions were written in HTML, the present version is written in XML throughout, which we thought would suit a handbook on how to use XML. All things considered, v. 2.0 of the handbook represents a major revision.

The [preface](#) to version 1.1 contains an overview of the background and the contributors to versions 1.0 and 1.1, which need not be repeated here. In the present version, Karl G. Johansson (Oslo) has written a new chapter on names, ch. 9.1, and Tarrin Wills (Sydney/Aberdeen) has made a thorough revision of ch. 9.2 on metrical encoding, making it compatible with the international [Skaldic project](#). He has also extensively revised ch. 7, which should be regarded as a completely new chapter. Furthermore, he has written a new introduction, ‘What is Menota’, aimed at readers who would like to get a brief overview of these guidelines.

The remaining chapters and appendices have been revised by Tone Merete Bruvik and Odd Einar Haugen. Tone Merete Bruvik has checked and validated all examples, given advice on all kinds of encoding questions and developed new TEI P5 conformant schemas. Thus, Appendix D is her work entirely. Due to her long-standing contribution to the handbook she is now included as one of the six authors of the book, as will appear on the new [title page](#). The remaining parts of the revision have been implemented by Odd Einar Haugen.

The major changes from v. 1.1 are the following:

1. The handbook now offers two schemas for validation of XML files: a DTD schema (as in the previous version) and a RELAX NG schema (which is new). We recommend using a RELAX NG schema rather than a DTD, since RELAX NG allows for namespaces.
2. The concept of namespace, which was introduced in TEI P5, has been implemented in the handbook. As a consequence, all Menota-specific elements and attributes are clearly marked as ‘me’, e.g. `<me:norm>` rather than only `<norm>`. See the new chapter 1.9 for an explanation and a complete list of additional elements and attributes.
3. As stated above, Tarrin Wills has written a new introduction, ‘What is Menota’.
4. In ch. 1, the poem ‘Upon Julia's Clothes’ has been replaced with what some thought was a more suitable example, a stanza from the Eddic poem ‘Thrymskvida’, and the ensuing discussion has been made TEI P5 conformant.
5. In ch. 2.2.2 and ch. 5.1 we now advise readers to use Unicode encoding also outside Basic Latin, rather than entities. We insist that entities be used for characters in the Private Use Area.
6. Ch. 2.2.3 on the element `<c>` is new.
7. Ch. 2.3 has been revised with reference to deviation in word division and encoding of word constituents.
8. Ch. 2.4 on punctuation and white space is new.
9. In ch. 3, we introduce the new TEI P5 elements `<ex>` (rather than `<expan>`) and `<am>` (rather than `<abbr>`), see also ch. 6.1.

10. The example in ch. 3.2 is new, and so is the image and the displays (using the Andron font by Andreas Stötzner, Leipzig).
11. In ch. 3.3 and 3.4 an important distinction has been drawn between ‘single-level transcriptions’ and ‘multi-level transcriptions’ (the Menota way, as it were). The new TEI element **<choice>** has been introduced as part of the multi-level transcription. This has also been done for **<sic>** and **<corr>** (see ch. 7).
12. Ch. 4.5 deals in more detail with prosimetrum texts, in which the elements **<p>** and **<lg>** are mixed.
13. Ch. 4.6 has a new solution to the problem of overlapping headings.
14. Ch. 4.8 on punctuation and hyphenation has been extensively revised.
15. Ch. 4.9 on initials and highlighted characters (*littera notabilior*) is new.
16. Ch. 4.10 on overlapping structures - which is a particularly thorny issue in XML - is completely new.
17. Ch. 5.1 is new, and part of ch. 5.2 has been moved from ch. 2 (which, as a consequence, has become simpler). Ch. 5 is now Unicode v. 5.0 compatible.
18. Ch. 6.1 is new, and the whole chapter Unicode v. 5.0 compatible.
19. As stated above, ch. 7 has been extensively revised by Tarrin Wills, and is for all intents and purposes a new chapter.
20. Ch. 8 has been extensively revised and introduces a new system for the general encoding of grammatical forms, using the **@me:msa** attribute. It also contains specifications for Old Norse and a discussion of lemmatisation of non-Nordic material. This should also be seen as a new chapter.
21. As stated above, ch. 9 has been extensively revised by Karl G. Johansson and Tarrin Wills.
22. Ch. 10 has been revised on a number of points, and now contains a discussion of the whole header (not only the manuscript description, as in v. 1.1).
23. The index is new.
24. The Menota schemas - DTD and RELAX NG - have been updated to TEI P5, and all examples in the handbook have been validated against a RELAX NG schema by Tone Merete Bruvik.
25. The discussion of headers in Appendix E has been revised and simplified. The downloadable headers in E.4 have been updated.

Tor Gjerde (Trondheim) has given detailed and valuable comments on the previous version of the handbook, which proved very useful in the process of revising the book. Vemund Olstad (Bergen) has used his considerable expertise with the FO processor in order to generate the PDF files of the handbook from the underlying XML files. Many thanks to both.

Bergen, 16 May 2008

Odd Einar Haugen (editor)

Introduction: What is Menota?

1. Electronic editing of medieval texts

The purpose of these guidelines is to define a framework for machine-readable editions of medieval Nordic texts. These guidelines are recommended for any scholar who wishes to produce detailed, machine-readable editions of primary works, that is, medieval Nordic manuscripts.

1.1. Menota and traditional editing practice

Editions may include a very great amount of information in addition to the basic text of the manuscript: introductory material, including textual and literary contexts; the textual content, including diplomatic and/or normalised text; a variant apparatus or various manuscript versions; notes and other forms of critical apparatus; glossaries and/or indices of names.

The present guidelines address all of these parts of an edition. The one exception is the textual or variant apparatus: as the approach of these guidelines is to encode different manuscript versions, the textual apparatus develops as each manuscript is encoded and aligned.

The approach taken here, however, differs from traditional editions in the way in which the additional information is included and consequently the possibilities of presentation. Traditional print editions rely on a very large amount of referencing between the text and the apparatus: note references may refer the reader to the notes section; glossaries and indices refer the reader back to the main text; the textual apparatus refers usually to line and/or page numbers; and aligned texts usually rely on visual parallels, such as facing pages. The approach taken here allows all of this information to be encoded without complex referencing, allowing information about a section of text to be checked, or presented, at the same time, depending on the capabilities of the display medium. The complexity of referencing, however, is replaced with a certain amount of complexity in encoding.

1.2. Machine-readable editions

The approach of Menota differs from the production of electronic texts using word-processing or desktop publishing software because the texts are machine-readable, that is, the texts are marked up in a way that meaningful entities within a text can be read and manipulated by a computer.

The approach taken here can be used to distinguish between different types of information in the text and consequently can extract and present the information of most interest to particular users, for example, students, literary scholars, linguists, palaeographers. A student may wish to read the normalised text; a linguist might only be interested in the word distribution, and so on.

Using this method, one can also produce editions for different media: printed and electronic books, interactive web applications, portable devices, CD-ROMs and so on.

1.3. Menota and other encoding schemes

Menota is based on the scheme defined by the Text Encoding Initiative. It defines further extensions based primarily on two major differences between Medieval Nordic texts and most other comparable corpora:

1. A very large degree of orthographical variation. This makes linguistic analysis difficult because of the difficulty in searching for words on the basis of a lemma. The compilation of glossaries, for example, cannot be done in any systematic way.
2. A very large degree of abbreviation of letters, groups of letters, words and so on.

The two problems are dealt with by breaking the text into 3 prototypical levels, where the text is encoded in its abbreviated form, in its expanded form and in a normalised orthography. These textual levels constitute the primary difference between Menota and standard TEI. Texts can be encoded on only one of these levels (typically the diplomatic), but can easily be extended to two or more levels, thus making it more versatile than traditional editions, which are restricted to representing the text in only one way.

2. How to use these guidelines

These guidelines provide a way of representing a text in a machine-readable and platform-independent way. They do not provide in themselves a way of publishing the text, but rather a way of encoding a text so that it can be published and analysed by other means in a variety of ways. In short, you can use these guidelines to represent characters, words and other meaningful units of text, in a way that is consistent and unambiguous. The approach is represented by the chapters:

1. An introduction to XML.

XML is the electronic language used to represent features of the edition. It differs from the languages used by, for example, word processors and typesetting engines, in that it is used to represent types of content rather than ways of displaying the text. XML is currently the most common way of encoding textual content. Learning how XML works is perhaps the most difficult aspect of these guidelines, but once a few fundamental concepts are grasped, it is a useful tool which can be applied to a range of other areas, such as web publishing.

2. How to encode the basic units of characters and words.

The encoding of characters in an unambiguous way represents the most basic step towards producing an exchangeable and machine-readable edition. It is in fact a fairly simple procedure which requires almost no knowledge of XML, but instead a basic idea of abstraction. The first thing to grasp is that the way characters are represented here is independent of individual fonts. One of the problems with many early electronic editions is that they have used non-standard fonts, and combinations of fonts in word processing programs. Once the font is obsolete, or if the software becomes obsolete, the electronic text is no longer of much use. The approach taken here overcomes these problems by representing characters either using a standard encoding or electronic references to different character types.

3. Levels of textual representation.

The guidelines discuss a simple and straightforward way of encoding text in a single-level transcription. However, in order to deal with the problems in section 1.3 above (frequent abbreviation and orthographic variation), the guidelines recommends a multi-level transcription. The text is divided into three 'levels': one which attempts to represent

the text as it appears in the manuscript, including abbreviation and significant letter forms (reduced to a partially-limited set, however); the second represents the text in the orthographical form of the manuscript, but expands abbreviations, generally providing a diplomatic representation of the text; and the third level involves normalisation to a set of letters, based on the actual orthographical system, but representing the phonological system as it was when the text was believed to have been composed.

Editors using these guidelines may wish to use any combination of the levels to encode the text.

Each level is encoded on a word-by-word basis.

4. Representing the structure of the document.

The guidelines go on to explain how to encode larger units and features, including linguistic structures such as chapters, paragraphs and headings; physical features such as pages and lines in the manuscript; verse material and punctuation. Such information is fairly straightforward to represent, and this chapter should not provide many difficulties to editors who have grasped the earlier material.

5-6. Encoding characters and abbreviations.

This chapter describes in detail how different letter-forms and abbreviations - and their expansions - are to be represented in a Menota-compliant edition.

7. Representing altered, corrected and unreadable text.

This chapter explains how to encode characters and words which fall outside of the normal flow of text, because they are altered by scribes, incorrect or illegible. Such features are frequent in primary sources, but need to be encoded unambiguously so that the edition represents the status of all the text in relation to the primary source and its scribes, and subsequent editors.

8. Lemmatisation.

This chapter provides an approach to the linguistic encoding of a text for editors who are interested in producing search engines and glossaries. Basically, lemmatisation is the process by which every word-form in the text is linked to a single word without grammatical variation - the equivalent to a dictionary head-word or lemma. Once this process is done, the text can be searched for words, regardless of morphological or orthographical variation. Each word can be linked to a glossary, and vice-versa.

9. Encoding additional features.

This chapter describes how editors may wish to make references between the encoded text and other types of information. For example, it may be useful for an editor to treat a name occurring in a text as both a linguistic entity (a word) and a potential reference to information about an individual. This is roughly the equivalent to indexing, where named entities - people, places and so on - are linked to the text.

10. Encoding front-matter and other meta-information.

The final chapter describes the additional 'meta' information about the edition is to be encoded in the document header. Such information makes it much easier to understand the relationship between the file and other types of documents by describing and categorising it. The header also contains information about the process and responsibility of creating the edition.

3. Basic content of the edition

The content of the edition is basically the same as a print edition of a primary work. It contains:

1. Front matter, including a title for the work, publication information, simple information about the editor(s) responsible, a description of the editorial approach taken, detailed acknowledgements of contributions and so on. All this information is encoded in the TEI header, described in chapter 10.
2. A table of contents; because the encoded document represents the parts of the text, including headings, in a machine-readable way, the table of contents can be generated automatically. (For comparison, the document you are reading is encoded in xml, with each section heading marked as such - the contents at the top of the document are generated automatically from this information.)
3. The text itself, including representation of: the orthography (described in chapters 2 and 5), abbreviations (chapter 6); linguistic information, including division into words (chapter 2) and textual levels for each word (chapter 3), lemmata for words (chapter 8); higher-level structures such paragraphs and chapters, physical pages and lines (chapter 4); alterations made to the text by scribes and editors (chapter 8); and references to people, places, and so on (chapter 9).
4. Back matter, potentially automatically-generated, including a glossary generated from word lemmata and indices generated from other encoded information such as names.

Chapter 1. Text encoding using XML

1.1 What is XML?

XML, Extensible Markup Language, is a recommendation, endorsed by the [World Wide Web Consortium](#), which defines a simple yet flexible generic syntax for document markup. XML, like its predecessor SGML, Standard Generalised Markup Language, developed by IBM in the 1970s and 1980s, allows for the definition of system-independent methods of representing texts of any kind in electronic form.

The term ‘markup’, originally used for the (hand-written) instructions added to a manuscript or typescript to indicate to the compositor how the printed text was to look in terms of spacing, font size, use of italics and so on, has been carried over into electronic document processing to describe the codes used to indicate these same features and other aspects of processing. A ‘markup language’ is therefore at its most simple a set of codes which are used to indicate or ‘tag’ certain features in the text, normally for formatting purposes. In most modern software packages the markup is generated with little or no conscious effort on the part of the user – in many modern word processing programs, such as the ubiquitous Microsoft Word, the user is not even given the option of viewing the codes. But they are there: and to see just how many one need only open a document produced in, say, Word or WordPerfect in a plain text editor such as Notepad. A text of even a few short lines will be prefaced by several dozen lines – possibly even pages – of code.

The problem is that every program has its own set of codes, and it is only rarely possible to convert files from one to another without at least some loss of formatting. And it isn't just the formatting that goes haywire – any exotic (read non-English) characters are also likely to mutate. SGML was originally developed in order to avoid these problems by being entirely platform independent – hence G for generalised. It achieves this by identifying the logical elements of the document rather than specifying the processing to be performed on it: the markup is descriptive, in other words, rather than procedural. With descriptive markup, the same document can be processed by many different pieces of software, each of which can apply different processing instructions to those parts of it which are considered relevant.

SGML's greatest success has been HTML, Hyper-Text Markup Language, the language of the World Wide Web. HTML restricts document authors to a finite set of tags, however, most of which are presentationally oriented, and is thus inappropriate for most things other than web design. XML is essentially ‘trimmed down’ SGML. It is not, in other words, a single, predefined markup language like HTML: like SGML it is a metalanguage – a language for describing other languages. The syntax is essentially the same as SGML, but some of the more complex and lesser used options have been removed.

The great advantage of XML is that it brings the power and flexibility of SGML to the Web; an XML document can be marked up entirely in accordance with the needs of the user and the result displayed in a standard web browser (see section 1.8 below). The implications for philologists are staggering.

In what follows, most of the more relevant areas of XML markup are touched upon. For a more thorough grounding, one of the many printed handbooks or websites devoted

to XML should be consulted. A good place to start would be the World Wide Web Consortium's own XML pages: <http://www.w3c.org/XML/>.

1.2 Appearance vs. structure

It is customary, in English and most other Western European languages, to use italic type in texts printed otherwise in plain roman to set certain things off the rest. *Hart's rules for compositors and readers at the University Press, Oxford* (39th ed.), for example, stipulates that the titles of books, films, plays, works of art and periodicals (but not chapters, shorter poems, articles) should be printed in italic, as should the names of ships (but not public houses), words and short phrases in foreign languages (other than those, such as quiche and blitzkrieg, that have been sufficiently anglicised so as to render this unnecessary), stage directions in plays, theorems in mathematical works and biological and zoological nomenclature. Although Hart's doesn't mention it, italic is also regularly used to indicate emphasis, for example in novels: 'I most certainly *didn't* ask him to come.' With ordinary word-processing software, all these things would be marked up in the same way, i.e. with the relevant codes for 'italic-on' and 'italic-off'. If you think of the computer as a glorified typewriter and are only interested in producing copy with the correct formatting, fine. If you wish to take advantage of the possibilities offered by sophisticated information retrieval systems, however, you're in trouble, since a search engine will not be able to distinguish foreign words from book titles or the names of ships, for the simple reason that procedural markup such as that produced by ordinary word-processing software only indicates how something is to be displayed, but not why is it to be displayed that way. With descriptive markup, on the other hand, elements in the text are tagged according to their function – titles as titles, foreign words as foreign words, stage directions as stage directions and so on. These can then be processed in whatever way one desires, for example displayed in italics. By concentrating on the structure of the document rather than its appearance a great many possibilities are opened up. Elements in the text can be marked up even where one has no desire to format them in any special way. One might wish, for example, to tag the names of persons, so that a search for 'King George', for example, would turn up only persons of that name rather than vessels or public houses.

1.3 Elements

The key concept in SGML/XML markup is the element. An element is essentially a textual unit, the idea being that texts, like houses, are made up of repeated occurrences of basic units arranged in a hierarchical structure; longer works in prose will be divided into chapters or sections, and these into sub-sections and then further into paragraphs, and there also may be lists and tables. Works of poetry may be divided into cantos or fits, and these into stanzas, and the stanzas into couplets, the couplets into lines, the lines into feet etc. The individual sections, whether chapters or cantos, will often have headings, which are not strictly speaking part of the main text, but nevertheless belong with it. Moreover, these elements will only combine in certain ways. A chapter will not begin in the middle of a paragraph, for example, or in a footnote. In SGML/XML pairs of tags are used to mark off these units, a start tag and an end tag, with the text in between being referred to as the element's content. Tags are placed within angle brackets, with a solidus to indicate an end tag. Chapters in a book, for example, could be demarcated by placing a **<chapter>** tag at the beginning of each one and a corresponding **</chapter>** tag at the end, while within each chapter there would be any number of paragraphs, tagged, say, **<paragraph>**. The way these two elements relate to each other hierarchically is determined by the *schema*

being used, which in this case would stipulate that a **<chapter>** must contain one or more **<paragraph>** elements (more on schemas in [ch. 1.8](#) below). SGML/XML syntax is really quite simple: for each element there is a declaration enumerating what other elements it may or must contain, how many of each, and if there are any constraints on the order. The more elements one has in one's system the more complicated, and subtle, that system becomes.

Let us take a concrete example, the two first stanzas of the Eddic poem *Þrymskviða*, rendered in normalised orthography (for simplicity of display, we are using ‘o with diaeresis’ rather than ‘o with tail’). The text is based on the edition by Jón Helgason (1955) and the translation is the one by Carolayne Larrington (1996):

Reiður var þá Vingþórr er hann vaknaði ok síns hamars um saknaði, skegg nam at hrista, skör nam at dýja, réð Jarðar burr um at þreifask.	(Thor was angry when he awoke, and missed his hammer; his beard bristled, his hair stood on end, the son of Earth began to grope around.
Ok hann þat orða alls fyrst um kvað: Heyrðu nú, Loki hvat ek nú mæli, er eigi veit jarðar hvergi né upphimins: áss er stolinn hamri!	And these were the first words that he spoke: ‘Listen, Loki, to what I am saying, what no one knows neither on earth, or in heaven: the hammer of the God is stolen.’)

The structure of this poem is clear enough: it is made up of two stanzas each of which contains eight short lines. This structure could be marked up in the following way:

```
<poem>
  <stanza>
    <line>Reiður var þá Vingþórr</line>
    <line>er hann vaknaði</line>
    <line>ok síns hamars</line>
    <line>um saknaði,</line>
    <line>skegg nam at hrista,</line>
    <line>skör nam at dýja,</line>
    <line>réd Jarðar burr</line>
    <line>um at þreifask.</line>
  </stanza>
  <stanza>
    <line>Ok hann þat orða</line>
    <line>alls fyrst um kvað:</line>
    <line>Heyrðu nú, Loki,</line>
    <line>hvat ek nú mæli,</line>
    <line>er eigi veit</line>
    <line>jarðar hvergi</line>
    <line>né upphimins:</line>
    <line>áss er stolinn hamri!</line>
  </stanza>
</poem>
```

If we abstract from this and attempt to describe the structure of poems in general we could say that a poem consists of one or more stanzas each of which is made up of one or more lines. This structure could be expressed in a Document Type Definition as follows:

```
<!ELEMENT poem (stanza+)>
```

```
<!ELEMENT stanza      (line+)>
<!ELEMENT line        (#PCDATA)>
```

The + sign after stanza and line means they are required and repeatable, i.e. can occur one or more times (a question mark would indicate an optional element, i.e. one which can occur zero or one time, while an asterisk would indicate that the element was optional and repeatable, i.e. can occur zero or more times; if there is no occurrence indicator, the element must occur once and only once). #PCDATA is ‘parsed character data’, which essentially means any number of valid characters. There is one obvious problem with this model, which is that it requires that all poems consist of at least one stanza, which is somewhat counter-intuitive, since it could be argued that a poem of only one stanza is made up only of lines. To remedy this, the content model for the poem element could be given as (line + | stanza+); the two are separated by a vertical bar, the ‘or’ connector, which shows that either can be used but not both, and each is marked with a plus sign to show that whichever is used, it can be used more than once. A poem, in other words, consists either of one or more lines or of one or more stanzas (each comprising one or more lines).

A poem will also normally have a title and be attributable to an author (even if that author – as in the case of *Prymskviða* – is the highly prolific ‘Anon.’). The name of the author will obviously not always appear with the poem, however, for example in a series or collection, where there are several poems by the same author. These could be added by redefining the content model of poem as:

```
<!ELEMENT poem (title, author?, (stanza+ | line+))>
```

Here, the pair of elements stanza and line are grouped together in round brackets (parentheses) to show they are to be treated together; the comma acts as ‘sequence connector’, indicating that the elements/groups must occur in this order. In plain prose this means that a poem must have a title, may have an author, and then will have either one or more stanzas or one or more lines. This is still not entirely satisfactory, however, since there will also be poems without titles (haikus for example), which would not be allowed with this content model. Nor would it be possible to reverse the order of author and title. The content model must therefore ideally allow for optional and non-repeatable title and author elements which can appear in either order, but only preceding the text of the poem itself. In order to allow for this kind of flexibility, the schema needs to become slightly more complex:

```
<!ELEMENT poem (((title | author?) | (author | title?)),
                 (stanza+ | line+))>
```

Here, the two possibilities are grouped together in round brackets with an or connector in between – title with an optional author or author with an optional title – and both possibilities are marked as optional.

A simpler way of dealing with this might be to define an element called, say, head, with the following content model:

```
<!ELEMENT head (#PCDATA | author | title)*>
```

This would allow one to preface the poem with plain text, author and title elements in any combination. This allows much greater flexibility, but also reduces greatly the amount of control one has over the document. With this content model, for example, there would be nothing to prevent one from having more than one title.

It is obvious too that a larger unit is required in order to accommodate more poems, for example <collection>. If one envisaged this collection as an anthology, one would probably wish to divide it into sections, in which poems by a particular poet were grouped

together; in the case of Eddic poems, one might make a division between mythological poems and heroic lays. Each of these sections would have a heading and possibly some prefatory matter, giving information on the author. Given the complex structure of all but the simplest documents, it is easy to see how a schema can quickly become very complicated indeed.

Other elements used in markup have less to do with the overall structural hierarchy of the document and are more free-floating, i.e. can appear in a variety of contexts. The principal use of tagging such as this is to enable searches: one marks things so as to be able to find them later. One might, for example, wish to indicate that ‘Vingþórr’ in line one of our poem is the name of a person:

```
<line>Reiðr var þá <name>Vingþórr</name></line>
```

1.4 Attributes

Without further information, the usefulness of such tagging is sometimes limited. More specific information about a particular element instance can be given as an attribute. Looking at the stanzas just cited, one might, for example, want to add attributes to the elements `<poem>` and `<stanza>`, using convenient typologies to indicate genre, form, metre or rhyme-scheme, and a `@number` attribute to the stanza and line elements. It might also be an advantage to indicate the type of name, in order to distinguish personal names from the the names of places, ships, swords etc., or perhaps even have a separate element `<person>`, with attributes such as `@gender` and `@role`:

```
<line number="1">
  Reiðr var þá <person gender="male" role="protagonist">Vingþórr</person>
</line>
```

One might want to identify the name in some more precise or uniform way, for example to make clear that ‘Vingþórr’ is identical to the heathen god otherwise known as ‘Þórr’; this could be done with an attribute called `@reg`, for ‘regularised’, as follows:

```
<person reg="Þórr">Vingþórr</person>
```

Like elements, attributes are declared in the schema, a list of possible attributes (ATTLIST) being given for each element; it is also possible to specify what kind of value is acceptable for each attribute, and if necessary a default value.

```
<!ELEMENT person (#PCDATA)>
<!ATTLIST person
  gender (male | female | unknown) "unknown"
  role CDATA #IMPLIED
  reg CDATA #IMPLIED>
```

When publishing this poem one might want to put all names in italics, small caps or another form of emphasis. Reproducing this would be an easy matter if all personal names were tagged as such, but the real advantage of this kind of tagging is for search purposes. One could, for example, search for references to Þórr in all Old Norse poems, regardless of whether he was called Vingþórr, Ásapórr, Bergþórr, or simply Þórr – or by a *kenning* such as ‘Jarðar burr’.

Attributes such as **type** and **subtype** are useful precisely because they allow for searches at varying degrees of abstraction. Markup such as the following:

```
I have a <dog>dog</dog>.
```

is all but pointless, since a free-text search for the word ‘dog’ would yield the same result. If, on the other hand, one moves to a greater level of abstraction, for example to:

```
I have a <animal class="mammalia" order="carnivora"
        family="canidae" genus="canis"
        species="familiaris">dog</animal>.
```

one could search for all animals, all mammals, all carnivores, all dog-like creatures and finally all dogs (wild and domestic), depending on how widely one wished to cast one's net. The use of standard international typologies such as this also allows for cross-linguistic searches.

1.5 Entities

The aspects of SGML/XML discussed so far are all concerned with the markup of structural elements within the document. SGML/XML also provides a mechanism for encoding and naming parts of the document's content: through entities. An entity is a kind of shorthand, a way of stating that a particular string of characters in the document should be replaced when the document is processed by some other string; this other string can be of any length, from a single character to a separate file containing millions of bytes, for example a text file or digital image. The name of the entity (entity reference) is placed between an ampersand and a semicolon: &entityname;. Such entities, which are known as General Entities, are defined in an external entity set which is itself referenced in the schema being used (see appendix D for examples). A simple declaration for a general entity looks like this:

```
<!ENTITY vth "Vingþórr">
```

Here, whenever the processing software (a parser or browser) encounters the entity &vth; it replaces it with the text 'Vingþórr'. In the case of our single poem, there is obviously no real advantage to treating the name of the protagonist in this way, but in longer documents or collections of documents it can be an extremely efficient way of dealing with repeated content. Entities may also contain XML markup (provided it is well-formed) as well as text:

```
<!ENTITY vth "<god type='áss'>Vingþórr</god>">
```

An entity can also refer to an external file, as in the following example:

```
<!ENTITY chapter1 SYSTEM "chapter1.xml">
```

Such entities are called system entities: instead of the replacement text, the SYSTEM keyword and a relative or absolute URL are given. The processing software will then replace the entity with the document found at the address given, i.e. insert that document into the existing document. The resulting document must be well-formed XML, so one must ensure that the document to be inserted is itself well-formed (although it need not have a single root element) and does not for example contain a prologue (i.e. XML and/or DOCTYPE declaration).

A third type of entities are called parameter entities; these are used inside markup declarations and need not concern us here.

Entities are particularly useful for providing descriptive mappings for non-standard (i.e. non-English) characters, such as the accented vowels (‘í’, ‘ê’, ‘ä’ etc.) used in many European languages, the German ‘ß’, Icelandic ‘Þ’ or Danish and Norwegian ‘ø’, which are notoriously non-portable. There are standard entity sets for characters used in the western-European languages (ISOLat1 and ISOLat2), as well as character sets for Greek (ISOgrk1), Cyrillic (ISOcyr1) and other alphabets. The Unicode standard covers all the world's languages, living and dead, and also allows for user-defined characters; the current

version 5.0 contains over 97,000 characters. Each of these characters is assigned a unique code point, which can be encoded in a variety of ways. The most common, mentioned above, is UTF-8. Numerical character references are either decimal or hexadecimal; decimal references begin with an ampersand and the number sign or hash mark (#), to which hexadecimal references add an x. Thus the hexadecimal character reference for the letter þ is `þ`, while the decimal reference is `þ`. These numerical character references are supported by standard browsers and do not need to be defined specially in the schema. One may, however, prefer to use entities which are more immediately intelligible to humans, for reasons of proof-reading or whatever, for example ‘þ’ for ‘þ’. It is a simple matter to define characters as entities, giving as the replacement text the numerical character reference.

```
<!ENTITY thorn "&#x00FE;">
```

Entity references can also be used for characters required for specific kinds of texts; the producer of a diplomatic text edition might want to distinguish between single and two-storey a, for example, by using separate entities. These entities could have the same replacement text, and thus appear identical when displayed, but still be available for search purposes.

Entities for Old Norse special characters are defined, with their Unicode values, in [ch. 5](#) and [ch. 6](#). See also [Appendix A](#).

1.6 Putting the pieces together

These, then, are the basic parts of a SGML/XML document. The key is the *schema*, in which the elements, with their attributes, are defined in terms both of their content and their relationship to other elements, and entities or entity sets are defined or referenced. In this handbook, we offer two closely related schemas, a Document Type Definition (DTD) schema and a RELAX NG schema. As of v. 2.0 of the handbook we recommend the RELAX NG schema, which is more flexible, yet at the same time somewhat stricter than a DTD. Those who are familiar with a DTD will not find the change dramatic at all.

Anyone can, if he or she so wishes, devise a schema to meet whatever encoding needs he or she may have. A host of XML authoring tools, parsers, browsers and search engines are available, many for free over the web. If all you want to do is make a searchable list of your CD collection, for instance, it is a relatively simple matter to create your own schema. Most people will prefer to use an existing application, however. As was said, the most successful implementation of SGML to date is HTML (the XML version is known as XHTML). But HTML is not flexible enough to deal with all but the most basic of texts: there simply aren't enough elements. Another fundamental weakness of HTML is that it has, despite its origins in SGML, decidedly procedural (rather than descriptive) tendencies, in that many elements are used for the effect they will produce when the document is displayed rather than to mark up structural features. `<p>`, for paragraph, for example, is used to give white space, rather than necessarily indicating the beginning of a paragraph, and ``, for unordered (i.e. unnumbered) list, is frequently used to produce indentation rather than for lists.

The first stanza of *Þrymskviða*, marked up in HTML, might look like this:

```
<html>
  <body>
    <h2>Þrymskviða</h2>
    <h3>Anonymous</h3>
    <p>Reiðr var þá <i>Vingþórr</i><br>
```

```

        er hann vaknaði<br>
        ok síns hamars<br>
        um saknaði,<br>
        skegg nam at hrista,<br>
        skör nam at dýja,<br>
        réð <i>Jarðar burr</i><br>
        um at preifask.<br>
    </p>
</body>
</html>

```

Marked-up in XML, the poem might look like this (referring to a DTD schema):

```

<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<!DOCTYPE poem SYSTEM "poem.dtd" [
]>
<poem type="eddic" subtype="epic-dramatic">
  <title>Þrymskviða</title>
  <author>Anonymous</author>
  <stanza number="1" form="fornyrðislag">
    <line number="1">Reiður var þá
      <person reg="Þórr">Vingþórr</person></line>
    <line number="2">er hann vaknaði,</line>
    <line number="3">ok síns hamars</line>
    <line number="4">um saknaði,</line>
    <line number="5">skegg nam at hrista,</line>
    <line number="6">skör nam at dýja,</line>
    <line number="7">réð <person reg="Þórr">Jarðar burr</person></line>
    <line number="8">um at preifask.</line>
  </stanza>
</poem>

```

Note that in this example, ‘Vingþórr’ has been indentified as Þórr, and the same has the kenning ‘Jarðar burr’, literally ‘son of Earth (the name of Þórr's mother)’. This is one of several ways of facilitating references in the text.

Displayed in a browser, there would not necessarily be any difference between this and the same text marked-up in HTML, but the underlying information is far greater. From the point of view of a search engine, all that can be said about the HTML text is that it consists of one paragraph with eight line breaks, but there is no indication as to why this should be, i.e. there is nothing that says ‘this is a poem’, ‘this is a stanza’, ‘this is a line of verse’. The words ‘Vingþórr’ and ‘Jarðar burr’, in the same way, will be rendered in italics, but there is nothing which indicates that they are referring to a person, and, in fact, to the same person.

The first line of our XML document is the XML declaration, which tells any processing software that the document is in XML; it is not strictly speaking necessary to have an XML declaration, since any XML-aware software can work out for itself whether a document is in XML or not (the *.xml file extension also does this), but every XML document should ideally begin with one. The value of the **@version** attribute is always "1.0"; it is possible that there will be further versions. The other two attributes are optional: **@encoding**, which specifies which encoding is to be used (the variable length encoding of the Unicode character set, UTF-8, is assumed by all the standard browsers), and **@standalone**, the possible values for which are "yes" and "no", which indicates whether the document makes use of an external schema. XML documents do not in fact require a schema, provided they are ‘well formed’, i.e. do not contain any errors in syntax (elements which overlap or are opened but not closed etc.); in cases where there is no schema the value of the **@standalone** attribute should be "yes". If the attribute is omitted, the value "no" is assumed. The second line, following the XML declaration, is the reference to the schema being used, in this case a Document Type Declaration. This line gives the root element

on various aspects of the problem. These were integrated into a first public draft, TEI P1 (P for ‘Proposal’), published in June 1990. A second draft (TEI P2) followed in 1992 and 1993, and the first official version of the guidelines (TEI P3) was published in May 1994. The next version, TEI P4, was released in June 2002. A fully XML-compliant version of TEI P4 is available in electronic form at the [TEI Guidelines](#) web site; a print edition is also available from the University of Virginia Press. V. 1.0 of the Menota handbook is conformant with TEI P4. On 1 November 2007, TEI P5 was released in electronic form only at [TEI Guidelines](#). This present version of the Menota handbook is conformant with TEI P5.

The TEI began as a research effort cooperatively organised by three scholarly societies (the Association for Computers and the Humanities, the Association for Computational Linguistics, and the Association for Literary and Linguistic Computing), and funded solely by research grants from the US National Endowment for the Humanities, the European Union, the Canadian Social Science Research Council, the Mellon Foundation and others. In December 2000, after a year's negotiation, a new non-profit corporation called the TEI Consortium was set up to maintain and develop the TEI standard. Four universities serve as hosts for this consortium, presently two in the United States and two in Europe. The Consortium is managed by a Board of Directors, and its technical work is overseen by an elected Council.

There are hundreds if not thousands of projects currently using the TEI encoding scheme; Menota is one of them.

1.8 The TEI schemas

TEI offers several schemas for defining the structure of an XML file. In TEI P4 and earlier releases, the only schema was the Document Type Definition (DTD) discussed above. As of TEI P5, a RELAX NG schema has been added. We offer both schemas in [Appendix D](#) to this handbook, but now recommend a RELAX NG schema. The function of a RELAX NG schema is the same as that of a DTD, but it allows users to make a clear distinction between TEI elements and attributes and local elements and attributes by way of establishing a namespace. Consequently, the encoding becomes more transparent. Adding a namespace is explained in [ch. 1.9](#) below.

One of the great strengths of the TEI schemas - whether a DTD or a RELAX NG - is that they actually consists of a number of different tag sets which can be used in a variety of combinations, according to the needs of the encoder and nature of the material being encoded. In this way the encoder can tailor the schema to his or her individual needs, selecting from the very large number of elements available those which are most relevant to the material to be encoded.

All TEI conformant documents must contain two elements, a header, tagged **<teiHeader>**, in which, as was mentioned, meta-data, information about the electronic document, is provided, and the text itself, tagged **<text>**. What elements go into the **<text>** is to a great extent determined by which base and additional tag sets have been chosen.

The TEI tagset for verse has, not surprisingly, elements corresponding to line, stanza and so on in the DTD presented above. The two first stanzas of *Þrymskviða*, tagged in TEI conformant XML, might look like this:

```
<?xml version="1.0"?>
<!DOCTYPE text SYSTEM
  "http://www.menota.org/guidelines-2/schemes/menotaP5.dtd">
```

```

<text>
  <body>
    <lg type="lyric" met="fornyrðislag">
      <head>Þrymskvíða</head>
      <lg n="1" type="stanza">
        <l>Reiðr var þá <name key="Þórr">Vingþórr</name></l>
        <l>er hann vaknaði</l>
        <l>ok síns hamars</l>
        <l>um saknaði,</l>
        <l>skegg nam at hrista,</l>
        <l>skör nam at dýja,</l>
        <l>réð <name key="Þórr">Jarðar burrr</name></l>
        <l>um at þreifask.</l>
      </lg>
      <lg n="2" type="stanza">
        <l>Ok hann þat orða</l>
        <l>alls fyrst um kvað:</l>
        <l>Heyrðu nú, <name key="Loki">Loki</name>,</l>
        <l>hvat ek nú mæli,</l>
        <l>er eigi veit</l>
        <l>jarðar hvergi</l>
        <l>né upphimins:</l>
        <l>áss er stolinn hamri!</l>
      </lg>
    </lg>
  </body>
</text>

```

The element **<person>** and the corresponding attribute **@reg** is not defined in TEI, so this has been replaced by the element **<name>** and the attribute **@key** (this is one of several ways of encoding names in TEI conformant XML). The value of the attribute **@key**, in this case ‘Loki’, would typically refer to a list, which may be part of the XML document or it may be an external list, e.g. a dictionary. The **@key** can also be used even if there is no such list.

The element **<l>** is used for a line of verse (as opposed to line breaks in written or printed texts, which are dealt with in another way; see [ch. 4](#)), while **<lg>**, for ‘line group’, is used for a group of lines functioning as a formal unit – here both the poem as a whole and the individual stanzas – with a **@type** attribute to indicate what sort of unit. The advantage of **<lg>** over our **<stanza>** is that, being more abstract, it is also more flexible; **<lg>** elements can also nest, i.e. appear within each other, which allows quite sophisticated markup of complex verse forms.

In addition to these structural elements the TEI also makes available a host of elements for indicating features of typography and layout; although these were originally intended for use in the description of printed materials most if not all are equally applicable to manuscripts. There are also tags which can be used for normalisation, grammatical information etc. The other chapters in this handbook explain in detail how they can be used.

1.9 The namespace: adding elements and attributes

In this handbook, we are following the recommendations in the TEI Guidelines P5 as closely as possible. We have, however, added a few elements and attributes in order to enhance the encoding of Medieval Nordic manuscripts (and, we believe, other medieval manuscripts). In TEI P5, any additions of this type should be defined as a *namespace*, and we have consequently set up a namespace ‘me’ for our usage (‘me’ being short for ‘Menota’).

The namespace must be specified at the very beginning of the XML file:

```
<?xml version="1.0" encoding="UTF-8"?>
<TEI xmlns="http://www.tei-c.org/ns/1.0"
      xmlns:me="http://www.menota.org/ns/1.0">
  ...
</TEI>
```

In the Menota XML files, all additional elements and attributes will be preceded by ‘me:’. For example, we recommend that a normalised transcription is contained in a new element, *norm*. This appears as **<me:norm>**, identifying it as an element belonging to the Menota namespace. The advantage of doing this, is that all additional elements and attributes stand out clearly in the encoding; anyone who just glances through a Menota XML file will understand which elements and attributes belong to TEI P5 and which are the additions by Menota.

The following is a complete list of additional elements and attributes in *The Menota handbook*:

1.9.1 Elements

<me:norm> for readings on a normalised level, cf. [ch. 3.2](#).

<me:dipl> for readings on a diplomatic level, cf. [ch. 3.2](#).

<me:facs> for readings on a facsimile level, cf. [ch. 3.2](#).

<me:pal> for readings on a paleographical level, cf. [ch. 3.4 \(end\)](#).

<me:expunged> for readings that are deleted by the editor (as opposed to deletions by the scribe, which are encoded by the *del* element), cf. [ch. 7.4.2](#).

<me:punct> for punctuation characters, cf. [ch. 3.4](#).

<me:textSpan/> for encoding any discontinuous structures, thus avoiding a full set of elements like *addSpan*, *delSpan*, *suppliedSpan*, *expungedSpan*, etc. Note that the attribute *category* is used to specify what type of textspan it is, e.g. addition, deletion, supplement, expunction, etc., cf. [ch. 4.10](#).

<me:all> for alliteration in encoding of verse, cf. [ch. 9.2](#).

<me:ass> for internal rhyme in encoding of verse, cf. [ch. 9.2](#).

1.9.2 Attributes

@me:msa for morphosyntactical analysis, i.e. for specifying the grammatical form of a word. This is an attribute to the *w* element, cf. [ch. 8.3](#).

@me:type for classification purposes. This is an attribute to the *ex* and *am* elements, cf. [ch. 6.1](#).

@me:level for identifying the level on which the text has been transcribed, i.e. facsimile, diplomatic or normalised (see above). This is an attribute to the *normalization* element used in the header, cf. [ch. 10.3](#).

@me:lemmatized for identifying those texts which have been lemmatised. This is an attribute to the *interpretation* element used in the header, cf. [ch. 10.3](#).

@me:morphAnalyzed for identifying those texts which have been morphologically analysed, i.e. given grammatical form. This is an attribute to the *interpretation* element used in the header, cf. [ch. 10.3](#).

@category for identifying type of text span. This is an attribute to the *me:textSpan* element used to encode overlapping structures, cf. [ch. 4.10](#).

@spanTo for identifying the end point of a text span. This is another attribute to the *me:textSpan* element used to encode overlapping structures, cf. [ch. 4.10](#).

1.10 Displaying the text

We have mentioned several times the possibility of displaying XML documents in standard web browsers. In order to do so, one final piece is necessary: a stylesheet. As has been said, XML elements describe, ideally at least, the semantic structure of the text, rather than its appearance (although there is obviously a degree of overlap). Web browsers have built-in stylesheets for displaying HTML and know that in an HTML document anything tagged `<i>` is to be displayed in italic, because HTML markup is essentially presentational: `<i>` means ‘display in italic’. XML markup is semantic (and the elements user-defined), and in order for a browser to display an XML document, it needs to know what formatting to apply to what elements. It needs to be told, for example, that things within `<title>` tags should be displayed in italic. A stylesheet does precisely that.

There are essentially two options, Cascading stylesheets (CSS) and Extensible style language transformations (XSLT). CSS is a simple, non-XML syntax used to describe the appearance of any element in a document. XSLT, on the other hand, is itself an XML application which specifies rules by which the XML document is transformed into another document, either another XML document or something else; its most obvious use is to take XML and turn it into something more browser-friendly, i.e. HTML (or XHTML). The original document retains its complexity, but for viewing purposes it is changed into something even older browsers can deal with. This transformation can be done either at the browser-end, by the webserver, when the XML document is called up by the user, or by the creator of the document, who may not wish to make it available in its original state.

The stylesheet to be associated with the document is indicated by a `xml-stylesheet` processing instruction (or stylesheet link), which comes after the XML declaration, either before or after the Document Type Declaration, if there is one, but before the root element.

