

Nordic Terminological Record Format (NTRF)

**Rådet for teknisk terminologi
Norwegian Council for Technical Terminology**



1999-02-09

Nordic Terminological Record Format (NTRF)

The documentation of NTRF is divided into three chapters:

- ¶ Principles
- Fields and Field Tags
- Special Characters and Functions

The documentation has been written by *Håvard Hjulstad*, Rådet for teknisk terminologi (Norwegian Council for Technical Terminology) and *Carol B. Eckmann*, Norwegian Ministry of Foreign Affairs.

Any comments relating to the documentation of NTRF are most welcome. Please send any communications by mail to the authors or by telefax:

Rådet for teknisk terminologi
Postboks 41 Blindern
N-0313 Oslo (Norway)
fax: +47-23198041
e-mail: hhj@nbs.no

Please make sure that you are using the latest version.

Note on revision

Since the 1995-05-23 version of the NTRF documentation only minor adjustments have been made to the technical contents. The following field tags have been added: **OBJECT** (5.5), **POSI** (6.2), **ADD** (6.21). RTT and TNC tags for each of the NTRF fields have been omitted, as each institution will have to maintain a table of tags corresponding to the NTRF tags. Some information on field contents has been added. A brief overview of terminology work has been added for the benefit of non-terminologist users. Since the 1996-11-04 only minor editorial changes have been done.



Principles

Principles

The purpose of this chapter is to lay down the principles for the application of the terminological record format which is being introduced as the common format for the central terminology institutions in Finland, Norway and Sweden. Its use by any institution and individual engaged in terminology and lexicography is welcome.

The format is called “Nordic Terminological Record Format” and abbreviated NTRF.

Some background

The central terminology institutions of Finland, Norway and Sweden have utilized slightly different, but basically compatible term record formats. The compatibility has been demonstrated through the cooperation on the CD-ROM *Termdok version 3 1992*.

The formats have each had their own weak points, and the differences have posed an obstacle to data transfer as well as to a joint software development.

Since the establishment of the various formats, the *Standard Generalized Markup Language (SGML)* has been adopted by the International Organization for Standardization (ISO). Much work has also been done on devising standard dictionary formats for general language dictionaries. It is essential that the terminology record format be compatible with SGML.

A number of other terminology record formats have been studied in connection with the work on NTRF. Among the most important of these is *MicroMATER* developed in the USA by experts at Brigham Young University, Provo, Utah, and Kent State University, Kent, Ohio. The development of an SGML application for terminology interchange is being carried out by ISO / TC 37 (see below).

Matters in connection with the record format have been discussed a number of times in Nordterm, the forum for terminology cooperation in the Nordic countries. A common record taxonomy has been worked out by Nordterm Working Group 2. NTRF is compatible with that taxonomy.

Terminology and terminology work

Users of NTRF will need to be familiar with some of the principles of terminology and the traditions and standards of terminology work. Under the International Organization for Standardization (ISO) the Technical Committee for terminology principles (TC 37) has developed and is developing a number of International Standards for terminology work. The following is a list of some of the standards that may be of particular interest in connection with NTRF. International Standards (marked ISO or IEC) and Draft International Standards (DIS) may be obtained from the national standardizing body. Working Drafts (WD) and Committee Drafts (CD) are not yet published.

ISO 639:1988 : *Code for the representation of names of languages.*

ISO 639-2:1998 : *Codes for the representation of names of languages – Part 2: Alpha-3 code.*

ISO 860:1996 : *Terminology work – Harmonization of concepts and terms.*

ISO 1087:1990 : *Terminology – Vocabulary.*

ISO/DIS 1087-1 : *Terminology Work – Vocabulary – Part 1: Theory and application.*

ISO/DIS 1087-2 : *Terminology Work – Vocabulary – Part 2: Computer applications.*

ISO 10 241:1992 : *International terminology standards – Preparation and layout.*

ISO/DIS 12 199 : *Alphabetical ordering of multilingual terminological and lexicographical data represented in the Latin alphabet.*

ISO/DIS 12 200 : *Computer applications in terminology – Machine-readable terminology interchange format (MARTIF) – Negotiated interchange.*

ISO/DIS 12 616 : *Translation-oriented terminography.*

ISO/DIS 12 620 : *Computer applications in terminology – Data categories.*

Some related International Standards of interest that are developed by other technical committees are:

ISO 3166-1:1997 : *Codes for the representation of names of countries and their subdivisions – Part 1: Country codes.*

ISO 3166-2:1998 : *Codes for the representation of names of countries and their subdivisions – Part 2: Country subdivision codes.*

ISO/DIS 3166-3 : *Codes for the representation of names of countries and their subdivisions – Part 3: Code for formerly used names for countries.*

ISO/IEC 8879:1986 : *Information Processing – Text and office systems – Standard Generalized Markup Language (SGML).*

ISO/IEC 10646-1:1993 : *Information technology – Universal Multiple-Octet Coded Character Set (UCS) – Part 1: Architecture and Basic Multilingual Plane.*

Concept-based terminology work and text-based terminology work

The building blocks of terminology work are *objects*, *concepts* and *terms*.

