

Chapter 5. Characters: typology and encoding

Version 2.0 (16 May 2008)

5.1 Introduction

[PDF of chapter 5.](#) 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 michaele hofud engli. at fylgia. adam
 ok ællum helgum hans at leida þa i paradisum 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&aacute;&rsccapdot; baud michaele hofud engli. at fylgia adam
ok &aolig;llum helgum hans at leida &thorn;a i paradi&long;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, ‘&rsccapdot;’, 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  &rsccapdot; baud michaele hofud engli. at fylgia adam
ok &aolig;llum helgum hans at leida þa i paradi#um h  na 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  #&combdot; baud michaele hofud engli. at fylgia adam
ok &aolig;llum helgum hans at leida þa i paradi#um h  na 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    EF22; baud michaele hofud engli. at fylgia adam
ok   EF93;llum helgum hans at leida þa i paradi#um h  na 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
a	LATIN SMALL LETTER A

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

Glyph	Descriptive name
ø	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
q	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
ø	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
ó	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 entities 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	a	0061	LATIN SMALL LETTER A
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 ornamenteally (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’, ‘T’, ‘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 (majuscules) 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
α	&aenl;	EEE0	LATIN ENLARGED LETTER SMALL A

Glyph	Entity	Unicode	Descriptive name
e	&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
f	&fins;	F10D	LATIN SMALL LETTER INSULAR F
v	&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
k	&kunc;	F208	LATIN SMALL LETTER K UNCIAL FORM
m	&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
ꝑ	&fMedrun;	16A0	RUNIC LETTER FEHU FEOH FE F
ꝑ	&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
ȝ	&rrot;	F20E	LATIN SMALL LETTER R ROTUNDA
ȝ	&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
	&aalig;	EF91	LATIN SMALL LIGATURE AA
	&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
	<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
	ø	00F8	LATIN SMALL LETTER O WITH STROKE

Glyph	Entity	Unicode	Descriptive name
Ø	Ø	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
ø	&oogon; = o + &combogon;	01EB = 006F + 0328	LATIN SMALL LETTER O WITH OGONEK = LATIN SMALL LETTER O + COMBINING OGONEK
ø	&ocurl; = o + &combcurl;	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
ø	&oloop;	F20D	LATIN SMALL LIGATURE OE WITH LOOP
ø	&aoligred;	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
á	á = a + &combacute;	00E1 = 0061 + 0301	LATIN SMALL LETTER A WITH ACUTE = LATIN SMALL LETTER A + COMBINING ACUTE ACCENT
ã	&adblac; = a + &combdblac;	E425 = 0061 + 030B	LATIN SMALL LETTER A WITH DOUBLE ACUTE = LATIN SMALL LETTER A + COMBINING DOUBLE ACUTE ACCENT
á	&aaligacute; = &aalig; + &combacute;	EFE1 = EF91 + 0301	LATIN SMALL LIGATURE AA WITH ACUTE = LATIN SMALL LIGATURE AA + COMBINING ACUTE ACCENT
ã	&aaligdblac; = &aalig; + &combdblac;	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
ÿ	&ydot; = y + &combordot;	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
ķ	&kdot; = k + &combordot;	E568 = 006B + 0307	LATIN SMALL LETTER K WITH DOT ABOVE = LATIN SMALL LETTER K + COMBINING DOT ABOVE
ġ	&gscapdot; = &gscap; + &combordot;	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 entities should be preferred.

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

Glyph	Entity name	Descriptive name
æ	æogon;	LATIN SMALL LIGATURE AE WITH OGONEK

Glyph	Entity name	Descriptive name
ø	øogonacute;	LATIN SMALL LETTER O WITH STROKE AND OGONEK AND ACUTE
æ	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
,	,	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.