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<!DOCTYPE poem SYSTEM "poem.dtd">
<?xml-stylesheet href="poem.css" type="text/css"?>
```

The value of **@href** is the URL (absolute or relative) where the stylesheet can be found, while the value of **@type** will either be `"text/css"` for cascading stylesheets or `"text/xml"` for XSL transformations. There are other (pseudo-) attributes, such as **@media**, but they need not concern us here. The style sheet referred to here is a CSS style sheet, which indicates how each of the elements is to be displayed:

```
body {
  font-family: "Book Antiqua";
}

poem {
  display: block;
  font-family: "Book Antiqua";
  margin: 25pt 15pt 15pt 45pt;
  font-size: 13pt;
  line-height: 15pt}

title {
  display: block;
  font-size: 18pt;
```



```
padding: 5pt}

author {
display:none;}

stanza {
display: block;
padding: 5pt}

line {display: block}

person {font-style: italic}
```

Displayed by an XML-aware browser, such as [Firefox](#) (Windows, Mac, Linux), [Opera](#) (Windows, Mac, Linux), [Safari](#) (Mac) or [Internet Explorer](#) (Windows), the two first stanzas of *Þrymskviða* will be displayed like this:



Note that browsers may display the same page slightly differently. If it does not look right in one browser, another browser may do the trick.

XSLT is more powerful than CSS. With CSS one can determine exactly how the content of an element is to be displayed, in terms of font, colour etc., or whether it is to be displayed at all (one might not, for example, wish to display some of the administrative information contained in the TEI header). CSS will also allow you to insert text before and/or after an element (using the **before** and **after** pseudo-element selectors). But that's about it. With XSLT, on the other hand, one can, for example, re-arrange the order of the elements or display the value of an element's attribute instead of its actual content (very useful, for example, if one wishes to produce normalised and unnormalised texts from a single marked-up file).

The above display of an Eddic stanza is the preferred one in many Nordic editions; each line occupies a line in the edition, whether it is a short line (as in fornyrðislag) or a full line (as in ljóðaháttur). In Continental editions such as the standard Neckel–Kuhn edition, a pair of short lines making up a long line is printed as a single line in the edition, though with a sizeable space between the two lines, thus:

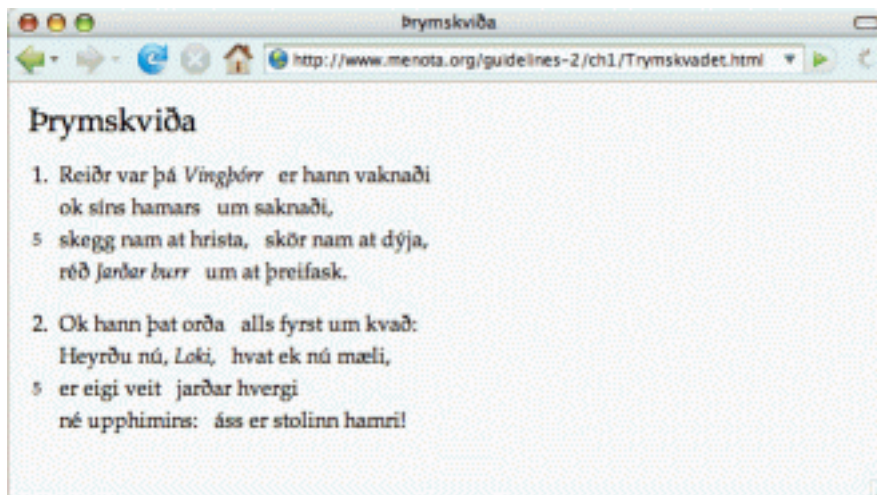
Reiðr var þá Vingþórr er hann vaknaði
ok síns hamars um saknaði

skegg nam at hrista, skör nam at dýja,
réð Jarðar burr um at þreifask.

For ease of reference, lines are numbered, but in stanzas of normal length only each fourth line (in ljóðaháttur) or each fifth line (in fornyrðislag) are numbered. In an eight-line display such as the one in the screenshot above, the fifth line of the first stanza is the one beginning with ‘skegg’. The same applies to the four-line display above, since each short line is counted, irrespective of whether it is displayed in conjunction with another short line or not. So to achieve a ‘Neckel–Kuhn display’ two operations are necessary, (a) every second short line in the encoded text is displayed on the same line as the previous short line, and with white space in between, and (b) lines are counted and a small number is positioned in the margin in front of every fifth line. This adds an element of transformation to the styling and is not easily done in CSS. In XSLT this is quite simple, even if the instructions may look difficult. An XSLT transforming the text as specified in (a) and (b) would look like this:

```
<xsl:template match="stanza">
<table class="stanza">
  <xsl:for-each select="child::line[ position() mod 2 = 1]">
    <tr>
      <xsl:choose>
        <xsl:when test="attribute::number mod 5 = 1">
          <!-- The first line -->
          <td>
            <xsl:value-of select="parent::stanza/attribute::number"/> .&#160;
          </td>
        </xsl:when>
        <xsl:when test="attribute::number mod 5 = 0">
          <!-- Line 5 -->
          <td>
            <xsl:attribute name="class">small</xsl:attribute>
            <xsl:value-of select="attribute::number"/>
          </td>
        </xsl:when>
        <xsl:otherwise>
          <td></td>
        </xsl:otherwise>
      </xsl:choose>
      <td><xsl:apply-templates/>&#160;&#160;
        <xsl:apply-templates select="following-sibling::line[1]"/>
      </td>
    </tr>
  </xsl:for-each>
</table>
</xsl:template>
```

Displayed in an XML-aware browser, the stanzas now look like this:



The display is different, but the XML encoding is not changed at all. It is only a matter of transforming the encoded text using XSLT and adding the required style with CSS. An XML document can also be transformed into a non-XML format, for example, a PDF, RTF or PostScript file. And the same XML file can be transformed again and again into dozens of different formats, without any effect on the content itself.

Chapter 2. Basic units: characters and words

2.1 Introduction

When transcribing a text, the transcriber will usually make a distinction between the individual characters, the white space between some of the characters, the words made up by sequences of characters, and the punctuation marks which are inserted between some of the words. The actual encoding can be as straightforward as the example in [ch. 1](#) above, in which characters, punctuation marks and spaces have been typed directly from the keyboard:

```
Reiðr var þá Vínþórr  
er hann vaknaði  
ok síns hamars  
um saknaði,  
skegg nam at hrista,  
skör nam at dýja,  
réð Jarðar burr  
um at þreifask.
```

In a more complex encoding, the transcriber might like to identify the basic units as such, so that a distinction easily can be drawn between single characters, words, punctuation marks and the white space surrounding them. This chapter will discuss these basic units and how they can be encoded specifically, if needed, using elements like `<c>` for individual characters and `<w>` for individual words.

2.2 Characters

The basic unit in any transcription of an alphabetic script is the individual letters. In a linguistic context a distinction is often drawn between the abstract entity of a **grapheme** and the representation of **graphs** in a written document. Variant forms are referred to as **allographs**, e.g. the Roman type of *s* and the Fraktur (black letter) type. The terminology is analogous to the distinction between *phonemes*, *phones* and *allophones*. For a general introduction to this terminology, see [Sture Allén 1971](#) or, more recently, [Manfred Kohrt 1985](#).

In this handbook we shall adopt the terminology of the [Unicode Standard](#). The fundamental distinction drawn is between **characters** and **glyphs**. Characters are, as Unicode defines it, ‘the smallest components of written language that have semantic value’, while glyphs are ‘the shapes that characters can have when they are rendered or displayed’ (cf. Unicode 4.0, ch. 2.2). What the transcriber sees in the source document is a series of individual glyphs, and the act of transcribing essentially involves connecting these glyphs to the characters at the transcriber's disposal.

The concept of a character is similar to, but not identical with the linguistic concept of a grapheme. These concepts are notoriously difficult, but for the purposes of this handbook we believe that the Unicode usage is robust and sufficiently well-defined.

The Unicode Standard puts great emphasis on the fact that individual characters may be represented by a number of glyphs, and is therefore reticent to accept as new characters what it perceives to be variant glyphs. It will be obvious to most people that the various

shapes of letters in printed type faces, such as Baskerville, Palatino, Helvetica etc., should not be seen as different characters, as shown in fig. 2.1.



Fig. 2.1 Various shapes (glyphs) of the characters ‘A’ and ‘a’ in Courier, Times and Lucida typefaces

Unicode draws a distinction between small (minuscule) characters such as ‘a’ and large (majuscule) characters such as ‘A’, since there is a possible semantic value attached to each set of characters. Thus, ‘the white house’ can refer to any house which is white in colour, while ‘the White House’ refers (normally) to one specific building. It can be argued that the same applies to the distinction between Roman types, ‘a’, and italics, ‘*a*’. For example, while ‘Metope’ refers a poem by the Norwegian author Olaf Bull, *Metope* (according to a widespread bibliographical practice) refers to the book in which this poem is published (a book which, co-incidentally, bears the same name as one of the poems contained in the book). However, Unicode does not regard italics (or bold type) as individual characters. There are good reasons for this, but the example serves to illustrate the fact that the definition of a character is not always clear-cut.

Medieval Nordic manuscripts were written in the Latin alphabet from the very beginning. The basic inventory is thus the characters a-z / A-Z. They were supplemented with a number of new (or borrowed) characters, several ligatures and a variety of diacritical marks. There was also a large number of abbreviation marks in use, especially in Old Icelandic and Old Norwegian manuscripts. We shall go through the inventory of ordinary characters, i.e. those based on the set a-z / A-Z, in [ch. 5](#) and abbreviation marks in [ch. 6](#), and we shall refer to both types as *characters*. In fact, some abbreviation marks behave as ordinary characters in the sense that they occupy a separate position on the base line. On the other hand, many components of ordinary characters are diacritical, i.e. placed above (or through or below) another character, and thus akin to typical abbreviation marks. This means that the rules for transcribing ordinary characters and abbreviation marks should be identical.

We believe that it is possible to identify a base line in all texts, as shown in fig. 2.2. We recommend that the transcriber identifies each separate character on the base line and record this in the same sequence as in the manuscript. Thus, the characters in fig. 2.2 would be transcribed as ‘abpp’ or ‘abpþ’. Note that the last character may be encoded with its Unicode code point, ‘p’ at 00FE, or with an entity, ‘þ’. Both encodings are strictly equivalent. Entities are explained in [ch. 1.5](#) and discussed further in [ch. 5.2](#).



Fig. 2.2 Position of characters on the base line

If there are marks of any sort placed above, through or below any base line character, we recommend that these marks (if they are to be interpreted as characters) are transcribed immediately *after* the base line character. In general, we refer to these marks as diacriticals. As mentioned above, abbreviation marks are also frequently written above

(and in some cases through or below) a base-line character. Assuming that the sign above ‘h’ should be referred to with the entity ‘&er;’, the transcription of the very first word in fig. 2.3 would be ‘h&er;’.

h̋ sér hān

Fig. 2.3 Diacritical marks and abbreviation marks

Diacritical marks are often seen as forming an integral part of a base line character and the whole encoded as a single character. This applies to accent marks, such as the one above ‘e’ in fig. 2.3. This combination of a base line character and a combining mark can be encoded as a single character, in Unicode referred to as LATIN SMALL LETTER E WITH ACUTE and the hexadecimal code value 00E9. As we shall see below, it is possible to decompose this letter in Unicode and refer to it as a combination of LATIN SMALL LETTER E and COMBINING ACUTE ACCENT. We would like to emphasize that both encodings are strictly equivalent.

Abbreviation marks, on the other hand, are usually treated as separate characters and encoded as characters in their own right. From a purely graphical point of view, the distinction between the acute accent in ‘é’ and abbreviation marks such as the ‘zigzag’ mark and the bar, both exemplified in fig. 2.3, is far from obvious, but the semantics are different. The acute accent may in some manuscripts be used to signify length, but it is often used quite freely, sometimes only to distinguish one minim character from another. Abbreviation marks have a definite (if sometimes ambiguous) meaning and can be expanded into one or more characters; the zigzag mark above ‘h’ in fig. 2.3 signifies ‘er’, and the bar above ‘n’ signifies another ‘n’.

2.2.1 Rules for encoding characters

We suggest the following basic rules for encoding characters, irrespective of whether they are ordinary (alphabetic) characters or abbreviation marks.

1. Each character is encoded according to its position in the direction of writing.
2. Alphabetical characters on the base line are encoded first:
 - 2.1 If the character belongs to the ordinary Latin character set a-z / A-Z (commonly known as ISO 646 or Basic Latin) it is always encoded as such.
 - 2.2 Characters outside Basic Latin should either be encoded by Unicode codepoints or by entities, e.g. either as ‘abpp’ or as ‘abpþ’.
 - 2.3 Characters which are not part of the Unicode Standard must always be encoded by entities. See [ch. 5](#) for a fuller explanation.
3. Abbreviation marks occupying a separate position on the base line are encoded in the same manner as alphabetical characters. This applies to e.g. LATIN SMALL LETTER P WITH STROKE THROUGH DESCENDER (for ‘per’ or ‘par’), as explained in [ch. 6](#) below.
4. Alphabetical characters with diacritical marks, e.g. ‘é’, are encoded in one of two equivalent ways:
 - 4.1 As a base line character + one or more combining marks. Thus the character ‘é’ would be encoded as ‘e’ + ‘&combacute;’ (the latter entity meaning COMBINING ACUTE ACCENT).

4.2 As a composite base line character and encoded with a single Unicode code point or an entity. Thus, the character ‘é’ would be encoded as either ‘é’ or as ‘é’.

5. Characters with abbreviation marks are encoded in the same manner as alphabetical characters, i.e. in one of two equivalent ways:

5.1 As a base line character + one or more combining marks. Thus the first character in fig. 3.2 above would be encoded as ‘h’ + ‘&er;’ (the latter entity meaning COMBINING ABBREVIATION MARK ‘ER’).

5.2 As a composite base line character and encoded with a single entity. Thus the above character might be encoded with a single entity, e.g. as ‘&her;’.

As a rule, we would recommend the first solution, since the number of combinations of base line characters and combining abbreviation marks is very high. Cf. the discussion in [ch. 6.4](#).

6. If there is more than one combining character, they are encoded in this order:

- (a) Combinations with the base line character within the x height of the base line character.
- (b) Combinations with the base line character outside its x height, but still in contact with it.
- (c) Combinations with the base line character outside its x height and without any contact with it.

7. If there is more than one combining character in any of the three positions defined in (6) above, they are encoded in a clockwise direction, beginning at 6 o'clock and moving through 9 o'clock, 12 o'clock etc.

2.2.2 Entities and Unicode values

By using entities it is possible to define as many characters as one believes are necessary for the transcription of a certain corpus of texts. However, since most applications now fully support Unicode, we recommend that characters in the Unicode Standard are encoded by their Unicode code points.

Note that the type of encoding is specified at the very beginning of an XML file. If the specification is

```
<?xml version="1.0" encoding="ISO-8859-1"?>
```

entities must be used for all characters outside Basic Latin and Latin-1 Supplement. Thus, ‘a’, ‘é’ and ‘þ’ can be entered directly, but characters like ‘#’ (LATIN SMALL LETTER O WITH OGONEK) must be encoded with an entity, ‘&oogon;’.

If, however, the encoding is specified as

```
<?xml version="1.0" encoding="UTF-8"?>
```

all characters in the Unicode Standard can be encoded with their Unicode code points, without resorting to entities.

In TEI P5, all entities must be declared in a separate list. A complete list of entities for Medieval Nordic texts is part of the Menota schema, and can be consulted in [Appendix D.1](#). An encoding using these entities will always be valid with respect to character encoding (but may, of course, be invalid for other reasons). In the Menota schema, entities are linked to code points defined in the [MUFI character recommendation](#), so that if a Menota text is displayed with a fully compliant [MUFI font](#), all entities will be displayed correctly.


If an encoder, for some reason, would like to encode a character which is not in the Menota list of entities, this character has to be declared in the header of the file, or by exchanging the Menota list of entities with an extended list.

The Basic Multilingual Plane of the Unicode Standard has 65,536 different code points. This includes a large Private Use Area (PUA), comprising some 6,000 code points. This area can be used for characters not defined in the Standard (so far). Our present recommendation is to use this area for characters not included in the Unicode Standard and to coordinate the allocation of codepoints with the recommendations by the [Medieval Unicode Font Initiative](#). It should be noted that the use of PUA is an interim solution. A long-term solution is obviously to apply to Unicode for the inclusion of additional characters and/or use other rendering techniques (such as OpenType).



Code points in Unicode are usually given in hexadecimal format, in which each digit spans a sequence of 16 positions, 0-1-2-3-4-5-6-7-8-9-A-B-C-D-E-F. Thus, 0001 equals 1 in the decimal system, 000F equals 16, 0010 equals 17 etc. The whole range thus goes from 0000 to FFFF (65,536). The PUA is located at E000-F8FF.

The Latin alphabet is the first to be described in the Unicode Standard. As was mentioned, many characters in Unicode can be defined in several ways, either as a single, composite character or as combination of a base line character and one or more combining marks.

(a) Commonly used characters have a single description in Unicode. This applies to all base line characters in the Latin alphabet.

Glyph	Encoding	Code point	Unicode descriptive name
	a	0061	LATIN SMALL LETTER A

(b) Composite characters may be described in more than one way. Thus ‘a with acute accent’ can be encoded as a combination of ‘a’ and a combining acute accent or as a single character, ‘a with acute accent’. Both descriptions are equivalent:

Glyph	Entity	Code point	Unicode descriptive name
	a + &combacute;	0061 + 0301	LATIN SMALL LETTER A + COMBINING ACUTE ACCENT
	á	00E1	LATIN SMALL LETTER A WITH ACUTE

(c) Some characters are not found in Unicode and must therefore be allocated to the Private Use Area (PUA), either as a character with its own code point or as a combination of an existing character and a combining diacritical mark in the PUA. The ligature ‘av’ is not included in the Unicode Standard (as of v. 5.0), and since we would rather not encode it as a sequence of ‘a’ + ‘zero width joiner’ + ‘v’, we have allocated it to a code point in the PUA, EF97.

Glyph	Entity	Code point	Descriptive name
ǰ	&avlig;	EF97	LATIN SMALL LIGATURE AV

Encoding with entities referring to the PUA may look unnecessarily complicated. It should be borne in mind, however, that the great majority of characters *are* defined in Unicode, and in many transcriptions the need for special characters in the PUA will not arise. With appropriate fonts, the transcriber does not need to spend much time on technicalities of this kind.

Finally, it should be noted that a text may be encoded with a mixture of Unicode code points and entities even for characters within the Unicode Standard. For the sake of clarity, some encoders might like to insert combining marks as entities. Thus, the example above might be encoded as:

```
h&er; sér han&bar;
```

even if both COMBINING ZIGZAG ABOVE and COMBINING OVERLINE are part of the Unicode Standard, at 035B and 0305 respectively. Some XML editors may not show combining characters in correct positions, so that it may be more legible to use entities for these characters, ‘&er;’ for the combining zigzag above and ‘&bar;’ for the combining bar above.

2.2.3 Encoding characters as such

In some cases, a character should be encoded as a character and not as a part of a word, e.g. in a grammatical discussion. The TEI P5 Guidelines recommend the element <c> for this type of encoding.

Element	Contents
<c>	(character) contains an individual character

A sentence like the following, from Einar Haugen's edition of the *First Grammatical Treatise*,

```
X, hann er samsettr i latinu af c ok s.
```

can be encoded as

```
<c>X</c>, hann er samsettr i latinu af <c>c</c> og <c>s</c>.
```

When displaying this text, the contents of the <c> element can be put in italics:

```
X, hann er samsettr í látinu af c ok s.
```

The <c> element should be restricted to contexts in which characters are cited as characters.

The encoding of initials and *littera notabilior* is discussed in [ch. 4.8](#) below.

2.3 Words

2.3.1 Basic mark-up

This chapter will introduce some important elements and attributes for the encoding of word or word parts, mostly based on [ch. 17.1 ‘Linguistic Segment Categories’](#) in the TEI P5 Guidelines.

Element / attribute	Contents
<w>	(word) contains an individual word
@lemma	states the lexical citation form of a word
<m>	(morpheme) contains a part of a word
@baseForm	states the base form of a morpheme
<seg>	(segment) groups one or more strings of text, e.g. words
@type	states the type of segmentation. Suggested values:
'nb'	no break
'enc'	enclitic

As a rule, medieval Nordic manuscripts in the Latin alphabet are written with a clearly identifiable space between each word. This obviously facilitates the work for the transcriber, since the word is a basic linguistic unit in grammars and dictionaries. In a simple transcription, word division can simply be entered by the space bar on the keyboard. Thus, a piece of text (from *Barlaams ok Josaphats saga* ch. 48) might be transcribed as

En ef ver fallum i hinar fornno syndir oc huerfum aptr til hinna fyrrv misverka sem
hundr til spyu sinnar þa kann lettlega at vera at oss kunni til hannda at berazt sem i
guðspialleno segir.

Here, each word is delimited by a space (or a punctuation mark). However, for a more detailed analysis it can be convenient to identify each word with a separate <w> element (for ‘word’). The <w> element functions as a container for information on levels of text representation (cf. [ch. 3](#) below) and morphological analysis (cf. [ch. 8](#)). In this example, each word has been identified by the <w> element, and the lemma (dictionary entry) specified as an attribute to the <w> element:

```
<w lemma="en">En</w>
<w lemma="ef">ef</w>
<w lemma="v&eacute;r">ver</w>
<w lemma="falla">fallum</w>
<w lemma="i">i</w>
<w lemma="hinn">hinar</w>
<w lemma="forn">fornno</w>
<w lemma="synd">syndir</w>
etc.
```

For practical reasons, each word has a separate line in this encoding. Unless otherwise specified, it is assumed that there is white space between each <w> element.

Ch. 3 will discuss further levels of transcription (facsimile and normalised), and ch. 8 how words can be marked for morphological categories.

2.3.2 Deviations in word division (words written together or apart)

Although words as a rule are separated by spaces in medieval Nordic manuscripts, there are many exceptions to this rule. For this reason, a distinction should be drawn between **graphical** words and **lexical** words. A graphical word is a sequence set out by space on either side, while a lexical word is a member of the set of word forms defined by grammars and dictionaries for the language in question. In the great majority of cases, graphical and lexical words are identical. However, we sometimes see that a preposition and its object may be written as a single word ('aveiðiskap' = 'á veiðiskap'), or that compounds are written as two separate words ('veiði kona' = 'veiðikona').

veiði kona mykyl hevir hon veret
ok miok agiarñ aveiðiskap

Fig. 2.4 Text adopted from *Barlaams saga ok Josaphats*, Holm perg. fol. nr. 6, f. 138

If the transcriber wishes to analyse two (or more) graphical words as a single lexical word, we suggest that this is done by putting the whole sequence within the <w> element:

```
<w>vei&eth;i kona</w>
```

Information on e.g. lemma can be given as an attribute to the <w> element:

```
<w lemma="vei&eth;ikona">vei&eth;i kona</w>
```

The sequence 'veiði kona' thus appears within a single element. In other words, the transcriber interprets it as one lexical word, 'veiðikona'. The space is left untouched, so that in a display of the transcription, the sequence will still show up as two graphical words, 'veiði' and 'kona'. However, since both graphical words are placed within a single element the *lemma* will refer to both parts.

The converse case is a single graphical word which the transcriber would like to analyse as two (or more) lexical words, e.g. 'aveiðiskap' = 'á veiðiskap'. Each lexical word should be placed within a <w> element, and information on lemma, morphological form etc. can be given within each <w> element. However, to generate a correct display of the text, i.e. a display with no space between each part, we suggest that the <seg> element is used with a type attribute. The value 'nb' would indicate that there is no break between the parts in the <w> element. If the lemma is given by way of an attribute, the encoding would look like this:

```
<seg type="nb">  
  <w lemma="&aacute;">a</w>  
  <w lemma="vei&eth;iskap">vei&eth;iskap</w>  
</seg>
```

In some rather marginal cases, a sequence may be encoded as both types. A simplified example from Codex Regius is 'aravk stola' which should be read as 'a ravkstola'. This sequence might be encoded in this way:

```
<seg type="nb">  
  <w lemma="&aacute;">a</w>  
  <w lemma="r&oogon;kst&oacute;ll">ravk stola</w>  
</seg>
```

This encoding shows that 'a' in 'aravk stola' is a lexical word, sc. the preposition 'á', and that 'ravk stola' is another lexical word, sc. the noun 'rökstóll' (for practical reasons, 'ö')

is used here rather than ‘o ogonek’). It will also allow a correct display of the sequence, since it specifies that there should be no space between ‘a’ and ‘rauk stola’, and the space between ‘rauk’ and ‘stola’ is also encoded (analogous to the encoding of ‘veiði kona’ above).

Enclitic words may be encoded in a similar way, e.g. ‘emk’ which should be read as ‘em’ + ‘(e)k’, ‘am I’:

```
<seg type="enc">
  <w lemma="vera">em</w>
  <w lemma="ek">k</w>
</seg>
```

2.3.3 Encoding of word constituents

The encoder might want to encode constituent parts of a word, e.g. prefixes, roots, derivational forms etc. We recommend using the **<m>** element (for ‘morpheme’) in such cases (cf. [ch. 17.1](#) in the TEI P5 Guidelines). This element may also be used for constituent parts such as ‘veiði’ and ‘kona’ in the examples above. The **<m>** element may contain information on level of text representation, lemma etc. We shall repeat the encoding of ‘veiði kona’ above:

```
<w lemma="vei&eth;ikona">vei&eth;i kona</w>
```

Now, if the encoder wishes to add lexicographical (or other) information to the two constituent parts, that can easily be done by inserting **<m>** elements in the **<w>** element:

```
<w lemma="vei&eth;ikona">vei&eth;i kona
  <m baseForm="vei&eth;i">vei&eth;i</m>
  <m baseForm="kona">kona</m>
</w>
```

This encoding would make a clear distinction between lemmata on the first level of encoding, in this case ‘veiðikona’, and the base form, **@baseForm**, of each constituent part, in this case ‘veiði’ and ‘kona’.

Lemmatisation is further discussed in [ch. 8](#) below and is here only given as an example of a word-based type of mark-up. Grammatical information can also be conveniently attached to the word through the **@msa** (morphosyntactical analysis) attribute. This is also discussed in [ch. 8](#).

2.4 Punctuation and white space

Having introduced elements for the encoding of individual characters and words, it can also be useful to tag punctuation marks specifically. The TEI P5 Guidelines do not have any punctuation element, so this has been added in the Menota namespace, **<me:punct>**. Note the prefix ‘me:’ which indicates that the element belongs to the Menota namespace and is not part of the elements defined in TEI P5. See [ch. 1.9](#) above on the use of namespaces in TEI schemes. Remember that namespaces are allowed with RELAX NG schemas, but not with a DTD (as in TEI P4). In the latter case, the prefix ‘me:’ should simply be dropped.

Element / attribute	Contents
<me:punct>	contains a punctuation mark

Element / attribute	Contents
<code><me:fac></code>	contains a reading on a facsimile level
<code><me:dipl></code>	contains a reading on a diplomatic level
<code><me:norm></code>	contains a reading on a normalised level
<code><num></code>	contains a number, including any delimiters

The three levels of text representation, *fac*, *dipl* and *norm*, will be explained in [ch. 3](#) below. Suffice it here to say that at the facsimile level, the manuscript is recorded in great detail, on the diplomatic level, it is somewhat normalised, and on the normalised level it is fully regularised according to standard grammars and dictionaries.

2.4.1 Punctuation

In [ch. 2.3.1](#) above, we said that a text can be encoded character by character. Punctuation marks are simply inserted where they occur in the manuscript, even if the position is wrong according to modern rules. If the actual punctuation in *Barlaams ok Jospahats saga* is added, the example above looks like this:

En ef ver fallum i hinar fornno syndir. oc huerfum aptr. til hinna fyrrv misverka sem
hundr til spyu sinnar. þa kann lettlega at vera. at oss kunni til hannda at berazt. sem i
guðspialleno segir.

In addition to punctuation marks like FULL STOP, COMMA, COLON, SEMICOLON and HYPHEN, there are a number of specific medieval punctuation marks, including an early form of the QUESTION MARK and a PUNCTUS ELEVATUS. A full list of additional punctuation marks can be found in the [MUFI character recommendation](#) with appropriate character entities. For example, the PUNCTUS ELEVATUS, which sometimes appear in Medieval Nordic texts, should be encoded with the entity ‘&punctelev;’.

If a text is encoded using the `<w>` element, we recommend using a `<me:punct>` element for punctuation marks:

```
<w>En</w>
<w>ef</w>
<w>ver</w>
<w>fallum</w>
<w>i</w>
<w>hinar</w>
<w>fornno</w>
<w>syndir</w>
<me:punct>.</me:punct>
<w>oc</w>
<w>huerfum</w>
<w>aptr</w>
<me:punct>.</me:punct>
etc.
```

The main reason for doing so will become clear in [ch. 3](#), in which several levels of transcription is discussed. At a diplomatic level, the transcriber should encode the punctuation marks exactly where they are in the source, but at a normalised level, some punctuation marks should be suppressed, some should be retained and some should be added. For a full discussion, please see [ch. 4.8](#).

2.4.2 White space

In a single-level transcription, spaces are simply inserted by the space bar. Note that in XML as well as in HTML any amount of white space (spaces, tabs and line breaks) are interpreted as a single space. It is not possible to encode a long space in the manuscript simply by hitting the space bar several times. Any distinctions in space length must be encoded specifically. In our experience, there is no significant variation in word spacing in Medieval Nordic manuscripts. If, however, a transcriber believes there are more than one length of the space, the simplest way of encoding this is probably to define the standard space, code point 0020, as the default space and to define deviating spaces with reference to the list of various space lengths in the Unicode chart *General Punctuation*, 2000-200B. For recommended entities, see the [MUFI character recommendation](#).

As for the interpretation and display of spaces in a multi-level transcription, we suggest the following three rules:

1. A transcription using the `<w>` and the `<me:punct>` element should be displayed with a space immediately after each element.

The example in [ch. 2.4.1](#) above would then be interpreted (e.g. by an XSLT style sheet) as

En ef ver fallum i hinar fornno syndir . oc huerfum aprtr .

This is correct in so far as there should be a space *after* each punctuation mark, but wrong in so far as there should not be a space *before* the punctuation mark. The following additions to the general rule must be made with respect to the `<me:punct>` element:

2. When displaying the text, there should not be any white space before a `<me:punct>` element.

The example above will then be correctly displayed as

En ef ver fallum i hinar fornno syndir. oc huerfum aprtr.

That is also true for any sequence of punctuation characters, e.g.

Hann segir, " Ek veit eigi."

In this example, no space is displayed before the comma nor before the final sequence of a full stop and a closing quotation mark. However, the position of the space in connection with the opening quotation mark is wrong. For this specific punctuation mark, the space should be *before*, not *after*:

Hann segir, "Ek veit eigi."

This will be taken care of by the XSLT style sheet, which treats opening quotation marks as an exception to rule (2).

Another exception are Roman numerals, which typically are delimited by a dot immediately before and after the number:

Hann er .xij. vetra gamall.

We recommend that the delimiters are encoded as part of the number, and thus contained in the `<num>` element:

```
<w>Hann</w>
<w>er</w>
<num>.xij.</num>
<w>vetra</w>
<w>gamall</w>
```

```
<me:punct>.</me:punct>
```

Similarly, if numbers are encoded as words, delimiters should be contained in the **<w>** element:

```
<w>Hann</w>
<w>er</w>
<w>.xij.</w>
<w>vetra</w>
<w>gamall</w>
<me:punct>.</me:punct>
```

However, if an ordinary punctuation mark is positioned immediately before a word rather than after the preceding word, we recommend that a **@rend** attribute is used with the value ‘rightlocation’. Thus,

Hann kemr .opt.

should be encoded as

```
<w>Hann</w>
<w>kemr</w>
<me:punct rend="rightlocation">.</me:punct>
<w>opt</w>
<me:punct>.</me:punct>
```

The XSLT style sheet will then be instructed to position the first punctuation mark accordingly, i.e. immediately in front of the following word.

Finally, the following addition to the general rule must be made with respect to the **<w>** element:

3. If two or more **<w>** elements are contained in a **<seg>** element (type="nb"), in the display on the **<facs>** and **<dipl>** levels there should not be any space after the **<w>** elements except for the last **<w>** element contained in the **<seg>** element.

Thus, the following sequence

```
<seg type="nb">
  <w>
    <me:facs>a</me:facs>
    <me:dipl>a</me:dipl>
    <me:norm>&aacute;</me:norm>
  </w>
  <w>
    <me:facs>lande</me:facs>
    <me:dipl>lande</me:dipl>
    <me:norm>landi</me:norm>
  </w>
</seg>
```

should be displayed as ‘alande’ on the **<me:facs>** and the **<me:dipl>** level, with no word division, but as ‘á landi’ on the **<me:norm>** level, with word division. In the latter case, rule (1) applies, which states that a space should be displayed after each **<w>** element. In the former case, rule (3) entails that there should not be displayed any space after the first of the two words in the **<seg>** element. Also see [ch. 2.3.2](#) above.

If the above-mentioned rules 1-3 are added to the XSLT style sheet, texts should be displayed correctly. See [Appendix F.2](#) for an example of how this can be implemented.

Chapter 3. Levels of text representation

3.1 Introduction

A transcription is basically a representation of a primary source in another format, such as paper or the electronic medium. Some transcriptions aim to reproduce the source text as closely as possible, others allow for a certain amount of generalisation. In transcriptions of speech, a distinction is usually drawn between narrow and broad transcriptions, depending on the amount of phonetic detail. The same perspective applies to transcriptions of manuscript texts. Close (or narrow) transcriptions are usually referred to as diplomatic, while regularised transcriptions are often referred to as normalised. This is the basic distinction drawn in e.g. [Wittgenstein's Nachlass: The Bergen Electronic Edition](#) (1998-2000). Here, all texts are available in two versions, a diplomatic transcription and a normalised one. For examples, please refer to [this page](#).

We suggest that medieval Nordic texts may be transcribed on up to three levels. In addition to the **normalised** level, we identify two closer levels. We shall refer to the narrowest level as the **facsimile** level, while the 'medium' level is designated as **diplomatic**. The three levels are exemplified in [ch. 3.2](#) below.

The distinction between three levels of text representation does not mean that a Menota transcription should contain all three levels. Many transcribers will probably choose a single level for their transcription. Our recommendation is to use these levels as a guide, so that a transcription can be described as following one of these levels. This information should be given in the header, and can optionally be given by use of specific elements in the transcription itself, as discussed in [ch. 3.2](#) below. If a transcriber wishes to deviate from any of these levels, and there may be good reasons to do so, we recommend that the deviations are specified in the header.

It is convenient to begin by looking at a Latin text example, *Passio et Miracula Beati Olavi*. An important source for this work is Corpus Christi College, Oxford MS 209, a vellum manuscript from the late 12th century. Below is a low resolution facsimile from the beginning of the *Passio*. For a facsimile of the whole manuscript in high resolution, please refer to [Early Manuscripts at Oxford University](#).

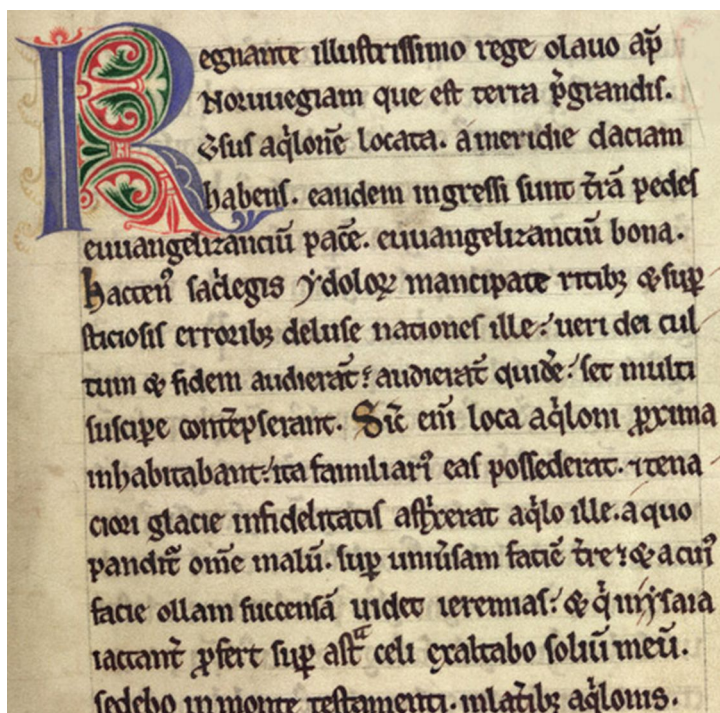


Fig. 3.1 CCC 209, fol. 57r., l. 1-15. © Corpus Christi College, Oxford

The following elements and attributes will be used in the encoding:

Element / attribute	Contents
<head>	contains a heading (or title)
<div>	contains a section of the text; can be nested hierarchically
@type	specifies the type of section
@n	specifies the number of a section
<p>	contains a paragraph of text
<hi>	contains a highlighted part of the text, e.g. by way of bold, italics, underlining
@rend	specifies how the highlighted text is rendered, e.g. by way of italics

Using these elements and attributes to describe the structure of the text, the manuscript can be transcribed straight away:

```
<head>Passio et miracula beati Olavi</head>

<div type="section" n="1">
  <p><hi rend="blue">R</hi>egnante illustrissimo rege Olauo
    apud Noruuegiam, que est terra pregrandis uersus aquilonem
    locata, a meridie Daciam habens, eandem ingressi sunt terram
    pedes euuangelizantium pacem, euuangelizantium bona.</p>
</div>

<div type="section" n="2">
  <p>Hactenus sacrilegis ydolorum mancipate ritibus et
    supersticiosi erroribus deluse nationes ille ueri Dei cultum
    et fidem audierant; audierant quidem, set multi suscipere
    contempserant.</p>
</div>
```



```

</div>

<div type="section" n="3">
  <p>Sicut enim loca aquiloni proxima inhabitabant, ita
    familiarius eas possederat et tenaciori glacie
    infidelitatis astrinxerat aquilo ille, a quo panditur omne
    malum super uniuersam faciem terre, et a cuius facie ollam
    succensam uidet Ieremias, et qui in Ysaia iactanter profert:</p>
</div>

<div type="section" n="4">
  <p><quote>Super astra celi exaltabo solium meum, sedebo in
    monte testamenti in lateribus Aquilonis.</quote></p>
</div>

```

(Adapted from an edition by Lars Boje Mortensen, University of Bergen. Cf. also the edition by [Frederick Metcalfe 1881](#).)

(When the illustrious King Óláfr ruled in Norway, a vast country located towards the north and having Denmark to the south, there entered into that land the feet of them that preach the gospel of peace and bring glad tidings of good things. The peoples of that country, previously subject to the ungodly rites of idolatry and deluded by superstitious error, now heard of the worship and faith of the true God – heard indeed, but many scorned to accept. Living in a region close to the north, it was the same north, from which comes every evil over the whole face of earth, that had possessed them all the more inwardly and gripped them all the more firmly in the ice of unbelief. From its face Jeremiah saw a seething pot; and in Isaiah there is the boaster who says, "I will exalt my throne above the stars of God: I will sit also upon the mount of the congregation, in the sides of the north.") [Translated by [Devra Kunin 2001](#).]

The transcription above is easily readable, even in its ‘raw’ XML format. In fact, if it was stripped for all elements, it would look like a plain ASCII text from any word processor:

Passio et miracula beati Olavi Regnante illustrissimo rege Olauo apud Noruuegiam, que est terra pregrandis uersus aquilonem locata, a meridie Daciam habens, eandem ingressi sunt terram pedes euuangelizantium pacem, euuangelizantium bona. Hactenus sacrilegis ydolorum mancipate ritibus et supersticiosiis erroribus deluse nationes ille ueri Dei cultum et fidem audierant; audierant quidem, set multi suscipere contempserant. Sicut enim loca aquiloni proxima inhabitabant, ita familiarius eas possederat et tenaciori glacie infidelitatis astrinxerat aquilo ille, a quo panditur omne malum super uniuersam faciem terre, et a cuius facie ollam succensam uidet Ieremias, et qui in Ysaia iactanter profert: Super astra celi exaltabo solium meum, sedebo in monte testamenti in lateribus Aquilonis.

With the help of an XML style sheet, the text could be displayed with a certain amount of formatting on the basis of the mark-up. For example, the title (**<head>**) might be shown in bold type, the initial might be rendered with an enlarged capital in blue colour, sections might be set out in separate paragraphs and numbered in bold type, and the Biblical quotation could be given in italics:

Passio et miracula beati Olavi

1. Regnante illustrissimo rege Olauo apud Noruuegiam, que est terra pregrandis uersus aquilonem locata, a meridie Daciam habens, eandem ingressi sunt terram pedes euuangelizantium pacem, euuangelizantium bona.

2. Hactenus sacrilegis ydolorum mancipate ritibus et supersticiosi erroribus deluse nationes ille ueri Dei cultum et fidem audierant; audierant quidem, set multi suscipere contempserant.
3. Sicut enim loca aquiloni proxima inhabitabant, ita familiarius eas possederat et tenaciori glacie infidelitatis astrinxerat aquilo ille, a quo panditur omne malum super uniuersam faciem terre, et a cuius facie ollam succensam uidet Ieremias, et qui in Ysaia iactanter profert:
4. *Super astra celi exaltabo solium meum, sedebo in monte testamenti in lateribus Aquilonis.*

Medieval Nordic texts need not contain any more mark-up than in this example, and they will still be fully valid XML. However, in order to comply with the Menota standard, it should follow the TEI guidelines. A few more elements and attributes will be needed for this:

Element / attribute	Contents
<code><TEI></code>	states that the contents of this element is a single TEI conformant document, comprising a header and a text
<code><teiHeader></code>	contains structured information on the text according to the recommendations by TEI
<code><text></code>	contains the text
<code><body></code>	contains the body of the text, excluding and front or back matter

The basic structure of the file is thus quite simple:

```
<?xml version="1.0" encoding="UTF-8"?>
<TEI xmlns="http://www.tei-c.org/ns/1.0"
      xmlns:me="http://www.menota.org/ns/1.0">
  <teiHeader>
    Here goes structured information on the text and the transcription.
  </teiHeader>
  <text>
    <body>
      Here goes the text as exemplified above.
    </body>
  </text>
</TEI>
```

For an example of a Menota header, please go to [Appendix E](#).

It is important to keep in mind that a transcription may be as straightforward and readable as this, and it would be fully acceptable as a Menota text.

However, not all primary sources are equally straightforward to transcribe. For most vernacular sources, entities will be required to deal with additional characters, and we might also like to transcribe the text at a more diplomatic level than in this example. For example, the last word on the very first line is transcribed as ‘apud’. In the facsimile above, we see that it has been written with the letters ‘ap’ and a superlinear abbreviation mark. Some transcribers might want to record the fact that there is an abbreviation mark at this point, or they might want to show how this abbreviation should be expanded. Two elements will be needed for this:

Element / attribute	Contents
<code><ex></code>	contains the text of an expanded abbreviation
<code><am></code>	contains an abbreviation marker

The characters that should be added, i.e. the expansion of the abbreviation, will be contained in the `<ex>` element:

```
ap<ex>ud</ex>
```

Other transcribers would like to encode the actual abbreviation mark being used, in this case a superlinear bar. This might be encoded with the help of an entity such as `&bar;`, meaning ‘a horizontal bar placed above the preceding character’. The element `<am>` (for ‘abbreviation marker’) indicates that the bar is an abbreviation mark, and not, for example, a sign for length (i.e. a macron) or a diacritical sign (like the bar sometimes used above ‘u’ to distinguish this character from ‘n’):

```
ap<am>&bar;i</am>
```

Yet other transcribers would like to encode the fact that the word has been abbreviated with a superlinear bar AND that this abbreviation should be expanded as ‘ud’ in this particular context. The superlinear bar is highly ambiguous; in this short extract alone, it should be expanded as ‘m’ in ‘*terram*’ (l. 4), ‘ut’ in ‘*Sicut*’ (l. 9), ‘ni’ in ‘*enim*’ (l. 9), ‘n’ in ‘*omne*’ (l. 12).

3.2 Levels of text representation

We believe that there are three focal levels of text representation for medieval Nordic texts and suggest that a transcription should reflect at least one of these levels. Furthermore, a transcription should be easily expandable so as to accommodate one or two additional levels. Since these levels have not been defined by TEI, they have been added in the Menota namespace, ‘me’:

Element / attribute	Contents
<code><me:fac></code>	contains a reading on a facsimile level
<code><me:dipl></code>	contains a reading on a diplomatic level
<code><me:norm></code>	contains a reading on a normalised level

Note: The ‘me’ prefix can only be used with a RELAX NG schema. It must be left out in texts which will be validated against a DTD.

This time, we shall use a short extract from an Old Icelandic manuscript, AM 233 a fol (first quarter of the 13th century) as an example:

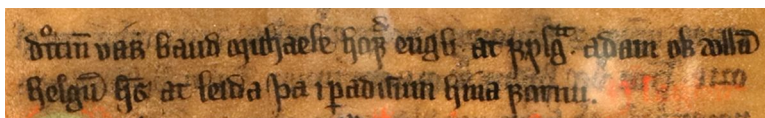


Fig. 3.2 AM 233a fol. 28v, l. 1-2. This is a fragment of *Niðrstigningar saga*, an Old Norse translation of the apocryphal *Evangelium Nicodemi*. A fuller example can be found in [Haugen and Pichler 2005](#), 220-22.

3.2.1 Facsimile level

On this level, the text is transcribed character by character, line by line. Allographic variation is to a great extent reflected in the transcription, and abbreviation marks are copied without any expansion. Thus, the text in fig. 3.2 would be transcribed as

```
&drot; &osup;ttin&bar; vá&rscapdot; bau&drot; michaele ho&fins;&dsup; engli.  
at &fins;ylg&ra; a&drot;am ok &aolig;llu&bar; helgu&bar; &hbar;s at lei&drot;a  
þa i &pbardes;a&drot;i&slong;um hína &fins;ornu.
```

and displayed (subject to an appropriate font) as

ðrōttinn vār baud michaele hof̊ engli. at fylg̊ adam ok ællū
helgū h̊s at leida þa i paradisum hína fornu.

Fig. 3.3 Facsimile rendering of the example text in fig. 3.2 using the font Andron.

At the facsimile level, the transcriber ought to encode the manuscript exactly as it reads, even if it contains obvious mistakes. Corrections can be made by inserting a note, or it can be left to the diplomatic or normalised level.

3.2.2 Diplomatic level

On this level, not all types of allographic variation are transcribed, and line divisions are usually not shown in the display of the transcription. In the transcription, expansions are set out by the element `<ex>` and in the display usually by italics. The text would then be transcribed as

```
d<ex>ro</ex>ttin<ex>n</ex> vá&rscapdot; baud michaele hof<ex>ud</ex> engli.  
at fylg<ex>ia</ex>. adam ok &aolig;llu<ex>m</ex> helgu<ex>m</ex> h<ex>an</ex>s  
at leida þa i p<ex>ar</ex>adi&slong;um hína fornu.
```

and displayed as (now disregarding the line break)

dróttinn vār baud michaele hofud engli. at fylgia. adam
ok ællum helgum hans at leida þa i paradisum hína fornu.

Fig. 3.4 Diplomatic rendering of the example text in fig. 3.2 (in Andron).

3.2.3 Normalised level

On this level, the orthography is regularised according to the norm found in grammars and dictionaries for the language in question. For Old Icelandic and Old Norwegian texts we recommend the normalisation rules in [AMKO's dictionary](#) (ONP). Abbreviations are expanded silently, and punctuation is regularised as well. Thus, the text in fig. 3.1 would be transcribed as