Objects may be concrete or abstract. Objects rarely find their way into terminology records, and NTRF does not really deal with objects in the terminology sense. The OBJECT information category in chapter 2, clause 5.5 is intended for objects in the information technology sense (in an NTRF document typically sound and pictures). In terminology, objects are the actual thing, for instance a 80×120 cm table that you are buying in a shop, or the melting of an ice-cube in your glass of water.

Concepts are always abstractions, ideas or notions. A concept would be the notion of “tables”, regardless of shape, colour and place, or the notion of “melting”, regardless of what is melting and where.

Concepts are normally ordered in systems. There are many types of tables, such as “kitchen tables” and “birch-wood tables”. And “tables” as a whole belong to the larger group of “furniture”. We talk about *broader concepts* and *narrower concepts*. In the example “kitchen table” is a narrower concept than “table”, and “table” is a broader concept than “kitchen table”.

In terminological dictionaries it is *concepts* that are defined, not terms. Terms serve as “labels” that are attached to the concepts. Terms are necessary to enable people to talk and write about concepts. Terms do not have any “meaning” in themselves.

This principle of terminology has an important bearing on the format used to represent terms and concepts. A terminological record will normally have no “source language”. The concept and the definition are the “source”. All terms are “target”. Consequently, all terms that designate a single concept, are stored in the same terminology record. Terminology records are frequently called “concept records” to emphasize this. Each record may contain any number of terms in any number of languages. Each record will normally contain only one definition (or possibly one definition in each language).

A terminology project that is based on these principles will normally start off by delimiting a set of concepts for inclusion. The concepts are organized into concept systems, and the definitions are drafted on the basis of these systems. In practice the terms are commonly added from the very beginning, but the final evaluation of the terms is done later in the process.

One important consequence of this principle is seen when the same term is used to denote more than one concept. The terminology record will *not* list the term once and list several “meanings”. On the contrary: The same term will be given in several records, each record representing a single concept. In a database system for terminology, “duplicate terms” *must* be allowed.

In translation-oriented terminology work (cf ISO 12 616) the starting point will always be the terms in the source language for the translation. Translation-oriented terminology work may be said to be text-based as opposed to concept-based. However, the translator will always need to study the terms and the concept in a systematic way: first establishing which concept is denoted by the source language term, then frequently looking at other related concepts to establish the true intension of the concept in question, and then deciding which target-language term fits the concept.

Even in a translation-oriented context the typical NTRF record will contain information about a single concept.

Sometimes non-typical terminology records will be useful, particularly in a translation-oriented context. Such records may include names, titles, fixed phrases, standard text segments, etc. Typical examples are “sincerely yours” and “on behalf of NN”.

Names and titles may be regarded as terms denoting “individual concepts”. In this connection it is important to note that e.g. the Ministry of Education in one country is *not* the same concept as the Ministry of Education in another. The concept description may be different, the structure (the “subordinate concepts”) will normally be quite different, and the corresponding name in another language may be different. Consequently, each institution or unit will normally be represented in one NTRF record.

Principles for the record format

SGML (cf ISO/IEC 8879) provides a very powerful and flexible system for the structuring of text. In SGML, as in NTRF and in any dictionary encoding system, the text is broken down into fields according to what type of information it contains.

SGML makes it possible to (1) group fields in any manner and to any depth; (2) nest fields to any depth; and (3) overlap fields. Each field in SGML is delimited by start and end tags, for instance: **<defi>** may be specified to mean “start of definition field”, and **</defi>** then would mean “end of definition field”. Overlapping of fields would be for instance: **<a>** text1 **** text2 **** text3 ****, where field type **a** would read “text1 text2”, and field type **b** would read “text2 text3”. Nesting of fields, on the other hand, would be for instance: **<a>** text1 **** text2 **** text3 ****, where field type **a** would read “text1 text2 text3” or “text1 text3” (depending on field specifications), and field type **b** would read “text2”.

In **NTRF** grouping and nesting of fields are permitted. Overlapping is not permitted. The effect of this is that the end of fields is indicated by a general tag, not a specific tag for each field. This simplifies the coding, which still is compatible with SGML.