```
Dróttinn vār bauð Michaele h&oogon;fuðengli at fylgja Adam ok &oogon;llum  
helgum hans at leiða þá í paradísum hina fornu.
```

and displayed as

Dróttinn várr bauð Michaelē hofuðengli at fylgja Adam
ok öllum helgum hans at leiða þá í paradísu hina fornu.

Fig. 3.5 Normalised rendering of the example text in fig. 3.2 (in Andron).

Note that at this level all characters have been encoded using official Unicode code points. So rather than encoding the character ‘ð’ with the entity ‘ð’ it has been encoded simply as ‘ð’, using its code point in Latin-1 Supplement, 00F0. The only exception here is the ‘o ogonek’, which for practical purposes has been encoded with the entity ‘&oogon;’, even if this character, too, has a Unicode code point, 01EB in Latin Extended-B. A suitable keyboard layout is helpful for the actual typing of some of these characters, but in general, all Medieval Nordic texts can be encoded without resorting to entities as long as the text is rendered on a normalised level. Many Old Swedish and Old Danish texts can also be encoded with a minimal amount of character entities.

For a more detailed discussion of the three levels discussed here, please refer to [Haugen 2004](#).

For keyboard layouts and for an overview of Unicode code charts, please see the [MUFI site](#).

3.3 Single-level transcriptions

A transcription of a Medieval Nordic manuscript may be as simple as the Latin example in section 3.1 above. In a typical diplomatic edition, abbreviations are expanded and sometimes proper names are capitalised. The text in fig. 3.2 above could thus be encoded as:

```
<p>Drottinn várr baud Michaelē hofud engli. at fylgia. Adam ok &aolig;llum  
helgum hans at leiða þa i paradísu hína fornu.</p>
```

Here, abbreviations have been expanded silently and proper names capitalised, but the punctuation and orthography remain unchanged. The small capital ‘R’ with a dot above in the second word has been interpreted as a geminate, ‘rr’, while the ligature of ‘a’ and ‘o’ is transcribed with an entity, since this character was not part of the Unicode Standard as of v. 5.0. A correct display of this character thus requires a specific font with a glyph in the Private Use Area (as explained in [ch. 2](#)).

If the text is going to be annotated on a lexicographical level, we recommend that each word is contained in a <w> element. Although not strictly necessary, it is also helpful to identify the level of transcription within each word. This is especially so if the text is going into an archive of texts transcribed on several levels. In the following example, it is clearly stated that each word has been transcribed on a diplomatic level, identified by the <me:dipl> element:

```
<w>  
  <me:dipl>d<ex>ro</ex>ttin<ex>n</ex></me:dipl>  
</w>  
  
<w>  
  <me:dipl>várr</me:dipl>  
</w>  
  
<w>  
  <me:dipl>baud</me:dipl>  
</w>
```

```

<w>
  <me:dipl>michaele</me:dipl>
</w>

<w>
  <me:dipl>hof<ex>ud</ex></me:dipl>
</w>

<w>
  <me:dipl>engli</me:dipl>
</w>

etc.

```

As a rule of thumb, if one removes all mark-up in a single-level transcription, the result is a fully readable text. The text of the example above is thus simply:

```
drottinn várr baud michaele hofud engli
```

This is equivalent to saving a formatted word processor file in a Text Only format. The text string is unchanged, but all information contained in the mark-up is lost.

3.4 Multi-level transcriptions

The transcriptions in [ch. 3.2](#) each reflect a specific level of text representation. However, we believe that the transcription should be expandable to accommodate more than one level. We recommend using the `<w>` element to group each lexical word in the transcription, as explained in [ch. 2.3](#). Within each `<w>` element, the `<choice>` element should be used to group levels of text representation. Each level is identified by descriptive elements: `<me:fac>` for the facsimile rendering (in which the element `<am>` is used for abbreviations), `<me:dipl>` for the diplomatic rendering (in which the element `<ex>` is used for expansions), and `<me:norm>` for the normalised rendering. This makes for a parallel encoding, in which up to three text strings co-exist within the boundaries of the `<w>` elements. Similarly, punctuation marks appear within the `<me:punct>` element.

Element / attribute	Contents
<code><choice></code>	groups a number of alternative encodings for the same point in the text
<code><me:punct></code>	contains a punctuation mark

Note: The ‘me’ prefix can only be used with a RELAX NG schema. It must be left out in texts which will be validated against a DTD.

For the sake of clarity, in the following example we have set out each word in a paragraph of its own:

```

<w>
  <choice>
    <me:fac>&drot; <am>&osup; </am>t tin<am>&bar; </am></me:fac>
    <me:dipl>d<ex>ro</ex>t tin<ex>n</ex></me:dipl>
    <me:norm>Dróttinn</me:norm>
  </choice>
</w>

<w>
  <choice>
    <me:fac>vá&rscapdot; </me:fac>
    <me:dipl>vá&rscapdot; </me:dipl>
  </choice>
</w>

```

```

    <me: norm>várr</me: norm>
  </choice>
</w>

<w>
  <choice>
    <me: facs>bau&drot;</me: facs>
    <me: dipl>baud</me: dipl>
    <me: norm>bauð</me: norm>
  </choice>
</w>

<w>
  <choice>
    <me: facs>michaele</me: facs>
    <me: dipl>michaele</me: dipl>
    <me: norm>Michael</me: norm>
  </choice>
</w>

<w>
  <choice>
    <me: facs>ho&fins;<am>&dsup;</am> engli</me: facs>
    <me: dipl>hof<ex>ud</ex> engli</me: dipl>
    <me: norm>h&oogon;fuðengli</me: norm>
  </choice>
</w>

<me: punct>
  <choice>
    <me: facs>.</me: facs>
    <me: dipl>.</me: dipl>
    <me: norm></me: norm>
  </choice>
</me: punct>

<w>
  <choice>
    <me: facs>at</me: facs>
    <me: dipl>at</me: dipl>
    <me: norm>at</me: norm>
  </choice>
</w>

<w>
  <choice>
    <me: facs>&fins;ylg<am>&ra;</am></me: facs>
    <me: dipl>fylg<ex>ia</ex></me: dipl>
    <me: norm>fylgja</me: norm>
  </choice>
</w>

<me: punct>
  <choice>
    <me: facs>.</me: facs>
    <me: dipl>.</me: dipl>
    <me: norm></me: norm>
  </choice>
</me: punct>

<w>
  <choice>
    <me: facs>a&drot;am</me: facs>
    <me: dipl>adam</me: dipl>
    <me: norm>Adam</me: norm>
  </choice>
</w>

```

```

<w>
  <choice>
    <me:facs>ok</me:facs>
    <me:dipl>ok</me:dipl>
    <me:norm>ok</me:norm>
  </choice>
</w>

<w>
  <choice>
    <me:facs>&aolig;llu<am>&bar;</am></me:facs>
    <me:dipl>&aolig;llu<ex>m</ex></me:dipl>
    <me:norm>&oogon;llum</me:norm>
  </choice>
</w>

<w>
  <choice>
    <me:facs>helgu<am>&bar;</am></me:facs>
    <me:dipl>helgu<ex>m</ex></me:dipl>
    <me:norm>helgum</me:norm>
  </choice>
</w>

<w>
  <choice>
    <me:facs>h<am>&bar;</am>s</me:facs>
    <me:dipl>h<ex>an</ex>s</me:dipl>
    <me:norm>hans</me:norm>
  </choice>
</w>

<w>
  <choice>
    <me:facs>at</me:facs>
    <me:dipl>at</me:dipl>
    <me:norm>at</me:norm>
  </choice>
</w>

<w>
  <choice>
    <me:facs>lei&drot;a</me:facs>
    <me:dipl>leida</me:dipl>
    <me:norm>leiða</me:norm>
  </choice>
</w>

<w>
  <choice>
    <me:facs>pá</me:facs>
    <me:dipl>pá</me:dipl>
    <me:norm>pá</me:norm>
  </choice>
</w>

<w>
  <choice>
    <me:facs>i</me:facs>
    <me:dipl>i</me:dipl>
    <me:norm>í</me:norm>
  </choice>
</w>

<w>
  <choice>
    <me:facs><am>&pbardes;</am>a&drot;i&slong;um</me:facs>

```

```

      <me:dipl>p<ex>ar</ex>adi&slong;um</me:dipl>
      <me:norm>paradísú</me:norm>
    </choice>
  </w>

  <w>
    <choice>
      <me:fac>hína</me:fac>
      <me:dipl>hína</me:dipl>
      <me:norm>hina</me:norm>
    </choice>
  </w>

  <w>
    <choice>
      <me:fac>&fins;ornu</me:fac>
      <me:dipl>fornu</me:dipl>
      <me:norm>fornu</me:norm>
    </choice>
  </w>

  <me:punct>
    <choice>
      <me:fac>.</me:fac>
      <me:dipl>.</me:dipl>
      <me:norm>.</me:norm>
    </choice>
  </me:punct>

```

Note: The sequence ‘hofud engli’ has been analysed as a single word and encoded as suggested in [ch. 2.3.2](#). Punctuation marks have been set out in the **<me:punct>** element; for a fuller discussion, see [ch. 4.8](#).

The display of the transcription is made by style sheets in XML:

- (a) the facsimile level is the content of the **<me:fac>** element, in which the **<am>** element describes abbreviation markers
- (b) the diplomatic level is the content of the **<me:dipl>** element, in which the **<ex>** element describes expanded abbreviations
- (c) the normalised level is the content of the **<me:norm>** element

ðóttuñ vár bauð michaele hof^ð engli. at fylg^ð adam ok ællū
 helgū h̄s at leida þa i paradísum hína fornu.

dróttinn vár bauð michaele hofud engli. at fylgia. adam
 ok ællum helgum hans at leida þa i paradísum hína fornu.

Dróttinn várr bauð Michaele hofuðengli at fylgja Adam
 ok öllum helgum hans at leida þá í paradísum hina fornu.

Fig. 3.6 A display of all levels contained in the multi-level transcriptions above.

As stated above, the elements **<me:fac>**, **<me:dipl>** and **<me:norm>** are not defined in TEI, but are part of the namespace we have defined for Menota texts. Please see the schemas in [Appendix D](#).

The three levels discussed here can be seen as **focal** in the sense that they are typical and often used levels of text representations in Medieval Nordic editions. A number of additional levels can be defined, e.g. a **<me:pal>** level for an even more detailed

paleographical encoding of the text. This level has been included in the Menota schemas, but is not seen as one of the focal levels.

Chapter 4. Document structure

4.1 Introduction: The structure of the manuscript vs. the structure of the work

Viewed as physical objects, rather than as vehicles for texts, manuscripts have a certain structural hierarchy. What is regarded as a single manuscript may in fact comprise more than one volume; Flateyjarbók, for example, is bound in two volumes, and the large rímur codex Acc. 22 in three. A manuscript book is made up of quires or gatherings, each of which contains a number of leaves, normally eight. Each leaf has a recto side and a verso side, and each side may be further divided into columns. The text is then written in lines across the page or column. In order to be able to locate a word quickly and easily, all, or at least most, of these structural divisions must be registered. We need to know that a given word appears in the fifth line of the right-hand or b column on the recto side of folio 34. As it is customary to foliate manuscripts without regard to their quire division, the quires will not normally need to be included in the hierarchical structure, but since the quiring can have implications for the text itself this division should be indicated, and will also generally form part of the `<msDesc>` element, found in the document header.

At the same time, of course, manuscripts obviously do contain texts, which is the reason why most of us are interested in them in the first place. A single manuscript will often contain more than one work, each of which may, in the case of lengthy prose works such as sagas, be divided into chapters or sections. In the case of poetry, rímur for example, a single work (rímnaflokkur) will usually consist of several cantos or fits, each containing a number of stanzas, made up of a number of lines. It may be necessary to group these lines in some other ways as well. The stanzas comprising the mansöngur should be distinguished from the main body of the fit, for example, while to facilitate certain types of metrical analysis it might be desirable to divide the individual stanzas into couplets. Some types of poetry, such as the vikivakakvæði, will have a refrain or burden, which should ideally also be distinguished from the narrative section(s) of the stanza.

XML has at its foundation the notion of a text as a single hierarchical structure, which means that it does not work well where there are several concurrent hierarchies, as is obviously the case when one wishes for example to indicate the line divisions both in a poem and in the manuscript in which the poem is contained. The TEI Guidelines offer various solutions to this problem, enabling both the structure of the document and the structure of the text to be encoded.

4.1.1 Hierarchical divisions

The principal means of representing hierarchy is the `<div>` (i.e. 'division') element. `<div>` elements may freely nest within each other. The `<div>` element has, in addition to the universally available `@id` and `@n` attributes, a `@type` attribute, which specifies the name conventionally given to the level of division, e.g. 'chapter', 'stanza', 'couplet', if attempting to represent the structure of the text, 'page', 'column', 'line' if the physical structure of the manuscript is to be preferred. It will be convenient to specify a value for the `@type` attribute in the `<div>` element at least each time a change of level occurs. The software, however, will keep count of the levels of nesting even if the type attribute is not used.

The complex structure of a work such as a set of rímur could be represented by using four levels of **<div>** elements, **<div type="canto">** for the cantos or fits, **<div type="part">** for the parts (for example the mansöngvar), **<div type="stanza">** for the stanzas, and **<div type="line">** for the lines. If the manuscript being encoded contains more than one set of rímur, as is frequently the case, it might be sensible to use **<div type="canto">** for each set. A simpler form of mark-up is possible, however. Instead of **<div>** elements, the tags **<l>** (for ‘line’) and **<lg>** (for ‘line-group’, i.e. a group of lines functioning as a formal unit) can be used, reserving the **<div>** element for larger structural units. The **@type** attribute is then used to identify the type of unit, e.g. ‘stanza’, ‘couplet’, like in **<lg type="stanza">**. Here again the type need only be defined once. Lines and line-groups can also be numbered and identified using the **@n** and **@id** attributes.

This type of markup focusses on the hierarchical structure of the text. The actual physical realisation of the text is considered of secondary importance – if of importance at all – when dealing with modern printed literary works: little significance is attached to the page and line breaks in the various editions of, say, Orwell's *Nineteen Eighty-Four*. In some cases, however, the early editions of Joyce's works, for example, supervised by the author himself, the physical make-up of the text can be of great consequence. It may also be necessary to maintain the pagination and lineation of standard editions of major works, as these are frequently used in citations in scholarly works. In the case of chirographically transmitted material, the physical organisation of the text is more likely to be recognised as being of importance and in need of encoding. This can be done hierarchically, as above, using **<div>** elements, which are then given the appropriate **@type** attributes, e.g. ‘page’, ‘column’ or ‘line’, but it seems more appropriate to reserve these elements for structural divisions in the text, while indicating the physical structure of the document through the use of so-called ‘milestone’ tags, i.e. **<pb/>**, **<cb/>** and **<lb/>**. These tags make up a separate hierarchy in the file and help to overcome the problem of overlapping structures in the mark-up; see also the discussion in [ch. 4.10](#) below.

The rest of this chapter presents how the text may be encoded at higher structural levels than characters and words. Important elements here are the larger divisions of the text, like chapters, paragraphs (with headings), and stanzas. This chapter also presents how pagination and foliation, together with column-breaks and line-breaks, may be encoded. The following [TEI](#) elements are presented:

Elements	Contents
<text> , <body>	Main divisions of the text,
<div>	division into chapters (multiple levels are encoded by nesting elements),
<p>	prose paragraphs,
<lg> , <l>	line groups and lines,
<head>	headings,
<pb/> , <cb/> , <lb/>	page-, column- and line-breaks.

4.2 Main divisions of a TEI document

The following presentation is based on [ch. 4 ‘Default Text Structure’](#) of the TEI P5 Guidelines.

A TEI document is always at its highest level enclosed by the start tag `<TEI>` and the end tag `</TEI>`. Within the `<TEI>` element, two other elements appear in a fixed order, namely the `<teiHeader>` and the `<text>` elements. Within the `<text>` element, the body text may appear, enclosed in the element `<body>`. If the text has front matter, there will be an element `<front>`, placed before `<body>` containing it. Similarly, there may be an element `<back>`, placed after `<body>` and containing back matter. The elements `<teiHeader>`, `<text>` and `<body>` are required in any TEI-conformant document, while `<front>` and `<back>` are optional. This, then, is the basic structure of a TEI document:

Elements	Contents
<code><TEI></code>	The TEI document begins here,
<code><teiHeader> ... </teiHeader></code>	the header goes here,
<code><text></code>	the text itself begins here,
<code><front> ... </front></code>	any front matter goes here,
<code><body> ... </body></code>	the main body of the text goes here,
<code><back> ... </back></code>	any back matter goes here,
<code></text></code>	the text ends here,
<code></TEI></code>	the TEI document ends here.

4.2.1 Another possible first division of the text: More than one `<text>` element

The transcriber may want to divide a document into more than one text. This can be done with the `<group>` element, which should be contained in the top level `<text>` element taking the place of `<body>` in the simpler scheme illustrated above. The following structure appears:

```

<text>
  <front> ... </front>
  <group>
    <text>
      <front> ... </front>
      <body> ... </body>
      <back> ... </back>
    </text>
    <text>
      <front> ... </front>
      <body> ... </body>
      <back> ... </back>
    </text>
  </group>
  <back> ... </back>
</text>

```

The main structure of the text, at the levels of work, first main division, second main division, first chapter of first main division, second chapter of first main division and so

on, can be encoded in different ways. If the electronic document consists of more than one work, the **<group>** structure illustrated above is the natural choice. In that case, one would get multiple sets of further structural divisions, one set within each of the **<body>** elements. If the electronic document is considered as a single work, and placed in one **<text>** element, there will only be a single **<body>** element that needs further divisions.

4.3 Chapters: **<div>**

Further division of the **<body>** block is achieved through **<div>** elements, with one level nesting inside the other as the transcriber moves down through the hierarchical structure of the text.

4.3.1 Type- and level-specified **<div>** elements

In a complex document, **<div>** elements may be specified by **@type** and **@n** attributes. In this example, the three first chapters of a work have been contained in **<div>** elements at the same hierarchical level (siblings):

Elements	Contents
<div type="chapter" n="1"> ... </div>	Chapter one goes here,
<div type="chapter" n="2"> ... </div>	chapter two goes here,
<div type="chapter" n="3"> ... </div>	chapter three goes here (and so on).

4.3.2 Unspecified **<div>** elements

It is also possible to use **<div>** elements without specifying their type:

Elements	Contents
<div> ... </div>	Chapter one goes here,
<div> ... </div>	chapter two goes here,
<div> ... </div>	chapter three goes here (and so on).

4.3.3 Nesting **<div>** elements

Note that **<div>** elements may nest inside each other. For example, the levels of work, chapter and then paragraph can be encoded in the following manner:

Elements	Contents
<div type="work">	The whole work starts here,
<div type="chapter">	the first subdivision starts here (nested),
<p> ... </p>	one paragraph of the subdivision goes here,

Elements	Contents
<code></div></code>	end of the subdivision,
<code></div></code>	end of the work.

While `<div>` elements may nest as shown here, `<p>` elements may not. They must be encoded sequentially, i.e. as siblings.

4.4 Paragraph text: `<p>`

The basic-level element for prose text is the paragraph, `<p>`. Typically, the deepest level `<div>` element will contain one or more `<p>` elements:

Elements	Contents
<code><div></code>	A new chapter starts here,
<code><head> ... </head></code>	this contains the heading,
<code><p> ... </p></code>	first paragraph,
<code><p> ... </p></code>	second paragraph,
<code><p> ... </p></code>	third paragraph,
<code></div></code>	the chapter ends here.

The `<p>` element may appear in other contexts, such as in the `<teiHeader>` element. It may also contain a number of other elements, but – as underlined above – it may not contain other `<p>` elements, i.e. it is not allowed to nest.

4.5 Metrical text: `<lg>` and `<l>`

The elements discussed here are defined and explained in [ch. 6 ‘Verse’](#) of the TEI P5 Guidelines.

Texts in verse should be encoded using `<lg>` (line group), which in turn contains one or more `<l>` elements (lines). As with `<div>`, `<lg>` elements can nest. According to the TEI Guidelines `<lg>` is a sibling of, i.e. at the same level as, `<p>`, and cannot be contained within it (unless it appears within a `<q>` element). Example:

Elements	Contents
<code>... </p></code>	A paragraph ends here,
<code><lg></code>	a line group starts here,
<code><l> ... </l></code>	first line,
<code><l> ... </l></code>	second line,

Elements	Contents
<code><l> ... </l></code>	third line,
<code></lg></code>	the line group ends here,
<code><p> ...</code>	and a new paragraph starts here.

Nesting of `<lg>` elements is useful for marking up longer poems. When a poem consists of two levels of line groups one may encode its structure as shown here:

Elements	Contents
<code><lg type="stanza"></code>	Here a line group on level one begins, a stanza,
<code><lg type="couplet"></code>	here a subgroup starts, a couplet,
<code><l> ... </l></code>	the first line,
<code><l> ... </l></code>	second line,
<code></lg></code>	and here the subgroup ends, the first of the couplets.
<code><lg></code>	Here a new subgroup starts,
<code><l> ... </l></code>	line,
<code><l> ... </l></code>	line,
<code></lg></code>	here the second subgroup ends,
<code></lg></code>	and here the level one line group ends.

The `<lg>` and `<l>` elements may have several attributes, among other things for encoding information about rhyme or other metrical phenomena. See [ch. 9.2](#) of this handbook for a more detailed presentation of metrical encoding.

Having `<p>` and `<lg>` as siblings can create problems for the encoding of prosimetrum texts, where lines or verse or even whole poems can appear within prose text, often as part of direct speech. However, rather than including `<lg>` directly within the `<p>` element, we recommend inserting the `<p>` and `<lg>` elements within `<div>` elements, using one `<div>` for each of them:

Elements	Contents
<code><div type="chapter" n="1"></code>	A chapter opens here,
<code><div type="text"></code>	beginning with some prose text, indicated by a <code><div></code> element.
<code><p> ... </p></code>	The text goes here,
<code></div></code>	and ends here, indicated by the <code><div></code> element.

Elements	Contents
<code><div type="stanza"></code>	Then a poem begins, indicated by a new <code><div></code> element
<code><lg></code>	with a linegroup (a stanza)
<code><l> ... </l></code>	containing some lines.
<code></lg></code>	The linegroup ends here,
<code></div> ...</code>	and the poem (i.e. the <code><div></code> element) also ends here.
<code><div type="text"></code>	A new piece of prose text begins, indicated by a new <code><div></code> element.
<code><p> ... </p></code>	The text goes here,
<code></div></code>	and ends here, indicated by the <code><div></code> element.
<code></div></code>	The chapter ends here.

4.6 Headings: `<head>`

The element `<head>` is used for containing headings on all levels of the document. If `<head>` is placed at the start of a `<div>` element, it typically contains a chapter heading:

Elements	Contents
<code><div></code>	Here a chapter begins,
<code><head> ... </head></code>	its heading,
<code><p> ... </p></code>	the first paragraph of the chapter,
<code><p> ... </p></code>	the second paragraph,
<code></div></code>	and here the chapter ends.

The level of a heading follows from the enclosing element. A `<head>` element within a level three `<div>` element, is a heading for a level three partition of the text.

An overlap problem may occur when, as is common in Old Norse manuscripts, headings for chapters are placed on the same text line as the last words of the preceding chapter. Graphically, the heading of a following chapter is in fact placed inside the text block of the preceding chapter. As we would like to place headings at the beginning of the textual divisions to which they logically belong, we must override the structure of the layout. One way to do that is to ignore the heading of the following chapter when transcribing the last lines of the preceding chapter. When that chapter is closed with an end tag `</div>`, we open the next chapter with its start tag `<div>`, go back one or two lines in the manuscript to where the heading starts and transcribe from there.

It is generally recommended (ch. 4.7 below) that line break elements `<lb/>` are inserted while transcribing the manuscript. Following that rule, it is obvious that one cannot keep a single series of line break elements through the intersection between the chapters in the case of a heading overlap. However, it is not invalid according to TEI that `<lb/>` elements

carrying the same number occur twice. Our recommendation is to use that possibility: When moving up again to encode the heading of the following chapter, then assign the actual number of that graphic line to its **<lb/>** element.

Consider the following column (line numbers in left margin):

```
05 .....
06 .... these are the last
07 Header for words of
08 chapter two chapter 1.
09 Here begins the text
10 of chapter two .....
11 .....
```

The example would be encoded this way (word tags omitted):

```
<div>
  <p> .....
    <lb n="6"/> ... these are the last
    <lb n="7"/>words of<lb n="8"/>chapter 1.</p>
</div>
<div>
  <head rend="inline left"><seg><lb n="7"/>Header for<lb n="8"/>
    chapter two</seg></head>
  <p>
    <lb n="9"/>Here begins the text<lb n="10"/>of chapter two ...
    <lb n="11"/> .....
  </p>
</div>
```

In this case is it important for the processing of the XML document that the **@rend** attribute in the **<head>** element gives the information that this headline is 'inline', and that it is located on the left side of the column. The element **<seg>** is used to encapsulate the **<lb/>** with the words that are on that particular line in the header. It is possible to make XSLT stylesheets to process this kind of encoding, but it is not simple.

When double numbering of line breaks is used in a transcription, one should make sure that any automatic numbering program that is run on the **<lb/>** elements is set up not to override manually given numbers.

4.7 Page, column and line breaks: **<pb/>**, **<cb/>**, **<lb/>**

4.7.1 Page breaks and column breaks

TEI uses the empty element **<pb/>** to indicate page breaks. This element has an attribute **@n** which can be used for the page numbers. As it is customary to refer to the manuscript leaves, rather than pages, the value of the **@n** attribute should indicate front or back pages (recto, verso). Column breaks, **<cb/>**, should also be indicated in manuscripts with two or more columns. Recommended values for the **@n** attribute of the **<cb/>** element are 'A', 'B' and so on. Example:

Elements	Contents
<pb n="1r"/>	Folio one, recto page, begins here,
<cb n="A"/>	the first column begins here,
<cb n="B"/>	and the second column begins here.

Elements	Contents

<code><pb n="1v"/></code>	Folio one, verso page, begins here,
<code><cb n="A"/></code>	the first column of the verso page begins here,
<code><cb n="B"/></code>	and the second column begins here.

Page break information from, for example, a printed standard edition, can be encoded in addition to the `<pb/>` tagging that refers to the manuscript itself. If one for example would like to add page break information from a standard edition, we recommend using the `@ed` attribute:

```
<pb ed="Standard Edition" n="1"/>
```

4.7.2 Line breaks

Line breaks are also indicated with an empty element, the `<lb/>`, which is placed at the beginning of a new line and may be numbered by using the `@n` attribute:

```
<lb n="1"/>Line number one begins here.
```

We recommend that each page, column and line be identified with an element at the very beginning. So for a manuscript with two columns, the three first lines in the first column on the back of the third leaf (folio) would be encoded in this manner:

```
<pb n="3v"/><cb n="A"/><lb n="1"/>This is the first line.
<lb n="2"/>This is the second line.
<lb n="3"/>This is the third line.
etc.
```

In other words, there should be as many `<pb/>` elements as there are pages, as many `<cb/>` elements as there are columns, and as many `<lb/>` elements as there are lines. We strongly discourage the use of the `<lb/>` element in the same way as the `
` element in HTML, in which there typically is one `
` element less than the number of lines (as the `
` element is inserted between the lines).

We recommend that `<lb/>` is used consistently for indicating the line breaks of the manuscript itself. One may include more than one layer of line break encoding, distinguishing them from each another with the `@ed` attribute, as shown in [ch. 4.7.1](#) above.

4.8 Punctuation and hyphenation

4.8.1 Punctuation

If a text has been encoded with each word within a `<w>` element, we recommend that punctuation is encoded within `<me:punct>` elements. This element permits the same levels of text representation as the `<w>` element, i.e. `<me:fac>`, `<me:dipl>` and `<me:norm>`. While punctuation on the `<me:fac>` and `<me:dipl>` levels in most cases will be identical, it is often radically different on the `<me:norm>` level. Here, many dots in the manuscript will simply be suppressed, while other punctuation marks will be added, including modern punctuation marks like quotation marks and exclamation marks. Suppressing a punctuation mark is simply done by leaving the element empty,

while any supplied marks are encoded by adding a new `<me:punct>` element in which the `<me:facs>` and possibly also the `<me:dipl>` element will be empty.

A text transcribed as

ok nu sagdi hann. þat er eigi sva. sem þu segir

on the `<me:dipl>` level would probably be rendered as

‘Ok nú,’ sagði hann, ‘Þat er eigi svá sem þú segir.’

on the `<me:norm>` level, allowing for some variation in the type of quotation marks and the order of comma or full stop and quotation mark. In a fully marked-up text, the dot after ‘sva’ would probably be suppressed on the `<me:norm>` level, while quotation marks would be added, and also a comma after ‘nu’. Finally, the dot after ‘hann’ would be changed into a comma:

```
<me:punct>
  <choice>
    <me:dipl></me:dipl>
    <me:norm>"</me:norm>
  </choice>
</me:punct>

<w>
  <choice>
    <me:dipl>ok</me:dipl>
    <me:norm>Ok</me:norm>
  </choice>
</w>

<w>
  <choice>
    <me:dipl>nu</me:dipl>
    <me:norm>nú</me:norm>
  </choice>
</w>

<me:punct>
  <choice>
    <me:dipl></me:dipl>
    <me:norm>,"</me:norm>
  </choice>
</me:punct>

<w>
  <choice>
    <me:dipl>sagdi</me:dipl>
    <me:norm>sagði</me:norm>
  </choice>
</w>

<w>
  <choice>
    <me:dipl>hann</me:dipl>
    <me:norm>hann</me:norm>
  </choice>
</w>

<me:punct>
  <choice>
    <me:dipl>.</me:dipl>
    <me:norm>,"</me:norm>
  </choice>
</me:punct>
```

```

<w>
  <choice>
    <me:dipl>pat</me:dipl>
    <me:norm>pat</me:norm>
  </choice>
</w>

<w>
  <choice>
    <me:dipl>er</me:dipl>
    <me:norm>er</me:norm>
  </choice>
</w>

<w>
  <choice>
    <me:dipl>eigi</me:dipl>
    <me:norm>eigi</me:norm>
  </choice>
</w>

<w>
  <choice>
    <me:dipl>sva</me:dipl>
    <me:norm>svá</me:norm>
  </choice>
</w>

<me:punct>
  <choice>
    <me:dipl>.</me:dipl>
    <me:norm></me:norm>
  </choice>
</me:punct>

<w>
  <choice>
    <me:dipl>sem</me:dipl>
    <me:norm>sem</me:norm>
  </choice>
</w>

<w>
  <choice>
    <me:dipl>pu</me:dipl>
    <me:norm>pú</me:norm>
  </choice>
</w>

<w>
  <choice>
    <me:dipl>segir</me:dipl>
    <me:norm>segir</me:norm>
  </choice>
</w>

<me:punct>
  <choice>
    <me:dipl></me:dipl>
    <me:norm>."</me:norm>
  </choice>
</me:punct>

```

In many cases, a dot should be interpreted as an abbreviation mark rather than a punctuation mark. In such cases, we recommend that the dot is encoded using the ordinary full stop in Basic Latin, but that it is placed within the **<am>** element. A text transcribed as

nu fann kgr. engan mann þar

on the **<me:facs>** level would probably be rendered as

nu fann *konongr* engan mann þar

on the **<me:dipl>** level. In a fully marked-up text, the abbreviationr ‘kgr.’ would be encoded within an **<am>** element, while it would be expanded into ‘onon’ (or ‘onun’) on the **<me:dipl>** level:

```
<w>
  <choice>
    <me:facs>nu</me:facs>
    <me:dipl>nu</me:dipl>
  </choice>
</w>

<w>
  <choice>
    <me:facs>fann</me:facs>
    <me:dipl>fann</me:dipl>
  </choice>
</w>

<w>
  <choice>
    <me:facs>kgr<am>.</am></me:facs>
    <me:dipl>k<ex>onon</ex>gr</me:dipl>
  </choice>
</w>

<w>
  <choice>
    <me:facs>engan</me:facs>
    <me:dipl>engan</me:dipl>
  </choice>
</w>

<w>
  <choice>
    <me:facs>mann</me:facs>
    <me:dipl>mann</me:dipl>
  </choice>
</w>

<w>
  <choice>
    <me:facs>þar</me:facs>
    <me:dipl>þar</me:dipl>
  </choice>
</w>

<me:punct>
  <choice>
    <me:facs></me:facs>
    <me:dipl>.</me:dipl>
  </choice>
</me:punct>
```

In some cases, a word abbreviated with a dot may occur at the end of a sentence, e.g.

nu fann hann eigi kgr.

This dot would be interpreted as an abbreviation mark and possibly also as a punctuation mark. On the **<me:facs>** level it would be encoded as no more than a dot, while on the **<me:dipl>** level it would be suppressed when ‘kgr.’ had been expanded to ‘konongr’. The

encoder might, however, add a dot as a punctuation mark within a **<me:punct>** element. That would certainly be the case on the **<me:norm>** level, possibly also on the **<me:dipl>** level:

```
<w>
  <choice>
    <me:fac>nu</me:fac>
    <me:dipl>nu</me:dipl>
    <me:norm>Nú</me:norm>
  </choice>
</w>

<w>
  <choice>
    <me:fac>fann</me:fac>
    <me:dipl>fann</me:dipl>
    <me:norm>fann</me:norm>
  </choice>
</w>

<w>
  <choice>
    <me:fac>hann</me:fac>
    <me:dipl>hann</me:dipl>
    <me:norm>hann</me:norm>
  </choice>
</w>

<w>
  <choice>
    <me:fac>eigi</me:fac>
    <me:dipl>eigi</me:dipl>
    <me:norm>eigi</me:norm>
  </choice>
</w>

<w>
  <choice>
    <me:fac>kgr<am>.</am></me:fac>
    <me:dipl>k<ex>onon</ex>gr</me:dipl>
    <me:norm>konungr</me:norm>
  </choice>
</w>

<me:punct>
  <choice>
    <me:fac></me:fac>
    <me:dipl>.</me:dipl>
    <me:norm>.</me:norm>
  </choice>
</me:punct>
```

On all three levels, a dot will be displayed after the word ‘konungr’, but the dot on the **<me:fac>** level is classified as an abbreviation mark (since it occurs within the **<am>** element), while the dot on the **<me:dipl>** and the **<me:norm>** levels is classified as a punctuation mark (since it occurs within the **<me:punct>** element).

The dot is by far the most common punctuation mark in Medieval Nordic sources. A question mark was sometimes used, while quotation marks and exclamation marks are post-medieval and only seen in normalised editions. There are a few additional punctuation marks, e.g. the *punctus elevatus* and the *virgula*. These marks can be encoded using entities, but should otherwise be kept within the **<me:punct>** element. See also [ch. 6.3.8](#) below.

4.8.2 Hyphenation

In medieval manuscripts, hyphens are frequently used at the end of a line to indicate that the word continues on the next line. In such cases, we recommend that the hyphen is entered immediately before the `<lb/>` element. This is what it would look like in a single-level transcription (cf. [ch. 3.3](#)):

```
<lb n="1"/>This is an example of how hyphen-
<lb n="2"/>ation can be encoded.
```

If the hyphen is missing in the manuscript, we suggest that the element `<supplied>` is used to contain the hyphen added by the transcriber:

```
<lb n="1"/>This is an example of how hyphen<supplied>-</supplied>
<lb n="2"/>ation can be encoded.
```

If the editor wants to display supplied hyphens differently from those found in the manuscript, that can easily be done by a stylesheet.

In a multi-level transcription, hyphenation would be contained in the `<me:punct>` element. Taking the word ‘hæ-góma’ as an example (from fig. 4.1 below, divided between line 3 and 4), the `<me:punct>` element would be placed within each textual level - facsimile, diplomatic and normalised.

```
<w>
  <choice>
    <me:fac>hæ<me:punct>-</me:punct><lb n="4"/>góma</me:fac>
    <me:dipl>hæ<me:punct>-</me:punct><lb n="4"/>góma</me:dipl>
    <me:norm>hæ<me:punct>-</me:punct><lb n="4"/>góma</me:norm>
  </choice>
</w>
```

In a display of the facsimile level, hyphens will always be rendered, while they may be suppressed on the diplomatic level, and they will always be suppressed on the normalised level.

If the hyphen does not occur in the manuscript but is supplied by the transcriber or editor, we recommend adding a `@type` attribute with the value ‘supplied’:

```
<w>
  <choice>
    <me:fac>hæ<me:punct type="supplied">-</me:punct>
    <lb n="4"/>góma</me:fac>
    <me:dipl>hæ<me:punct type="supplied">-</me:punct>
    <lb n="4"/>góma</me:dipl>
    <me:norm>hæ<me:punct type="supplied">-</me:punct>
    <lb n="4"/>góma</me:norm>
  </choice>
</w>
```

Note that a single line break will appear several times in a multi-level transcriptions, if it occurs within a word. Great caution must therefore be taken with automatic numbering of `<lb/>` elements.

4.9 Initials and highlighted characters

Medieval manuscripts often have initials, sometimes quite large and often decorated in various ways. It is also quite common to find a highlighted capital at the beginning of a section in the text, a *littera notabilior*. Some transcribers would simply transcribe an initial and a *littera notabilior* with capitals and refer to a facsimile for the way they have

been drawn. Other transcribers would like to encode these traits of the manuscript. For this purpose, we recommend using the `<c>` element with a `@type` and a `@rend` attribute.

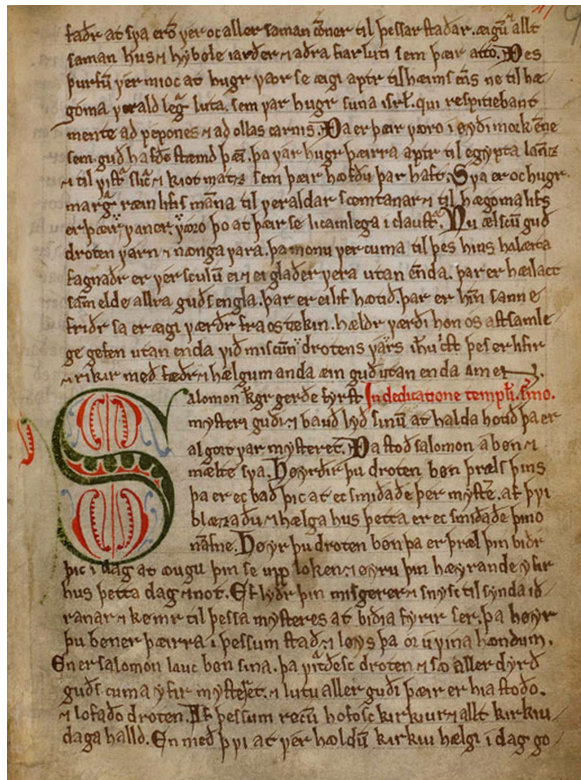


Fig. 4.1 AM 619 4to, fol. 47r. Note the decorated initial 'S' and the *littera notabilior*, beginning with a capital *eth*, 'Ð', in the last word of line 2.

Elements / attributes	Contents
<code><c></code>	contains a character
<code>@type</code>	specifies the type of character, e.g. 'initial', 'littNot'
<code>@rend</code>	specifies how the character has been rendered in the source

In fig. 4.1, the last word of line 2 can be encoded as

```
<c type="littNot" rend="black">&ETH;</c>es
```

while the first word of line 16 can be encoded as

```
<c type="initial" rend="red and green">S</c>alomon
```

This type of encoding is more relevant for the facsimile and possibly the diplomatic level, but not for the normalised level of text representation.

4.10 Overlapping structures

There are no simple ways of encoding overlapping structures in XML, since XML is a strict tree structure in which every element must be part of a single 'parent' element. For example, a word or sentence may be written over two manuscript pages. If we represent the manuscript page as an element, the words will not belong to a single page and a parser error will occur.

This problem is dealt with in the current chapter by using empty elements to represent page breaks in the manuscript, rather than a page of text (cf. [ch. 4.7](#) above). The same is true for columns and lines, where words, sentences and paragraphs routinely overlap with the physical features of the manuscript. These elements, `<pb/>`, `<cb/>` and `<lb/>`, are empty in the sense that they are inserted at a specific point in the structure without any extension. For this reason, they are often referred to as milestones. Note the position of the slash in these elements.

In [ch. 11 ‘Representation of Primary Sources’](#) in the TEI P5 Guidelines the elements `<addSpan/>`, `<delSpan/>` and `<damageSpan/>` are defined. These elements are counterparts to the elements `<add>`, `` and `<damage>`, but are all empty, and should be used when the feature to be encoded crosses structural divisions. There are in fact many more elements which can cross structural divisions, e.g. `<sic>`, `<corr>`, `<unclear>` and `<supplied>`, but there are no corresponding `<sicSpan>`, `<corrSpan>`, `<unclearSpan>` and `<suppliedSpan>`. Rather than adding these and several other elements we recommend using one generic empty element to cover all cases of overlapping structures. We have called this new element `<me:textSpan/>` and given it attributes from the classes ‘att.spanning’, ‘att.transcriptional’, ‘att.typed’ and ‘att.global’, and the attribute `@me:category`:

Elements / attributes	Contents
<code><me:textSpan/></code>	A generic element to handle overlapping text structures
<code>@category</code>	Specifies the type of span, restricted to this list of values:
'add'	for contents that would otherwise be contained by the <code><add></code> element, cf. ch. 7.2.1
'corr'	for contents that would otherwise be contained by the <code><corr></code> element, cf. ch. 7.4.3
'del'	for contents that would otherwise be contained by the <code></code> element, cf. ch. 7.2.2
'damage'	for contents that would otherwise be contained by the <code><damage></code> element, cf. ch. 7.5.1
'gap'	for contents that would otherwise be contained by the <code><gap/></code> element, cf. ch. 7.3.1
'me:expunged'	for contents that would otherwise be contained by the <code><me:expunged></code> element, cf. ch. 7.4.2
'sic'	for contents that would otherwise be contained by the <code><sic></code> element, cf. ch. 7.4.3
'supplied'	for contents that would otherwise be contained by the <code><supplied></code> element, cf. ch. 7.4.1
'unclear'	for contents that would otherwise be contained by the <code><unclear></code> element, cf. ch. 7.3.2
'other'	for any other contents
<code>@spanTo</code>	Specifies the end point of the text span, using values like:

Elements / attributes	Contents
'an1'	anchor 1
'an2'	anchor 2, etc.
<anchor/>	An empty element (milestone) which attaches an identifier to a point within a text
@xml:id	Specifies the identifier corresponding to the one used in the @spanTo attribute of the preceding <me:textSpan> element, using values like:
'an1'	anchor 1
'an2'	anchor 2, etc.

We will discuss an example of an overlapping structure in AM 673 b 4to (*Plácitusdrápa* 1):

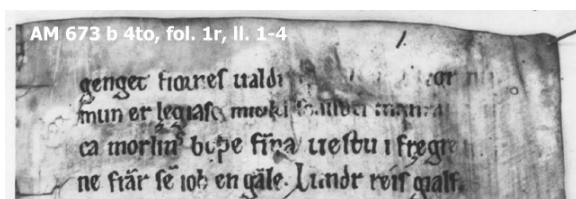


Fig. 4.2 AM 673 b 4to, fol. 1r, ll. 1-4

The first three lines read approximately:

genget fiornes ualdr [quap.....fr]legr nu | mun er lægiasc miuks scalldu manra[un
sli] | ca morlins boþe finna uestu i frægre f[rest]

The letters in brackets were read by earlier editors, especially Finnur Jónsson in 1889. For this section, we will discuss the text at the end of the second line and at the start of the third. It is clear that part of each word is missing, but the damaged manuscript forms a single feature. Text can be supplied from Finnur Jónsson's transcription, but we want to represent both the damage and the supplied text as a single feature, which overlaps with the middle of the two words. The simple encoding, without the unclear text marked or the supplied text, would be:

```
<w>manra<gap/></w>
<w><gap/><lb n="3"/>ca</w>
```

With the supplied text encoded in the conventional way, the following would produce an error:

```
<!-- WRONG: -->
<w>manra<supplied resp="FJ">aun</w>
<!-- the processor stops here because this is not well-formed XML -->
<w>sli</supplied><lb n="3"/>ca</w>
```

The <unclear> and <supplied> elements, if used in their conventional way, would overlap with the <w> elements, meaning that the word tag would close before an element inside it had closed. That would stop an XML processor from proceeding any further with the document.

In these guidelines, we offer two solutions to the problem of overlapping structures. The first is more complex, but more robust. The second is simpler, but is less machine-readable

and may affect the validation of the document structure in other respects. Even so, we recommend the latter solution.

4.10.1 Linked segments

The following approach is more sound from the point of view of an XML document, but creates extra tagging. The feature is encoded in a series of separate elements, linked together.

In order to encode linked segments, the encoder should break the overlapping feature into parts which fit within the XML structure (usually within the word or dipl/facs/norm elements). Each part is identified using the **@xml:id** attribute, and they are linked together using the following attributes:

Elements / attributes	Contents
@xml:id	provides a unique identifier for the element bearing the attribute
@next	used at the start and in the middle: an IDREF pointing to the element which marks the next tag of the same feature
@prev	used in the middle and at the end: an IDREF pointing to the element which marks the previous tag of the same feature

The two-word example above is encoded thus:

```
<w>man<supplied source="FJ" xml:id="sup1.1" next="sup1.2">raun
  </supplied></w>
<w><supplied xml:id="sup1.2" prev="sup1.1">&slong;li</supplied>
  <lb n="3"/>ca</w>
```

Adding all three textual levels, including the unclear text encoded at the facs level, we would have:

```
<w>
  <choice>
    <me:facs>man<unclear xml:id="uncl.1" next="uncl.2">
      <gap extent="8"/></unclear></me:facs>
    <me:dipl>man<supplied source="FJ" xml:id="sup1.1" next="sup1.2">raun
      </supplied></me:dipl>
    <me:norm>manraun</me:norm>
  </choice>
</w>
<w>
  <choice>
    <me:facs><unclear xml:id="uncl.2" prev="uncl.1">&slong;li</unclear>
      <lb n="3"/>ca</me:facs>
    <me:dipl><supplied xml:id="sup1.2" prev="sup1.1">&slong;li</supplied>
      <lb />ca</me:dipl>
    <me:norm>slíka</me:norm>
  </choice>
</w>
```

It is recommended that the additional information for the feature (such as the editor responsible, type, etc.) be only included in the first element, but editors may wish to include the attributes in all elements.

For the purposes of display, the start of a feature can be marked by selecting the element with the 'next' attribute set, but not the 'prev'; and the end can be marked by selecting the element with the 'prev' attribute set but not the 'next'.

4.10.2 Boundary marking with empty elements

Another solution is to encode the beginning and end of a text span with empty elements. This method has been described in [ch. 20 'Non-hierarchical Structures'](#) of the TEI P5 Guidelines and will be applied here in a slightly modified version. As outlined above, we have introduced a generic element `<me:textSpan/>` which is specified by way of a `@category` attribute. If, for example, the overlapping structure to be encoded is a piece of supplied text, this fact is expressed through the value of the `@category` attribute:

```
<me:textSpan category="supplied"/>
```

Thus, all instances of supplied text in the file will either be contained in supplied elements (in non-overlapping contexts) or in `<me:textSpan category="supplied">` elements (in overlapping contexts).

In addition to inserting the empty `<me:textSpan/>` element at the beginning of the textual span, an attribute `@spanTo` is added with a suitable index, e.g.

```
<me:textSpan category="supplied" spanTo="an1"/>
```

It now remains to mark the end of the span, i.e. the extent of the supplied text, with another empty element, the TEI `<anchor/>` element. This must be specified with an `@xml:id` attribute having the same index as the `@me:spanTo` attribute at the beginning of the span:

```
<anchor xml:id="an1"/>
```

The full encoding will be like this:

```
<w>man<me:textSpan category="supplied" spanTo="an1"/>raun</w>
<w>&slong;li<anchor xml:id="an1"/><lb n="3"/>ca</w>
```

Note that the value of `@xml:id` attribute must be unique within the whole document.

There is no simple answer to the problem of non-hierarchical structures in XML encoding. However, we believe that using empty elements as boundary markers may prove to be the simplest and most general encoding, and it is therefore the solution we recommend. With either technique, only one method should be used in each document.

Chapter 5. Characters: typology and encoding

5.1 Introduction

The basic characters a-z / A-Z in the Latin alphabet can be encoded in virtually any electronic system and transferred from one system to another without loss of information. Any other characters may cause problems, even well established ones such as Modern Scandinavian ‘æ’, ‘ø’ and ‘å’. In v. 1 of *The Menota handbook* we therefore recommended that all characters outside a-z / A-Z should be encoded as entities, i.e. given an appropriate description and placed between the delimiters ‘&’ and ‘;’. In the last years, however, all major operating systems have implemented full Unicode support and a growing number of applications, including most web browsers, also support Unicode. We therefore believe that encoders should take full advantage of the Unicode Standard, as recommended in [ch. 2.2.2](#) above.

As of version 2.0, the character encoding recommended in *The Menota handbook* has been synchronised with the recommendations by the [Medieval Unicode Font Initiative](#). The [character recommendations](#) by MUFI contain more than 1,300 characters in the Latin alphabet of potential use for the encoding of Medieval Nordic texts. As a consequence of the synchronisation, the list of entities which is part of the Menota scheme is identical to the one by MUFI. In other words, if a character is encoded with a code point or an entity in the MUFI character recommendation, it will be a valid character encoding also in a Menota text. For more information on this synchronisation, please refer to [Appendix A](#).

From an encoding point of view, three major classes of characters should be kept apart :

(1) Basic Latin (a-z / A-Z). These characters can be encoded as they are, without resorting to entities. Note, however, that a few characters in Basic Latin are used for specific purposes in XML encoding, so if these characters are going to be encoded as such, only entities will do. These characters are the *ampersand*, ‘&’, which must be encoded as ‘&’; the *less-than sign*, ‘<’, which must be encoded as ‘<’ and the *greater-than sign*, ‘>’, which must be encoded as ‘>’.

(2) All characters in the Unicode Standard outside Basic Latin. All of these characters can be encoded directly with their Unicode codepoints, e.g. using one of the keyboards offered on the [MUFI](#) site. MUFI compliant fonts probably contain all characters that are needed, e.g. the free Andron web font (see the [MUFI font page](#)). However, as explained in [ch. 2.2](#) above, one may refer to all characters outside Basic Latin with entities, and one may mix Unicode encoding and encoding with entities in the same document.

(3) Characters in the Private Use Area. A number of characters in Medieval Nordic manuscripts are not part of the Unicode Standard, even if a substantial number of central characters recently was proposed for Unicode and became part of the standard as of v. 5.1 (April 2008). Characters in the Private Use Area are coordinated by MUFI, and as explained above, Menota synchronises its list of characters with the one by MUFI.

The following example will illustrate how these rules should be interpreted:

drottinn vár baud michaelē hofud engli. at fylgia adam
ok ællum helgum hans at leida þa i paradifum hína fornu.

Fig. 5.1 Text example from *Niðrstigningar saga* in AM 233a fol, 28v, l. 1-2 (cf. [ch. 3.2](#) above).

If entities are used for all characters outside Basic Latin, the example above would look like this (transcribed on a diplomatic level, with silent expansion of abbreviations):

```
drottinn v&acute;&rscapdot; baud michaelē hofud engli. at fylgia adam
ok &aolig;llum helgum hans at leida &thorn;a i paradi&slong;um
h&iacute;na fornu.
```

Four of these characters need not be encoded with entities since they are part of the Unicode Standard, i.e. ‘á’ and ‘í’ (available in nearly all fonts), ‘þ’ (available in most fonts) and ‘#’ (‘long s’, available in some fonts). Two characters are not part of the standard and must be referred to by entities, i.e. the small capital ‘R’ with a dot above, ‘&rscapdot;’, and the ligature of ‘a’ and ‘o’, ‘&aolig;’. They are both located in the Private Use Area. The transcription immediately becomes more legible:

```
drottinn v&rscapdot; baud michaelē hofud engli. at fylgia adam
ok &aolig;llum helgum hans at leida þa i paradi#um hína fornu.
```

The small capital ‘R’ with a dot could in fact be encoded without resorting to the Private Use Area. It would then have to be decomposed, i.e. encoded as a sequence of a small capital R, 0280 in Unicode, and a dot above, 0307 in Unicode. This combination may not display well in all editors or browsers, so some encoders would prefer to use the 0280 code point for the small capital ‘R’, but encode the dot above with the entity ‘&combdot;’. The small capital ‘R’ are not found in all fonts so it may not display properly, but the encoding would be correct (and with a suitable font, the character would display properly):

```
drottinn v&#x0280&combdot; baud michaelē hofud engli. at fylgia adam
ok &aolig;llum helgum hans at leida þa i paradi#um hína fornu.
```

The three encoding examples above are all valid according to the Menota schemes. The major thing to remember is not to use code points for characters in the Private Use Area. The following encoding is valid, but not advisable:

```
drottinn v&#xEF22; baud michaelē hofud engli. at fylgia adam
ok &#xEF93;llum helgum hans at leida þa i paradi#um hína fornu.
```

In this example, the Private Use Area code point for LATIN LETTER SMALL CAPITAL R WITH DOT, EF22, and for LATIN SMALL LIGATURE AO, EF93, have been used. This transcription, too, is valid, and subject to an appropriate font it will display correctly. However, since code points in the Private Use Area can change we strongly recommend using entities. Entities can easily be reinterpreted, for example in the case of a character which are accepted by Unicode. If this happened to LATIN SMALL LIGATURE AO, the only change to be effected would be a change in the entity list in the Menota scheme, from:

```
<!ENTITY aolig "&#xEF93;"> <!-- LATIN SMALL LIGATURE AO -->
```

to, say,

```
<!ENTITY aolig "&#x2C7A;"> <!-- LATIN SMALL LIGATURE AO -->
```

In the encoded text, the entity ‘&aolig;’ could be retained and the display would still be correct.

5.2 Naming and referring to characters


Entities are needed at the bottom level, as it were, in an XML transcription of a text. This is parallel to the source code of a typical HTML file, which can be inspected in most HTML editors and browsers, but is usually not shown. Although a number of characters will have to be referred to with entities, it is important to note that the transcriber does not have to type in entities when s/he is transcribing a manuscript or doing proof reading. With appropriate software and fonts the transcription can be displayed on screen and printed out with all (or at least most) entities shown as readable and recognizable characters.

The characters a-z / A-Z are seen as base line characters, i.e. characters occupying a separate position on the base line of a primary source (typically a manuscript) and transcribed one by one in the order they stand. In addition to the characters a-z / A-Z there are a number of ligatures, i.e. combination of two (or in principle more) characters making up a new base line character, such as ‘æ’. There are also a number of variant base characters, e.g. a round form of ‘r’ (r rotunda), or a tall form of ‘s’, and there is even a whole set of small capitals to be reckoned with, especially in Old Icelandic script. Furthermore, the base line characters can be modified by a number of diacritics (accents, dots, hooks, strokes etc.), so that the theoretical number of combinations for any character is very high.


For practical reasons, all characters needed for the transcription of medieval Nordic manuscripts should be given descriptive names. We have found the naming scheme in the Unicode Standard to be a good model. There are, however, a considerable number of characters which so far have not been defined and described in Unicode. For these characters we must resort to the Private Use Area, and we need rules for the naming of such characters.

Descriptive names have basically the same syntax as in rules (6) and (7) in [ch. 2.2.1](#) above. The following examples refer to characters in the official Unicode Standard and thus serve to illustrate the naming scheme.


1. Base line character.

Glyph	Descriptive name
	LATIN SMALL LETTER A


2. Modification of a base line character within its x-height.

Glyph	Descriptive name
	LATIN SMALL LETTER O WITH STROKE


3. Modification of a base line character touching the base character outside its x-height. As explained in [ch. 2.2.2](#) above, this character can be encoded and described in two equivalent ways.

Glyph	Descriptive name
	LATIN SMALL LETTER O + COMBINING OGONEK = LATIN SMALL LETTER O WITH OGONEK

4. Modification of a base line character not touching the base line character itself. Also this character can be encoded and described in two equivalent ways.

Glyph	Descriptive name
	LATIN SMALL LETTER O WITH STROKE + COMBINING ACUTE ACCENT = LATIN SMALL LETTER O WITH STROKE AND ACUTE

5. More than one modification. Here, there are in fact three equivalent ways of encoding and describing this character.

Glyph	Descriptive name
	LATIN SMALL LETTER O + COMBINING OGONEK + COMBINING ACUTE ACCENT = LATIN SMALL LETTER O WITH OGONEK + COMBINING ACUTE ACCENT = LATIN SMALL LETTER O WITH OGONEK AND ACUTE

In general, we believe that the number of variants should be minimised, whether of base characters or of diacritics. There is, for example, only one base line character ‘a’, although this letter may have various forms in the manuscripts, i.e. ‘single-storeyed’ (with a neck) or ‘double-storeyed’ (closed without a neck). We regard this type of variation as paleographical, and suggest that it is not encoded, but that it is described elsewhere, e.g. in the TEI header or in the front matter of the electronic edition.

We would like to stress that the characters in this chapter should not be taken as an instruction of minimal and necessary distinctions to be made by the transcriber. We have defined two types of ‘s’, a low (or round) one and a long one. This does not mean that the transcriber should use both characters in the encoding of whichever manuscript exhibiting them, only that if s/he wishes to make the distinction, we suggest how that can be done.

5.2.1 Glyphs

Glyphs are the typical shape of a character. In this chapter, they are displayed in the font Andron by Andreas Stötzner (Leipzig). The regular version of this font can be downloaded from the [MUFI font page](#).

5.2.2 Entity names

All characters outside the range a-z / A-Z are referred to with entity names placed within the delimiters ‘&’ and ‘;’. We recommend that entities as far as possible conform to the standard ISO entity sets. However, the ISO set only covers a minor selection of the entites we believe are necessary for the full transcription of medieval Nordic manuscripts. This

chapter thus discusses a number of additional characters with accompanying entities. We have tried to adhere to the inventory and syntax of ISO entities. For a summary of the entity naming scheme, please refer to [ch. 5.6](#) below.

5.2.3 Unicode values

We have supplied code points from *Unicode 5.0* for all characters (or parts of characters) defined in this standard. For the remaining characters we have defined code points in the Private Use Area. These are shown in bold type (and dark blue). The [MUFI character recommendation](#) v. 2.0 contains Unicode values for a large selection of characters.

5.2.4 Descriptive names

Each character is described according to the naming scheme in Unicode, as explained above. We also suggest descriptive names for those characters not included in the Unicode standard.



5.3 Base line characters

Base line characters are unmodified characters occupying a separate position on the base line, i.e. characters which are not clearly modified by diacritical marks or being part of a ligature.

5.3.1 Base line characters in the Modern English alphabet

These characters are described in ISO 646 and are found on the keyboard of virtually any Western computer. They are identical to US ASCII positions 32-126 and are often referred to as Basic Latin. Characters in Basic Latin are encoded without use of entity references.

Unicode 5.0 defines these characters as belonging to the range [Basic Latin](#) (positions 0020-007E).

Glyph	Letter	Unicode	Descriptive name
	a	0061	LATIN SMALL LETTER A
	A	0041	LATIN CAPITAL LETTER A

etc.

Note that the distinction between minuscule (lowercase) and majuscule (uppercase) characters is an inherent trait of the coding scheme; it is not shown by entity names such as ‘&amin;’ for ‘a’ and ‘&amaj;’ for ‘A’. However, when it comes to the question of small capitals and enlarged minuscules it will be necessary to introduce entity names, as discussed in [ch. 5.2.3](#) and [ch. 5.2.4](#) below.

5.3.2 Base line characters in the Modern Icelandic alphabet

Modern Icelandic has two characters for dental fricatives, ‘þ’ (thorn) and ‘ð’ (eth). In ISO 8859-1 they are referred to with the entity names ‘þ’ and ‘ð’, also adopted here.

Unicode 5.0 defines ‘þ’ (thorn) and ‘ð’ (eth) in the range [Latin-1 Supplement](#).

Glyph	Entity	Unicode	Descriptive name
ð	ð	00F0	LATIN SMALL LETTER ETH
Ð	Ð	00D0	LATIN CAPITAL LETTER ETH
þ	þ	00FE	LATIN SMALL LETTER THORN
Þ	Þ	00DE	LATIN CAPITAL LETTER THORN

In addition to ‘þ’ and ‘ð’, Modern Icelandic has seven vowels with diacritical marks, ‘á’, ‘é’, ‘í’, ‘ó’, ‘ú’, ‘ý’ and ‘ö’, and one ligature, ‘æ’. These will be treated as modified characters and discussed below.

5.3.3 Small capitals

Small capitals have the same form as majuscules (capital letters), but are usually drawn with the same height as a minuscule (small letter) such as ‘x’. Small capitals were used in Old Icelandic to denote geminates, i.e. long consonants, or they were used ornamentally (often so in Old Norwegian). The letters ‘B’, ‘D’, ‘G’, ‘M’, ‘N’, ‘R’, ‘S’ and ‘T’ were often used as geminates, while these and other letters might also be used as ornaments in the whole or in parts of highlighted words. Some of the small capitals, e.g. ‘O’ and ‘C’, are difficult to distinguish from minuscule letters. We suggest that small capitals receive the suffix ‘scap’ (for ‘small capital’) in the entity name.

Unicode 5.0 has defined nine small capitals in the [IPA Extensions](#) range, sc. ‘B’, ‘G’, ‘H’, ‘I’, ‘L’, ‘N’, ‘#’, ‘R’ and ‘Y’, and sixteen in the [Phonetic Extensions](#) range, sc. ‘A’, ‘Æ’, ‘C’, ‘D’, ‘ETH’, ‘E’, ‘J’, ‘K’, ‘M’, ‘O’, ‘P’, ‘T’, ‘U’, ‘V’, ‘W’ and ‘Z’. For the remaining small capitals we will have to resort to the Private Use Area, i.e. ‘F’, ‘Q’, ‘S’, ‘THORN’ and ‘X’. Cf. [Appendix A](#) for reference to the complete overview in the [MUFI character recommendation](#).

Glyph	Entity	Unicode	Descriptive name
G	&gscap;	0262	LATIN LETTER SMALL CAPITAL G
M	&mscap;	1D0D	LATIN LETTER SMALL CAPITAL M



etc.

We recommend that small capitals are transcribed as such, irrespective of whether they are being used for geminates or for ornamental purposes. Cf. [ch. 6.3.10](#).

5.3.4 Enlarged minuscules

Some scholars believe that enlarged minuscules should be transcribed as separate characters. The traditional view is to interpret these characters as variants of capitals (majuscles) and encode them as such. There are comparatively few characters which appear as enlarged minuscules, and it is sometimes difficult to decide whether a minuscule character is enlarged or not. We recommend that enlarged minuscules are transcribed as capitals in cases where it seems obvious that they function as a capital and as ordinary minuscules elsewhere. If, however, the transcriber wishes to make a distinction between capitals and enlarged minuscules, we recommend the suffix ‘enl’ (for ‘enlarged’) in the entity name.

Unicode 5.0 does not recognise enlarged minuscules as separate characters. A small selection of enlarged minuscules has been included in the Private Use Area, e.g. ‘a’ and ‘e’. Cf. [Appendix A](#) for reference to the complete overview in the [MUFI character recommendation](#).



Glyph	Entity	Unicode	Descriptive name
	&aenl;	EEE0	LATIN ENLARGED LETTER SMALL A
	&eenl;	EEE6	LATIN ENLARGED LETTER SMALL E

etc.

5.3.5 Insular characters

A few characters have distinct Insular forms, e.g. ‘r’, ‘f’ and ‘v’. These characters are sometimes transcribed as separate characters, as opposed to their Carolingian counterparts. We suggest using the suffix ‘ins’ (for ‘Insular’).

Unicode 5.0 does not recognise Insular characters as separate characters, with the exceptions of ‘g’ and ‘w’ (wynn) in [Latin Extended-B](#). A few Insular characters have been included in the Private Use Area, e.g. ‘f’ and ‘v’.

Glyph	Entity	Unicode	Descriptive name
	&fins;	F10D	LATIN SMALL LETTER INSULAR F
	&vins;	F211	LATIN SMALL LETTER INSULAR V

etc.

Insular ‘g’ is to our knowledge not found in medieval Nordic manuscripts.

As a rule, characters should be given identical names across various scripts (Carolingian, Insular, Gothic etc.). However, when clearly identifiable letter forms from one script appear within the context of another, as is the case with some Insular letter forms in Nordic Carolingian script, they may be singled out by the transcriber, if s/he wishes to do so.

5.3.6 Uncials

A few characters may appear with a typical Uncial form, especially ‘e’ and ‘m’. These characters are sometimes transcribed as separate characters, as is the case with Insular letter forms. We suggest using the suffix ‘unc’ in the entity name. Note that some Uncial forms may also be characterised as round, cf. 5.3.8 below.

Unicode 5.0 does not recognise Uncial characters as separate characters. A small selection of Uncial characters has been included in the Private Use Area, e.g. ‘e’, ‘k’ and ‘m’. Cf. [Appendix A](#) for reference to the complete overview in the [MUFI character recommendation](#).


Glyph	Entity	Unicode	Descriptive name
	&eunc;	F218	LATIN SMALL LETTER E UNCIAL FORM
	&kunc;	F208	LATIN SMALL LETTER K UNCIAL FORM
	&munc;	F225	LATIN SMALL LETTER M UNCIAL FORM

etc.

5.3.7 Runes

Runes are normally not used in conjunction with the Latin alphabet, but when they appear in isolated instances – e.g. in *The third grammatical treatise* – they should be transcribed with appropriate entity names. We suggest using the suffix ‘Medrun’ (for ‘Medieval runes’).

Unicode 5.0 has defined a selection of 81 runes from the Older and Younger Futhark in the [Runic](#) range. Note that the descriptive names given below are those chosen by Unicode.

Glyph	Entity	Unicode	Descriptive name
	&fMedrun;	16A0	RUNIC LETTER FEHU FEOH FE F

Glyph	Entity	Unicode	Descriptive name
ſ	&mMedrun;	16D8	RUNIC LETTER LONG-BRANCH-MADR M

etc.

Note that the runes ‘m’ and ‘f’ may also be used as abbreviation signs, cf. [ch. 6.3.6-7](#).

5.3.8 Other variants of base line characters

Some base line characters have commonly recognised variants. In general, we recommend that variants, e.g. ‘single storeyed a’ and ‘two storeyd a’, are not transcribed as separate entities. In many cases it is difficult to decide which of the variants to choose from. However, there are a few variants which are very distinctive and often recognised in transcriptions. This applies to ‘tall s’ and ‘round r’, for which we suggest the suffixes ‘tall’ and ‘rot’ (for ‘rotunda’) respectively.

Unicode 5.0 recognises ‘long s’ as part of the [Latin Extended-A](#) range, but ‘round r’ is not recognised. This has been allocated to code point **F20E** in the Private Use Area.

Glyph	Entity	Unicode	Descriptive name
ſ	&slong;	017F	LATIN SMALL LETTER LONG S
ð	&drot;	F109	LATIN SMALL LETTER D ROTUNDA
ʀ	&rrot;	F20E	LATIN SMALL LETTER R ROTUNDA
ƿ	&trot;	F129	LATIN SMALL LETTER T ROTUNDA

etc.

5.4 Ligatures

Ligatures are two base line characters which are joined so that they form a new, composite base line character. Some consist of two identical characters, e.g. ‘a+a’, others of different characters, e.g. ‘a+v’. Ligatures may be used to denote length, ‘a+a’, diphthong, ‘a+v’, or a distinct vowel quality, often mutation (Umlaut), ‘a+v’. A well known example is the ligature ‘æ’, formed of ‘a’ and ‘e’, encoded as ‘æ’ in ISO 8879. In analogy with this usage we suggest that ligatures receive the suffix ‘lig’ following those base line characters which make up the ligature.

Unicode 5.0 does not recognise ligatures in the Latin alphabet as base characters. The only exceptions are ‘æ’, ‘#’ and ‘ij’ (not used in Nordic). For ‘æ’ see the Unicode range [Latin-1 Supplement](#), and for ‘#’ [Latin Extended-A](#). Other ligatures must be defined in the Private Use Area. Cf. [Appendix A](#) for reference to the complete overview in the [MUFI character recommendation](#).

Glyph	Entity	Unicode	Descriptive name
æ	&aalig;	EF91	LATIN SMALL LIGATURE AA
av	&avlig;	EF97	LATIN SMALL LIGATURE AV

etc.

We recommend that only ligatures with a distinctive value should be given an entity name of their own, i.e. only those ligatures which possibly reflect a phonological opposition. We regard ligatures which are motivated by graphic economy as sporadic ligatures and recommend that they should be transcribed as separate characters. To this group belong ligatures such as ‘b+b’, ‘p+p’ etc. Especially in late Gothic script there are many examples of junctures (fusion of bows) which can be interpreted as ligatures, but which in our opinion should be encoded as individual characters.

If a transcriber wishes to transcribe sporadic ligatures as ligatures, we suggest using the element `<seg>` with the attribute `@type="ligature"`, e.g.

Glyph	Encoding
pp	<code><seg type="ligature">pp</seg></code>



5.5 Modified characters

Modified characters are base line characters with diacritical marks. They are described according to rule (4) in [ch. 2.2.1](#). If there is more than one modification, they are listed in the sequence specified in rule (6).

5.5.1 Strokes (slashes)

The character ‘ø’ is still being used in Modern Danish and Norwegian, and is encoded as ‘ø’ in ISO 8879. In some manuscripts the stroke may be horizontal and in others diagonal, but in general we do not believe it is relevant to distinguish between variant strokes.

Unicode 5.0 has defined ‘ø’ as part of the [Latin-1 Supplement](#) range.



Glyph	Entity	Unicode	Descriptive name
	<code>&oslash;</code>	00F8	LATIN SMALL LETTER O WITH STROKE
	<code>&Oslash;</code>	00D8	LATIN CAPITAL LETTER O WITH STROKE

etc.

5.5.2 Hooks and loops



A few vowels, especially ‘o’ and ‘e’, may have a hook. The latter combination, ‘e caudata’, is common in Latin manuscripts, in which the letter form alternates with the ligature ‘æ’. The hook may be placed below or above the base line character, facing either to the right or to the left. Of these combinations, the distinction between left- and right-turning hooks may simply be accidental. The two ‘canonical’ forms are the hook below to the right and the hook above to the left. We recommend using ‘ogon’ for the hook below and ‘curl’ for the hook above (since ‘hook’ possibly is more ambiguous).

Unicode 5.0 recognises ‘a’ and ‘e’ with hooks in the range [Latin Extended-A](#), and ‘o’ with hook in [Latin Extended-B](#). In Unicode, the hook is referred to as ‘ogonek’, a Polish word for ‘little tail’. The ogonek is also defined as a combining character, 0328 in the range [Combining Diacritical Marks](#). The hook above may be identified with the tone mark in Vietnamese, 0309 in the range [Combining Diacritical Marks](#). This mark, however, has a slightly different form (comparable to the recognised distinction between the cedilla and the ogonek). For this reason, we suggest using a separate code point in the Private Use Area, **F1C4**.

Glyph	Entity	Unicode	Descriptive name
	<code>&oogon;</code> = o + <code>&combogon;</code>	01EB = 006F + 0328	LATIN SMALL LETTER O WITH OGONEK = LATIN SMALL LETTER O + COMBINING OGONEK
	<code>&ocurl;</code> = o + <code>&combcurl;</code>	E7D3 = 006F + F1C4	LATIN SMALL LETTER O WITH CURL = LATIN SMALL LETTER O + COMBINING CURL

Loops are in most cases reduced forms of ‘a’ or ‘o’ and can thus be interpreted as ligatures.

Unicode 5.0 does not recognise loops, either as separate characters or as combining diacritical marks.

Glyph	Entity	Unicode	Descriptive name
	<code>&oloop;</code>	F20D	LATIN SMALL LIGATURE OE WITH LOOP
	<code>&aoligred;</code>	F206	LATIN SMALL LIGATURE AO NECKLESS

5.5.3 Single and double accents

Single and double acute accents are quite common in Nordic script. A single acute accent is encoded with the suffix ‘acute’ in ISO 8879, e.g. ‘á’, while double acute is encoded with the suffix ‘dblac’. This usage is adopted here.

Unicode 5.0 defines ‘a’, ‘e’, ‘i’, ‘o’, ‘u’ and ‘y’ with acute accents in the [Latin-1 Supplement](#) range, and ‘æ’ and ‘ø’ in the [Latin Extended-B](#) range. The vowels ‘o’ and ‘u’ are defined with double acute accents in the [Latin Extended-A](#) range. Other accented characters must be encoded as a combination of a base line character and 0301 COMBINING ACUTE ACCENT or 030B COMBINING DOUBLE ACUTE ACCENT from the range [Combining Diacritical Marks](#). As explained in [ch. 2.2](#) this ‘decomposed’ encoding can also be used for the precomposed vowels mentioned above.

Glyph	Entity	Unicode	Descriptive name
	<code>&aacute;</code> = a + <code>&combacute;</code>	00E1 = 0061 + 0301	LATIN SMALL LETTER A WITH ACUTE = LATIN SMALL LETTER A + COMBINING ACUTE ACCENT
	<code>&adblac;</code> = a + <code>&comdblac;</code>	E425 = 0061 + 030B	LATIN SMALL LETTER A WITH DOUBLE ACUTE = LATIN SMALL LETTER A + COMBINING DOUBLE ACUTE ACCENT
	<code>&aaligacute;</code> = &aalig; + <code>&combacute;</code>	EFE1 = EF91 + 0301	LATIN SMALL LIGATURE AA WITH ACUTE = LATIN SMALL LIGATURE AA + COMBINING ACUTE ACCENT
	<code>&aaligdblac;</code> = &aalig; + <code>&comdblac;</code>	EFEB = EF91 + 0301	LATIN SMALL LIGATURE AA WITH DOUBLE ACUTE = LATIN SMALL LIGATURE AA + COMBINING DOUBLE ACUTE ACCENT

Double acute accent sometimes resembles a circumflex, ‘^’, cf. [Seip 1954](#), p. 145.

Grave accent sporadically appears in comparatively young Icelandic manuscripts, especially ‘è’, while double grave accent to our knowledge is not found in medieval Nordic script at all. If necessary, we suggest using the suffix ‘grave’, e.g. ‘è’, for the single grave accent.

5.5.4 Single and double dots

Single and double dots are quite common in Old Norse script. Single dots appear over vowels as well as consonants, double dots usually only above vowels. In ISO 8879 the suffixes ‘dot’ and ‘uml’ (for ‘Umlaut’) refer to single and double dots respectively. This usage is adopted here (although double dots in no way restricted to the original mutated vowels).

Unicode 5.0 defines a number of consonants with a single dot above, sc. ‘b’, ‘d’, ‘f’, ‘h’, ‘m’, ‘n’, ‘p’, ‘r’, ‘s’, ‘t’, ‘w’, ‘x’ and ‘long s’, and also the vowel ‘y’, all in the [Latin Extended Additional](#) range. Other dotted characters must be encoded as a combination of a base line character and 0307 COMBINING DOT ABOVE or 0308 COMBINING DIAERESIS from the range [Combining Diacritical Marks](#). As is the case with accents, ‘decomposed’ encoding can also be used for the precomposed characters mentioned here.

Glyph	Entity	Unicode	Descriptive name
ȳ	&ydot; = y + &combdot;	1E8F = 0079 + 0307	LATIN SMALL LETTER Y WITH DOT ABOVE = LATIN SMALL LETTER Y + COMBINING DOT ABOVE
ö	ö = o + &combuml;	00F6 = 006F + 0308	LATIN SMALL LETTER O WITH DOUBLE DOT ABOVE = LATIN SMALL LETTER O + COMBINING DIAERESIS

Single dots also appear over a number of consonants:

Glyph	Entity	Unicode	Descriptive name
ḱ	&kdot; = k + &combdot;	E568 = 006B + 0307	LATIN SMALL LETTER K WITH DOT ABOVE = LATIN SMALL LETTER K + COMBINING DOT ABOVE
ḡ	&gscapdot; = &gscap; + &combdot;	EF20 = 0262 + 0307	LATIN LETTER SMALL CAPITAL G WITH DOT ABOVE = LATIN LETTER SMALL CAPITAL G + COMBINING DOT ABOVE

Single dots above can be seen as a type of abbreviation, since the dot usually signifies gemination of the characters it is placed above. Cf. [ch. 6.4.8](#).




5.6 Complex characters

The discussion in [ch. 5.3-5.5](#) has shown that entity names are built up in a strict sequence with a limited number of possible values. The syntax and inventory is shown in the table below. Note that not all slots need to be filled in; in most cases only one or two slots are used.

Base line character	Main type	Variant	Ligature	Fixed modification	Loose modification
a A	comb enl ins run scap unc	long rot	lig ligred	ogon slash	acute dblac dot curl grave uml

Please note that if there is a conflict between the standard ISO entities and the syntax suggested here, ISO entites should be preferred.



On the basis of this table we can name and describe a number of complex characters (not necessarily occuring in medieval Nordic script). Some examples:

Glyph	Entity name	Descriptive name
	æogon;	LATIN SMALL LIGATURE AE WITH OGONEK
	øogonacute;	LATIN SMALL LETTER O WITH STROKE AND OGONEK AND ACUTE
	æogonuml;	LATIN SMALL LIGATURE AE WITH OGONEK AND DIAERESIS

5.7 Punctuation marks

The punctuation marks in medieval Nordic script are basically the same as in the Modern European languages, but their use was less consistent, and many manuscripts only used a single mark, the dot. There was also some special types of punctuation marks.

Unicode 5.0 has the marks in the table below in the ranges [Basic Latin](#) and [Latin-1 Supplement](#), with the exception of the inverted semicolon, the pause mark and the triangular dots.

Glyph	Character	Unicode	Descriptive name
	.	002E	FULL STOP
	·	00B7	MIDDLE DOT

Glyph	Character	Unicode	Descriptive name
,	,	002C	COMMA
:	:	003A	COLON
;	;	003B	SEMICOLON
⁂	&punctelev;	F161	PUNCTUS ELEVATUS
?	?	003F	QUESTION MARK
⸔	&punctinterlemn;	F1F1	PUNCTUS INTERROGATIVUS LEMNISKATE FORM
-	-	002D	HYPHEN
/	/	002F	SOLIDUS
⸌	&bidotscomposit;	F1F2	TWO DOTS OVER COMMA POSITURA
⸎	&tridotsupw;	F1EF	ONE DOT OVER TWO DOTS PUNCTUATION

5.8 List of characters

An extensive list of characters (including punctuation and abbreviation marks) is found in the MUFI character recommendation, cf. [Appendix A](#) below.

Chapter 6. Abbreviations: typology and encoding

6.1 Introduction

Abbreviations are a common feature of medieval manuscripts. In the medieval Nordic tradition, abbreviations were used most frequently in Norwegian and Icelandic manuscripts, and particularly in the latter. In some Icelandic manuscripts as many as a third of the words may be abbreviated, some of them with several abbreviation marks. The system of abbreviations was inherited from English and Continental practice, but the adoption of this system also meant that the usage of some abbreviation marks was extended and it led to the development of some new types.

The encoding of abbreviations is discussed in the [TEI P5 Guidelines](#) in [ch. 11](#), particularly [ch. 11.3.2](#). As of this version, four elements are defined:

Element	Contents
<code><abbr></code>	(abbreviation) contains an abbreviated word
<code><am></code>	(abbreviation marker) contains the actual abbreviation
<code><expan></code>	(expansion) contains an expanded word
<code><ex></code>	(expansion marker) contains the actual expansion

TEI P5 recommends that abbreviations spanning a whole word is encoded with the `<abbr>` element, while the actual abbreviation can be encoded with the `<am>` element, e.g.

```
<abbr>xpc</abbr>
han<am>&bar; </am>
```

In the first line of this example, the sequence ‘xpc’ is an abbreviation for ‘christus’. This is a *nomen sacrum* using originally Greek characters and should therefore be interpreted as a special abbreviation character (brevigraph). The abbreviation in the second line is by far the most common one in Medieval Nordic sources. Here, a part of the word, an ‘n’, has been abbreviated by way of putting a horizontal bar above the preceding character. Even if the element `<abbr>` can be used for the first type and `<am>` for the second, we suggest that the element `<am>` should be used in both cases. An abbreviation of the whole word can simply be seen as a borderline case of an abbreviations of a word part.

A similar distinction is drawn in TEI P5 between the `<expan>` element, which contains an expansion of a whole word, and the `<ex>` element, containing the expanded part of the word, e.g.

```
<expan>christus</expan>
han<ex>n</ex>
```

In the first line of this example, the abbreviation ‘xpc’ has been expanded as ‘christus’, meaning that there are no overlapping characters between the abbreviation (the brevigraph) and the expansion. In the second line, the horizontal bar has been expanded

as ‘n’. We recommend using the `<ex>` element in both cases, for similar reasons as for the use of the `<am>` element.

In a multi-level transcription, the `<am>` element typically belongs to the *fac*s level, while the `<ex>` element belongs to the *dipl* level. The *norm* level usually have none, e.g.