An NTRF record may look like this:

```

enTE  English term
POS   noun
frTE  terme <GEND m> anglais
enDEF  typical definitions frequently contain
       cross-references to <RCON another term>
CX     A context where the English term is found
SOURF  Source-document-1
CREA   1996-10-24 HHj
APPR   1996-11-03 CBE
=

```

An NTRF file consists of one or more records, each consisting of one or more fields. The first record starts at the beginning of the file. Each record ends with an end-of-record tag, which is a line starting with the = character and followed by zero or more (redundant) space or tab characters. The last record of the file must also end with the end-of-record tag.

Fields in NTRF are of two kinds: **top level fields** and **embedded fields**. A top level field may be subordinate to another top level field, but not to an embedded field. An embedded field is always subordinate to a top level field or another embedded field.

Top level field tags start at the far left (first position of the line). All characters up to, but not including, the first of one or more space or tab characters are considered to be part of the field tag. The field starts immediately after one or more space or tab characters. A field may continue over more than one line. Continuation lines start with one or more space or tab characters. A top level field ends when a new line starts with a non-space/non-tab character.

In the example above the fields enTE, POS, frTE, enDEF, CX, SOURF, CREA, and APPR are top level fields.

Embedded fields are delimited by **<** and **>** and nested into another field. The field tag is given immediately following the **<** character, followed by one or more space characters (or carriage return followed by one or more space or tab characters). Embedded fields may be nested to any depth, but must be back to top level before the end of the top level field.

In the example above the fields GEND and RCON are embedded fields.

Grouping of top level fields is specified in the documentation of field tags (see chapter 2). Grouping is not indicated in the record. Field groups shall be kept together during processing.

In the example above the pairs enTE + POS and CX + SOURF represent grouping of top level fields.

Any field may be used at more than one position within the grouping system. All permitted positions must be documented. Groups within groups are permitted. Groups and single fields may appear recursively.

For use with some software, in particular database systems, duplicate fields may be prohibited. For this purpose the system allows for the numbering of fields using one or two digit numbers immediately following the field type tag (see below). The principles of the numbering may be specified for each application. The numbering is redundant for the purpose of data exchange, and should only be used when necessary. Programs written for NTRF should accept numbering, but ignore the contents. The development of software allowing duplicate fields is encouraged.

Empty fields may be needed in certain cases. All higher level fields in a field group must be present. If a higher level field contains no data, the symbol #: (two characters) shall be entered. If an empty field contains an embedded field, the symbol #: is not needed.

No field is *required* in any NTRF record as specified in this documentation. However, for given purposes certain fields may be specified as being required. If no content is available for a required field, the use of “dummy content”, e.g. “xxx”, is recommended. As it in most applications will be simpler to search for a “dummy” than to search for a record with a missing field, this method may for instance be used in cases where it has been decided that all records in a project shall include terms in a specified language. E.g. the tag and field “enTE xxx” will indicate that the required English term is missing.

Field tags consist of three parts:

First part: a **language symbol** consisting of two lowercase letters a–z. The language symbols as standardized in ISO 639 shall be used. Some of the symbols are given in the documentation of field tags. The language symbol is omitted in language independent fields (for instance classification, dates and project information). The language symbol may also be omitted in embedded fields, when the language symbol is given in the top level field tag.

Second part: a **field type tag** consisting of one or more uppercase letters A–Z. This part of the field tag is obligatory.

Third part: an optional one or two digit **number** to avoid duplicate fields, if duplicate fields are prohibited in a particular application. The use of the numbering must be documented for each application. The numbering is redundant.

Files

For the purpose of data exchange standard sequential data files are to be used. The character set should be specified in each case, but should normally be the ASCII character set. Unless otherwise specified, lines should be no longer than 80 characters, including the field tags and following space or tab characters.

Programs, applications

Computer programs written to process NTRF data should, whenever possible, be made general enough also to handle other sets of data structured according to the same principles. Programs should to the largest possible extent be offered to other users of NTRF on a “freeware” or “shareware” basis.

The authors of this documentation will distribute information about computer programs and applications of NTRF to registered NTRF users.

•

Fields

and

Field

Tags

Fields and Field Tags

This chapter specifies information categories and field tags for terminology use. The principles and details may be used for other sets of data. Corresponding documentation must be written for any other types of data.

Field tags

The principles of the field tags are described in the **Principles** chapter (see chapter 1). Field tags consist of three parts: a language symbol, a field type tag, and an optional number.