```
<w>
  <choice>
    <me:fac>han<am>&bar;</am></me:fac>
    <me:dipl>han<ex>n</ex></me:dipl>
    <me:norm>hann</me:norm>
  </choice>
</w>
```

The `<am>` element may have a `@me:type` attribute specifying what kind of abbreviation it is. The same applies to the `<ex>` element. We have not given examples of these attributes in the present chapter, but users may refer to the typology in [ch. 6.2](#) below if they would like to make a more detailed encoding.

Element	Contents
<code><am></code>	contains the actual abbreviation
<code>@me:type</code>	specifies the type of abbreviation (optional)
<code><ex></code>	contains the actual expansion
<code>@me:type</code>	specifies the type of expansion (optional)

In this chapter, we shall give a typology of abbreviation and then exemplify a number of cases.

6.2 Typology

Abbreviations are usually divided into four categories (see e.g. [Hreinn Benediktsson 1965](#), p. 85 and, for a more detailed classification, [Kristian Kålund 1907](#), pp. viii-x):

- (1) **Suspensions.** The first part of the word, often the initial letter only, is written out, followed by a dot or similar mark. The plural may be represented by a doubling of the initial letter, e.g. ‘ss.’ = synir (sons).
- (2) **Contractions.** Some letters are left out, but the initial and final letters are written out, often one or more of the intermediate as well. The abbreviation is often indicated with a horizontal bar above the word.
- (3) **Interlinear marks.** The interlinear abbreviation is usually a vowel representing either ‘r’ or ‘v’ + the vowel itself or a consonant representing ‘a’ + the consonant itself.
- (4) **Special signs (brevigraphs).** These signs are usually placed on the base line and are thus akin to ordinary letters. The Tironian *notae* belong to this category.

The typology in [ch. 6.3](#) below takes as its point of departure the location of the abbreviations. The main distinction is drawn between abbreviation signs placed on the base line and those placed above (or through or below) a base line character. We suggest that letter-sized characters on the base line are referred to as **signs**, while combining abbreviation marks (above, through or below another character) are referred to as *marks*.

For the sake of simplicity, however, we shall refer to both categories as **marks** in this chapter.

6.2.1 Glyphs

Glyphs are displayed in the Andron font by Andreas Stötzner (Leipzig). The regular version of this font can be downloaded from the [MUFI font page](#).

Since abbreviation marks typically appear as parts of words and are frequently associated with a base line character we have chosen to illustrate each mark within the context of a whole word.

6.2.2 Entity names

All abbreviations are referred to with entity names, with the exception of full stop, ‘.’, and colon, ‘:’. Entity names are placed within the delimiters ‘&’ and ‘;’, and we have tried to give as short and mnemonic names as possible. As a rule, we have based the entity name on the typical expansion of the abbreviation. Thus, the cross mark which is an abbreviation for ‘kross’ is given the entity name ‘✗’.

We aim at synchronizing our use of entities with those recommended by ISO, but since there presently are no abbreviation entities in ISO, we are left to our own devices in this chapter.

6.2.3 Unicode values

Unicode 5.0 has only defined a handful of abbreviation characters and only a few of interest for our use. The great majority of abbreviation characters must therefore be defined as code values in the Private Use Area. The only exceptions are the full stop, colon and semicolon, which are part of the range [Basic Latin](#) in Unicode, and the Tironian sign for *et*, in the range [General Punctuation](#).

For a complete list of suggested Unicode values, see [Appendix A](#) below.

6.2.4 Descriptive names

As is the case with ordinary characters (cf. [ch. 5](#)) we adhere to the naming scheme in Unicode. Since *Unicode 5.0* only defines one abbreviation mark in the Latin alphabet, the TIRONIAN SIGN ET in the range [General Punctuation](#), and only one in each of the Armenian, Syriac, Devanagari, Thai and Khmer alphabets, we do not have completely clear examples of descriptive names. We suggest ABBREVIATION SIGN ‘000’ as a general name for abbreviations occupying a separate position on the base line, and COMBINING ABBREVIATION MARK ‘000’ for those typically placed above, through or below a base line character.

6.3 Abbreviation marks on the base line

Abbreviation marks on the base line behave as any other character. The typology of these abbreviation marks is discussed and exemplified below.

6.3.1 The ‘et’ mark

The Tironian *nota* resembling the number ‘7’ (or the character ‘z’ with or without a crossbar) is often used for the conjunction ‘ok’ / ‘oc’ (in Latin ‘et’). We recommend using the entity name ‘&et;’, reflecting the Latin origin of the abbreviation.

In *Unicode 5.0* this character is located at 204A in the range [General Punctuation](#).

Abbreviated form	Expanded form	Encoding	Abbreviation mark code point
7	(et)	<am>&et;</am>	204A

There are two major variants of this sign. If the transcriber wishes to make a distinction between these, we suggest using ‘&et;’ for the sign without a crossbar and ‘&etslash;’ for the sign with a crossbar. The code point for the latter is F158.

6.3.2 The ‘ed’ mark

The semicolon was used for ‘e’ + dental consonant, often in the preposition ‘með’. We recommend ‘&sem;’ as entity name.

In *Unicode 5.0* the semicolon is located at 003B in the range [Basic Latin](#). When the semicolon is used as a punctuation mark, it should be transcribed as such, i.e. simply as ‘;’. When it is used as an abbreviation mark we recommend that it is transcribed with an entity, ‘&sem;’. Note that there is another form of this abbreviation mark, looking like the number ‘3’. This is included in the [MUFI character recommendation](#) v. 2.0 at code point F155 and can be encoded with the entity ‘&etfin;’.

Abbreviated form	Expanded form	Encoding	Abbreviation mark code point
m;	m(eð)	m<am>&sem;</am>	F1AC

6.3.3 The ‘con’ mark


A sign resembling a backwards ‘c’ was often used for ‘con’ in Latin and ‘kon’ in Nordic words. This ‘con’ mark is similar to 0254 LATIN SMALL LETTER OPEN O in the range [IPA Extensions](#) of *Unicode 5.0* and may be identified with this character.

Abbreviated form	Expanded form	Encoding	Abbreviation mark code point
ɔa	(kon)a	<am>&oopen;</am>a	0254

See the [MUFI character recommendation](#) v. 2.0 for other variants of the ‘con’ mark (descending and with a dot).

6.3.4 The ‘rum’ mark


The sequence ‘rum’ was often abbreviated with a character resembling a small version of the number 4 (in fact, it is the round ‘r’ with a stroke across its tail). We recommend the entity name ‘&rum;’ and a separate code point in the Private Use Area.

Abbreviated form	Expanded form	Encoding	Abbreviation mark code point
	eo(rum)	eo<am>&rum;</am>	F154

6.3.5 The cross mark

The word ‘kross’ was sometimes abbreviated with the cross symbol, which we suggest calling ‘✗’.


This ‘kross’ mark can be identified with 271D LATIN CROSS in the range [Dingbats](#) of *Unicode 5.0*.

Abbreviated form	Expanded form	Encoding	Abbreviation mark code point
	(kross)	<am>✗</am>	271D

6.3.6 The ‘m’ rune

The runic character for ‘m’ was sometimes used for the word ‘maðr’ (including case forms with the stem ‘mann-’). We recommend the entity name ‘&mMedrun;’, as introduced in [ch. 5.3.7](#).

Unicode 5.0 has defined a selection of 81 runes from the Older and Younger Futhark in the [Runic](#) range. This range includes the ‘m’ rune.


Abbreviated form	Expanded form	Encoding	Abbreviation mark code point
	(maðr)	<am>&mMedrun;</am>	16D8

The runic character may appear with interlinear marks (‘a’, ‘i’, ‘e’, ‘n’, ‘z’) for various inflected forms of the word ‘maðr’, e.g. ‘manna’, ‘manni’/‘manne’, ‘mann’, ‘mannz’. The encoding of this type is discussed in [ch. 6.4.7](#) below.

6.3.7 The ‘f’ rune

The runic character for ‘f’ was sometimes used for the word ‘fé’. In analogy with the use of the ‘m’ rune, we suggest the entity name ‘&fMedrun;’.

The ‘f’ rune is included in the [Runic](#) range of *Unicode 5.0*.

Abbreviated form	Expanded form	Encoding	Abbreviation mark code point
	(fé)	<am>&fMedrun;</am>	16A0

6.3.8 Dot (full stop)

Dots were often used as abbreviation marks, typically for suspensions, e.g. ‘s.’ for ‘sonr’ (or ‘segja’, ‘svara’). They may sometimes appear on both sides of the abbreviated word, ‘.s.’. We recommend that the dot is transcribed in the same manner as a full stop, i.e. with the ‘.’ mark in [Basic Latin](#). Thus, no entity name is called for.

Abbreviated form	Expanded form	Encoding	Abbreviation mark code point
.s.	s(onr)	<am>.s.</am>	002E
.kgr.	k(onun)gr	<am>.kgr.</am>	002E

If the transcriber wishes to distinguish between the dot used as an abbreviation mark and the dot used as a punctuation mark, we suggest that the entity name ‘.’ could be used in the former case and ‘.’ in the latter. However, we believe that there will arise a number of cases where it is difficult to decide whether the dot in the manuscript is a mark of abbreviation, punctuation or both, e.g. when a suspended word is the last word in a sentence. We therefore believe it is better to accept that the full stop is an ambivalent mark, as is also (although to a much lesser extent) the case with the colon and the runic characters ‘f’ and ‘m’. When the encoder believes that the full stop is an abbreviation mark that should be indicated simply by using the <am> element, as shown here.

6.3.9 Colon

The colon is sometimes, though not often, used as a mark of suspension, in the same manner as the dot (full stop). In analogy with the encoding of dots we suggest transcribing the colon simply as a colon, i.e. without using an entity.

Abbreviated form	Expanded form	Encoding	Abbreviation mark code point
Rognv:	Rognv(aldr)	Rognv<am>:</am>	003A

6.3.10 Small capitals

In Old Icelandic, small capitals were used to denote geminated (long) consonants or they were simply used ornamentally (especially in Old Norwegian). In [ch. 5.3.3](#) above we recommended that they were encoded as entities in both cases. The use of small capitals can be seen as a form of abbreviation, but there will be a number of cases where the usage is open to interpretation. We recommend that the transcriber copies the text as it is, transcribing a small capital as a small capital irrespective of whether it is being used to denote gemination or as an ornament. Thus, exactly the same entities will be used here as introduced in [ch. 5.3.3](#).

Abbreviated form	Expanded form	Encoding	Abbreviation mark code point
heRa	heRa	he&rscap;a	0280

For the encoding of small capitals with dot above, please see [ch. 6.4.8](#) below.

6.4 Combining abbreviation marks

The majority of abbreviation marks are placed above, through or below a base line character. It could be argued that they really refer to the whole word, but from an analytical point of view we recommend that they are encoded immediately after the base line character to which they seem most closely associated. Cf. the rules in [ch. 2.2.1](#).

It is sometimes difficult to decide whether a sign is placed on the base line or above another base line character. For example, the ‘us’ mark (cf. [ch. 6.4.3](#) below) may sometimes occupy a position of its own, although slightly raised above the base line. The classification in this chapter is based on what we believe are the prototypical positions of the abbreviation marks.

6.4.1 Horizontal bar

The horizontal bar is from a historical point of view the earliest form of an abbreviation mark and it is also the most ambiguous type. It is commonly used for ‘m’ or ‘n’ and is often referred to as a ‘nasal stroke’, but it is also used in a number of other contexts, as a mark of suspension or contraction. We recommend using the same entity name in all instances, ‘&bar;’. The unmarked position of the bar is above the immediately preceding character.


This horizontal bar is partially similar to 0304 COMBINING MACRON and 0305 COMBINING OVERLINE in the range [Combining Diacritical Marks](#) of *Unicode 5.0*, and may be identified with the latter.

Abbreviated form	Expanded form	Encoding	Abbreviation mark code point
hañ	han(n)	han<am>&bar;</am>	0305
p̄	p(restr)	p<am>&bar;</am>	0305
þ̄	p(at)	þ<am>&bar;</am>	0305

In the last example, the bar crosses the ascender of the character ‘þ’. In our view, this is only a coincidence, since the bar in all cases is placed above the x height of the base line character. If there is a character with an ascender, the bar will simply cross this stroke.

The unmarked position of the bar is above the base line character, and this is therefore part of the definition of the entity ‘&bar;’. In some cases the bar may be placed below the base line character. Here, we suggest the entity name ‘&barbl;’ (for ‘bar below’).


The horizontal bar below is partially similar to 0331 COMBINING MACRON BELOW or 0332 COMBINING LOW LINE in the range [Combining Diacritical Marks](#) of *Unicode 5.0*, and may be identified with the latter.

Abbreviated form	Expanded form	Encoding	Abbreviation mark code point
	p(er)	p<am>&barbl;</am>	0332

It is possible to identify various shapes of the horizontal bar. In general we recommend that the transcriber should not make more distinctions than strictly necessary. If the transcriber for some reason would like to create a typology of bar forms, we suggest that this is done by numbering, ‘&bar-1;’, ‘&bar-2;’, ‘&bar-3;’, etc. The meaning of each entity must be explained in the header of the transcription and specified in the entity list (cf. [Appendix D](#) below)


6.4.2 Flourish

The flourish may be described as a horizontal bar with a return. It appears in the abbreviation of the Latin word ‘pro’ in contradistinction to ‘per’, which typically is abbreviated with a simple horizontal bar. We suggest using the entity name ‘&combflour;’ and recommend that it is given a separate code point in the Private Use Area.

Abbreviated form	Expanded form	Encoding	Abbreviation mark code point
	p(ro)fat	p<am>&combflour;</am>&fins;at	F1C6


6.4.3 The ‘us’ mark

Originally a Tironian *nota*, a mark resembling a small version of the number ‘9’ is often used for ‘us’. It is usually placed in a raised position, though not always clearly above the preceding character. Since the typical position of this mark is above the base line, we regard it as a combining mark and suggest the entity name ‘&us;’ and recommend that it is given a separate code point in the Private Use Area.

Abbreviated form	Expanded form	Encoding	Abbreviation mark code point
	la(us)	la<am>&us;</am>	F15B



6.4.4 The ‘er’ mark

A mark resembling a zigzag was frequently used as abbreviation of a front vowel (including diphthongs) + ‘r’, e.g. ‘ir’, ‘er’, ‘eir’, ‘ær’. The earliest form resembles a horizontal stroke with a descender to the left and an ascender to the right. It later acquired a zigzag-like form and even later resembles the letter ‘u’ turned upside-down. This abbreviation mark has now become part of the Unicode Standard (based on its usage in Lithuanian) in the range [Combining diacritical marks](#).

Abbreviated form	Expanded form	Encoding	Abbreviation mark code point
	v(er)	v<am>&er;</am>	035B


6.4.5 The ‘ra’ mark

Originally an open form of the character ‘a’, this mark was used as an abbreviation for ‘ra’ or ‘va’. One variant resembles the Greek omega-sign and another variant the omega-sign with a horizontal bar above. We suggest using the entity name ‘&ra;’ for the first type and ‘&rabar;’ for the second. We recommend that both marks are given separate code points in the Private Use Area.

Abbreviated form	Expanded form	Encoding	Abbreviation mark code point
	s(va)	s<am>&ra;</am>	F157
	f(ra)	&fins;<am>&rabar;</am>	F1C1

6.4.6 The ‘ur’ mark






The syllable ‘ur’ (sometimes ‘yr’) can be abbreviated by a mark resembling a small version of the number 2. A second form of this mark resemble a tilde, and a third form a horizontal version of the number 8 (equal to the lemniskate symbol), cf. [Hreinn Benediktsson 1965](#), p. 91. Due to the considerable variation in form we suggest that it might be useful to distinguish between three main forms, using the entity ‘&urrot;’ for the first type, ‘&ur;’ for the second and ‘&urlemn;’ for the third. The code points are respectively F153, F1C3 and F1C2 (all in the Private Use Area).

Abbreviated form	Expanded form	Encoding	Abbreviation mark code point
	ock(ur)	ock<am>&urrot;</am>	F153


6.4.7 Interlinear characters

Interlinear characters are a common type of abbreviation. An interlinear vowel typically represents a consonant (often ‘r’) + the vowel itself, while an interlinear consonant typically represents a vowel (often ‘a’) + the consonant itself. We suggest that interlinear abbreviation marks are named by the character itself + ‘sup’ (for ‘superscript’), e.g. ‘&asup;’ (interlinear ‘a’), ‘&osup;’ (interlinear ‘o’), ‘&rscapsup;’ (interlinear small capital ‘r’), etc.

Unicode 5.0 includes a selection of 13 superscript characters, namely ‘a’, ‘e’, ‘i’, ‘o’, ‘u’, ‘c’, ‘d’, ‘h’, ‘m’, ‘r’, ‘t’, ‘v’, ‘x’. They are located at the end of the range [Combining diacritical marks](#), 0363-036F. We suggest that these characters are used to display interlinear characters and that characters outside this selection are given separate code points in the Private Use Area.

Abbreviated form	Expanded form	Encoding	Abbreviation mark code point
	b(or)g	b<am>&osup;</am>g	0366
	m(anna)	m<am>&asup;</am>	0363
	v(ir)þa	v<am>&isup;</am>þa	0365
	þeg(ar)	þeg<am>&rsup;</am>	036C
	Otta(rr)	Otta<am>&rscapsup;</am>	F026

The runic character ‘m’, which itself can be used as an abbreviation (cf. [ch. 6.3.6](#) above), can appear with an interlinear abbreviation mark. The encoding follows the pattern above.


Abbreviated form	Expanded form	Encoding	Abbreviation mark code point
	(manna)	&mMedrun;<am>&asup;</am>	16D8 + 0363

Since the first entity, ‘&mrun;’, is defined as a base line character and the second, ‘&asup;’, as an interlinear mark placed above the immediately preceding base line character, there will be no doubt as to the positioning.


6.4.8 Superscript dots

Superscript dots are sometimes used to denote length. It is a moot question whether this is a type of abbreviation, but in any case the transcriber should use an entity for the encoding. We recommend that superscript dots are transcribed in analogy with other combining abbreviation marks and suggest using the entity name ‘&combdot;’ (for ‘combining dot above’).

Unicode 5.0 has a combining dot above in the range [Combining diacritical marks](#).

Abbreviated form	Expanded form	Encoding	Abbreviation mark code point
	leg(g)ia	leg<am>&combdot; </am>ia	0307

Sometimes the dot is used above small capitals. Since small capitals themselves are a way of representing gemination, the dot above is redundant. The encoding will simply be the same as above. Cf. [ch. 6.3.10](#) above.


Abbreviated form	Expanded form	Encoding	Abbreviation mark code point
	var(r)	va&rscap;<am> &combdot;</am>	0307


6.5 Special cases

6.5.1 Nomina sacra

In some cases the whole word must be analysed as an abbreviation. This applies to the traditional *nomina sacra*, i.e. abbreviations for sacred words such as ‘iesus’ and ‘christus’. These contain characters which originally were Greek but might be taken for Latin characters. For example, the ‘p’ in ‘xpm’ is originally a Greek ‘rho’ (‘r’).

We believe these abbreviations should be encoded as a sequence of the individual base line characters and one or more combining bars above. In the examples below, the originally Greek base line characters have been identified with the similar-looking Latin characters. Greek characters might also have been used in the encoding (such as ‘&igr;’ for GREEK SMALL LETTER IOTA, etc.).

Abbreviated form	Expanded form	Encoding	Abbreviation mark code point
	(iesus)	<am>i&bar;h&bar;c &bar;</am>	0305 (+ 0305 + 0305)

Abbreviated form	Expanded form	Encoding	Abbreviation mark code point
	(christum)	<code><am>x&bar;p&bar;m &bar;</am></code>	0305 (+ 0305 + 0305)

Note that the combining bar above has been encoded more than once in these examples. That ensures an appropriate display of the manuscript text, since the bar will be shown as extending over the whole word. However, it may be argued that there is only a single bar in each example, and that this bar simply happens to extend over more than one character. This problem is discussed more fully in [ch. 6.5.5](#) below.

6.5.2 Interlinear characters in other contexts


Interlinear (superscript) characters are used in various ways, not always as abbreviations. According to [de Leeuw van Weenen 2000](#): 36-43 there are four types:

(a) as abbreviation

This type is discussed in [ch. 6.4.7](#) above. Here, we recommend the usage of entities such as ‘&asup;’.


(b) as addition

When interlinear characters are used for adding characters which were left out by the scribe we recommend that this is encoded by use of the element `<add>` and the attribute `@place="supralinear"` (cf. [ch. 7.2](#)). There is no need for an entity of the type ‘&asup;’ since the location of the character is indicated by the element.

Manuscript form	Expanded form	Encoding
	han`a´	<code>han<add place="supralinear">a</add></code>


(c) as complementation of Roman numbers

Inflected forms of Roman numbers are sometimes specified by interlinear characters. In these cases, the interlinear characters are not placed above any base line character but merely raised above the base line. We suggest using the element `<seg>` and the attribute `@type="superscript"`.

Manuscript form	Expanded form	Encoding
	v.`ti´	<code>v.<seg type="superscript">ti</seg></code>


(d) as space savers

Especially at the end of a line one or more characters may be placed above the last word to save place and complete the line. We suggest the same encoding as in (c) above.

Manuscript form	Expanded form	Encoding
	e`s´	e<seg type="superscript">s</seg>


6.5.3 Missing abbreviation mark

From time to time one can find examples of a word that obviously is abbreviated but where there is no trace of the abbreviation mark. There is then no alternative but transcribing the text as it reads in the manuscript.

Manuscript form	Expanded form	Encoding
	d(rottning)	<am>d</am>

6.5.4 Nesting (stacking) of abbreviation marks


There are a few examples of base line characters which are abbreviated with an abbreviation mark which is itself abbreviated. An example is the base line character ‘m’ with an interlinear ‘o’ which in turn has a horizontal bar. According to rule 7 in [ch. 2.2.1](#) above this abbreviation should be encoded as the sequence ‘m’ + ‘&osup;’ + ‘&bar;’.

Manuscript form	Expanded form	Encoding
	m(onnom)	m<am>&osup;&bar;</am>


Since ‘&osup;’ is defined as a combining character, it follows that it is placed above the immediately preceding character, in this case ‘m’, and since ‘&bar;’ is also defined as a combining character, it follows that it is placed above ‘&osup;’. There is therefore no doubt as to the positioning of each part.

6.5.5 Extension of abbreviation marks

As a rule, combining abbreviation marks are associated with a single base line character. Thus, the sequence ‘m&osup;’ means that the interlinear character ‘o’ is seen as being placed above ‘m’ and not above any other character. However, some abbreviation marks extend over more than one character. For example, the word ‘k(ir)kia’ may be abbreviated with a horizontal bar crossing both the first and the second ‘k’. We believe it is sufficient to associate the abbreviation mark with only one of these characters, preferably the first.

Manuscript form	Expanded form	Encoding
	k(ir)kia	k<am>&bar;</am>kia


It is possible to encode this word so that the bar is associated with both characters. This is in a sense closer to the manuscript form, but it means that a single abbreviation mark may appear as two distinct marks (unless it is somehow stated that the two marks belong together). Thus, this is a more complex and possibly misleading solution.

Manuscript form	Expanded form	Encoding
	k(ir)kia	k<am>&bar; </am>k<am>&bar;</am>ia


On the other hand, it should be noted that this a case where 0305 COMBINING OVERLINE is appropriate, since it connects to left and right. Cf. the reference in [ch. 6.4.1](#) above.



6.5.6 Sporadic ligatures with abbreviation marks

In [ch. 5.4](#) we recommended that sporadic ligatures should not be encoded by use of separate entities but by the element <seg> with the attribute @type="ligature". A sporadic ligature is basically a joining of two base line characters which together do not reflect a separate phonological value. This is the case with ligatures such as 's+k' and 'p+p' which in this respect are identical to 's' + 'k' and 'p' + 'p'.



Manuscript form	Expanded form	Encoding
	(pp)	<seg type="ligature">pp</seg>

However, some ligatures are formed in such a manner that it is difficult to distinguish the separate parts. That applies to the ligature of long s + h, k and p. In these cases, we suggest that it is advisable to use individual entities. These characters must be referred to the Private Use Area.

Abbreviated form	Expanded form	Encoding	Abbreviation mark code point
	h(an)s	<am>&hslonglig;</am>	EBAD


Abbreviated form	Expanded form	Encoding	Abbreviation mark code point
	k(onung)s	<code><am>&kslonglig; </am></code>	EBAE
	p(es)s	<code><am>&thornslonglig; </am></code>	E734

Sometimes, a horizontal bar is used across these ligatures. The bar may be encoded separately with its usual entity, `&bar;` (cf. [ch. 6.4.1](#) above) or with a character located in the Private Use Area.

Abbreviated form	Expanded form	Encoding	Abbreviation mark code point
	k(onung)s	<code><am>&kslonglig; &bar;</am></code>	EBAE + 0305
	k(onung)s	<code><am>&kslongligbar; </am></code>	E7C8


6.5.7 The character ‘r’ as interlinear ligature

A quite special type of abbreviation is interlinear ‘r’ in ligature with e.g. ‘p’. We suggest encoding this as a sporadic ligature of ‘p’ and interlinear ‘r’.

Abbreviated form	Expanded form	Encoding	Abbreviation mark code point
	p(ar)	<code><seg type="ligature"> &thorn;&rsup;</seg></code>	00FE + 036C

6.5.8 Sharp ‘s’

In late Old Norwegian, the ‘sharp s’ appears in a number of abbreviations, e.g. for ‘skilling’, ‘smør’ and ‘son’. The German character ‘sharp s’ is defined in *Unicode 5.0* as 00DF LATIN SMALL LETTER SHARP S in the range [Latin-1 Supplement](#). We recommend using the ISO entity ‘ß’ also when this character is used as an abbreviation mark. The element `<am>` will indicate clearly that it is an abbreviation mark, not an ordinary character. See the discussion on the full stop in [ch. 6.3.8](#) above.

Abbreviated form	Expanded form	Encoding	Abbreviation mark code point
	Hakon(son)	<code>Hakon<am>&ssharp; </am></code>	00DF

6.6 List of abbreviation marks

An extensive list of abbreviation characters is found in the MUFI character recommendation, cf. [Appendix A](#) below.

Chapter 7. Altered, corrected and unreadable text

7.1 Introduction

This chapter deals with the encoding of additions, deletions and corrections made in the manuscript by the scribe or later users, or similar changes made in the transcription, e.g. by the transcriber or encoder of the manuscript text. Further, the chapter deals with the encoding of damage to the manuscript that affects the reading of the manuscript text. In [ch. 7.2](#) corrections, deletions and additions made by the scribe or later users of the manuscript are treated. In [ch. 7.3](#) damage to the manuscript that affects the reading of the manuscript text is treated. [Ch. 7.4](#) treats corrections, deletions and additions made by the transcriber of the manuscript text that have been made e.g. from other text witnesses or earlier editions of the text. [Ch. 7.5](#) contains a summary of all elements and attributes discussed in this chapter.

7.1.1 Structure

Because the features described in this chapter are largely non-linguistic, they have more relevance to the textual levels (encoded by the elements `fac`s, `dipl` and `norm`) than to the linguistic encoding of texts, parts of texts, words and so on. In many cases, therefore, the encoding of these features will form an incompatible hierarchy with the structure of the text. In such cases, and in cases where it occurs within a word, the encoder will have to encode the features at a particular textual level. These features will, in general, be marked up at the `fac`s level when they correspond to the physical manuscript. Changes to the text made by the editor/transcriber, on the other hand, should be encoded at the `dipl` level where necessary. The `norm` level contains the final, corrected text without non-linguistic markup. The following table indicates at what level each feature should be marked up, and/or its contents included, where relevant:

Feature	Element	<code>fac</code> s	<code>dipl</code>	<code>norm</code>
addition in ms.	<code><add></code>	text, markup	text	text
deletion in ms.	<code></code>	text, markup	(removed)	(removed)
illegible text	<code><gap/></code>	markup	—	—
blank space	<code><space/></code>	markup	—	—
unclear text in ms.	<code><unclear></code>	text, markup	text	text
text supplied by ed.	<code><supplied></code>	—	text, markup	text
error in ms. text	<code><sic></code>	text, markup	(removed or corrected)	(removed or corrected)
text corrected by ed.	<code><corr></code>	—	text, markup	text

Here 'Element' also refers to the value of the 'category' attribute when the `<me:textSpan/>` element is used (see [ch. 7.5](#)).

7.1.2 Elements

The encoding recommended here is based on [ch. 11 'Representation of Primary Sources'](#) of the TEI P5 Guidelines, where the following elements are defined:

Elements	Contents
<code><add></code>	contains letters, words, or phrases inserted in the manuscript text or in the margins of the manuscript by an author, scribe, annotator or corrector.
<code></code>	contains a letter, word or passage deleted, marked as deleted, or otherwise indicated as superfluous or spurious in the manuscript text by an author, scribe, annotator or corrector.
<code><gap/></code>	indicates a point where material has been omitted in a transcription, normally because the manuscript text is illegible, but potentially for some other reason.
<code><space/></code>	indicates a significant or deliberate space in the manuscript.
<code><unclear></code>	contains a word, phrase or passage which cannot be transcribed with certainty because it is illegible in the manuscript.
<code><supplied></code>	signifies text supplied by the transcriber, encoder or editor in place of text which cannot be read, either because of physical damage or loss in the original or because it is illegible for any reason.
<code><sic></code>	contains text reproduced in the transcription although apparently incorrect or inaccurate.
<code><corr></code>	contains the correct form of a passage apparently erroneous in the manuscript text. This element should only be used for corrections made in the transcription or encoding of the manuscript text. It should not be used for corrections made within the manuscript (e.g. by the scribe or a later hand).

7.1.3 Philological introduction

In a discussion on editorial practice for Old Norse texts Helle Jensen, with reference to [Stefán Karlsson 1963](#), LXVII f., outlines aspects of the manuscript text which should be noted in an edition ([Jensen 1988](#)). Jensen's suggestions start with structural markup of e.g. linebreaks in the manuscript. She also gives special signs for each of the features that has to do with scribal or later changes in the manuscript as follows (Jensen 1988, 102 f.):

Sign	Explanation
˘	Includes something that has been added above the line in the manuscript.
˘˘	Includes something that has been added in the margins. Unless stated in a footnote the addition is considered to be the work of the hand that has written the main text.

Sign	Explanation
-	Text that has been struck through, underdotted or erased is placed within these brackets.
- -	Text that has been written twice without being marked by the scribe in the manuscript is placed within these brackets.
< >	Text not present in the exemplar, but supplied in the edition by the editor.
*	The following word is corrected by the editor. In a footnote the original form is given.
[]	The text of the manuscript is illegible due to use or damage. The text included could be supplied from another manuscript or be a conjecture made by the transcriber or editor. If the addition is made from another manuscript it should be given diplomatically, if from other sources, such as editions or transcriptions, it should be rendered in a form normalized in accordance with the manuscript text.
[[]]	Characters within double brackets have been read for the first time in the present transcription.
000	Unreadable characters or characters lost e.g. through damage to the manuscript. The number of zeros corresponds to the number of characters presumed missing.
000...000	The number of unreadable characters is not known.

In addition, Helle Jensen suggests that uncertain readings should be subpunctuated. In editions from the Arnamagnæan Institutes in Reykjavík and Copenhagen these suggestions are in general followed, and in most editions of medieval Scandinavian texts similar systems are used. This gives us a starting point when we are transcribing Old Icelandic and Old Scandinavian manuscripts.

The principles presented in this handbook are based on the tradition of producing scholarly editions of texts and individual manuscripts. The system for printed editions outlined by Helle Jensen can therefore very often be translated into the electronic markup language presented in this chapter.

Text written in the margins can be of various kinds and of varying interest for our knowledge about the main text and the history of the manuscript. Notes on the main text in the margins are of course valuable when we are interested in the text tradition. Other notes could indicate that someone at a certain stage has used it for example in a transcription of the text.

In medieval manuscripts, however, we often also find notes in the margins that have nothing whatsoever to do with the manuscript text. These notes can at first sight seem to be of no value to philological investigation, but in a larger context they can sometimes give information as to where a manuscript has been at a certain stage of its history. If e.g. the same type of scribbles are found in a group of manuscripts where one of the manuscripts can be geographically pin-pointed, this could indicate the whereabouts of the whole group. Information of this kind can also lead to the establishing of new connections between manuscripts that were not previously seen as connected. There are thus good arguments for including information also on this kind of marginal note, but these are more

properly contained in the manuscript description in the header (cf. [ch. 10.2.2](#)) than within the encoded transcription.

The first kind of notes, i.e. comments or additions to the main text, are often treated in foot-notes in printed editions. They are considered relevant to the reading of the text, and are therefore given in relation to the main text. Marginal notes that indicate the owner or user of the manuscript in any obvious way are often treated in the introduction to the edition as they are considered relevant to the history of the text or manuscript.

The third category of notes, the ones that do not seem to give any relevant information, is often excluded or treated only briefly in the introduction. This is of course a rational way to handle these scribbles when the printed edition sets the limits, and the information often is obscure and cannot be easily related to parallel information concerning other manuscripts. In the electronic transcription of a manuscript, however, there is no reason to make this limitation. The information can be given in the same way as for the other categories, and thereby give us the possibility to search for all kinds of obscure information.

Medieval manuscripts have often become damaged through use, sometimes with relevance for our reading of the text. Pieces of parchment may for example have been torn out, leaving a physical gap in the manuscript. Parts of the text may be illegible because of use or deliberate erasure, or they may be darkened to such an extent that the text is no longer readable. In printed editions, unreadable sections of a text are marked as suggested by Helle Jensen. In the introductions to printed editions problems related to illegible text and damage to the manuscript are often discussed at length. If there are other text witnesses these are often used to replace missing stretches of text. In a diplomatic transcription of a manuscript text, however, the missing or unreadable parts are most often just marked as such. In the following sections the relation between the traditional markup of these kinds of textual and editorial difficulties and electronic encoding will be obvious. It is therefore relevant to take traditional transcription and editing as a starting point for the electronic encoding of transcriptions of manuscript texts.

The primary aim of the following sections are to give recommendations for the transcription and encoding of manuscript texts. It does, however, in some instances also give recommendations for editorial encoding, e.g. markup that refers to corrections or additions made by the transcriber or encoder. It is therefore important to keep the transcription and encoding of the manuscript text on the one hand and on the other hand the editorial changes consistently separated, so that the former provides a starting point for the editorial work.

7.2 Scribal features: Additions, deletions and substitutions

In the manuscript text and in the margins of the manuscript we often find different kinds of corrections, deletions and additions that we want to encode. These changes can be divided into different groups depending on the nature of the change and its relevance for the reading of the manuscript text or our knowledge about the manuscript. The main division is between additions or substitutions to the manuscript text, within the text or in the margins, and deletions made in the manuscript text. The former should be marked with the `<add>` element while the latter should be marked with the `` element. Additions and substitutions made by the transcriber or editor are treated in the last section ([ch. 7.4](#)).

7.2.1 Additions

This section deals with additions *made by a scribal hand only*. It is a common mistake to use elements designed for this purpose to mark up additions made by an editor - such features are covered below in [ch. 7.4](#).

The following elements are recommended for describing additions made by the author of the text, a compiler, scribe, annotator or corrector in the manuscript text. The TEI P5 Guidelines recommend the use of the **<add>** element to describe additions in the manuscript ([ch. 11.3](#)). In the following the use of **<add>** in relation to our recommended encoding of the individual word within the element **<w>** and on the three different levels **<me:facs>**, **<me:dipl>** and **<me:norm>** is treated.

Elements	Contents
<add>	Contains letters, words or phrases inserted in the manuscript text or in the margins of the manuscript by an author, scribe, annotator or corrector. Attributes include:
@hand	Signifies the agent which made the addition. The value is an XML IDREF, referring to a <handNote> element included in the header under <handDesc> . See the Menota header in Appendix E .
@resp	Signifies the transcriber or editor responsible for identifying the hand. The value is an XML IDREF, referring to an agent described in the header (cf. also ch. 10).
@place	Indicates where the addition is made. Suggested values include:
'inline'	The addition is made in a space originally left empty by the scribe.
'supralinear'	The addition is made above the line.
'infralinear'	The addition is made below the line.
'margin-left'	The addition is made in the left margin.
'margin-right'	The addition is made in the right margin.
'margin-top'	The addition is made in the top margin.
'margin-bot'	The addition is made in the bottom margin.
'nextPage'	The addition is made on the next page.
'previousPage'	The addition is made on the previous page.

Additions which can be ascribed to the author of a text are rare in medieval Nordic manuscripts. The additions being described with the above-mentioned attribute **@hand** will therefore primarily be ascribed to the values 'scribe', 'compiler', 'annotator' or 'corrector'. Scribal additions are probably the most common changes to be recorded in the transcription and encoding of a manuscript text. The list of hands in the header (cf. [ch. 10](#)) should identify the individual hand, either as anonymous or, if possible, by name. The main hand in a manuscript will normally be marked as 'mainscribe'.

If the addition consists of a series of complete words, the **<add>** tag should be surrounding the word(s). The following example contains a marginal addition:

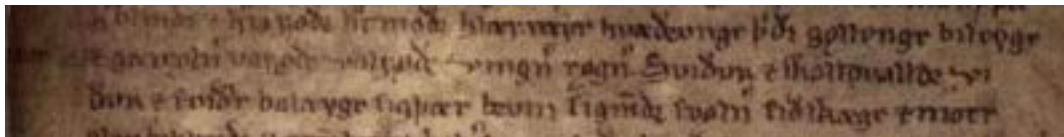


Fig. 7.1 AM 748 I b 4to, fol. 18r, ll. 5-7

Here (AM 748 I b 4to, fol. 18r, ll. 5-7) the scribe (identified as 'mainscribe' in the header) has inserted the word 'sigavtr' in the left margin (now partially cut off by the binding) with a mark after 'skollvalldr'. It should be encoded as follows. Because the addition is a whole word, the tag should enclose the word (Note that for the sake of clarity we have limited the use of encoding to the relevant sequence and simplified the orthography):

```
<!-- after skollvalldr -->
<add place="left" hand="mainscribe">
  <w>
    <choice>
      <me:fac>sig&avlig;tr</me:fac>
      <me:dipl>sig&avlig;tr</me:dipl>
    </choice>
  </w>
</add>
```

Note that to ensure correct rendering of the addition, no space is included between the **<add>** and **<w>** elements.

In cases where the addition forms part of a word, the markup should normally be restricted to the facsimile level. In any case, the addition need not be marked up as part of the normalised text. Cf. the following:

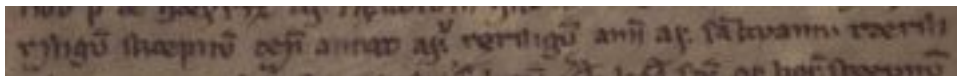


Fig. 7.2 AM 748 I b 4to, fol. 1r, l. 15

```
annat af
<w>
  <me:fac><add hand="scribe" place="supralinear">v</add>
  r&eogon;riligv<am>&bar;</am></me:fac>
  <me:dipl>vr&eogon;riligv<ex>m</ex></me:dipl>
  <me:norm>uhr&aelig;riligum</me:norm>
</w>
annat
```

The location of the addition in the above markup is indicated by the attribute **@place**; in this case, the addition is made above the line of the manuscript text and therefore uses the value 'supralinear'.

The diplomatic and normalised text will normally include the addition if made by a scribal hand. Additions made by later hands will normally be omitted from the diplomatic and normalised text without markup.

Additions are sometimes made by an annotator, i.e. comments to the text. This kind of additions could be encoded as the marginal note “vantar ekkert F. J.” by Finnur Jónsson in Codex Wormianus (AM 242 fol. p. 60):

```
<add hand="FJ">
  <w><me:fac>vantar</me:fac></w>
  <w><me:fac>ekkert</me:fac></w>
  <w><me:fac>F&dot;</me:fac></w>
  <w><me:fac>J&dot;</me:fac></w>
</add>
```

It is also possible to indicate with the attribute **@place** where on the manuscript page the annotation is made. Finnur Jónsson's annotation is made in the bottom margin, and should be encoded as follows:

```
<add hand="FJ" place="bottom">
  <w><me:facs>vantar</me:facs></w>
  <w><me:facs>ekkert</me:facs></w>
  <w><me:facs>F&dot;</me:facs></w>
  <w><me:facs>J&dot;</me:facs></w>
</add>
```

Changes in scribal hands are not considered additions. For the markup of such phenomena, use the empty **<handShift/>** element. Note that this element is parallel to empty elements like **<pb/>**, **<cb/>** and **<lb/>** in that it only points to a break in the text (cf. the discussion in [ch. 4.7](#) and [ch. 4.10](#) above).

See §7.5 below for examples of how to mark up additions which span linguistic (**<w>**, etc.) boundaries.

7.2.2 Deletions

This section deals with deletions made by a scribal hand only. Text suppressed by the editor is dealt with under 'corrections' ([ch. 7.4.2](#)) below.

The TEI P5 Guidelines specify the use of the **** element to describe additions in the manuscript ([ch. 11.3](#)). In the following the use of **** in relation to our recommended encoding of the individual word within the element **<w>** and on the three different levels **<me:facs>**, **<me:dipl>** and **<me:norm>** is treated.

Elements	Contents
	Contains a letter, word or passage deleted, marked as deleted, or otherwise indicated as superfluous or spurious in the manuscript text by an author, scribe, annotator or corrector. Attributes include:
@hand	Signifies the agent which made the deletion. The value is an XML IDREF, referring to a <handNote> element included in the header under <handDesc> .
@resp	Signifies the editor or transcriber responsible for identifying the hand of the restoration. The value is an XML IDREF, referring to an agent described in the header (cf. ch. 10). This information can also be given in the header.
@rend	Display rendering information in TEI. This attribute is used here specifically to classify the deletion as displayed, using any convenient typology. Sample values include:
'overstrike'	The text has been struck through.
'erasure'	The text has been erased.
'bracketed'	Deletion indicated by brackets in the text or margin.
'subpunction'	Deletion indicated by dots beneath the letters deleted.

Deletions that can be ascribed to the author of a manuscript text are rare in medieval Nordic manuscripts. The deletions being described with the above mentioned attribute **@hand** will therefore primarily be ascribed to 'scribe' or 'corrector'.

Deletions should normally only be marked up at the 'facs' level. The text should be removed without indication at the other textual levels.

The example in [ch. 7.2.1](#) (Figure 7.1) contains the word 'skolldvalldr' in which the first 'd' is marked with a dot below signifying deletion. It should be marked up as follows:

```
<w>
  <choice>
    <me:facs>&slong;koll<del hand="mainscribe" rend="subpunction">d</del>
      valld&rrot;</me:facs>
    <me:dipl>&slong;kollvalld&rrot;</me:dipl>
  </choice>
</w>
```

Deletions of one or more words made by the scribe(s) or corrector(s) of a manuscript are encoded as in the passage from the Third Grammatical Treatise below. Note that we for clarity limit the use of encoding to the relevant manuscript line, and that only the **<me:facs>** and **<me:dipl>** levels are shown, and a few of the words have been suppressed.

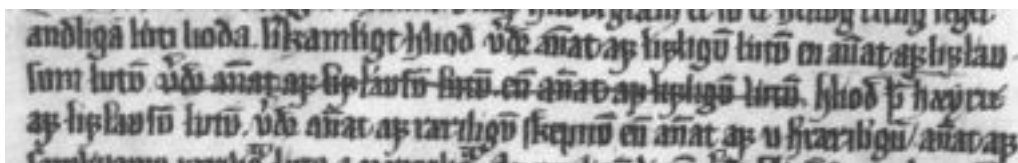


Fig. 7.3 AM 242 fol., p. 94, ll. 11-13

```
<lb n="94:12" />
<w>
  <choice>
    <me:facs>liflau<lb/>sum</me:facs>
    <me:dipl>liflausum</me:dipl>
  </choice>
</w>
<w>
  <choice>
    <me:facs>lvtv<am>&bar;</am>.</me:facs>
    <me:dipl>lvtv<ex>m</ex>.</me:dipl>
  </choice>
</w>
<del hand="mainscribe" rend="overstrike">
<w>
  <choice>
    <me:facs>v<am>&er;</am>rdr</me:facs>
    <me:dipl>v<ex>er</ex>dr</me:dipl>
  </choice>
</w>
<w>
  <choice>
    <me:facs>an<am>&bar;</am>at</me:facs>
    <me:dipl>an<ex>n</ex>at</me:dipl>
  </choice>
</w>
<!-- skipping: 'af liflausum lutum. enn annat af lifligum lutum' ... -->
<w>
  <choice>
    <me:facs>lvtv<am>&bar;</am>.</me:facs>
    <me:dipl>lvtv<ex>m</ex>.</me:dipl>
  </choice>
</w>
</del>
```

```

<w>
  <choice>
    <me:facs>Hlioð</me:facs>
    <me:dipl>Hlioð</me:dipl>
  </choice>
</w>
<w>
  <choice>
    <me:facs>þ<am>&bar;</am></me:facs>
    <me:dipl>þ<ex>at</ex></me:dipl>
  </choice>
</w>
<w>
  <choice>
    <me:facs>hæyriz</me:facs>
    <me:dipl>hæyriz</me:dipl>
  </choice>
</w>

```

In the TEI P5 Guidelines cited above there are a number of possible types of deletion described with the attribute **@type**. These could be applied to deletions made both by scribe(s) and corrector(s). If a deletion is made e.g. by 'overstriking' the deleted text it could be encoded as (here only presented on the **<me:facs>** level):

```

en tuenner flokkar þeirar þioðar er<lb n="3r:15"/>
<del hand="mainscribe" rend="overstrike">
  <w>
    <me:facs>liguri</me:facs>
  </w>
  <w>
    <me:facs>hæita</me:facs>
  </w>
  <w>
    <me:facs>er</me:facs>
  </w>
</del>
traceum hæiter.

```

This could then be displayed on the computer screen or in a printed edition in the manner suggested above (ch. 7.1):

```

14 ...en tuenner flokkar &thorn;eirar &thorn;io&eth;ar er
15 |-liguri h&aelig;ita er-| traceum h&aelig;iter...

```

The text that is marked as deleted must be at least partly legible in the manuscript so that it can be read by the transcriber. If the deleted text is not legible the deletion should be marked up with the **<gap/>** element, described below (7.3.1). The **<gap/>** element could be enclosed in the **** element to indicate that the gap is in some way intentional. Parts of the deleted text that are legible could be indicated by the **<unclear>** element in combination with the **<gap/>** element as described below (ch. 7.3.2).

As for the markup of additions using **<add>**, if the deletion is of part of a word, it should normally only be marked up at the **<me:facs>** level. If the deletion is by a scribal hand, the deleted text will be omitted from the **<me:dipl>** and **<me:norm>** levels without markup.

In some cases, the deletion will conflict with word boundaries. ch. 7.5 below describes how to mark up such deletions.

7.2.3 Substitutions

This section describes the markup of text substituted by a scribe, that is, where a scribe deletes text and replaces it with some other text. In medieval manuscripts a rather common

phenomenon is the combination of deleted text and added text. It is not always possible, however, to ascertain the relation between the two. If someone has deleted the originally written text inline this does not automatically mean that a corresponding addition above the line or in the margin is made by the same scribe. It can therefore not be stated as certain whether the correspondence is intentional or not. There is no specific element for this type of feature. We suggest that substitutions made in the manuscript should be marked primarily with the two core tags `` and `<add>`. In cases where we can be relatively sure about the agent of the whole substitution this could be indicated with a combination of the `` and the `<add>` elements as illustrated below.

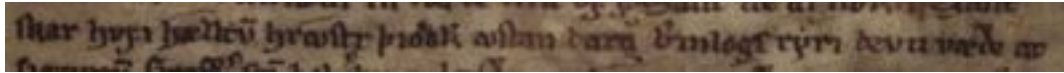


Fig. 7.4 AM 748 I b 4to, fol. 13r, l. 25

Here, the seventh word 'barv' has been altered by the scribe, such that the original 'a' at the end of the word is marked as deleted by subpunctuation and a 'u' has been added as the replacement letter above the line. This word should be marked up as follows (facs and dipl levels only):

```
<w>
  <choice>
    <me:facs>bar<del hand="mainscribe" rend="subpunction">a</del>
      <add hand="mainscribe" place="supralinear">v</add></me:facs>
    <me:dipl>barv</me:dipl>
  </choice>
</w>
```

As for the `` and `<add>` elements, markup within words should only be included at the facs level, and the dipl and norm levels should include the text which the scribe(s) appears to have intended, as in the example above.

The same type of markup can be used for substitutions which span structural boundaries. This type of markup is discussed in more detail in [ch. 7.5](#).

7.3 Damage and illegibility

The following section deals with text omitted in the transcription or editing of text due to damage or illegibility in the manuscript, and text supplied from other sources such as other text witnesses or earlier editions.

The following detail of a manuscript page will serve to illustrate the markup in the following sections.

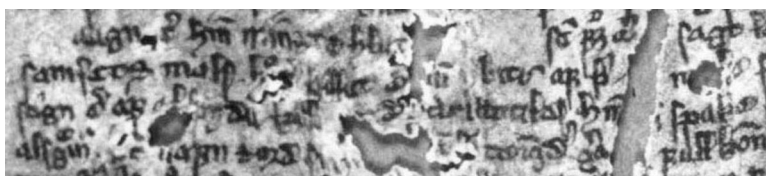


Fig. 7.5 AM 757 a 4to, fol. 2r, ll. 22-25

7.3.1 Text omitted from the transcription

When the manuscript is illegible we suggest the use of the elements `<gap/>` and `<supplied>` to indicate the illegible text, its extension and how it has been supplied (for the `<supplied>` element see [ch. 7.4.1](#)). The `<space/>` element is used to represent

deliberate omissions from the manuscript which have some significance, e.g. spaces left for decorated initials or words.

Elements	Contents
<gap/>	Is an element without extension in the encoded manuscript text. It indicates a point where material has been omitted in a transcription because the manuscript text is illegible. Attributes include:
@reason	Gives the reason for omission. Sample values include: 'sampling', 'illegible', 'irrelevant', 'cancelled', 'cancelled and illegible'.
@quantity	Indicates approximately how much text has been omitted from the transcription, in the way that has been suggested by Helle Jensen referred to above (ch. 7.1). Values can be given as e.g. number of signs, number of lines or number of pages in the manuscript.
@resp	Indicates the transcriber, encoder or editor responsible for the decision not to provide any transcription and hence the application of the <gap/> element.
@hand	In instances where text is omitted from the transcription because of deliberate deletion by an identifiable hand, this attribute signifies the hand which made the deletion.
@agent	In the case of text omitted because of damage, categorizes the cause of the damage, if it can be identified.
<space/>	Is an element without extension in the encoded manuscript text. It indicates a point in a transcription of a manuscript where the manuscript has a deliberate omission. Attributes include:
@quantity	The extent of the space. Values can be given as e.g. number of signs, number of lines or number of pages in the manuscript.
@unit	Names the unit used for describing the extent of the gap.

In medieval manuscripts we often find sections that for some reason are illegible. This can be due to e.g. damage or use. In the transcription we primarily wish to register the sections that are illegible and the extent of the illegibility. We suggest that the illegible sections should be indicated by the **<gap/>** element. The extent of the illegible section could be encoded as the following text from the manuscript above, where wearing has occurred next to a hole (bottom left of image):

```
<w>
  <choice>
    <me:fac>m<lb n="25"/>a<slong>g<am>&esup;</am>in<am>&rsup;</am>
    </me:fac>
    <me:dipl>m<lb n="25"/>a<slong>g<ex>re</ex>in<ex>ar</ex></me:dipl>
  </choice>
</w>
<!-- we know from the other mss that a whole word is missing -->
<w>
  <choice>
    <me:fac><gap /></me:fac>
    <me:dipl><!-- we can insert text here - see below --></me:dipl>
  </choice>
</w>
```



```

<w>
  <choice>
    <me:facs>nafn</me:facs>
    <me:dipl>nafn</me:dipl>
  </choice>
</w>

```

With this markup the extent of the illegible section is not defined. It can be presented on the computer screen or in a printed edition in the manner suggested above ([ch. 7.1](#)):

malsg(re)in(ar) 00...00 nafn

If the transcriber or encoder of the text wishes to define the section more accurately it can be done as in the following example. The number of missing signs is given as a value to the attribute **@quantity**. It should be noted that the number given in the example is not intended as an exact evaluation of the number of signs missing in the present manuscript.

```

<w>
  <choice>
    <me:facs>m<lb n="25"/>al&slong;g<am>&esup;</am>in<am>&rsup;</am>
    </me:facs>
    <me:dipl>m<lb n="25"/>al&slong;g<ex>re</ex>in<ex>ar</ex></me:dipl>
  </choice>
</w>
<!-- we know from the other mss that a whole word is missing -->
<w>
  <choice>
    <me:facs><gap extent="2"/></me:facs>
    <me:dipl><!-- we can insert text here - see below --></me:dipl>
  </choice>
</w>
<w>
  <choice>
    <me:facs>nafn</me:facs>
    <me:dipl>nafn</me:dipl>
  </choice>
</w>

```

This could be represented as follows on the computer screen or in a printed edition. As the accuracy of this kind of evaluation is questionable it should not have the highest priority to display this in e.g. a printed edition.

malsg(re)in(ar) 00 nafn

See below ([ch. 7.4.1](#)) for an example of text supplied in place of the gap by the editor using the **<supplied>** element.

A deliberate space in a manuscript should be indicated by the **<space/>** element. For example, in the image above, an initial letter has been omitted, presumably for a coloured or decorated initial to be added later. This is encoded as follows:

```

<lb/><w>
<!-- initial 'S' is left blank by the scribe here -->
  <me:facs><space quantity="1" dim="horizontal"/>augn</me:facs>
</w>

```

In this example, the text has only been encoded on the *facs* level, so the **<choice>** element would be redundant.

If the text omitted in the space is supplied using the **<supplied>** element, the **<space/>** tag will normally be used at the *facs* level and the **<supplied>** element will be used at the *dipl* level (with the **@reason="space"** attribute set). See [ch. 7.4.1](#).

7.3.2 Uncertain readings in the manuscript

In medieval manuscripts we often encounter problems of illegibility due to use or damage. In the following the encoding of such sequences is treated. To some extent this has already been treated in the above section ([ch. 7.3.1](#)). In cases where the text is readable to some extent the `<gap/>` and `<supplied>` elements should not be used. The TEI P5 Guidelines ([ch. 11.5](#) of the TEI P5 Guidelines) recommend that the `<unclear>` element is used for encoding damage and illegibility where the text of the damaged or illegible area can be read with some, but not full, certainty.

Elements	Contents
<code><unclear></code>	Contains a letter, word, phrase or passage which cannot be transcribed with certainty because it is illegible in the manuscript text. Attributes include:
<code>@reason</code>	Indicates why the material is hard to transcribe.
<code>@resp</code>	Indicates the individual responsible for the transcription of the letter, word, phrase or passage contained within the <code><unclear></code> element.
<code>@hand</code>	Signifies the hand responsible for the action where the difficulty in transcription arises from action (partial deletion, etc.) assignable to an identifiable hand. Note that this attribute has the same function in the <code></code> element above (ch. 7.2.2).
<code>@agent</code>	Where the difficulty in transcription arises from an identifiable cause, signifies the causative agent.
<code>@rend</code>	Indicates how the element in question was rendered or presented in the source text.

The example given above from AM 757 a 4to includes a section on the second line which can only be partially read. The following is the encoding for the section, only represented at the facs level:

```

<w>
  <me:facs>mal&slong;</me:facs>
</w>.
<w>
  <me:facs>h<am>&osup;</am></me:facs>
</w>
<w>
  <me:facs><gap extent="2"/></me:facs>
</w>
<unclear reason="worn" resp="TW">
<w>
  <me:facs>k&ocurl;llut</me:facs>
</w>
<w>
  <me:facs>ein<am>&bar;</am></me:facs>
</w>
<w>
  <me:facs>hlutr</me:facs>
</w>
</unclear>
<w>
  <me:facs>af</me:facs>
</w>

```

The text may also include fully illegible text, as represented above with the `<gap/>` element. With this encoding the text could be presented with subpunction or as grey text for all the words that the editor can not read with absolute certainty.

7.4 Editorial interventions

When transcribing medieval material we often encounter words or longer sequences of text that we consider corrupt in one way or another. Sometimes it may also be obvious that text is missing in the manuscript we are transcribing or that the scribe has made a mistake. The transcriber of the manuscript text may in these instances wish to indicate the mistake or even correct the text, either directly from other versions of the same text or based on already existing editions of the text. Sometimes the transcriber or editor may also wish to make obvious grammatical corrections in the text without having any other text witness or precedence in an earlier edition. In the following the encoding of corrections made by the transcriber of the text or by an editor are treated. Note that we do not recommend the use of the attribute `@hand` for the changes made in transcription or encoding of the manuscript text. The attribute `@resp` should be used consistently for corrections or additions made in the transcription or encoding of the text to distinguish clearly between what is found in the manuscript text and what is made in the transcription and encoding of the text.

7.4.1 Additions made by the transcriber or editor

If text is obviously missing in the manuscript text we may wish to supply it. This could be based on, for example, another text witness or on a earlier edition of the text. The markup of such additions should give information about the source as well as about the responsibility for the addition. To encode additions made in the transcription we recommend the use of the `<supplied>` element as described in the TEI P5 Guidelines ([ch. 11.3.7](#)). In the following the use of `<supplied>` in relation to our recommended encoding of the individual word within the element `<w>` and on the three different levels `<me:fac>`, `<me:dipl>` and `<me:norm>` is treated.

Elements	Contents
<code><supplied></code>	Signifies text supplied by the transcriber, encoder or editor in place of text which cannot be read, either because of physical damage or loss in the original or because it is illegible for any reason. Attributes include:
<code>@source</code>	States the source of the supplied text if this can be located.
<code>@resp</code>	Indicates the individual responsible for the addition of letters, words or passages contained within the <code><supplied></code> tag. It can be given values like:
'transcriber'	The person responsible for the transcription of the manuscript text.
'encoder'	The person responsible for the encoding of the manuscript text.
'editor'	The editor of the text used for the addition or responsible for the addition in editing the manuscript text.
<code>@reason</code>	Indicates why the text has had to be supplied
<code>@agent</code>	Where the presumed loss of text leading to the supplying of text arises from an identifiable cause, signifies the causative agent.

If the transcriber or editor wishes to supply text that is missing in the transcribed manuscript text from for example another text witness, this can be handled with the **<supplied>** element. The interpolated text could be transcribed as in this instance from the detail of AM 757 a 4to above. Note that we for clarity limit the use of encoding to the relevant sequence:

```
<w>
  <choice>
    <me:fac>m<lb n="25"/>a<slong>g<am>&esup;</am>n<am>&rsup;</am>
    </me:fac>
    <me:dipl>m<lb n="25"/>a<slong>g<ex>re</ex>n<ex>ar</ex></me:dipl>
  </choice>
</w>
<w>
  <choice>
    <me:fac><gap extent="2"/></me:fac>
    <me:dipl><supplied resp="TW" source="AM 748 I b 4to">&thorn;at
    </supplied></me:dipl>
  </choice></w>
<w>
  <choice>
    <me:fac>nafn</me:fac>
    <me:dipl>nafn</me:dipl>
  </choice>
</w>
```

This could then be displayed on the computer screen or in a printed edition in the manner suggested above (ch. 7.1):

malsg(re)n(ar) <pat> nafn

In the above example there are other sources available for the illegible text. This kind of editorial change is, however, not suggested as compulsory. In a primary transcription and encoding the use of the **<gap/>** element should only give the essential manuscript information. The attributes to **<gap/>** and **<supplied>**, such as **@source** or **@resp**, can of course be included voluntarily and to the extent that information is available.

7.4.2 Deletions made by the transcriber or editor

If a piece of text obviously should be deleted, e.g. duplicated text in a dittography, the transcriber or editor might want to make a deletion. This is the converse action of adding text, and should be distinguished from similar actions made by the scribe. While the elements **<add>** and **** describe actions by the scribe himself or other scribes, the editorial additions and deletions should be singled out by separate elements. For additions, TEI recommends the element **<supplied>**, but there is no parallel to the **** element. We suggest the element **<me:expunged>**, since the noun *expunction* and the verb *expunge* are commonly used for editorial deletion.

Elements / attributes	Contents
<me:expunged>	Contains text which the transcriber or editor believes should be expunged.
@resp	Indicates the individual responsible for the expunction of letters, words or passages contained within the <me:expunged> . It can be given values like:
'transcriber'	The transcriber responsible for the expunction.

Elements / attributes	Contents
'encoder'	The encoder responsible for the expunction.

Note that expunged text will not be deleted from the transcription, but will be contained by the **<me:expunged>** element. The editor of the text may decide to display it with no comments, put it in brackets or suppress it. In a multi-level transcription, expunction will typically not be found on the **<me:dipl>** and **<me:norm>** levels, but not on the **<me:facs>** level.

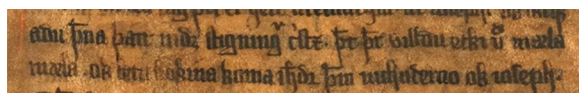


Fig. 7.6 AM 233 a fol, fol. 28vB, l. 39-40

In this example from *Niðrstigningar saga*, AM 233 a fol, the word ‘menn’ has been supplied and thereafter the duplicated word ‘mæla’ has been expunged:

```

<w>
  <choice>
    <me:facs>&thorn;<am>&bar;</am>t</me:facs>
    <me:dipl>&thorn;<ex>uia</ex>t</me:dipl>
  </choice>
</w>
<w>
  <choice>
    <me:facs>&thorn;<am>&bar;</am>r</me:facs>
    <me:dipl>&thorn;<ex>ei</ex>r</me:dipl>
  </choice>
</w>
<w>
  <choice>
    <me:facs>villdu</me:facs>
    <me:dipl>villdu</me:dipl>
  </choice>
</w>
<w>
  <choice>
    <me:facs>eck</me:facs>
    <me:dipl>eck</me:dipl>
  </choice>
</w>
<w>
  <choice>
    <me:facs>v<am>&dsup;</am></me:facs>
    <me:dipl>v<ex>id</ex></me:dipl>
  </choice>
</w>
<w>
  <choice>
    <me:facs></me:facs>
    <me:dipl><supplied resp="OE" source="AM 645 4to">menn</supplied>
    </me:dipl>
  </choice>
</w>
<w>
  <choice>
    <me:facs>m&aelig;la</me:facs>
    <me:dipl>m&aelig;la</me:dipl>
  </choice>
</w>
<lb n="40" />

```

```
<w>
  <choice>
    <me:facs>m&aelig;la</me:facs>
    <me:dipl><me:expunged>m&aelig;la</me:expunged></me:dipl>
  </choice>
</w>
```

Depending on the style sheet, the diplomatic text might be displayed this way:

puiat þeir villdu ecki vid <menn> mæla -|mæla|-

7.4.3 Corrections

In the manuscript it is not always possible to say anything with certainty about the intention of changes in the text. When transcribing the text, however, corrections of obvious mistakes in the manuscript text could be marked with the following tag set recommended in the TEI P5 Guidelines (ch. 11.3). In the following the use of <sic> and <corr> in relation to our recommended encoding of the individual word within the element <w> and on the three different levels <me:facs>, <me:dipl> and <me:norm> is treated.

Elements / attributes	Contents
<sic>	Contains text reproduced although apparently incorrect or inaccurate.
<corr>	Contains the correct form of a passage apparently erroneous in the manuscript text.
@resp	Indicates the individual responsible for the correction of letters, words or passages contained within the <corr> and <sic> elements. It can be given values like:
'transcriber'	The person responsible for the transcription of the manuscript text.
'encoder'	The person responsible for the encoding of the manuscript text.
'editor'	Signifies the editor responsible for suggesting the correction.
'rend'	Indicates how the element in question was rendered or presented in the source text.

In a first-level transcription it can be relevant just to mark the obviously corrupted instances in the manuscript text. This could be done with the <sic> element as in this instance from AM 242 fol:

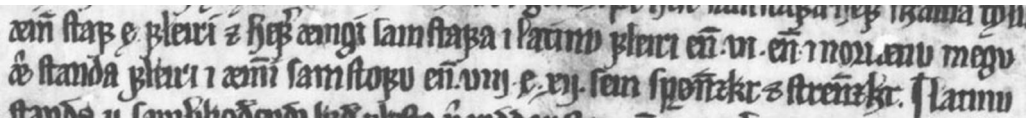


Fig. 7.7 AM 242 fol, p. 98, ll. 3-4

Here the second numreral 'xij.' is written instead of 'ix.', obvious from the context and other manuscripts. The error should be signalled at the facs level and corrected at the dipl level:

```
<w>
  <choice>
    <me:facs>en<am>&bar;</am></me:facs>
    <me:dipl>en<ex>n</ex></me:dipl>
```

```

    </choice>
  </w>
  <num>
    <choice>
      <me:fac>.viiij.</me:fac>
      <me:dipl>.viiij.</me:dipl>
    </choice>
  </num>
  <w>
    <choice>
      <me:fac><am>.</am>e<am>.</am></me:fac>
      <me:dipl>e<ex>ðā</ex></me:dipl>
    </choice></w>
  <num>
    <choice>
      <me:fac><sic>.xij.</sic></me:fac>
      <me:dipl><corr>.ix.</corr></me:dipl>
    </choice>
  </num>

```

With this markup it is possible to show the text on the computer screen or in a printed edition in accordance with the suggestions above ([ch. 7.1](#)):

en(n) .viiij. e(ðā) *.ix.

with the corrected form from the manuscript text underneath the edited text:

*.xij.

It is also possible to include information about the person responsible for the correction with the attribute **@resp** and its values.

In this example of a multi-level transcription, the **<sic>** element lies on the *fac*s level, while the **<corr>** element is introduced on the *dipl* level, and perhaps silently on the *norm* level. In a single-level transcription (cf. [ch. 3.3](#)), the **<choice>** element should be used to group the **<sic>** and **<corr>** elements. If, for example, a source has the reading

Please look left now!

in which ‘left’ should be corrected to ‘right’, this would be the appropriate encoding:

```
Please look <choice><sic>left</sic><corr>right</corr></choice> now!
```

In a single-level transcription, the **<choice>** element thus groups **<sic>** and **<corr>** elements, while in a multi-level transcription it groups readings on different levels, i.e. the **<me:fac>**, **<me:dipl>** and **<me:norm>** elements. The **<choice>** element is simply a neutral mechanism to group alternative readings.

7.5 Reference: elements and attributes

Because of the large number of elements and attributes introduced in this chapter, the following reference material is supplied.

7.5.1 Elements

The following is a list of elements to encode non-linguistic features. In the table, ‘level’ indicates the textual level at which the feature should be encoded, if it occurs within a word. No such features should be encoded at the ‘norm’ level, as this level should not include non-linguistic aspects of the text. Some elements are included which are not treated in this chapter. Encoders should refer to the TEI P5 guidelines for more information.

element	level	section	description	attributes
<add>	facs	7.2.1	addition made by scribe outside the normal flow of text	type , hand , place , resp , cert
<corr>	dipl	7.4.3	corrected text inserted by editor	resp , cert
<damage>	facs	N/A	physical damage to the ms	type , hand , extent , agent , degree , resp
	facs	7.2.2	deletion made by the scribe	type , hand , resp , cert , status
<me:expunged>	dipl	7.4.2	identified by editor to be suppressed: spurious or superfluous	type , resp
<gap/>	facs, dipl	7.3.1	illegible text in ms	type , hand , extent , agent , reason , resp
<restore>	facs	N/A	text deleted and then restored	type , hand , resp , cert
<sic>	facs	7.4.3	incorrect text to be marked as such	
<space/>	facs	7.3.1	deliberate space in ms	extent , dim , resp
<supplied>	dipl	7.4.1	text supplied by the editor	type , hand , agent , reason , source , resp
<unclear>	facs	7.3.2	unclear or partially legible text	hand , agent , reason , resp , cert

7.5.2 Descriptive attributes

The following attributes have similar meanings, regardless of which element they are used with.

attribute	used with	content
@agent	damage, gap, supplied, unclear	The cause of damage or illegibility in the ms. Suggested values include 'hole', 'cut off', 'worn'.
@cert	add, corr, del, restore, unclear	The degree of certainty of the editor in identifying the feature being marked up.
@degree	damage	The extent to which the ms is damaged. This attribute should only be used where the text may be read with some confidence.

attribute	used with	content
@dim	space	Indicates whether the space is horizontal or vertical. Only two possible values: 'horizontal' or 'vertical'.
@quantity	damage, gap, space	The physical extent of the feature. An integer representing the approximate number of letters is the recommended unit, although millimetres, minims or other units may also be used.
@hand	add, damage, del, gap, restore, supplied, unclear	The hand responsible for the feature encoded, or in the case of supplied text, the hand responsible for the feature which necessitates the supplied text. This attribute is in the form of an IDREF to an entry in the handDesc in the header.
@place	add	The location of the addition made by the scribe. Suggested values include 'inline' (a space originally left by the scribe), 'supralinear' (above the line), 'infralinear' (below the line), 'left' (in the left margin), 'right' (in the right margin), 'top' (in the top margin), 'bottom' (in the bottom margin), 'verso' (overleaf).
@reason	gap, supplied, unclear	The reason for the gap, unclear text or supplied text in the ms. Suggested values: 'sampling', 'illegible', 'irrelevant', 'cancelled'.
@rend	(all)	Information on how the element is rendered in the source text. This attribute should not be used where the 'type' attribute is more appropriate.
@resp	add, corr, damage, del, expunged, gap, restore, space, supplied, unclear	The editor or transcriber responsible for identifying and describing the feature encoded; or (in the case of expunged, corr, supplied) supplying or altering the text. This attribute is an IDREF referring to the <code>teiHeader</code> .
@source	supplied	The source of the supplied text, for example, another manuscript or an edition.
@status	del	May be used to indicate faulty deletions, e.g. strikeouts which include too much or too little text.
@type	add, damage, del, expunged, gap, restore, supplied	The type of feature being encoded. The typology relates to the element itself, for example 'del' may use the values 'subpunction' or 'strikethrough'.

Ch. 8. Lemmatisation of manuscript text

8.1 Introduction

In [ch. 2.3](#) we suggested that the word, <w>, is a basic unit in any transcription. Each <w> element in a manuscript text can easily be supplied with information about the dictionary entry and the grammatical analysis of the word in question. We recommend that this information is provided by two attributes, **@lemma** for the dictionary entry and **@me:msa** for the grammatical form:

Element / attribute	Contents
<w>	delimits a grammatical word.
@lemma	states the lemma (lexical entry) of the word
@me:msa	states the grammatical (morphosyntactical) form of the word.

It is essential that the lemmatisation of Medieval Nordic manuscript text is done in adherence to the principles developed for handling large corpora in linguistic research. We have found the guidelines provided by [EAGLES \(1996\)](#) to be particularly useful, but have decided to deviate somewhat from these guidelines in order to produce a more self-explanatory, although slightly more verbose, system.

The model provided here is aimed at Medieval Norwegian and Icelandic texts. For Medieval Swedish and Danish texts and also for later Norwegian texts, we can expect a radical levelling in the grammatical system, e.g. in the nominal and verbal inflections. The model provided here will therefore overgenerate when applied to Medieval Swedish and Danish texts, and to late Medieval Norwegian texts.

This chapter is intended as a discussion of the basic principles for lemmatisation and grammatical encoding of manuscript text. It should be read as a suggestion rather than as definite guidelines.

Medieval Nordic texts sometimes include words, phrases or even whole passages in other languages, particularly in Latin. The encoding of such passages is discussed in [ch. 8.7](#) below.

8.2 The attribute @lemma

The element <w> can be supplied with several lexicographical attributes for each word in a transcription. The attribute **@lemma** provides the lexical form of each word based on the entries in standard dictionaries. For Medieval Norwegian and Icelandic texts we suggest that the word-list produced by the Arnamagnæan Commission's [Ordbog over det norrøne prosasprog \(ONP\)](#) at the University of Copenhagen is used to create the lemma base. The attribute would then be marked up as in this example, which states that the word 'hefir' has 'hafa' as its lemma:

```
<w lemma="hafa">hefir</w>
```

Lemmatised texts are useful for any language, and in particular for languages with complex morphology or variable orthography. The morphology of Old Norse is more complex than that of the modern Nordic languages, but not particularly difficult – it is rather like the morphology of Modern German. The orthography, however, was far from fixed, and since many transcriptions are likely to be fairly diplomatic, any lemma may be instantiated by a large number of orthographic forms. For example, the pronoun ‘hann’ has only three forms in the normalised orthography of Old Norse: ‘hann’ (nominative and accusative), ‘hans’ (genitive), and ‘honum’ (dative). In an actual transcription, however, a dozen or more forms may occur, as shown in the table below.

Form	Lemma	Grammatical form
hann	hann	Nominative
han<am>&bar;</am>		
h<am>&bar;</am>		
h<am>&bar;</am>n		
ha scap;	hann	Genitive
hans		
han&slong;		
h<am>&bar;</am>s		
h<am>&bar;</am>&slong;	hann	Dative
honum		
honom		
h<am>&bar;</am>m		

In [ch. 2.3](#) the use of <w> for the encoding of graphic words and information concerning their description is treated. Note the use of entities for special characters, such as ‘&fins;’ and ‘ scap;’, or abbreviations such as ‘&bar;’. These are described in [ch. 5](#) and [ch. 6](#).

As stated in [ch. 3](#), a text may be encoded on a single level of transcription, as exemplified with ‘hefir’ above. If the text is transcribed on more than one level there is no need for any further attributes, since each word is contained within a single <w> element and the attribute is valid for the whole contents:

```
<w lemma="hafa">
  <choice>
    <me:facs>ha&fins;i</me:facs>
    <me:dipl>ha&fins;i</me:dipl>
    <me:norm>hafi</me:norm>
  </choice>
</w>
```

The next example is slightly more complicated since it contains an abbreviation on the facsimile level and a corresponding expansion in the diplomatic level, but the **@lemma** attribute is unchanged:

```
<w lemma="koma">
  <choice>
    <me:fac>co<am>&bar i</am></me:fac>
    <me:dipl>co<ex>m</ex></me:dipl>
    <me:norm>kom</me:norm>
  </choice>
</w>
```

In cases where a graphic word is included partially or completely in the element **<unclear>** this can be encoded within the element **<w>** and be related to the attribute **@lemma**.

```
<w lemma="sv&aacute;">
  <choice>
    <me:fac><unclear reason="faded">s<am>&ra i</am></unclear></me:fac>
    <me:dipl><unclear>s<ex>ua</ex></unclear></me:dipl>
    <me:norm>sv&aacute i</me:norm>
  </choice>
</w>
```

Text included within the element **<supplied>** is not lemmatized. The following example shows how a character, word or phrase that has been supplied is encoded with the element **<w>**, but without any **@lemma** attribute as the text is not transcribed from the manuscript itself.

```
<w>
  <choice>
    <me:fac><supplied reason="illegible" resp="KGJ">lei
    </supplied>kti</me:fac>
    <me:dipl><supplied reason="illegible" resp="KGJ">lei
    </supplied>kti</me:dipl>
    <me:norm>leikti</me:norm>
  </choice>
</w>
```

This means that the forms that are not marked will not be included in the searchable database under the category **@lemma**. We hereby avoid the problem of contamination between forms that are from the manuscript text and forms that have been supplied by a transcriber or encoder of the text. A basic principle is that the lemmatized text should be from the manuscript text.

8.3 The attribute **@me:msa**

The attribute **@me:msa** (for **morphosyntactical analysis**) adds information about the grammatical form of a word. To be able to make this analysis it is necessary to create a model which includes all possible morphological forms of each lemma. As stated above, the model is based on the morphology of Medieval Norwegian and Icelandic, as expounded in standard grammars of Old Norse or ‘norrønt’.

We recommend a scheme in which the attribute **@me:msa** contains a set of name tokens, one for each morphological category. White space separates each name token. We further recommend that the order of the name tokens should be fixed, and that there should be one specific order for each word class, as specified in [ch. 8.5](#) below. For words with inflection, the first token specifies the word class and the following tokens the morphological categories relevant for this specific word class. Words belonging to word

classes with no inflection, such as prepositions and subjunctions, will only receive a single name token for the word class itself. In addition to tokens for morphological categories such as gender, number and case, tokens for inflection class may be added.

Each name token consists of two parts. The first part specifies the category itself and is represented by a single lower-case letter. The second part specifies the value of the category and is given in one or more upper-case letters. As far as possible, mnemonic characters are used, e.g. ‘c’ for ‘case’ and ‘G’ for ‘genitive’. The name token ‘cG’ is thus to be understood as ‘case: genitive’ and is applicable to all words which can be inflected in genitive, such as nouns, adjectives, pronouns/determiners, numerals and verb participles.

In Old Norse, nouns are inflected for gender, number, case and species (definiteness). Below is an example of the mark-up for the word ‘hestum’, dative plural indefinite of the masculine noun ‘hestr’. The **@me:msa** attribute opens with a name token for the word class, ‘xNC’ for ‘noun, common’, moving on to ‘gM’ for ‘gender: masculine’, ‘nP’ for ‘number: plural’, ‘cD’ for ‘case: dative’ and finally ‘sI’ for ‘species: indefinite’.

```
<w lemma="hestr" me:msa="xNC gM nP cD sI">hestum</w>
```

Prepositions, which are not inflected, will receive a much simpler encoding, consisting of a single name token, ‘xAP’, in which ‘x’ denotes word class and ‘AP’ the actual class, prepositions.

```
<w lemma="fyrir" me:msa="xAP">fyrir</w>
```

Old Norse has the most complex morphology of the Nordic vernaculars and is therefore a suitable starting point. For texts with less complex morphology it is simply a case of making a selection of relevant categories from the repertoire in this chapter. Cf. the discussion on zero values in [ch. 8.4.3](#) below.

8.3.1 Invariable properties

Words in inflectional languages exhibit variable and invariable properties. Word class is the prime example of an invariable property, since a word can belong to one and only one word class – the noun ‘hestr’ can not be inflected in adjectival and verbal forms. For nouns, gender is an invariable property – once again, ‘hestr’ can not be inflected in feminine or neutral forms. Adjectives, on the other hand, are inflected in gender, so for this word class gender is a variable property. Other categories, such as case, number, grade etc., are all variable.

Information on inflectional classes can be added to the **@me:msa** attribute, e.g. strong vs. weak verbs, stem classes of nouns etc. These are also invariable properties.

The name tokens will, in any case, make it clear which tokens refer to invariable properties and which refer to variable properties.

8.3.1.1 Word class

Word class is denoted by a name token consisting of the character ‘x’ + an uppercase two-letter abbreviation for each class, including commonly recognised subclasses (such as the division between common and proper nouns). Inevitably, there will be some conflict of categorisation, especially among the pronouns and determiners. They will be discussed in [ch. 8.5](#) below.

Name token	Word class	Inflection
xNC	Noun, common	Yes
xNP	Noun, proper	
xAJ	Adjective	
xPE	Pronoun, personal	
xPQ	Pronoun, interrogative	
xPI	Pronoun, indefinite	
xDP	Determiner, possessive	
xDD	Determiner, demonstrative	
xDQ	Determiner, quantifier	
xPD	Pronoun/ Determiner	
xNA	Numeral, cardinal	
xNO	Numeral, ordinal	
xVB	Verb	
xAV	Adverb	
xAT	Articles	
xAP	Preposition (adposition)	No
xCC	Conjunction, coordinating	
xCS	Conjunction, subordinating	
xIT	Interjection	
xIM	Infinitive marker	
xRP	Relative particle	
xUA	Unassigned	–

8.3.1.2 Inflectional class

Inflectional class is another invariable property and can usually be derived from a combination of the lemma and the word class. Thus, the lemma ‘fara’ belonging to the word class ‘xVB’ (verbs) will be classified as being a strong verb of the 6th class,

according to most grammars of Old Norse. This is information which might be found in a dictionary or a lexicographical database of Old Norse.

If the encoder wishes to include information on the inflectional class we recommend that this is being done by adding to the **@me:msa** attribute a name token consisting of the lowercase character ‘i’ + an uppercase abbreviation for each class. The table below contains examples for the verb class, but can easily be extended to other classes. Incidentally, the distinction between strong and weak inflection also applies to nouns.

Name token	Inflectional class
iST	Strong
iWK	Weak
iRD	Reduplicating
iPP	Preterito-Presentic
etc.	

Since inflectional class is an invariable property of the word there is no compelling reason to specify it as part of the morphosyntactical analysis. The major verb classes listed above are a possible exception, since there are some pair verbs which must be disambiguated by way of inflectional class, e.g. the weak (and transitive) verb ‘brenna’ vs. the homonymuous strong (and intransitive) verb ‘brenna’.

The distinction between strong and weak inflection is an invariable property in verbs and nouns, i.e. a verb or a noun has either weak or strong inflection. For example, the noun ‘armr’ has a strong inflection, while ‘granni’ has weak inflection. What has been termed ‘species’ (or ‘finiteness’) here, is a variable property. This applies to nouns and adjectives, e.g. ‘hestr’ vs. ‘hestrinn’ and ‘hvítr [hestr]’ vs. ‘[inn] hvíti [hestr]’. Cf. [ch. 8.3.2.4](#) below.

8.3.2 Variable properties

The list of variable properties is rather long for an inflectional language such as Old Norse. Note that the very first category in this list, gender, is a borderline case, since it is an invariable (inherent) property for nouns. For other word classes, such as adjectives, pronouns/determiners, numerals, articles and verb participles, it is a variable property. The remaining categories are variable.

8.3.2.1 Gender

This category applies to nouns, adjectives, pronouns/determiners, numerals and verb participles. Gender is denoted by a name token consisting of the lowercase character ‘g’ + an uppercase abbreviation for each gender. The character ‘U’ indicates unspecified cases.

Name token	Value
gM	Masculine
gF	Feminine

Name token	Value
gN	Neuter
gU	Unspecified

Some nouns may have two genders, e.g. ‘hungr’ (hunger), which is either masculine or neutral. For words of this type we suggest using name tokens with more than one value, ‘gMF’, ‘gMN’ and ‘gFN’.

We recommend that gender is ascribed on the basis of standard dictionaries. Even if a text at a certain point may point to a specific gender, e.g. in the collocation ‘mikill hungr’ (meaning that ‘hungr’ is masculine), any disambiguation is of limited value. So rather than trying to distinguish between (a) sure cases of ‘hungr’ being masculine, gM, (b) sure cases of ‘hungr’ being neuter, gN, and (c) ambiguous cases, gMN, we recommend the classification ‘gMN’ in all cases (since this is what the dictionary states).

Name token	Value
gMF	Masculine or Feminine
gMN	Masculine or Neuter
gFN	Feminine or Neuter

For words with three possible genders, we suggest using the character ‘U’, meaning that the value is unspecified, ‘gU’.

8.3.2.2 Number

This category applies to nouns, adjectives, pronouns/determiners and verbs. Number is denoted by a name token consisting of the lowercase character ‘n’ + an uppercase abbreviation for each number. The dual form occurs only in the inflection of personal pronouns. The character ‘U’ indicates unspecified cases.

Name token	Value
nS	Singular
nD	Dual
nP	Plural
nU	Unspecified

8.3.2.3 Case

This category applies to nouns, adjectives, pronouns/determiners and numerals. Case is denoted by a name token consisting of the lowercase character ‘c’ + an uppercase abbreviation for each case. The character ‘U’ refers to words that cannot be specified for case.

Name token	Value
cN	Nominative
cG	Genitive
cD	Dative
cA	Accusative
cU	Unspecified

8.3.2.4 Species

This category applies to nouns and adjectives. Species (or definiteness) is denoted by a name token consisting of the lowercase character ‘s’ + an uppercase abbreviation for each type of species. The character ‘U’ indicates unspecified cases.

In Old Norse, nouns and adjectives can have either indefinite or definite forms, e.g. ‘hestr’ (indefinite noun) vs. ‘hestrinn’ (definite noun) or ‘hvítr [hestr]’ (indefinite adjective) vs. ‘[inn] hvíti [hestr]’ (definite adjective).

Name token	Value
sI	Indefinite
sD	Definite
sU	Unspecified

8.3.2.5 Grade

This category applies to adjectives and adverbs. Grade is denoted by a name token consisting of the lowercase character ‘r’ + an uppercase abbreviation for each grade. The character ‘U’ indicates unspecified cases.

Memory hint: since the character ‘g’ has been reserved for ‘gender’, the character ‘r’ can be interpreted as ‘relative’, which refers to an aspect of the category of grade.

Name token	Value
rP	Positive
rC	Comparative
rS	Superlative
rU	Unspecified

8.3.2.6 Person

This category applies to verbs and some of the pronouns. Person is denoted by a name token consisting of the lowercase character ‘p’ + an uppercase abbreviation for each person. The character ‘U’ indicates unspecified cases.

Name token	Value
p1	1. person
p2	2. person
p3	3. person
pU	Unspecified

8.3.2.7 Tense

This category applies only to verbs. Tense is denoted by a name token consisting of the lowercase character ‘t’ + an uppercase abbreviation for each tense. The character ‘U’ indicates unspecified cases.

Name token	Value
tPS	Present
tPT	Preterite
tU	Unspecified

Preterite-present verbs are classified according to their logical tense, not their historical formation. Thus, ‘veit’ has the present tense of ‘vita’ (even if it has a preterite formation) and ‘vissti’ the preterite tense.

8.3.2.8 Mood

This category applies only to verbs. Mood is denoted by a name token consisting of the lowercase character ‘m’ + an uppercase abbreviation for each mood. The character ‘U’ indicates unspecified cases.

Name token	Value
mIN	Indicative
mSU	Subjunctive
mIP	Imperative
mU	Unspecified

8.3.2.9 Voice

This category applies only to verbs. Voice is denoted by a name token consisting of the lowercase character ‘v’ + an uppercase abbreviation for each type of voice. The character ‘U’ indicates unspecified cases.

Name token	Value
vA	Active
vR	Reflexive
vU	Unspecified

8.3.2.10 Finiteness

This category applies only to verbs. Finiteness is denoted by a name token consisting of the lowercase character ‘f’ + an uppercase abbreviation for each type of finiteness. The character ‘U’ indicates unspecified cases.

Name token	Value
fF	Finite
fI	Infinite (non-finite)
fP	Participle (non-finite)
fU	Unspecified

8.3.2.11 Enclitics

Personal pronouns may be attached to finite verbs, e.g. ‘emk’ for ‘em ek’ or ‘fórtu’ for ‘fórt þú’. From a morphological point of view, this process is similar to the suffixation in definite noun forms, e.g. ‘hestr + inn’ = ‘hestrinn’, or reflexive verb forms, e.g. ‘kalla + s(i)k’ = ‘kallask’. However, it may be argued that the enclitic pronoun retains its character as a word to a larger extent than the suffixed determiner ‘inn’ or the reflexive pronoun ‘s(i)k’. For this reason, we suggest that enclitic forms are encoded with the <seg> element, as suggested in [ch. 2.3.2](#) above.:

```
<seg type="enc">
  <w lemma="vera">em</w>
  <w lemma="ek">k</w>
</seg>

<seg type="enc">
  <w lemma="fara">fort</w>
  <w lemma="&thorn;&uacute;">u</w>
</seg>
```

The segmentation is in several cases open to discussion. Thus, the ‘t’ in ‘fortu’ may be seen as part of the verb form or as part of the pronoun. From a phonological point of view, it is an assimilation product of the final ‘t’ in the verb and the initial ‘þ’ in the pronoun. It is therefore useful to supply these verb and pronoun forms with a marker for enclitication. We suggest a name token ‘eE’ for this purpose, to be used in the @me:msa attribute of both words:

Name token	Value
eE	Enclitic pronoun

This category is only relevant for combinations of a verb and an enclitic pronoun. In all other cases, the name token is simply not used.

8.3.2.12 Government

In the Old Norwegian lemmatised corpus, prepositions are encoded for the case which they govern. This is valuable syntactic information, but it is really not a morphological category. We therefore recommend that prepositions, which have no inflection in Old Norse (or possibly not in any other language), are only encoded for word class in the **@me:msa** attribute, 'xAP'.

However, to accomodate the information provided in the Old Norwegian lemmatised corpus without introducing attributes for syntactic categories we suggest using a name token for government, consisting of the lowercase character 'y' + an uppercase abbreviation for each type of case government. This category would apply to prepositions, verbs and some adjectives.

Name token	Value
yG	Governing Genitive
yD	Governing Dative
yA	Governing Accusative
yU	Unspecified government

In the Old Norwegian lemmatised corpus, also conjunctions (i.e. subjunctions) are encoded for the mood which they govern. This is not a morphological category, but the information can be retained by adding a name token for government, consisting of the lowercase character 'y' + an uppercase abbreviation for each type of mood government.

Name token	Value
yIN	Governing Indicative
ySU	Governing Subjunctive
yU	Unspecified government

8.4 Homography and zero values

Two or more words sometimes have the same spelling, but different meanings. This is usually referred to as 'homography' and it is a basic problem for all morphological analysis. We shall distinguish between two types of homography, external and internal.

The first case must be handled by the **@lemma** attribute, the second by the **@me:msa** attribute.

For the discussion in this chapter, we shall adopt the distinction between **word form**, **grammatical form** and **lemma** (lexeme). The word form is the word as it is spelt in the text, whether normalised or unnormalised. The grammatical form is a specific morphological value of the word, referred to by the attribute **@me:msa**. The lemma is the common denominator for all of these forms, typically given as a dictionary entry and referred to by the attribute **@lemma**.

8.4.1. External homography

External homography means that one grammatical word can be mapped onto two or more lemmata. In some cases the alternative lemmata are different words from a semantic and etymological point of view, such as the feminine noun **þýða** ‘friendship’ in nominative singular and the verb **þýða** ‘interpret’ in infinitive. In all but a few cases, a semantic analysis will disambiguate these forms.

In other cases it is a questions of related words with variant forms, such as the neutral nouns **líf** and **lífi**. In dative singular they happen to have the same form, **lífi**:

Lemma	Word form	Grammatical form
líf	lífi	xNC gN nS cD sI
lífi		

For this case of external homography we recommend encoding each of the possible lemmata in full, using the vertical bar, ‘|’, as delimiter (for the sake of simplicity we are using ‘í’ rather than ‘í’):

```
... <w lemma="líf | lífi" me:msa="xNC gN nS cD sI |  
xNC gN nS cD sI">lífi</w> ...
```

Note that for each possible **lemma** value there must be a corresponding **me:msa** value, even if they happen to be identical (as in this example). Thus, the first possible lemma is ‘líf’ and the corresponding me:msa value is ‘xNC gN nS cD sI’. The second possible lemma is ‘lífi’ and the corresponding me:msa value ‘NC gN nS cD sI’. The general form is thus:

```
... <w lemma="alt.1 | alt.2" me:msa="alt.1 | alt.2">homograph</w> ...
```

A search engine would be able to pick out both ‘líf’ and ‘lífi’ as possible lemmata for ‘lífi’, and also to keep this example separate from unambiguous ones, such as the genitive ‘lífs’, which can only be mapped to the lemma ‘líf’, or the nominative ‘lífi’ which can only be mapped to the lemma ‘lífi’.

8.4.2 Internal homography

Internal homography means that one word form can be mapped onto two or more grammatical words. This is often referred to as syncretism, and is frequently found in many languages, typically as the result of linguistic change (such as phonological mergers). The levelling of the morphological system in Medieval Nordic (except Icelandic) produced a large amount of syncretism.

The feminine noun ‘kona’ is a case in point. It has the same form, ‘konu’, in all three non-nominative (oblique) cases in singular:

Lemma	Word form	Grammatical form
kona	kona	xNC gF nS cN sI
	konu	xNC gF nS cG sI
		xNC gF nS cD sI
		xNC gF nS cA sI

The encoder may choose to see these forms as syncretistic and simply encode case as unspecified for this word, using the value ‘U’:

```
<w lemma="kona" me:msa="xNC gN nS cU sI">konu</w>
```

This encoding entails that the word ‘kona’ has case as a relevant category, but that the exact value has not been determined by the encoder. A search engine would be able to list the form as an example of e.g. a feminine noun in singular, but not as an example of a feminine noun in dative.

In most cases, however, a syntactic or semantic analysis will yield a unique result. For example, in the phrase ‘til konu’ the word form ‘konu’ would be analysed as genitive since the preposition ‘til’ only governs this particular case:

```
<w lemma="til" me:msa="xAP">til</w>
<w lemma="kona" me:msa="xNC gF nS cG sI">konu</w>
```

In another phrase, e.g. ‘fyrir konu’, the encoder might not be willing to make a definitive choice, since the preposition ‘fyrir’ governs both accusative and dative. The encoder might therefore decide to list both alternatives in the **@me:msa** attribute, using the vertical bar as a delimiter:

```
<w lemma="fyrir" me:msa="xAP">fyrir</w>
<w lemma="kona | kona" me:msa="xNC gF nS cA sI | xNC gF nS cD sI">konu</w>
```

Note that also in this case should there be a lemma value specified for each **@me:msa** value, even if the lemma values are identical, i.e.

```
... <w lemma="alt.1 | alt.2" me:msa="alt.1 | alt.2">homograph</w> ...
```

A search engine would be able to pick out this instance of ‘kona’ as a possible example of accusative and of dative. The presence of the delimiter would also make it possible to identify this as an instance of syncretism, so that this example would not be counted among the unambiguous examples of either accusative or dative. The order of the alternatives is arbitrary; the encoding above does not imply that accusative is more likely than dative.

Finally, it should be pointed out that it is a moot question whether ‘konu’ should be seen as a single word form, or as a three homographic word forms representing three distinct grammatical forms, ‘konu-GEN’, ‘konu-DAT’ and ‘konu-ACC’. The answer to this question depends on the morphological analysis of the linguistic stage in question. One might possibly claim, for example, that in Medieval Norwegian case is a relevant distinction to make for all nouns, but that in Late Medieval Norwegian the case distinction

has collapsed, and that the lemma ‘kona’ only has two grammatical forms, the nominative ‘kona’ and the non-nominative (oblique) ‘konu’.

8.4.3 Combinations of external and internal homography

In more complex cases, there may be a combination of external and internal homography. For example, the word form ‘sinni’ may be a dative of the noun ‘sinn’ or it may be either dative or accusative of the noun ‘sinni’. In other words, the combinations are:

Lemma	Word form	Grammatical form
sinn	sinni	xNC gN nS cD sI
sinni		xNC gN nS cD sI
		xNC gN nS cA sI

A unique way of encoding this structure would be to list the three alternatives in such an order that the first lemma value corresponds to the first me:msa value, the second lemma value corresponds to the second me:msa value, and the third lemma value corresponds to the third me:msa value. In other words:

```
... <w lemma="alt.1 | alt.2 | alt.3" me:msa="alt.1 | alt.2 | alt.3">
  homograph</w> ...
```

or

```
... <w lemma="sinn | sinni | sinni" me:msa="xNC gN nS cD sI
  | xNC gN nS cD sI | xNC gN nS cA sI">sinni</w> ...
```

This way of encoding homography is verbose, but it is unambiguous and simple to process.

8.4.4 Zero values

We believe it is convenient to distinguish between two types of zero values in morphological encoding, **not applicable** and **not specified**.

(a) Not applicable

No words have the complete set of morphological categories listed in 8.3 above. For example, although verb participles belong to the verb class, they are not inflected for mood. There is no need to encode participles for ‘mood:zero’ – it is sufficient to leave out the name token for mood. In other words, the absence of the name token implies that mood is not a relevant category for the word in question.

(b) Not specified

In other cases, a word is inflected for a certain category, but the encoder is not able to specify a value. This may be the case with some proper nouns, for which no gender can be given. This is a different type of ‘zero’ value, and we therefore suggest to indicate these cases with the character ‘U’ to be read as ‘unspecified’. An example:

```
<w lemma="Byblos" me:msa="xNP gU">Byblos</w>
```

This encoding entails that the word in question is a noun and that it does have a gender (it is thus not a case of non-applicability), but that the encoder does not know which gender that would be.

Another example: In Old Norse, there is no gender distinction in genitive or dative plural of any adjective or determiner. It is possible to encode adjectives and determiners for gender based on concord with a noun (if there happens to be one), so that in a genitive plural phrase like ‘spakra manna’ the adjective ‘spakra’ might be ascribed masculine gender on the basis of the noun *maðr*, which is masculine. From experience, we know that this is time-consuming and not really informative encoding. A less specified option would be to use the character ‘U’ to indicate non-specification:

```
<w lemma="spakr" me:msa="xNC gU nP cG sI">spakra</w>
```

A search engine would be able to pick out ‘spakra’ as an example of an adjective in genitive plural, but not as an adjective in masculine (or feminine, or neutral) gender.

8.5 General model for Medieval Nordic

This chapter contains examples of encoding for each word class in a Medieval Nordic text. As pointed out in the introduction, the model is based on the grammar of Old Norse, and will thus be more detailed than needed for Old Danish and possibly also for Old Swedish. For these linguistic stages and for Middle Norwegian, the model can be scaled down, but we believe that the general framework will still be useful.

We strongly recommend a fixed order of name tokens for each class, beginning with the name token for the word class itself. Note, however, that non-relevant categories can simply be left out, as recommended in [ch. 8.4.3](#) above. Thus, for late Medieval texts the encoding of many word classes may be shorter than the one exemplified here.

8.5.1 Nouns (NC and NP)

Nouns are divided into two subgroups, **common noun** (xNC) and **proper nouns** (xNP). They are further encoded for **gender**, **number**, **case** and **species**

Example: Encoding of the noun ‘ymr’ in the phrase ‘þá heyrðu þeir ym mikinn ok gny’:

```
<w lemma="ymr" me:msa="xNC gM nS cA sI">ym</w>
```

Word class	Gender	Number	Case	Species
xNC xNP	gM gF gN gU	nS nP nU	cN cG cD cA cU	sI sD sU

Possibly, a separate name token for oblique case, ‘cO’, might be added. The concept of the oblique case covers all non-nominative cases, i.e. genitive, dative and accusative.

8.5.2 Adjectives (AJ)

Adjectives are encoded for **grade**, **gender**, **number**, **case** and **species**.

Example: Encoding of the adjective ‘langr’ in the phrase ‘seint er um langan veg at spyrja tíðenda’:

```
<w lemma="langr" me:msa="xAJ rP gM nS cA sI">langan</w>
```

Word class	Grade	Gender	Number	Case	Species
xAJ	rP rC rS rU	gM gF gN gU	nS nP nU	cN cG cD cA cU	sI sD sU

Note that in the comparative form, adjectives only have weak (indefinite) inflection. Nevertheless, we recommend that they are encoded for species, ‘sI’, throughout. Also note that some adjectives have defect comparison, but we still recommend that they are encoded for grade.

8.5.3 Pronouns proper (PE, PQ and PI)

In recent grammars the traditional category pronoun is usually divided into **pronouns** in a strict sense (words replacing a noun) and ‘determiners’ (adjunct words), and that is our recommendation as well, cf. [ch. 8.5.3](#) and [8.5.4](#) below. However, in some projects (i.e. the Old Norwegian lemmatised corpus) there is only a single category pronoun, and we have therefore added in [ch. 8.5.5](#) a combined category, **pronouns and determiners**.

Although pronouns in the strict sense of ‘words replacing a noun’ is a smaller category than the traditional one, there are nonetheless three distinct sub-categories. In the following these are treated separately to provide an over-view.

8.5.3.1 Personal pronouns (PE)

Personal pronouns are encoded for **person**, **gender**, **number** and **case**. Note that only personal pronouns in 3. person have a gender distinction; for pronouns in 1. and 2. person this category is simply left out.

Example: Encoding of the personal pronoun ‘vit’ in the phrase ‘vit erum fegnir’ (leaving out the gender category):

```
<w lemma="vit" me:msa="xPE p1 nD cN">vit</w>
```

Word class	Person	Gender	Number	Case
xPE	p1 p2 p3 pU	gM gF gN gU	nS nD nP nU	cN cG cD cA cU

8.5.3.2 Interrogative pronouns (PQ)

Interrogative pronouns are encoded for **gender**, **number** and **case**. Memory hint: in the name token ‘xPQ’ the last character stands for ‘question’.

Example: Encoding of the interrogative pronoun ‘hverr’ in the phrase ‘Frigg spurði hverr sá v#ri með ásum’:

```
<w lemma="hverr" me:msa="xPQ gM nS cN">hverr</w>
```


Word class	Gender	Number	Case
xPQ	gM gF gN gU	nS nD nP nU	cN cG cD cA cU

8.5.3.3 Indefinite pronouns (PI)

Indefinite pronouns are encoded for **gender**, **number** and **case**.

Example: Encoding of the indefinite pronoun ‘einnhverr’ in the phrase ‘vill hann taka til at þreyta drykkju við einhvern mann’:

```
<w lemma="einnhverr" me:msa="xPI gM nS cA">einhvern</w>
```

Word class	Gender	Number	Case
xPI	gM gF gN gU	nS nP nU	cN cG cD cA cU

8.5.4 Determiners (DP, DD and DQ)

The contents of the word class determiners vary between languages and grammars. In the present analysis, determiners comprise a large part of the traditional word class pronouns (as defined in many grammars of Old Norse) with the exception of pronouns proper. Determiners have three subcategories: possessives, demonstratives and quantifiers.

Note that articles and numerals are often analysed as determiners, but these traditional classes have been retained here.

8.5.4.1 Possessives (DP)

Possessives are encoded for **gender**, **number** and **case**.

Example: Encoding of the possessive ‘sinn’ in the phrase ‘hann hugðisk þá at reyna afl sitt’:

```
<w lemma="sinn" me:msa="xDP gN nS cA">sitt</w>
```

Word class	Gender	Number	Case
xDP	gM gF gN gU	nS nD nP nU	cN cG cD cA cU

8.5.4.2 Demonstratives (DD)

Possessives are encoded for **gender**, **number** and **case**.

Example: Encoding of the demonstrative ‘hinn’ in the phrase ‘hitt fjall er hátt’:

```
<w lemma="hinn" me:msa="xDD gN nS cN">hitt</w>
```

Word class	Gender	Number	Case
xDD	gM gF gN gU	nS nD nP nU	cN cG cD cA cU

8.5.4.3 Quantifiers (DQ)

Quantifiers are encoded for **gender**, **number** and **case**. This category may overlap with Indefinite pronouns.

Example: Encoding of the demonstrative ‘mar(g)t’ in the phrase ‘mart folk hefir komit hér’:

```
<w lemma="margr" me:msa="xDQ gN nS cN">mart</w>
```

Word class	Gender	Number	Case
xDQ	gM gF gN gU	nS nD nP nU	cN cG cD cA cU

8.5.5 Pronouns/determiners (PD)

This is the traditional category of ‘pronoun’, as defined in the grammars of e.g. [Noreen 1923](#) and [Iversen 1973](#). From a inflectional point of view this is a heterogenous category, but since it has been used in much lexicographical work, it is given here as an alternative to the two classes pronouns proper (8.5.3) and determiners (8.5.4).

Pronouns/determiners are encoded for **person** (only personal pronouns), **gender**, **number** and **case**.

Example: Encoding of the pronoun ‘engi’ in the phrase ‘ormrinn er sl#gari en ekki annat kvikendi’ (no name token for person, since this category is not relevant):

```
<w lemma="engi" me:msa="xPD gN nS cN">ekki</w>
```

Word class	Person	Gender	Number	Case
xPD	p1 p2 p3 pU	gM gF gN gU	nS nD nP nU	cN cG cD cA cU

8.5.6 Numerals (NA and NO)

The numerals are divided into two sub-categories: ‘cardinals’ (NA) and ‘ordinals’ (NO). The character U is used for ‘unspecified’, so that ‘xNU’ comprises both cardinal and ordinal numerals - the case for the Old Norwegian lemmatised corpus.

Numerals are encoded for **gender** (only the cardinals 1-4), **number** (only ordinals), **case**, and **species** (only relevant for the numerals ‘einn’, ‘fyrstr’, and ‘annarr’). Memory hint: since the obvious candidate ‘NC’ for ‘numeral, cardinal’ has been reserved for ‘nouns, common’, the character ‘A’ in ‘NA’ can be seen as referring to the vowel ‘a’ which occurs two times in the word ‘cardinal’.

The numerals *hundrað* ‘one hundred (and twenty)’ and *þúsund* ‘one thousand (two hundred)’ are treated as nouns.

Example: Encoding of the numeral ‘sjaundi’ in the phrase ‘in sjaunda borg’:

```
<w lemma="sjaundi" me:msa="xNO gF nS cN sD">sjaunda</w>
```

Word class	Gender	Number	Case	Species
xNA xNO xNU	gM gF gN gU	nS nP nU	cN cG cD cA cU	sI sD sU

8.5.7 Articles (AT)

In recent grammars the traditional word class ‘articles’ is usually classified as part of the word class ‘determiners’. However, in some projects (i.e. the Old Norwegian lemmatised corpus) articles are treated as a separate class, and we suggest that as an alternative they may be classified as such.

Articles are encoded for **gender**, **number**, **case**, and **species**.

Example: Encoding of the article ‘einn’ in the phrase ‘ein kona’:

```
<w lemma="einn" me:msa="xAT gF nS cN sI">ein</w>
```

Word class	Gender	Number	Case	Species
xAT	gM gF gN gU	nS nP nU	cN cG cD cA cU	sI sD sU

8.5.8 Verbs (VB)

Verbs are either **finite** or **infinite**. In the former category, they are inflected for **tense**, **mood**, **person**, **number** and **voice**. In the latter category, participles are basically inflected as adjectives, while infinitives have a very restricted inflection. For practical reasons, we recommend that finite and infinite forms are treated separately.

8.5.8.1 Finite forms

Finite verbs are encoded for **tense**, **mood**, **person**, **number**, and **voice**. Optionally, verbs may be encoded for **inflectional class**. This may prove practical since Old Norse has some ‘pair verbs’ with identical lemmatic forms such as the strong verb ‘brenna’ and the weak verb ‘brenna’. In the Old Norwegian lemmatised corpus, verbs are divided into four inflectional classes, as exemplified in the table below.

Example: Encoding of the verb ‘telja’ in the phrase ‘hon taldi’ (leaving out inflectional class):

```
<w lemma="telja" me:msa="xVB fF tPT mIN p3 nS vA">taldi</w>
```

Word class	Finiteness	Tense	Mood	Person	Number	Voice	Infl. class
xVB	fF	tPS tPT tU	mIN mSU mIP mU	p1 p2 p3 pU	nS nP nU	vA vR vU	iST iWK iRD iPP iU

8.5.8.2 Infinite forms

Infinite forms are either participles or infinitives, and may be distinguished by the name token **finiteness** with ‘fP’ for participles and ‘fI’ for infinitives.

(a) Participles

Participles are inflected for the verbal categories **tense** and **voice**, and for the nominal categories **gender**, **number**, **case** and **species** and **voice** (in supinum). Optionally, participles may be encoded for **inflectional class**.

Note that present participles only have weak (definite) declension. Preterite (perfect) participles usually have strong (indefinite) declension, but may sometimes occur with weak (definite) forms. Voice is only relevant for supinum, cf. e.g. ‘hann hefir kallat’ vs. ‘hann hefir kallazk’.

Example: Encoding of the verb ‘koma’ in the phrase ‘hann er kominn’:

```
<w lemma="koma" me:msa="xVB fP tPT vA gM nS cN sI">kominn</w>
```

Word class	Finiteness	Tense	Voice	Gender	Number	Case	Species	Infl. class
xVB	fP	tPS tPT tU	vA vR vU	gM gF gN gU	nS nP nU	cN cG cD cA cU	sI sD sU	iST iWK iRD iPP iU

(b) Infinitives

Infinitives are inflected only for the verbal categories **tense** and **voice**, and **tense** only applies to three verbs, ‘munu’, ‘skulu’ and ‘vilja’ (which have preterital forms). Optionally, participles may be encoded for **inflectional class**.

Example: Encoding of the verb ‘fara’ in the phrase ‘hann mun fara’ (with optional information on inflectional class):

```
<w lemma="fara" me:msa="xVB fI tPS vA iST">fara</w>
```

Word class	Finiteness	Tense	Voice	Infl. class
xVB	fI	tPS tPT tU	vA vR vU	iST iWK iRD iPP iU

8.5.9 Adverbs (AV)

Adverbs are only encoded for **grade**.

Example: Encoding of the adverb ‘sterkliga’ in the phrase ‘hann svaf ok hraut sterkliga’:

```
<w lemma="sterkliga" me:msa="xAV rP">sterkliga</w>
```

Word class	Grade
xAV	rP rC rS rU

Note that some adverbs have defect comparison, but we still recommend that they are encoded for grade.

8.5.10 Prepositions and particles (AP and VP)

‘Prepositions’ are not inflected and only encoded for word class, xAP. The latter is an abbreviation for ‘adposition’, which is the hyponymous term for ‘preposition’ and ‘postposition’ (found in e.g. Japanese, but not in the Nordic languages).

Example: Encoding of the preposition ‘at’ in the phrase ‘koma þeir at kveldi til eins búanda’:

```
<w lemma="at" me:msa="xAP">at</w>
```

There is seldom any doubt about the word class for prepositions in prepositional phrases like ‘í hendi’, ‘á landi’, ‘til þings’, etc. However, when prepositions appear without complementation (in absolute position) or as verbal particles, it is convenient to have an alternative word class. We suggest xVP for this use of prepositions.

Word class	Specification
xAP	all prototypical prepositions
xVP	in absolute or adverbial use (e.g. as verbal particles)

The words ‘of’ and ‘um’ are frequently used as so-called expletive particles in Eddic poems. This usage is so specific that many encoders would like a separate class for this type. See [ch. 8.3.2.15](#) below

As stated in [8.3.2.12](#) above, prepositions in the Old Norwegian lemmatised corpus are encoded for the case they govern. Using the name token ‘y’ + case, the example above would receive this encoding:

```
<w lemma="at" me:msa="xAP yD">at</w>
```

Word class	Government
xAP	yN yG yD yA yU

8.5.11 Conjunctions and subjunctions (CC and CS)

In recent grammars, the traditional word class ‘conjunctions’ is usually divided into two separate classes, ‘conjunctions’ (e.g. ‘ok’, ‘en’) and ‘subjunctions’ (e.g. ‘at’, ‘ef’). The former category connects phrases on the same syntactical level, while the latter category typically introduces clauses. In traditional terminology, this is reflected in the subdivision of conjunctions into ‘coordinating’ and ‘subordinating’. We recommend making a distinction between conjunctions proper = coordinating conjunctions (xCC) and subjunctions = subordinating conjunctions (xCS).

However, in some schemes (i.e. the Old Norwegian lemmatised corpus) only a single word class ‘conjunctions’ is recognised. In that case, the word class may be designated ‘xCU’ using the character ‘U’ for ‘unspecified’.

Example: Encoding of the conjunction ‘ok’ in the phrase ‘Logi hafði etit slátr allt ok beinin með’:

```
<w lemma="ok" me:msa="xCC">ok</w>
```

Example: Encoding of the subjunction ‘at’ in the phrase ‘hon sagði at Baldr hafði þar riðit’:

```
<w lemma="at" me:msa="xCS">at</w>
```

Word class
xCC xCS xCU

As stated in [8.3.2.12](#) above, conjunctions in the Old Norwegian lemmatised corpus are encoded for the mood they govern. This information can be retained by adding a name token for government, consisting of the lowercase character ‘y’ + an uppercase abbreviation for mood.

Word class	Government
xCU	yIN ySU yU

8.5.12 Interjections (IT)

‘Interjections’ are not inflected and only marked for word class, xIT.

Word class
xIT

8.5.13 Infinitive marker (IM)

The **infinitive marker** is not inflected and encoded as xIM. In Old Norse it usually has the form ‘at’.

Word class
xIM

8.5.14 Relative particle (RP)

The *relative particle* is not inflected and only marked as xRP. In Old Norse it usually has the form ‘er’ or ‘sem’. Some grammarians would classify the relative particle as a subjunction, while others tend to look upon it as a pronoun.

Word class
xRP

8.5.15 Expletive particle (EX)

The *expletive particles* ‘of’ and ‘um’ are frequently found in Eddic poems. From one point of view, they can be seen as prepositions in absolute position. However, the specific usage in Eddic poems has led many grammarians to distinguish them from the prepositions ‘of’ and ‘um’. We suggest that they are classified as expletive particles, xEX.

Word class
xEX

8.5.16 Unassigned (UA)

Some words are corrupt, difficult to analyse, belong to another language or are for other reason indeterminate. These words are marked as unassigned, xUA. See, however, the discussion of non-Nordic words in [ch. 8.7](#) below.

Word class
xUA

8.6 Specifications for Old Norse

In the previous chapter, we have given a few alternative analyses, especially the choice between a broad class of pronouns and a smaller class of pronouns and a new class of determiners. We have also pointed out that Old Swedish and particularly Old Danish texts may require a simpler analysis. There is thus a need for further specification. This chapter will deal with Old Norse, i.e. Old Icelandic up to ca. 1550 and Old Norwegian up to ca. 1350. This is the same period as defined by [Ordbog over det norrøne prosasprog](#).

8.6.1 Normalised orthography

There is some variation in the normalised orthography of Old Norse in standard grammars, dictionaries and editions. We recommend that the orthography of the [ONP](#) dictionary in Copenhagen is taken as normative, irrespective of whether the source is Old Icelandic or Old Norwegian. The lemma is above all an address, and as such there should be no variation. Thus, in a Old Norwegian text, the word ‘hnakki’ might be normalised to ‘nakki’ (or even ‘nakke’) in an edition, but the lemma should be ‘hnakki’. Otherwise, Norwegian and Icelandic examples of this word will appear under two different lemmata, ‘nakki’ and ‘hnakki’.

The main points in the ONP orthography are the following:

1. All long vowels have accents, including ‘#’ (not just ‘æ’) and ‘#’ (not ‘œ’).
2. The asyllabic semivowel is spelt ‘j’, not ‘i’, e.g. ‘jafn’, ‘hjarta’.
3. The privative prefix is spelt ‘ó-’, e.g. ‘ójafn’.
4. No lengthening of stressed vowels in words like ‘sjalfr’ and ‘holmi’.
5. The consonant cluster ‘pt’ should be rendered with ‘ft’, thus ‘oft’ and ‘eftir’ rather than ‘opt’ and ‘eptir’.

The last point is a recent decision by ONP. An updated list of lemmata is kept by ONP, and should be consulted before finalising a lemmatisation.

8.6.2 Choice of lemma

We recommend that the lemmatisation is coordinated with the list of lemmata in [ONP](#). At present, three volumes have been published (a-em), and in addition, the dictionary has made available a complete list of planned lemmata for the remaining volumes.

Alternative lemmata. In some cases, ONP has two or more lemmata, e.g. ‘blóðigr, blóðugr’. We recommend using the first lemma.

Hypothetical lemmata. Some lemmata are not attested in the sources. This applies to a few verbs with no known infinitive, a few adjectives with no known positive form,

and some nouns with no known singular form. For example, ONP lists the singular noun forms ‘ørlag’ and ‘skap’ rather than the plural forms ‘ørl#g’ and ‘sk#p’. However, ONP has not listed the hypothetical singular form ‘dur’ as lemma for the attested plural form ‘dyrr’ (door). We have identified a few words where we would like to deviate from ONP:

ørl#g (xNC)

sk#p (xNC)

This list is preliminary and will be supplied.

8.6.3 Word classes

The word classes in [ONP](#) should be taken as normative. In the great majority of cases, there will be no doubt as the word class identification. Some problems remain, though. This is a list of the most frequent ones.

Pronouns vs. determiners. Recent grammars make a distinction between pronouns in the original sense of the word (pro nomen) and determiners. This would also apply to Medieval nordic, from a syntactical point of view as well as a morphological point of view. The inflection of determiners is clearly different from that of the pronouns. However, there is a long-standing tradition for a broad definition of the word class pronouns, and since this is used in standard dictionaries like *Norrøn ordbok* and the *ONP dictionary*, we recommend using the wordclass xPD in the encoding of Old Norse.

Prepositions vs. adverbs. Prepositions in absolute position (i.e. with no complementation) can be analysed as adverbs or verbal particles. We recommend that prepositions with complementation, e.g. ‘í hendi’, ‘til matar’, ‘undir honum’, are classified as xAP, while prepositions without any complementation are classified as xVP. This is a simple rule, and there should be no need for the encoder to distinguish in the latter case between cases where a complementation can be recovered from the context (i.e. prepositions in absolute positions) and cases where there is no obvious complementation (i.e. prepositions as verbal particles). Finally, the expletive particles ‘of’ and ‘um’ in Eddic poems should be recognised as a class of its own, xEX.

Adjectives in adverbial usage. Adjectives in neuter are often used as adverbs, e.g. ‘hann kallaði hátt’ (he called loudly), in which the adjective ‘hár’ has the form neuter singular accusative, i.e. xAJ xRP gN nS cA. Some encoders would like to indicate the adverbial usage by encoding xAJ xRP gN nS cA | xAV. However, we believe that the simplest solution is to encode the adjective as an adjective, and leave the rest for a syntactical analysis of the text. In other words, only xAJ xRP gN nS cA.

Supinum. In periphrastic constructions, the verb ‘hafa’ is typically followed by supinum, e.g. ‘hann hefir keypt hús’ (he has bought a house). From a morphological point of view, this form is identical with the perfect participle in neuter singular accusative, i.e. xVB fP tPT gN nS cA, and we would recommend to analyse supinum this way. Note that this analysis also applies to the older construction of verb + object + object predicative, e.g. ‘hann hefir hús keypt’ (literally, he has a house in bought condition).

Cardinal numbers. With the exception of ‘einn’, which has plural forms, there is no need to encode cardinal numbers (i.e. ‘tveir’, ‘þrír’, ‘fjórir’, ‘fimm’, etc.) for number, nP. For cardinal numbers above one, plural is inherent.

Roman numerals. Roman numerals are frequent in Medieval Nordic texts, and should be encoded as numbers using the <num> element, e.g. <num>.iv.</num>. They should not be lemmatised. Cf. the discussion in [ch. 2.4.2](#) above.

Participles vs. adjectives. If a participle can be referred to a verb, the infinitive should be used as the lemma. Thus, ‘búa’ should be the lemma for ‘búinn’, even if this participle is on the verge of being lexicalised as an adjective in Old Norse. For a participle like ‘ítrborinn’ there is no corresponding verb ‘ítrbera’, so ‘ítrborinn’ should be chosen as lemma and the word class must be adjective, xAJ xRP gN nS cM sI.

As a rule, we recommend encoders to avoid duplication of words. So rather than distinguishing between the numeral ‘einn’, the pronoun ‘einn’ and the article ‘einn’, we recommend mapping this word to a single word class. Only in cases where there is a morphological distinction, should potentially homonymuous words be disambiguated. One example is the verb ‘brenna’, which is inflected as a weak verb when transitive (‘hann brenndi húsit’), and as a strong verb when intransitive (‘húsit brann’). Inevitably, there will be some variance between encoded texts in this matter, but as long as the same lemma has been used, this should not cause any major problems. Users should, however, be aware of some recurring borderline cases:

Lemma	Recommended word class	Alternative word class(es)
einn	xPD	xAT, xNU
fyrstr	xAJ (superlative of ‘fyrr’)	xNU, xPD
annarr	xPD	xNU
inn/enn	xPD	xAT
allr	xPD	xAJ
margr	xPD	xAJ

8.7 Lemmatisation of non-Nordic material

The dominant language in a transcription should be specified as an attribute to the <text> element. For a Menota transcription, that will typically be one of the Medieval Nordic languages. In this example, the text is specified as Old Swedish (‘osw’):

```
<text xml:lang="osw">
  <body>The whole text of the source comes here.</body>
</text>
```

If there is only one language in the text, no further specification is needed. If there are words, phrases or passages in another language, they should be set out by the @xml:lang attribute, preferably one for each word. Since the other language most likely will have a different morphology from Medieval Nordic (in the case of Latin and Greek, a more complex one) we recommend a simplified morphosyntactical analysis, perhaps only identifying the word class. For example, the phrase ‘per omnia saecula saeculorum’ might be encoded in this manner:

```
<w lemma="per" me:msa="xAP" xml:lang="lat">per</w>
<w lemma="omnis" me:msa="xPD" xml:lang="lat">omnia</w>
<w lemma="saecula" me:msa="xNC" xml:lang="lat">saecula</w>
<w lemma="saecula" me:msa="xNC" xml:lang="lat">saeculorum</w>
```

If there is a lengthy passage in another language, the attribute can also be given at a higher level in the encoding, e.g. to a **<div>** element.

All **@xml:lang** attributes should be defined in the header. This is part of the **<profileDesc>** element, which must contain a list of all languages referred to in the encoded text. We recommend this standard set of Medieval Nordic languages plus Greek and Latin:

```
<langUsage>
  <language ident="oic">Old Icelandic</language>
  <language ident="onw">Old Norwegian</language>
  <language ident="oda">Old Danish</language>
  <language ident="osw">Old Swedish</language>
  <language ident="lat">Latin</language>
  <language ident="grc">Ancient Greek</language>
</langUsage>
```

Note that the Profile Description may list more languages than actually referred to in the text.

The three-letter language codes for Latin and Ancient Greek are conformant with the ISO 639-2 standard, while the codes for the Medieval Nordic languages are not. ISO 639-2 only has 'non' for Old Norse, which in our view is not sufficient.

Ch. 9. Additional features: Names and metrical encoding

9.1 Encoding of names

Medieval manuscripts contain an abundance of names: personal names for historical, fictional and mythological beings, place names relating to historical as well as to mythological locations, animal names for domestic, wild and mythological creatures, and artefact names for weapons, ships, buildings, etc. In the following, the encoding of names is presented according to the recommendations in [ch. 13 ‘Names, Dates, People, and Places’](#) of the TEI P5 Guidelines.

The basic element for encoding names of any type is `<name>`, supplied with the `@type` attribute.

Elements & attributes	Explanation
<code><name></code>	contains a name, i.e. a proper noun or a noun phrase
<code>@type</code>	Indicates what type of name it is

Some examples from Old Norse sources:

```
<name>Egill Skallagrímsson</name>
<name>Borg</name>
<name>Sleipnir</name>
<name>Skiðblaðnir</name>
```

A distinction may be drawn between e.g. personal names, place names, animal names and artefact names by using the `@type` attribute:

```
<name type="person">Egill Skallagrímsson</name>
<name type="place">Borg</name>
<name type="animal">Sleipnir</name>
<name type="artefact">Skiðblaðnir</name>
```

For many encoders, the `<name>` element is all that is needed for a simple identification of names in the text. The `@type` attribute may be added simultaneously or at a later stage.

For a more detailed encoding of personal names and place names, we recommend using the elements `<persName>` and `<placeName>` respectively. They should be treated as strictly equivalent with the `<name>` element and the `@type` attribute with the values ‘person’ and ‘place’:

```
<name type="person">Egill Skallagrímsson</name>
= <persName>Egill Skallagrímsson</persName>

<name type="place">Borg</name>
= <placeName>Borg</placeName>
```

The element `<persName>` can contain several other elements and can thus be used for a more detailed name analysis, e.g. for making a distinction between forenames, patronymica and surnames. This level of detail is recommended in the header (cf. [ch. 10](#) below), and will also be necessary for any encoding of a text which should make the basis for a name index. In a similar way, the element `<placeName>` can contain elements for

various types of geographical locations. For both elements, TEI P5 offers an additional tagset, which will be discussed and exemplified below.

9.1.1 Personal names

Personal names may be divided into several categories, depending on the source and the naming conventions of the time.

Elements & attributes	Explanation
<code><persName></code>	The name of a person, consisting of one or more words.
<code><forename></code>	The first name of a person.
<code><addName></code>	An additional name of a person.
<code>@type</code>	Indicates whether the name is a patronym, a metronym, a nickname or an epithet.
<code><surname></code>	The family name of a person, excluding patronyms and metronyms.
<code><roleName></code>	The name for a role held by a person.
<code>@type</code>	Indicates the role of the person, e.g. in the form of a title.

Forenames and patronymica

Forenames are encoded with the element `<forename>` contained in the `<persName>` element:

```
<persName>
  <forename>Egill</forename>
</persName>
```

We recommend that patronymica, i.e. a father's name such as Haraldsson or Haraldsdóttir, should be encoded with the element `<addName>` and the `@type` attribute set to 'patronym':

```
<persName>
  <forename>Egill</forename>
  <addName type="patronym">Skallagrímsson</addName>
</persName>
```

It could probably be questioned whether Egil Skallagrímsson should be considered a historical or a fictional person when he appears in *Egils saga Skallagrímssonar*. The `@type` attribute might be used to specify this:

```
<persName type="historical">
  <forename>Egill</forename>
  <addName type="patronym">Skallagrímsson</addName>
</persName>
```

Mythological and legendary names could be seen as forming a group, once again using the `@type` attribute to establish the category. This example is found in the *Poetic Edda*:

```
<persName type="mythological">
  <forename>Loki</forename>
  <addName type="metronym">Laufeyjarson</addName>
```

```
</persName>
```

As Loki has the second name from his mother Laufey, the value of the attribute is ‘metronym’ (meaning that it is derived from the name of the person’s mother).

Surnames

In the Scandinavian and Old Icelandic material, surnames were introduced in the 14th and 15th centuries, in most cases for people belonging to the elite. These names could be encoded with the **<surname>** element:

```
<persName>
  <forename>Bengt</forename>
  <addName type="patronym">Jönsson</addName>
  <surname>Oxenstierna</surname>
</persName>
```

Nicknames, Epithets and Titles

Medieval texts contain an abundance of nicknames, epithets and titles. With the additional tagset these could be treated in the same manner as forenames and surnames, with the elements **<addName>** and **<roleName>**. In *Egils saga Skallagrímssonar*, a man named Þorgils gjallandi ‘the screaming’ appears. His nickname might be encoded as a type of **<addName>**:

```
<persName>
  <forename>Þorgils</forename>
  <addName type="nickname">gjallandi</addName>
</persName>
```

Epithets indicating the providence of a person should also be encoded with the **<addName>** element, specified by the **@type** attribute. Thus, Eyvindr austmaðr Bjarnarson in *Njáls saga* has ‘austmaðr’ as an epithet and ‘Bjarnarson’ as his patronym:

```
<persName>
  <forename>Eyvindr</forename>
  <addName type="epithet">austmaðr</addName>
  <addName type="patronym">Bjarnarson</addName>
</persName>
```

Titles such as ‘konungr’, ‘jarl’ and ‘hersir’ could be encoded by using the **<roleName>** element in addition to the **<addName>** element:

```
<persName>
  <forename>Óláfr</forename>
  <roleName type="political">konungr</roleName>
  <addName type="patronym">Tryggvason</addName>
</persName>
```

The **@type** attribute indicates that the title is considered to refer to the political system of the Old Norwegian society.

9.1.2 Place names

The encoding of place names should be rather straight-forward, but there are a few things that need to be mentioned and exemplified. The starting point is the above-mentioned encoding of all place names with the element **<name>** and an attribute **@type** with the value ‘place’:

```
<name type="place">Borg</name>
```

A more detailed encoding is achieved by the additional tagset. We recommend that this tagset is used in cases when place names are to be more thoroughly encoded:

Elements & attributes	Explanation
<placeName>	A name of a specific location.
<settlement>	The name of the smallest component of a place name expressed as a hierarchy of geo-political or administrative units.
<region>	Larger or administratively superior to the settlement and smaller or administratively less important than the country.
<country>	Larger or administratively superior to the region.

The additional element **<settlement>** adds information about administrative units, i.e. farms, villages or cities. In this example, we have chosen Skara, one of the oldest cities of modern-day Sweden:

```
<placeName>
  <settlement>Skara</settlement>
</placeName>
```

A region is a larger administrative unit than the district or settlement, as e.g. the province of Västergötland, in the medieval material referred to as ‘Vestra Gautland’:

```
<placeName>
  <region>Vestra Gautland</region>
</placeName>
```

The region of Västergötland forms a part of the country referred to as Svíþjóð in medieval texts:

```
<placeName>
  <country>Svíþjóð</country>
</placeName>
```

9.1.3 Other names

There are other groups of names that might be singled out in the encoding. In the following, we present suggestions as to how animal names and names of artefacts could be treated. The **@type** attribute is used liberally to enhance searchability.

Animal names

Medieval texts contain names for horses, dogs and other domestic animals. These could be encoded with the **<name>** element and the **@type** attribute. On a basic level we would suggest that all animal names are marked with the attribute value ‘animal’:

```
<name type="animal">Freyfaxi</name>
```

It should also be possible to give more specific information in the **@type** attribute. The name ‘Freyfaxi’ might be specified as the name of a horse:

```
<name type="horse">Freyfaxi</name>
```

There are also names of animals in relation to myths and legends. For example, Sigurðr Fáfnisbani had a horse named Grani. If we wish to encode a mythological or legendary

name this could be done by specifying the value ‘horse’ of the **@type** attribute, and ‘legendary’ in the **@subtype** attribute:

```
<name type="horse" subtype="legendary">Grani</name>
```

There is no element **<animalName>** on par with **<persName>** and **<placeName>**. For all but the most detailed encodings, we suggest that the type value ‘animal’ is sufficient.

Artefact names

There are a number of names for artefacts in the medieval material, e.g. for weapons and ships. These names could be encoded in the same fashion as the ones described above. The ship belonging to the god Freyr in *Snorra Edda* could then be encoded as follows:

```
<name type="artefact">Skiðblaðnir</name>
```

This artefact could be specified as a ship:

```
<name type="ship">Skiðblaðnir</name>
```

And it could further be encoded as a mythological name:

```
<name type="ship" subtype="mythological">Skiðblaðnir</name>
```

There is no element **<artefactName>**. In most cases, we believe that the attribute value ‘artefact’ is sufficient.

9.2 Encoding of metrical structures

It is recommended that passages of verse in manuscripts should be encoded as such. The basic encoding of verse is covered in chapter 4.5 above, where it is recommended that verse be encoded in line groups (i.e. stanzas) and lines.

The following section extends the basic mark-up of verse to include: (1) references to the relevant stanza within the whole poetic corpus, so that poetry can be cross-referenced between manuscript versions; and (2) markup of metrical features within verses and lines.

There are good reasons to establish a system for the encoding of metrics in the medieval manuscripts even if this structure is not generally represented graphically in the manuscripts. For many users of the established text the stanzas are of great interest and it is therefore practical to mark them in a way that makes it possible to find and delimit them from the surrounding text. A more detailed encoding of the stanzas can open up for new ways of research on different metrical variants, concerning e.g. alliteration, internal rhyme and stress.

The same rules that apply for prose are relevant also for the encoding of stanzas. In addition to these rules we suggest codes that facilitate the search for and identification of stanzas. We also give guidelines as to the encoding of the metrics. It should be pointed out, however, that these codes would have to be given a more detailed form if the stanzas should be analysed in a more detailed way.

The verse will normally be marked up only at the **<me:dipl>** and **<me:norm>** levels of the transcription. For the **<me:facs>** level, verse will be included as if prose, in accordance with the practice of the medieval manuscripts.

On the primary level we recommend that the stanzas are encoded with the following elements and attributes:

Elements / attributes	Explanation
<lg>	Marks the stanza in relation to the surrounding prose text.
@n	Indicates the identity of the stanza within the manuscript, i.e. its number in the manuscript.
@xml:id	Indicates the identity of the stanza within the medieval poetic corpus. The id should refer to a standard edition of the works. Menota recommends using the sigla for verses used by the Skaldic Poetry project (http://skaldic.arts.usyd.edu.au) for non-Eddic verse, and the numbering in Neckel and Kuhn 1983 for Eddic verse. If no standard corpus contains the verse (e.g. rímur), it should be indicated by a separate typology.
@type	Indicates the general metrical form of the stanza, e.g. 'dróttkvætt' or 'fornyrðislag.'
<l>	Marks the line within the stanza.
@n	Indicates the line number within the stanza. Lines are broken according to Norse-Icelandic conventions, that is, alliterative lines are treated as two lines, with a break at the caesura.
@type	Indicates the type of line for formatting purposes; the implied value, 'normal', is not indented; use 'b-line' for the b-line of eddic metres which should have the caesura represented by a long space; use 'ljod-long' for ljóðaháttir long lines, which should have a line break and be indented.
@met	Indicates the metrical form of the line. The form should be according to a standard typology, e.g. types A-E of the common germanic verse form. Sub-types can also be represented, using, e.g. Gade 1995. Alternatively, the actual scansion of the line can be represented using a series of symbols (e.g. '/' for a lift, '\' for a secondary stress, 'x' for a dip and ' ' for a syntactic caesura; or cf. MUFI recommendation for metrical symbols).
<me:all>	Indicates the alliteration of the line.
<me:ass>	Indicates the internal rhymes of the line, where relevant.

The following example is from *Guta saga* (ed. by Peel 1999, 2), in which the correction of the word 'reð' has been suppressed for the sake of simplicity:

Hann reþ draum þinna so:
 Alt ir baugum bundit
 Boland al þitta varþa
 ok faum þria syni aiga

The basic encoding of a verse within a prose work is thus:

```
<p><!-- ... --> <w>hann</w> <w>rep</w> <w>dra<lb n="16"/>um</w>
  <w>pinna</w> <w>so</w>.</p>
<lg n="1" xml:id="Guta-v1" type="germ">
  <l n="1"><w>Alt</w> <w>ír</w> <w>baugum</w> <w>bundit</w>
    <lb n="17"/></l>
  <l n="2"><w>bo land</w> <w>al</w> <w>pitta</w> <w>warpa</w></l>
  <l n="3"><w>oc</w> <w>faum</w> <pb n="1v"/> <w>pria</w> <w>syni</w>
    <w>aiga</w></l>
</lg>
```

The verse is numbered as the first in the manuscript, and the value of **@xml:id** is according to its own typology (this verse does not occur in most poetic corpora). The value of **@type** simply represents that this is in the common Germanic metre. No attempt has been made here to do more detailed metrical analysis.

The following example is of a more complex, skaldic example from *Skáldskaparmál* in AM 748 II 4to. The verse (from Bragi Boddason's *Ragnarsdrápa* (stanza 5) is thus:

þar sua at giordu gyrdan
 golfhaukuis sa fylkis
 segls naglfara siglur
 saums anduanar standa
 urdu snemst ok saurli
 samrada þeir hamdir
 halum herdi mylum
 hergautz vínum bardir.

The transcription is encoded thus (long 's' is normalised to 's'; encoded at the diplomatic level):

```
<lg n="3" xml:id="Bragi-Rdr-5" type="dróttkvætt">
  <l n="1" met="A3-2">
    <w><me:dipl>þ<ex>ar</ex></me:dipl></w>
    <w><me:dipl>s<ex>ua</ex> at</me:dipl></w>
    <w><me:dipl><me:all>g</me:all><me:ass>iord</me:ass>u</me:dipl></w>
    <w><me:dipl><me:all>g</me:all><me:ass>yrd</me:ass>an</me:dipl></w>
  </l>
  <l n="2" met="X">
    <w><me:dipl><me:all>g</me:all>olfh<me:ass>aul<lb n="19"/>k</me:ass>
      uis</me:dipl></w>
    <w><me:dipl>sa</me:dipl></w>
    <w><me:dipl>f<me:ass>ylk</me:ass>is</me:dipl></w>
  </l>
  <l n="3" met="D2">
    <w><me:dipl><me:all>s</me:all><me:ass>egl</me:ass>s</me:dipl></w>
    <w><me:dipl>naglf<ex>ar</ex>a</me:dipl></w>
    <w><me:dipl><me:all>s</me:all><me:ass>igl</me:ass>ur</me:dipl></w>
  </l>
  <l n="4" met="D2">
    <w><me:dipl><me:all>s</me:all>aums</me:dipl></w>
    <w><me:dipl><me:ass>and</me:ass>uana<add place="supralinear">r</add>
      </me:dipl></w>
    <w><me:dipl>st<me:ass>and</me:ass>a</me:dipl></w>
  </l>
  <l n="5" met="X">
    <w><me:dipl>urdu</me:dipl></w> <lb n="20"/>
    <w><me:dipl><me:all>s</me:all>nemst</me:dipl></w>
    <w><me:dipl><ex>ok</ex></me:dipl></w>
    <w><me:dipl><me:all>s</me:all>aurli</me:dipl></w>
  </l>
  <l n="6" met="X">
```

```

    <w><me:dipl><me:all>s</me:all><me:ass>am</me:ass>rada</me:dipl></w>
    <w><me:dipl>p<ex>ei</ex>r</me:dipl></w>
    <w><me:dipl>h<me:ass>am</me:ass>d<ex>ir</ex></me:dipl></w>
  </l>
  <l n="7" met="A1-1">
    <w><me:dipl><me:all>h</me:all><me:ass>al</me:ass>um</me:dipl></w>
    <w><me:dipl><me:all>h</me:all><ex>er</ex>di</me:dipl></w>
    <w><me:dipl>m<me:ass>yl</me:ass>um</me:dipl></w>
  </l>
  <l n="8" met="X">
    <w><me:dipl><me:all>h</me:all><ex>er</ex>gautz</me:dipl></w>
    <w><me:dipl>vín<lb n="21" />um</me:dipl></w>
    <w><me:dipl>bardir</me:dipl></w><me:punct>.</me:punct>
  </l>
</lg>

```

The value of @n is 3, as this is the third verse recorded in the manuscript. The verse's @xml:id is the siglum from the [skaldic project](#). The type of metre is 'dróttkvætt', but this categorisation is unnecessary, as the link to the skaldic project also provides information on the metrical category of each verse. Each line is encoded with the <l> element, with the line number given and the metrical categorisation (here, from Gade 1995; 'X' means uncategorised).

The alliterative staves are indicated using the <me:all> element, and the internal rhymes using <me:ass>. Both elements are defined by Menota and belong to the Menota namespace (cf. [ch. 1.9](#) above).

Ch. 10. The header

10.1 Introduction

This chapter deals with the first major part of any Menota XML file, the header. The header should describe the file so that the text itself, its source, its encoding and its revisions are sufficiently documented. It has four major parts:

Elements / attributes	Contents
<code><fileDesc></code>	A file description
<code><encodingDesc></code>	An encoding description
<code><profileDesc></code>	A text profile
<code><revisionDesc></code>	A revision history

This chapter will discuss the minimal amount of information for each of the four parts.

10.2 The file description

The file description is a mandatory part of the header and must include information on the title, on the publication and on the source, cf. [ch. 2.2 ‘The File Description’](#) of the TEI P5 Guidelines. It contains a number of elements, several of which were discussed in [ch. 9](#) above (`<name>`, `<persName>`, `<forename>`, `<surname>` and `<addName>`).

10.2.1 Title, edition, extent and publication statement

This part of the header supplies the necessary bibliographical information about the text. The main editor(s) should always be identified, but since most electronic editions are the result of a teamwork (or of an accumulative work) all major contributors should be listed.

10.2.1.1 Title statement

Elements / attributes	Contents
<code><titleStmt></code>	Information on the title, editor and other people who have been responsible for the edition
<code><title></code>	The title of the work
<code><editor></code>	The editor of the encoded work
<code>@role</code>	The role of the editor, e.g. person, institution or project
<code><orgName></code>	The name of an organisation
<code>@affiliation</code>	The affiliation of the editor in the institution

Elements / attributes	Contents
<respStmt>	A statment of responsibility
<resp>	Type of responsibility, e.g. transcription, conversion, proof-reading

The title of the work should specify the primary source (manuscript) on which the transcription is based, and, where applicable, the title of the work. We recommend that the title states that the present text is an electronic edition. In single-text manuscripts, the title may be fairly short:

```
<title>Holm perg 6 fol : Barlaams ok Josaphats saga : an electronic edition</title>
```

In multi-text manuscripts, the title will by necessity be longer:

```
<title>AM 242 fol (Codex Wormianus) : Snorra-Edda, the four grammatical treatises,
Rígsþula, Mariúkvæði, and Ókennd heiti : an electronic edition</title>
```

The full list of work titles will be given in the **<sourceDesc>** below, so at this stage, the title may be somewhat abbreviated.

Manuscript sigla are given according to various standards, so that “Holm perg 6 fol” in many contexts is referred to as “Sth. perg. 6 fol”. For Old Norse sources we recommend using the sigla in the index volume of Ordbog over det norrøne prosasprog (also accessible on the ONPs web page). The manuscript siglum should always be given in full.

In addition to the title, the **<titleStmt>** must also list the editor(s) and other contributors to the edition. We recommend that one or more people (or institutions) are identified as the main editor(s) of the text in the **<editor>** element.

```
<editor>
  <name>
    <persName>
      <forename>Magnus</forename>
      <surname>Rindal</surname>
    </persName>
    <orgName type="affiliation">University of Bergen</orgName>
  </name>
</editor>
```

Editors should be listed either in alphabetical order or in order of importance. Institutions as well as individuals may be given as editors:

```
<editor role="institution">
  <name>
    <orgName>Gammelnorsk Ordboksverk / Enhet for digital
    dokumentasjon</orgName>
  </name>
</editor>
<editor>
  <name>
    <persName>
      <forename>Christian-Emil</forename>
      <surname>Ore</surname>
    </persName>
    <orgName type="affiliation">University of Oslo</orgName>
  </name>
</editor>
```

In this case, the attribute ‘role’ explains the fact that the first editor is an institution rather than an individual.

Any other contributors are listed chronologically in one or more responsibility statements, **<respStmt>**, with a specification of what their contribution has been. The editors may also be added to this list of responsibility statements.

```
<respStmt>
  <resp>Lemmatisation and morphological encoding</resp>
  <name>
    <persName>
      <forename>Jon Erik</forename>
      <surname>Hagen</surname>
    </persName>
    <orgName type="affiliation">University of Bergen</orgName>
  </name>
</respStmt>

<respStmt>
  <resp>Conversion to Menotic XML</resp>
  <name>
    <persName>
      <forename>Christian Emil</forename>
      <surname>Ore</surname>
    </persName>
    <orgName type="affiliation">University of Oslo</orgName>
  </name>
</respStmt>
```

If a contributor is responsible for several activities, this may be specified in more than one **<resp>**, given in chronological order:

```
<respStmt>
  <resp>Transcription of primary source</resp>
  <resp>Conversion to XML</resp>
  <resp>Lemmatisation</resp>
  <name>
    <persName>
      <forename>Karl G.</forename>
      <surname>Johansson</surname>
    </persName>
    <orgName type="affiliation">University of Oslo</orgName>
  </name>
</respStmt>
```

As discussed in [ch. 9](#) above, patronymica should not be encoded as surnames, but rather as additional names:

```
<name>
  <persName>
    <forename>Guðvarður Már</forename>
    <addName type="patronym">Gunnlaugsson</addName>
  </persName>
  <orgName type="affiliation">University of Reykjavík</orgName>
</name>
```

When listing persons in alphabetical order, a surname should be given before any forenames, e.g. ‘Rindal, Magnus’. In the absence of a surname, a forename is given before an additional name, e.g. ‘Guðvarður Már Gunnlaugsson’.

The TEI P5 Guidelines also recommends that the element **<author>** is included in the **<titleStmt>** ([ch. 2.2.1 ‘The Title Statement’](#)). Since almost all Medieval Nordic texts are anonymous we believe this element is not required.

10.2.1.2 Edition statement

Elements / attributes	Contents
<editionStmt>	A statment of the edition
<edition>	A description of the edition, typically giving it a number
@n	The number of the edition

The **<editionStmt>** should be used to specify whether the present text is a new or a revised edition of the electronic text as described in the title statement above. Here, “edition” is to be understood as “version”. The version number should be given in the **@n** attribute with the usual number system, i.e. 1.0, 1.0.1, 1.1, 1.2, etc., while the date of the version should be given as year, month and day, e.g. 2004-02-01.

A complete edition statement may be as simple as this:

```
<editionStmt>
  <edition n="1.0">First draft, <date when="2004-02-01">
    1 February 2004</date>.</edition>
</editionStmt>
```

10.2.1.3 Extent

Elements / attributes	Contents
<extent>	The size of the file, preferably specified in words
@n	The number of words (or any other measure)

The **<extent>** element specifies the size of the file. The exact number of words should be given in the **@n** attribute as well as in plain text within the element, e.g.:

```
<extent n="76411">76411 words</extent>
```

10.2.1.4 Publication statement

Elements / attributes	Contents
<publicationStmt>	A statment of the publication
<distributor>	A reference to the distributor, e.g. Menota
<idno>	A reference, e.g. ‘Ms. 1’
@type	The type of reference

Elements / attributes	Contents
<date>	The date for the publication of the edition
@value	The date, preferably in the year-month-day format, e.g. 2004-03-01
<availability>	A description of the conditions for the distribution and use of the text
@status	The type of availability, typically with the values 'free', 'restricted' or 'unknown'.

The **<publisher>** specifies the body (publisher, archive) which has made the text available, e.g. the Medieval Nordic Text Archive (Menota).

The **<idno>** is a unique identification of the text. For texts in the Menota archive the attribute value will be Menota, and the contents of the element will be an acquisition number, beginning with ms. 1. Note that this information will be supplied by Menota, if the text is being deposited in this archive.

The **<availability>** element specifies the accessibility of the text. We recommend adding a **@status** attribute with one of the three values 'free', 'restricted', 'unknown' (cf. [ch. 2.2.4 'Publication, Distribution, etc.'](#) of the TEI P5 Guidelines). Further specifications can be added in a **<p>** element. For texts in the Menota archive, we suggest this description: 'This text is available for purposes of academic research and teaching only. Re-distribution in any form without prior permission is prohibited. Short extracts may be cited with full acknowledgment of the source.'

Thus, a complete publication statement may look like this:

```
<publicationStmt>
  <distributor>Medieval Nordic text Archive</distributor>
  <idno type="Menota">Ms. 1</idno>
  <date when="2004-03-01">1 March 2004</date>
  <availability status="restricted">
    <p>This text is available for purposes of academic research and
      teaching only.Re-distribution in any form without prior
      permission is prohibited. Short extracts may be cited with full
      acknowledgment of the source.</p>
  </availability>
</publicationStmt>
```

10.2.2 Source description

The **<sourceDesc>** is a mandatory part of the header (cf. [ch. 2.2.7 'The Source Description'](#) of the TEI P5 Guidelines). With TEI P5, this part of the header includes a specific element for manuscript description, based chiefly on the work of the EU-funded MASTER project (1999-2001) and the TEI Medieval Manuscripts Description Work Group (1998-2000). For more detailed information on the manuscript description module, see [ch. 10 'Manuscript Description'](#) of the TEI P5 Guidelines.

The **<msDesc>** element is the framing element into which the manuscript description is put. The description need not consist of more than the basic information necessary to identify the source, i.e. its location, both geographical and institutional, and its shelfmark or other identifying number or name (e.g. Oslo, Universitetsbibliotek, UB 1042 8vo), but it is also possible to provide a detailed description of the source, analogous to what one would find in the introduction to a scholarly edition. (Note that while the **<msDesc>**

element will normally appear within **<sourceDesc>** in the document header, it can also appear anywhere within the body of a TEI conformant document, in the same way as the bibliographic elements **<bibl>**, **<biblStruct>** and **<biblItem>**.)

Within **<msDesc>** the following seven elements are available, of which only the first is required:

Elements	Contents
<msIdentifier>	Groups information that uniquely identifies the manuscript, i.e. its location, holding institution and shelfmark.
<head>	A standard TEI element, used to provide a brief unstructured description of the manuscript, including, for example, a uniform or supplied title, information on place and date of origin etc.
<msContents>	Contains an itemised list of the intellectual content of the manuscript or manuscript part, either as a series of paragraphs or as a series of structured manuscript items including transcriptions of rubrics, incipits, explicits etc., as well as primary bibliographic references.
<physDesc>	Groups information concerning all physical aspects of the manuscript or manuscript part, its material, size, format, script, decoration, binding, marginalia etc.
<history>	Provides information on the history of the manuscript or manuscript part, its origin, provenance and acquisition by its holding institution.
<additional>	Groups other information about the manuscript, in particular, administrative information relating to its availability, custodial history, surrogates etc.
<msPart>	Contains in essence a nested <msDesc> , in cases of composite manuscripts now regarded as constituting a single unit but made up of two or more parts which were originally physically distinct; since the contents, physical description and history of the individual parts will normally be quite different, an <msPart> element can contain all the elements listed above, with the exception that instead of <msIdentifier> the <altIdentifier> element is used to provide a shelfmark or other identifying name or number.

Within each of these elements a number of sub-elements is available; **<msContents>**, for example, will normally consist of one or more **<msItem>** elements, each in turn containing specific elements for **<rubric>**, **<incipit>**, **<explicit>** and **<colophon>**, as well as the standard TEI elements **<author>**, **<title>** and **<bibl>**. The contents need not be this structured, however, since with all the elements listed above, apart from **<msIdentifier>**, there is also the option of using ordinary prose, marked up with the **<p>** element. Doing so would limit greatly the possibilities both for processing and searching the data, but could be preferable when dealing with pre-existing descriptions (so-called ‘legacy data’), the exact form of which one may wish, or be required, to maintain.

10.2.2.1 The manuscript identifier and manuscript heading elements

The only mandatory element, as was said, is **<msIdentifier>**. Within it a number of sub-elements is available: **<country>**, **<region>**, **<settlement>** (the TEI term for what most people would call city), **<institution>**, **<repository>**, **<collection>** and **<idno>**, all of

which are self-explanatory. Although not required it is strongly recommended that at least the elements `<settlement>`, `<repository>` and `<idno>` be included, since they provide what is, by common consent, the minimum amount of information necessary to identify a manuscript. In many cases, no other elements are needed, as common sense will suffice to distinguish, say, Paris, France from Paris, Texas, as the location of the *Bibliothèque Nationale*. For search purposes, however, it is probably a good idea to include as much information as possible, such as `<country>` and, where applicable, `<region>`. There are two further sub-elements, `<altIdentifier>`, which contains an alternative structured identifier used for a manuscript, such as a catalogue number or former shelfmark, and `<msName>`, which contains any form of unstructured alternative name used for a manuscript; the manuscript Uppsala, Universitetsbiblioteket, DG 1, for example, is far better known under the name *Codex Argenteus* or the *Silver Bible*. There are many examples of such nicknames among the manuscripts in the Arnamagnæan Collection, as Árni Magnússon frequently gave his manuscripts names based on the places where they had been made or where he got them from or the people whom he knew to have produced or possessed them. Occasionally a manuscript can have several such names, or perhaps rather several forms of the name, typically in different languages. These can be dealt with through the `@xml:lang` attribute, which is available on all TEI elements.

A typical `<msIdentifier>` for a manuscript in the Arnamagnæan collection looks like this:

```
<msIdentifier>
  <country key="DK">Danmark</country>
  <settlement>København</settlement>
  <repository>Den Arnamagnæanske Samling</repository>
  <idno>AM 45 fol.</idno>
  <altIdentifier type="KKKat">
    <idno>59</idno>
  </altIdentifier>
  <msName type="nickname" xml:lang="la">Codex Frisianus</msName>
  <msName type="nickname" xml:lang="is">Fríssbók</msName>
</msIdentifier>
```

The value of the `@key` attribute on `<country>`, which is the standard international two-letter code, is for search purposes, enabling one to find all manuscripts in Danish repositories regardless of whether the cataloguer has given the name of the country as ‘Denmark’ or ‘Danmark’ (or, for that matter, ‘Dänemark’, ‘Dinamarca’ or ‘#####’). There are many such attributes in the manuscript description tagset which allow for cross-language searches.

The `<head>` element is intended to provide a short summary description of the manuscript. Any phrase-level elements, such as `<author>` and `<title>`, or the specialized elements `<origPlace>`, `<origDate>`, can also be used within `<head>`, but it should be remembered that the `<head>` element is intended primarily to provide a heading for a manuscript description, rather than the description itself, and more structured information concerning the intellectual content, physical form or history of the manuscript should be given within the specialized elements available for that purpose, described below. The `<note>`, which can be used to provide information on the manuscript which is of particular importance or interest but not covered by the other elements. The `<note>` element is repeatable, and can be given a `@type` attribute, if it is thought necessary to distinguish between different kinds of notes, but again if one feels the need to distinguish between many different kinds of notes, it is probably preferable to use the specific tags described below). The following, the `<head>` element for the Arnamagnæan manuscript AM 1 e # I fol., is typical:

```
<head>
  <title type="uniform" xml:lang="is">Sögubrot af nokkrum fornkonungum
    í Dana ok Svía veldi</title>;
```

```

<origPlace>Iceland</origPlace>, <origDate>c. 1300</origDate>.
<note>This manuscript and some fragments of <title>Knýtlinga saga</title>
  in AM 20b I fol. are presumed originally to have belonged together.
  Together these fragments constitute the work known as
  <title>Skjöldunga saga</title>.</note>
</head>

```

10.2.2.2 Intellectual content

Although it is possible to use the **<title>** element in **<head>**, it would normally be used there for a uniform title (e.g. *Brennu-Njáls saga*) or a supplied title, which describes the contents of the manuscript as a whole (e.g. 'Collection of rímur'). Detailed description of a manuscript's contents is put in the **<msContents>** element, which consists of one or more **<msItem>** elements (prefaced, if desired, by a **<summary>** element, when only some of the items are to be described in detail).

<msItem> elements are allowed to 'nest', by which is meant that an **<msItem>** can contain other **<msItem>** elements; this is useful where separate items in a manuscript are grouped under a single title or rubric, for example in collections of prayers.

A **@defective** attribute, with possible values of 'true', 'false', 'unknown' or 'unspecified', is available on **<msItem>**, providing a useful means of distinguishing between texts which are fragmentary and those which are not. The attribute is also available on the specialised elements for **<incipit>** and **<explicit>**. When dealing with collections of fragments, each fragment may be given as a separate **<msItem>** and the first and last words of each transcribed as defective incipits and explicits, as in the following example, a manuscript containing four fragments of a single work:

```

<msContents>
  <msItem defective="true"><locus from="1r" to="9v">1r-9v</locus>
    <title>Knýtlinga saga</title>
    <msItem n="1.1"><locus from="1r:1" to="2v:30">1r:1-2v:30</locus>
      <incipit defective="true">dan<ex>n</ex>a a engl<ex>an</ex>di
      </incipit><explicit defective="true">en meðan har<ex>aldr</ex>
      hein hafði k<ex>onung</ex>r v<ex>er</ex>it yf<ex>ir</ex>
      danmark</explicit>
    </msItem>
    <!-- msItems 1.2 to 1.4 -->
  </msItem>
</msContents>

```

The standard TEI element **<bibl>** (and the grouping parent element **<listBibl>**) is also available within **<msItem>**. This should be used to provide bibliographical information on the **<msItem>** level, i.e. concerning editions of the item in question. Bibliographical information pertaining to the manuscript as a whole can be placed in the **<additional>** element, described below.

10.2.2.3 Codicological features

The next major element in an **<msDesc>** is **<physDesc>**, i.e. physical description. The first sub-element within **<physDesc>** is **<objectDesc>**, which relates specifically to the text-bearing object and contains two further sub-elements, **<supportDesc>** and **<layoutDesc>**; **<supportDesc>** in turn contains the elements relating to the physical object, or vehicle, on which the text is inscribed: **<support>**, i.e. whether written on parchment, paper etc., and a description thereof, **<extent>**, the number and size of leaves, **<foliation>**, how and, if known, when and by whom the manuscript has been paginated/ foliated, **<collation>**, a description of the quire structure, any missing leaves and so on, and **<condition>**, for a description of the present physical state of the manuscript. **<layoutDesc>** contains one or more **<layout>** elements, detailing the way(s) in which

the text is organised on the page, the number of columns, dimensions of the written area, number of lines per page/column etc.

The second group of elements within a structured physical description concerns aspects of the writing, illumination or other notation (notably, music) found in a manuscript, including additions made in later hands – the text, as it were, as opposed to the carrier. The elements are: **<handDesc>**, containing one or more **<handNote>** elements, **<musicNotation>**, containing one or more paragraphs, **<decoDesc>**, containing one or more **<decoNote>** elements and **<additions>**, containing one or more paragraphs.

The **<handDesc>** element is intended for a description of the scribal hand or hands of the manuscript. This may simply contain one or more **<p>** elements, but can also consist of a series of **<handNote>** elements, each containing a prose description of one of the hands. The level of detail in these descriptions is determined entirely by the scholar or cataloguer. The following is an example of a short **<handDesc>** element:

```
<handDesc hands="1">
  <p>Written in <term type="script">Gothic hybrid</term>.
  The scribe is unknown, but the same hand is found on sections
  of AM 23 4to and Gl. kgl. S. 25 fol.</p>
</handDesc>
```

The use of the TEI element **<term>** with its attribute **@type** allows for more precise searching than would be possible with free text, but is obviously dependent on there being a commonly agreed taxonomy.

In the following, where a **<handNote>** element is used to describe an individual hand (one of six in the manuscript), the **@script** attribute is used to indicate the type of script. Note also the **@scope** attribute, with possible values of 'major', 'minor' and 'sole'.

```
<handDesc hands="6">
  <handNote script="Hybrida" scope="major">
    <p>The main hand (Hand 1) writes <locus>ff. 1r-9r
    and 16r-118v</locus> in a practised Gothic hybrid.</p>
  </handNote>
  <!-- more handNote elements -->
</handDesc>
```

Here the **<locus>** element, which we saw above in **<msItem>**, is used to indicate specifically which parts of a manuscript are written in a given hand.

As the content of the **<handNote>** element is 'p+', i.e. one or more paragraphs, there is no limit to the amount of information which may be given on any single hand. Thus, a detailed analysis of palaeographical and orthographical features ('/a/ is of the two-storey kind' etc.) is perfectly possible within this overall structure.

There is a corresponding element, **<decoDesc>** for the description of illumination and other decorative features in the manuscript. **<decoDesc>**, like **<handDesc>**, may simply contain one or more paragraphs or a sequence of topically organised sub-elements, called **<decoNote>**s, each describing either a decorative component of a manuscript (e.g. a single illuminated initial) or a homogenous class of such components (e.g. illuminated initials generally).

Two attributes are available on **<decoNote>** (in addition to those globally available), **@type** (e.g. 'initial') and **@subtype** (e.g. 'historiated'), which may be used in order to facilitate sophisticated searches, although here again it requires that there be a commonly agreed taxonomy.

The following is an example of a typical **<decoNote>**:

```
<decoNote type="secondary" subtype="initial">
  <p>There are red initials on ff. 4r, 5v, 8r, 9lr, 95r, 100r, 101r,
    102r, 104r, 107r, 108r, 110r, 111r, 112r, 113 and 116r.</p>
</decoNote>
```

The standard TEI **<list>** element can also be used if one wishes to list separately the individual instances of a particular type of decoration, rather than using separate **<decoNote>** elements:

```
<decoNote type="miniature">
  <p>The manuscript is decorated with 48 framed miniatures depicting
    scenes from the life of Christ and the life of the Virgin.
  <list>
    <item n="1"><locus>2v</locus><term>Pietà</term>; the dead Christ
      supported by the Virgin Mary.</item>
    <!-- other items -->
  </list>
</p>
</decoNote>
```

Finally, the **<additions>** element can be used to list or describe any marginalia or other additions to the manuscript which may be considered to be of interest or importance. Such additions may also be discussed or referenced elsewhere, for example as part of the **<history>** element in cases where the marginalia provide evidence of ownership.

```
<additions>
<p>The manuscript contains the following marginalia:
<list>
  <item>Fol. <locus>4v</locus>, left margin: <q xml:lang="is">hialmadr
    <ex>ok</ex> <lb/>brynjadr</q>, in a fifteenth-century hand, imitating
    an addition made to the text by the scribe at this point.</item>
  <item>Fol. <locus>5r</locus>, lower margin: <q xml:lang="is">
    þ<ex>e</ex>tta þiki m<ex>er</ex> v<ex>er</ex>a gott blek en<ex>n</ex>da
    kan<ex>n</ex> ek ecki betr sia</q>; fifteenth-century hand, probably
    the same as that on the previous page.</item>
  <item>Fol. <locus>9v</locus>, bottom margin: <q xml:lang="is">þessa
    bok uilda eg <sic>gæt</sic> lært með<lb/>an Gud gefe myr Gott ad
    <lb/>læra</q>; seventeenth-century hand.</item>
</list>
</p>
</additions>
```

The third group of elements pertains to things which have happened to the manuscript after it came into being, and are thus less integrally a part of it; the elements included here are **<bindingDesc>**, containing a description of the state of the present and former bindings of a manuscript, given either as one or more paragraphs or as a series of distinct **<binding>** elements, **<sealDesc>**, which supplies information about the seal(s) attached to a document, again either as one or more paragraphs summarising the overall nature of the seals, or as one or more **<seal>** elements, and **<accMat>**, for describing and/or transcribing any material not originally part of the manuscript but bound with it or otherwise accompanying it, for example the small paper slips on which Árni Magnússon frequently noted details on the manuscripts in his collection, how he had come to possess them, anything he had been able to discover about its previous owners and so on, which are now kept with the manuscript in question, usually bound into the front or back.

10.2.2.4 The history of the manuscript

The **<history>** element contains information on the history of the manuscript. Available within it are just three sub-elements: **<origin>**, for information on when, where and, if

known, by whom the manuscript was written, <provenance>, in which any evidence of ownership and use is provided, and <acquisition>, which describes when and how the manuscript was acquired by its holding institution. Each of these elements contains one or more paragraphs. Alternatively, as with the other major elements in a manuscript description, the <history> element may itself consist simply of one or more paragraphs in which the entire history of the manuscript is given (or, as the case may be, not given, if nothing is known of the manuscript's previous history).

The principal source of information on the history of manuscripts in the Arnarnagðæ collection will be Árn Magnússon's notes, found either the paper slips kept with the manuscript, mentioned above, or separately in the manuscript AM 435 a 4to. One may wish to provide a full transcription of these comments within the <provenance> (or <acquisition>) element, as in the following example:

```
<provenance>
  <p>According to AM 435 a 4to, ff. 54v-56v, the manuscript had been owned
  by <name type="person" subtype="owner">Sr. Þórður Jónsson á
  <name type="place">Staðastað</name> (1672-1720)</name>, who had got it
  from <name type="person" subtype="owner">Jón Hákonarson að
  <name type="place">Vatnshorni</name> (c. 1658-1748)</name>, who had in
  turn got it from <name type="person" subtype="owner">Þorgeir Jónsson
  (c. 1661-1742)</name>, <foreign>ráðsmaður</foreign> at <name type="place">
  Hólar</name> and brother of Bishop <name type="person">Steinn Jónsson
  </name>. Þorgeir had got the manuscript, probably in 1696 or 97, at
  <name type="place">Kalastaðir</name>, <name type="place">Hvalfjarðarströnd
  </name> from <name type="person" subtype="owner">Þórður Illugason</name>,
  son of <name type="person">Illugi Vigfússon</name> (c. 1570-1634), son of
  <name type="person">Vigfús Jónsson, <foreign>sýslumaður</foreign>
  (d. c. 1595)</name>. Þorgeir's wife, <name type="person">Margrét
  Guðmundsdóttir</name>, and Þórður Illugason, who had no children of his
  own, were related (<foreign>þrímenningar</foreign>).</p>

  <p>The full text of Árn's comments reads:
  <q><p>Compendium Historiæ
  Norvegicæ, undiqve mutilum, alias fragmentum rarissimum. 4to minori.
  Komid til min fra Þordi Jonssyne. en<ex>n</ex> fyrer þ<ex>ad</ex> var þad
  i eigu Þorgeirs Jonssonar, sem þad feck...</p>
  <pb/>
  <p>Fragmentum historiæ Norvegicæ in octavo /:þad sem eg feck af Þorde
  Jonssyne, en<ex>n</ex> han<ex>n</ex> af Jone Hakonarsyne/: eignadest
  Þorgeir Jonsson /:mägur Guðmundar Arnarsonar i Heynese/: ä Kalastødum
  ä Hvalfiardar strönd fyrer 10. eda 11. ärum (fra 1707. ad reikna) þad
  hafdi næst f<ex>irir</ex> han<ex>n</ex> ätt Þordur Jllu<pb/>gason
  Vigfussonar, brodurson Orms i Eyum, og høfdu þesse blød vered langfedga
  eign þeirra fedga allt fra Vigfuse Jonssyne fordum Syslum<ex>anni</ex>
  i Kios, secundum traditionem þ<ex>ess</ex> folks.</p>

  <p>Þegar han<ex>n</ex> feck þesse blød, voru þau eins mutila & nu
  eru þau. <del rend="overstrike">hefur</del> var & þar sem Þorgeir
  þau feck, eck<ex>er</ex>t <pb/> meira, ecke helldr neinstadar þar um
  kring ä ströndinne, so vött Þorgeir inqvirerad gat, sem han<ex>n</ex>
  segest m<ex>ed</ex> flid giørt hafa.</p>
  <p>Eingar utskrifter ætlar Þorgeir þar af vera, ad vösu seigest
  han<ex>n</ex> eckert slikt nockurn tíma sied hafa. Dixit coram
  1707.</p>

  <p>Þorgeir atti eigi leinge þetta fragment, helldur feck þ<ex>ad</ex>,
  so mutilum sem
  <pb/>
  þad var, Jone Hakonar syne, en<ex>n</ex> h<ex>an</ex> Þorde Jons syne
  sem adr er sagt.</p>
  <p>Jon Hakonar son af mi<ex>er</ex> adspurdr, meinat eingar utskrifter
  þar af vera i landinu, og seigest alldri þvilíkt neitt, fyrr edur sidar,
  sied hafa.</p>
```

```

</q></p>

<p>This agrees with the information found on the second (of four)
Arnarnagnæan slip, which reads:
<q>Eignarm<ex>en</ex>n þ<ex>ess</ex>a
fragm<ex>en</ex>ts hafa nylegast vered
<list>
  <item>Þorgeir Jonsson.</item>
  <item>Jon Hakonarson.</item>
  <item>Þordr Jonsson.</item>
  <item>Eg.</item>
</list>
</q></p>
</provenance>

```

It should be noted the various mechanisms for the transcription of primary sources described elsewhere in this handbook, expansion of abbreviations and so on, may be employed here as well.

10.2.2.5 Other information

The final large grouping element in a manuscript description is, appropriately enough, the **<additional>** element. The first subsection of this element is called **<adminInfo>**, which, as its name suggests, contains information pertaining to the curation and management of the manuscript. Such information would not normally form part of the introduction to a scholarly edition, but there is no reason why it could not be included in the document header. Sub-elements available here include **<custodialHist>**, in which information can be given on such matters as conservation, loans and exhibitions and so on, either as a series of paragraphs or one or more dated **<custEvent>** elements, and the standard TEI element **<availability>**, for information on the availability of the manuscript, for example any restrictions on its use or access etc.

Also available within **<additional>** is a **<surrogates>** element for information on photographic reproductions. Here it would be possible to provide information on, and links to, any digital reproductions which may be available of the manuscript.

Finally, the element **<listBibl>** is available within **<additional>** for bibliographical information pertaining to the manuscript as a whole, rather than individual text-items, which, as was mentioned above, should rather be given under the appropriate **<msItem>**.

10.2.2.6 Names of persons, places and institutions; bibliographical references

Most of the elements that have been mentioned so far have the character of boxes into which information of a certain type can be fitted. But it will be noted in the examples cited that there are other kinds of elements which can appear anywhere within the document, so-called ‘phrase-level elements’, of which there is a large number available within any TEI-conformant document. These are primarily used in order to facilitate certain types of processing and/or for search purposes. All names, for example, can be tagged using the **<name>** element, with a **@type** attribute to indicate whether they are the names of persons, places or organisations (such as religious orders). More detailed information about persons can be provided in a **<listPerson>** element within the header#s **<profileDesc>**, using the standard TEI **<person>** element, to which the value of the **@key** attribute refers. The individual **<person>** elements provide information on birth, death, residence, occupation and so on, either as one of more paragraphs of running prose, or through the use of specialised sub-elements, and there are also attributes to indicate the gender and role of the person.

In the description of the provenance of AM 435 a 4to, cited above, instead of providing birth and death dates and so on for each of the persons mentioned, one could refer using the **@key** attribute on name to an external **<person>** element, such as the following, for Þórður Jónsson:

```
<particDesc>
  <person xml:id="ThorJon" sex="1" role="owner">
    <persName xml:lang="is">Þórður Jónsson</persName>
    <birth notBefore="1672-01-01" notAfter="1672-12-31">1672</birth>
    <death when="1720-08-21">21 August 1720</death>
    <residence>
      <placeName>
        <settlement type="farm">Staðastaður</settlement>
        <region type="parish">Staðarsveit</region>
        <region type="county">Snæfellsnessýsla</region>
        <region type="compass">Western</region>
        <country key="IS">Iceland</country>
      </placeName>
    </residence>
    <occupation>Clergyman</occupation>
  </person>
</particDesc>
```

Treating names in this way means that each person is uniquely identified with an ID, to which all individual instances of that person's name then refer, whatever form those instances take. This solves the problem not only of variant spellings but also where, for example, a medieval author is known by a Latin name and any number of vernacular forms, many or all of which may have claims to 'authenticity'. In order to ensure uniformity, the method generally employed in the library world has been to accept the form found in some authority file, for example that of the American Library of Congress, as the 'base' or 'neutral' form. Feelings can run high on this matter, however, and people are frequently reluctant to accept as 'neutral' an overtly 'foreign' form of the name of some local saint or hero. Within the **<person>** tag any number of variant forms of a name can be given, with no prioritisation, and hence, less likelihood of offense. The chief advantage of treating persons in this way, however, is for searching, in particular once one has put together a large body of material. It is possible not only to search for persons with a particular name, but also born in a particular place at a particular time. The **<person>** elements taken as a whole can also function as a reference tool, a veritable *Who's Who* in medieval and early-modern Scandinavia. The possibilities as regards scribes are especially exciting, as it would be a relatively easy matter to add images to the **<person>** elements showing the hand or hands of each scribe, making it possible eventually to produce a register of all known scribes, searchable in terms of date, location etc.

It is possible to treat bibliographical references in a similar way. Since many of the same works are likely to be referred to again and again it would seem most sensible to provide full bibliographical information only once, in a separate bibliography, to which all bibliographical references in the individual records could then point.

The following is a typical bibliographical record as found in the separate bibliography file:

```
<biblStruct xml:id="StudIsl24">
  <analytic>
    <author>Ólafur Halldórsson</author>
    <title level="m">Helgafellsbækur fornar</title>
    <title level="s">Studia Islandica</title>
  </analytic>
  <monogr>
    <imprint>
      <biblScope type="vol">XXIV</biblScope>
      <pubPlace>Reykjavík</pubPlace>
```



```

        <date>1966</date>
    </imprint>
</monogr>
</biblStruct>

```

While in the description of AM 238 VII fol., one of the manuscripts discussed in the article, the bibliographical reference is given using a **<ref>** element within **<bibl>**, as follows:

```

<bibl><ref target="StudIsl24">Ólafur Halldórsson 1966</ref>,
    pp. 18 and 22</bibl>

```

As with the **<listPerson>** file, the bibliography file – which can in effect become an authorised bibliography of studies in the medieval Scandinavian philology – can be searched and browsed separately, making it a valuable tool for scholars.

10.3 The encoding description

The **<encodingDesc>** should document the relationship between the electronic edition and the source it is based upon. It is an optional part of the header, but we recommend that it contains information on the standard of encoding and level of quality. It should contain two elements: a **<projectDesc>** and an **<editorialDecl>**.

The **<projectDesc>** can be used to specify in prose the standard of the encoding, e.g. “This text has been encoded according to the standard set out in *The Menota handbook*, version 2.0, at <http://www.menota.org/guidelines>.”.

The **<editorialDecl>** uses the **<correction>** element with the **@status** attribute to specify the level of quality control. Attribute values (according to TEI) are ‘high’, ‘medium’, ‘low’, ‘unknown’. Except for the attribute value the element may be empty. The TEI P5 Guidelines ([ch. 2.3.3 ‘The Editorial Practice Declaration’](#)) has these definitions:

high: the text has been thoroughly checked and proofread

medium: the text has been checked at least once

low: the text has not been checked

unknown: the correction status of the text is unknown

A further specification can be given in prose within a **<p>** element.

Next within the **<editorialDecl>** element, a **<normalization>** element with a **@me:level** attribute is used to specify the level on which the text is encoded. The prototypical levels are ‘facs’, ‘dipl’ and ‘norm’, but other levels can also be specified, e.g. a ‘pal’ level. See [ch. 3.2](#) for a discussion of these levels. Also here, a description in prose may be added in a **<p>** element. Note that more than one level may be specified:

```

<editorialDecl>
  <normalization me:level="facs dipl norm">
    <p>This text has been encoded on three levels: facsimile, diplomatic
      and normalised.</p>
  </normalization>
</editorialDecl>

```

Finally within the **<editorialDecl>** element, an **<interpretation>** element is used to specify the amount of lexical and grammatical information in the encoded text. We suggest two attributes, **@me:lemmatized** and **@me:morphAnalyzed**, both with the values ‘completely’, ‘partly’ and ‘none’. A lemmatised text will have lemmata (i.e. dictionary entries) added in the **@lemma** attribute of the **<w>** element, while a morphologically

analysed text will have grammatical forms specified in the **@me:msa** of the same element. See [ch. 2.3](#) for a general overview and [ch. 8](#) for details on this lexical and morphological encoding. A description in prose may be added in a **<p>** element.

A complete **<encodingDesc>** may look like this:

```
<encodingDesc>
  <projectDesc>
    <p>This text has been encoded according to the standard set out in
      <title>The Menota handbook</title>, version 2.0,
      at http://www.menota.org/guidelines.&rdquo;</p>
  </projectDesc>
  <editorialDecl>
    <correction status="high">
      <p>This text was proofread by Magnus Rindal and colleagues
        before the publication of the printed version in 1981. It is
        unlikely that it contains any significant number of errors.
        However, it can not be ruled out that the subsequent conversion
        of the file may have introduced some systemic errors.
      </p>
    </correction>
    <normalization me:level="dipl">
      <p>This text has been encoded on a diplomatic level, according
        to the editorial practice by Norsk Historisk
        Kjeldeskrift-Institutt.
      </p>
    </normalization>
    <interpretation me:lemmatized="completely"
      me:morphAnalyzed="completely">
      <p>The complete text has been lemmatised and morphologically
        analysed according to the rules specified in ch. 8 of the
        Menota Handbook, v. 2.0.
      </p>
    </interpretation >
  </editorialDecl>
</encodingDesc>
```

10.4 The profile description

The **<profileDesc>** is an optional part of the header. We recommend that it is used to specify the number of hands in the source (if more than one). It should also be used to list language names outside the list in ISO 639-2.

The languages referred to in the encoding are given as a list in the **<langUsage>** element with three-letter abbreviations as values of the **@ident** attribute.

ISO 639-2 contains a list of three-letter abbreviations of language names. In addition to the modern languages ‘dan’ (Danish), ‘ice’ or ‘isl’ (Icelandic), ‘nor’ (Norwegian) and ‘swe’ (Swedish), there is only one abbreviation for Medieval Nordic, sc. ‘non’ (Old Norse, i.e. Old Icelandic and/or Old Norwegian). Since Old Norse is a problematic term and the abbreviation ‘non’ is idiosyncratic, we recommend introducing the values ‘oda’ (Old Danish), ‘oic’ (Old Icelandic), ‘onw’ (Old Norwegian), ‘osw’ (Old Swedish). In cases of uncertainty, a hyphen may be used, e.g. ‘oic-onw’ for a manuscript which is either Old Iceland or Old Norwegian (but most probably Old Icelandic), ‘onw-oic’ the other way round, etc. Please note that this usage is not ISO conformant.

For Latin we recommend the abbreviation ‘lat’, and ‘grc’ for Ancient Greek (both in in ISO 639-2).

The **<handNotes>** element specifies the number of hands recognised in the encoding of the source (if more than one).

A complete **<profileDesc>** may look like this:

```
<profileDesc>
  <langUsage>
    <language ident="oic">Old Icelandic</language>
    <language ident="onw">Old Norwegian</language>
    <language ident="osw">Old Swedish</language>
    <language ident="oda">Old Danish</language>
    <language ident="oic-onw">Old Icelandic with Old Norwegian
      traits</language>
    <language ident="onw-oic">Old Norwegian with Old Icelandic
      traits</language>
    <language ident="lat">Latin</language>
    <language ident="grc">Ancient Greek</language>
  </langUsage>
  <handNotes>
    <handNote xml:id="h1"/>
    <handNote xml:id="h2"/>
  </handNotes>
</profileDesc>
```

10.5 The revision description

Even if this is an optional part of the header, it is essential that all changes to the file are recorded. Each change is described within a **<change>** element. Here, the **<date>** is first given, then the **<name>** of the revisor (preferably with affiliation), and, finally, a description in prose of the actual change.

A single **<change>** may look like this:

```
<revisionDesc>
  <change>
    <date>2006-04-18</date>
    <name>
      <persName>Tone Merete Bruvik</persName>
      <orgName type="affiliation">Aksis</orgName>
    </name>
    : Revised the transcription in accordance with
      v. 2.0 of the Menota handbook.
  </change>
</revisionDesc>
```

10.6 Minimal headers

Two complete examples of Menota headers can be accessed in [Appendix E](#). One header is for a single-text source, such as Holm perg 6 fol (Barlaams ok Josaphats saga) while the other is for a multi-text source, such as AM 242 fol (Codex Wormianus).

Literature

Allén, Sture. 1971. *Introduktion i grafonomi*. Data linguistica 2. Stockholm: Almqvist & Wiksell.

AMKO's dictionary = *A Dictionary of Old Norse Prose / Ordbog over det norrøne prosasprog*. København: Den Arnamagnæanske Kommission, 1989–. Presently a volume of indices (1989) and three dictionary volumes have been published, *a–bam* (1995), *bam–da* (2000) and *de–em* (2004).

Bugge, Sophus, utg. 1867. *Norræn Fornkvæði. Islandsk Samling aff folkelige Oldtidsdigte om Nordens Guder og Heroer almindelig kaldet Sæmundar Edda hins Fróða*. Christiania. – Rpt., Oslo 1965.

Den norsk-islandske skjaldedigtning. See Finnur Jónsson 1912–15 below.

EAGLES. Recommendations for the Morphosyntactic Annotation of Corpora. 1996. EAGLES Document EAG-TCWG-MAC/R. – <http://www.ilc.cnr.it/EAGLES96/annotate/annotate.html>

Finnur Jónsson. 1912–15. *Den norsk-islandske skjaldedigtning*. A: *Tekst efter håndskrifterne*, 2 vols. B: *Rettet tekst*, 2 bd. København. – Rpt., København, 1967–73.

Gade, Kari Ellen. 1995. *The Structure of Old Norse Dróttkvætt Poetry*. Islandica XLIX. Ithaca and London: Cornell University Press.

Haugen, Odd Einar. 1995. 'Constitutio textus. Intervensjonisme og konservatisme i utgjevinga av norrøne tekster.' *Nordica Bergensia* 7: 69–99.

Haugen, Odd Einar. 2004. 'Parallel views: Multi-level encoding of medieval Nordic primary sources.' *Linguistic and Literary Computing* 19: 73–91.

Haugen, Odd Einar, and Alois Pichler. 2005. 'Fra kombinerte utgaver til dynamisk utgivelse: Erfaringer fra edisjonsfilologisk arbeid med Wittgensteins filosofiske skrifter og nordiske middelaldertekster.' *Læsemåder: Udgavetyper og målgrupper*, eds. Per Dahl, Johnny Konderup and Karsten Kynde, 178–224, 240–49. Nordisk Netværk for Editionsfilologer. Skrifter 6. København: Reitzel.

Hreinn Benediktsson. 1965. *Early Icelandic Script*. Reykjavík: The Manuscript Institute of Iceland.

Iversen, Ragnvald. [1923] 1973. *Norrøn grammatikk*. 7th ed., rev. by Eyvind Fjeld Halvorsen. Oslo: Aschehoug.

Jensen, Helle. 1988. 'Profilering og standardisering af udgivelsespraksis'. In: *Tekstkritisk teori og praksis*, eds. Bjarne Fidjestøl, Magnus Rindal and Odd Einar Haugen, 101–115. Oslo: Novus.

Jón Helgason, ed. 1955. *Eddadigte*. 2. ed. Nordisk Filologi. Serie A. Tekster. København: Munksgaard.

- Kohrt, Manfred. 1985. *Problemgeschichte des Graphembegriffs und des frühen Phonembegriffs*. Reihe germanistische Linguistik 61. Tübingen: Niemeyer.
- Kuhn, Hans, ed. 1983. *Edda. Die Lieder des Codex regius nebst verwandten Denkmälern. Herausgegeben von Gustav Neckel*. 5. ed. by Hans Kuhn. Heidelberg: Winter.
- Kunin, Devra, tr. 2001. *The Passion and Miracles of the blessed Óláfr*. In: *A history of Norway and The Passion and Miracles of the blessed Óláfr*, ed. Carl Phelpstead, 26–74. Text series 13. London: Viking Society for Northern Research.
- Kålund, Kristian. 1905. *Palæografisk Atlas. Oldnorsk-islandsk afdeling*. København: Gyldendal.
- Kålund, Kristian. 1907. *Palæografisk Atlas. Ny serie. Oldnorsk-islandske skriftprøver c. 1300–1700*. København: Gyldendal.
- Larrington, Carolyne, ed. 1996. *The Poetic Edda*. Translated with an introduction and notes by Carolyne Larrington. Oxford: Oxford University Press.
- de Leeuw van Weenen, Andrea. 2000. *A Grammar of Möðruvallabók*. CNWS 85. Leiden: School of Asian, African, and Amerindian Studies.
- Metcalf, Frederick, ed. 1881. *Passio et miracula Beati Olavi: edited from a twelfth-century manuscript in the library of Corpus Christi College, Oxford*. Oxford: Clarendon Press.
- Noreen, Adolf. 1923. *Altnordische Grammatik*. 4th ed. Halle: Niemeyer.
- Ordbog over det norrøne prosasprog*. See AMKO's dictionary.
- Peel, Christine, ed. 1999. *Guta saga. The history of the Gotlanders*. Viking Society for Northern Research. Text Series 12. London: Viking Society, University College.
- Rindal, Magnus, ed. 1981. *Barlaams ok Josaphats saga*. *Norrøne tekster* 4. Oslo: Norsk Historisk Kjeldestrift-Institutt.
- Robinson, Peter. 1994. *The Transcription of Primary Textual Sources using SGML*. Office for Humanities Communication Publications 6. Oxford: Oxford University Computing Services.
- Seip, Didrik Arup. 1954. *Paleografi. B. Norge og Island*. *Nordisk kultur* 28 B. Oslo: Aschehoug.
- Sievers, Eduard. 1893. *Altgermanische metrik*. *Sammlung kurzer Grammatiken germanischer Dialekte*, ed. W. Braune. *Ergänzungsreihe*, vol. 2. Halle: Niemeyer.
- Skj = Finnur Jónsson. 1912-15. See above, *Den norsk-islandske skjaldedigtning*.
- Sperberg-McQueen, C.M, and Lou Burnard. 1994. *Guidelines for Electronic Text Encoding and Interchange (TEI P3)*. Chicago, Oxford: Text Encoding Initiative. – Version P4 published in March 2002. – Version P5 published in November 2007 (only in digital form).
- Stefán Karlsson, ed. 1963. *Islandske originaldiplomer indtil 1450. Tekst*. *Editiones Arnarnagæanæ A*: 7. København: Munksgaard.

Index

This index covers ch. 1-10 of the handbook. Note that attributes are cited immediately below the elements they occur with; for example, the element **<add>** has the attributes **@hand**, **@place**, **@resp** and **@supralinear**.

1 TEI elements

Element / attribute	Chapter
<abbr>	ch. 6.1
<acquisition>	ch. 10.2.2.4
<add>	ch. 6.5.2, 7.1, 7.2.1, 7.6.1
@hand	ch. 7.2.1
@place	ch. 6.5.2, 7.2.1
@resp	ch. 7.2.1
@supralinear	ch. 6.5.2
<additional>	ch. 10.2.2, 10.2.2.5
<additions>	ch. 10.2.2.3
<addName>	ch. 9.1.1, 10.2.1.1
@type	ch. 9.1.1, 10.2.1.1
<addSpan>	ch. 4.10
<adminInfo>	ch. 10.2.2.5
<altIdentifier>	ch. 10.2.2.1
<am>	ch. 3.1, 4.9, 6.1, 6.3, 6.4, 6.5, 8.2
@me:type	ch. 1.9, 6.1
<anchor/>	ch. 4.10, 7.5.2
@xml:id	ch. 4.10, 7.5.2
<author>	ch. 10.2.2.1

Element / attribute	Chapter
<availability>	ch. 10.2.1.4, 10.2.2.5
@status	ch. 10.2.1.4
<back>	ch. 4.2
<bibl>	ch. 10.2.2.2, 10.2.2.6
<body>	ch. 3.1, 4.1, 4.2
<c>	ch. 2.1, 2.2.3, 4.9
@rend	ch. 4.9
@type	ch. 4.9
<cb/>	ch. 4.1, 4.7, 4.10
@ed	ch. 4.7
@n	ch. 4.7
<change>	ch. 10.5
<choice>	ch. 3.4, 4.8, 6.1
<collation>	ch. 10.2.2.3
<collection>	ch. 10.2.2.1
<condition>	ch. 10.2.2.3
<corr>	ch. 7.1, 7.4.3, 7.6.1
@resp	ch. 7.4.3
<correction>	ch. 10.3
@status	ch. 10.3
<country>	ch. 9.1.2, 10.2.2.1
@key	ch. 10.2.2.1
<custEvent>	ch. 10.2.2.5
<custodialHist>	ch. 10.2.2.5
<damage>	ch. 7.6.1
<damageSpan>	ch. 4.10
<date>	ch. 10.2.1.4, 10.5
@value	ch. 10.2.1.4
<decoDesc>	ch. 10.2.2.3

Element / attribute	Chapter
<decoNote>	ch. 10.2.2.3
@subtype	ch. 10.2.2.3
@type	ch. 10.2.2.3
	ch. 7.1, 7.2.2, 7.6.1
@hand	ch. 7.2.2
@place	ch. 7.2.2
@resp	ch. 7.2.2
<delSpan>	ch. 4.10
<distributor>	ch. 10.2.1.4
<div>	ch. 3.1, 4.1, 4.3, 4.4, 4.6
@id	ch. 4.1
@n	ch. 3.1, 4.1, 4.3
@type	ch. 3.1, 4.1, 4.3, 4.5
<edition>	ch. 10.2.1.2
@n	ch. 10.2.1.2
<editionStmt>	ch. 10.2.1.2
<editor>	ch. 10.2.1.1
@role	ch. 10.2.1.1
<editorialDecl>	ch. 10.3
<encodingDesc>	ch. 10.1, 10.3
<ex>	ch. 3.1, 4.9, 6.1
@me:type	ch. 1.9, 6.1
<expan>	ch. 6.1
<explicit>	ch. 10.2.2.2
@defective	ch. 10.2.2.2
<extent>	ch. 10.2.1.3, 10.2.2.3
@n	ch. 10.2.1.3
<fileDesc>	ch. 10.1
<foliation>	ch. 10.2.2.3

Element / attribute	Chapter
<forename>	ch. 9.1.1, 10.2.1.1
<front>	ch. 4.2
<gap/>	ch. 7.1, 7.3.1, 7.6.1
@agent	ch. 7.3.1
@hand	ch. 7.3.1
@quantity	ch. 7.3.1
@reason	ch. 7.3.1
@resp	ch. 7.3.1
<handDesc>	ch. 10.2.2.3
@hands	ch. 10.2.2.3
<handNote>	ch. 10.2.2.3, 10.4
@script	ch. 10.2.2.3
@xml:id	ch. 10.4
<handNotes>	ch. 10.4
<head>	ch. 3.1, 4.4, 4.6, 10.2.2
<hi>	ch. 3.1
@rend	ch. 3.1
<history>	ch. 10.2.2
<idno>	ch. 10.2.1.4, 10.2.2.1
@type	ch. 10.2.1.4
<incipit>	ch. 10.2.2.2
@defective	ch. 10.2.2.2
<institution>	ch. 10.2.2.1
<interpretation>	ch. 10.3
@me:lemmatized	ch. 1.9, 10.3
@me:morphAnalyzed	ch. 1.9, 10.3
<l>	ch. 4.1, 4.5, 9.2
@met	ch. 9.2
@n	ch. 9.2

Element / attribute	Chapter
@type	ch. 9.2
<language>	ch. 8.7, 10.4
@ident	ch. 8.7, 10.4
<langUsage>	ch. 8.7, 10.4
<layout>	ch. 10.2.2.3
<layoutDesc>	ch. 10.2.2.3
<lb/>	ch. 4.1, 4.7, 4.10
@ed	ch. 4.7
@n	ch. 4.7
<lg>	ch. 4.1, 4.5, 9.2
@n	ch. 9.2
@type	ch. 4.5, 9.2
@xml:id	ch. 9.2
<list>	ch. 10.2.2.3
<listBibl>	ch. 10.2.2.2, 10.2.2.5
<listPerson>	ch. 10.2.2.6
<locus>	ch. 10.2.2.3
<m>	ch. 2.3
@baseForm	ch. 2.3
<msContents>	ch. 10.2.2
<msDesc>	ch. 4.1
<msIdentifier>	ch. 10.2.2
<msItem>	ch. 10.2.2.2, 10.2.2.5
@defective	ch. 10.2.2.2
<msName>	ch. 10.2.2.1
@type	ch. 10.2.2.1
@xml:lang	ch. 10.2.2.1
<msPart>	ch. 10.2.2
<musicNotation>	ch. 10.2.2.3

Element / attribute	Chapter
<name>	ch. 9.1, 10.2.1.1, 10.2.2.4, 10.2.2.6, 10.5
@subtype	ch. 10.2.2.4
@type	ch. 9.1, 10.2.2.4, 10.2.2.6
<normalization>	ch. 10.3
@me:level	ch. 1.9, 10.3
<num>	ch. 2.4
<objectDesc>	ch. 10.2.2.3
<orgName>	ch. 10.2.1.1, 10.5
@type	ch. 10.2.1.1, 10.5
<origDate>	ch. 10.2.2.1
<origin>	ch. 10.2.2.4
<origPlace>	ch. 10.2.2.1
<p>	ch. 3.1, 4.1, 4.3, 4.4, 4.5, 4.6
<pb/>	ch. 4.1, 4.7, 4.10
@ed	ch. 4.7
@n	ch. 4.7
<persName>	ch. 9.1.1, 10.2.1.1, 10.5
<person>	ch. 10.2.2.6
<physDesc>	ch. 10.2.2
<placeName>	ch. 9.1.2
<profileDesc>	ch. 10.1, 10.2.2.6, 10.4
<projectDesc>	ch. 10.3
<provenance>	ch. 10.2.2.4
<publicationStmt>	ch. 10.2.1.4
<ref>	ch. 10.2.2.6
<region>	ch. 9.1.2, 10.2.2.1
<repository>	ch. 10.2.2.1
<resp>	ch. 10.2.1.1
<respStmt>	ch. 10.2.1.1

Element / attribute	Chapter
<restore>	ch. 7.6.1
<revisionDesc>	ch. 10.1, 10.5
<roleName>	ch. 9.1.1
@type	ch. 9.1.1
<seg>	ch. 2.3, 5.4, 6.5.2, 6.5.6, 6.5.7, 8.3.2.11
@type	ch. 2.3, 5.4
<settlement>	ch. 9.1.2, 10.2.2.1
<sic>	ch. 7.1, 7.4.3, 7.6.1
@resp	ch. 7.4.3, 7.6.1
<space/>	ch. 7.1, 7.3.1, 7.6.1
@quantity	ch. 7.3.1
@unit	ch. 7.3.1
<supplied>	ch. 7.1, 7.4.1, 7.5, 7.6.1
@agent	ch. 7.4.1
@reason	ch. 7.4.1
@resp	ch. 7.4.1
@source	ch. 7.4.1
<support>	ch. 10.2.2.3
<supportDesc>	ch. 10.2.2.3
<summary>	ch. 10.2.2.2
<surname>	ch. 9.1.1, 10.2.1.1
<surrogates>	ch. 10.2.2.5
<TEI>	ch. 3.1, 4.2
<teiHeader>	ch. 3.1, 4.2
<text>	ch. 3.1, 4.1, 8.7
@xml:lang	ch. 8.7
<title>	ch. 10.2.1.1, 10.2.2.1
@type	ch. 10.2.2.1
@xml:lang	ch. 10.2.2.1

Element / attribute	Chapter
<titleStmt>	ch. 10.2.1.1
<unclear>	ch. 7.1, 7.3.2, 7.6.1
@agent	ch. 7.3.2
@hand	ch. 7.3.2
@reason	ch. 7.3.2
@rend	ch. 7.3.2
@resp	ch. 7.3.2
<w>	ch. 2.1, 2.3, 4.2, 4.8, 6.1, 8.1-7, 10.3
@lemma	ch. 2.3, 8.1-7, 10.3
@me:msa	ch. 1.9, 8.1-7, 10.3
@xml:lang	ch. 8.7

2 Menota elements

Element / attribute	Chapter
<me:all>	ch. 1.9, 9.2
<me:ass>	ch. 1.9, 9.2
<me:dipl>	ch. 1.9, 2.4, 4.8
<me:expunged>	ch. 1.9, 7.4.2, 7.6.1
@resp	ch. 7.4.2, 7.6.1
@type	ch. 7.6.1
<me:facs>	ch. 1.9, 2.4, 4.8
<me:norm>	ch. 1.9, 2.4, 4.8
<me:pal>	ch. 1.9, 3.4
<me:punct>	ch. 1.9, 2.4, 4.8
<me:textSpan>	ch. 1.9, 4.10, 7.5.2