Language symbols

Language symbols are standardized in ISO 639. Some of the language symbols that are most commonly used the Nordic terminology institutions are:

da	Danish	ja	Japanese
de	German	la	Latin
en	English	nl	Dutch
es	Spanish	no	Norwegian (see below)
fi	Finnish	ru	Russian
fo	Faeroese	pl	Polish
fr	French	pt	Portuguese
is	Icelandic	se	Sami
it	Italian	sv	Swedish

The Norwegian language has two written forms. Because of the frequency of use, the non-standardized symbols **nb** and **nn** may be used to denote “bokmål” and “nynorsk” respectively.

Field types

The field type tags given here are to be used for data exchange. Any user may make his or her own set of field type tags based on the same principles, to make the field type tags mnemotechnically suitable in his or her language.

Field type tags are made up of one or more uppercase letters (A–Z). Special letters (such as Æ Ä Å) should not be used.

In some cases information on **typical field contents** is given for the purpose of clarity. The terms *finite list* and *format restrictions* are used:

finite list : In each case (field, project, etc.) a list of permissible instances will have to be established and documented. Depending on the specification, the field may contain one or more members of the list.

format restrictions : Special use of certain characters will have to be established and documented.

The list of field type tags is divided into six categories: **1** Field type tags for information relating to terms, **2** Field type tags for information relating to concepts, **3** Field type tags for information relating to concept relations, **4** Field type tags for information relating to classification, **5** Field type tags for information relating to presentation, and **6** Field type tags for administrative information.

1 Field type tags for information relating to terms

1.1 **TE** **primary term** : Primary term(s) in each language. Normally there should be only one **TE** field for each language in each record.

1.2 **SY** **synonym term** : Term designating the concept, but not regarded as the primary term. There may be one or more **SY** fields only if there is a **TE** field in the same language.

- 1.3 **DTE** **deprecated term** : Term designating the concept, but which for some reason should not be used. The reason should preferably be given in a **TYPT** or a **ADD** field (1.16, 6.21).
- 1.4 **EXTE** **excerpted term** : Contextual term which is not yet evaluated as primary term, synonym or deprecated term.
- 1.5 **DES** **designation** : Designation of the concept which is not regarded as a term, and which normally will be language independent.
- 1.6 **ACRO** **acronym** : Acronym, in particular for a name. Acronyms may be given in the **ACRO** field or in another term field (**TE** or **SY**) with the indication “acronym” given in a subordinate or embedded **TYPT** field. The use of the **ACRO** field may facilitate retrieval regardless of language.
- 1.7 **GRAM** **grammatical information** : Information about part of speech, gender, etc. The **GRAM** field may be subordinate to or embedded in any term field (1.1–1.6 and 5.6). Instead of the **GRAM** field one or more of the four following fields (1.8–1.11) may be used. The contents of the **GRAM** field will normally be restricted to a finite list.
- 1.8 **POS** **part of speech** : Information about part of speech. The **POS** field may be subordinate to or embedded in any term field (1.1–1.6 and 5.6). The contents of the **POS** field will normally be restricted to a finite list, e.g. “noun”, “verb”, “adj”, etc.
- 1.9 **GEND** **gender** : Information about gender of a noun. The **GEND** field may be subordinate to or embedded in any term field (1.1–1.6 and 5.6). The contents of the **GEND** field will normally be restricted to a finite list, e.g. “f”, “m”, “n”, etc.
- 1.10 **INFL** **inflection** : Information about inflection of a word or a term. The **INFL** field may be subordinate to or embedded in any term field (1.1–1.6 and 5.6). The contents of the **INFL** field may be restricted to a finite list or limited by format restrictions, e.g. “-ing, -ed”.
- 1.11 **SYNT** **syntactical information** : Syntactical information in connection with a word or a term, e.g. transitivity, valency etc. The **SYNT** field may be subordinate to or embedded in any term field (1.1–1.6 and 5.6). The contents of the **SYNT** field may be restricted to a finite list or limited by format restrictions.
- 1.12 **PRON** **pronunciation** : Pronunciation of a word or a term. The **PRON** field may be subordinate to or embedded in any term field (1.1–1.6 and 5.6).
- 1.13 **PHR** **phrase** : Phraseological context to a term. To indicate parallel **PHR** fields in different languages, one **PHR** field may be embedded in another.
- 1.14 **ETYM** **etymology** : Information about etymology (word history). The **ETYM** field may be subordinate to or embedded in any term field (1.1–1.6 and 5.6). The contents of the **ETYM** field will normally be limited by format restrictions, e.g. “from lat.”.
- 1.15 **GEOG** **geographical area of use** : Information about the geographical area (country etc.) in which a designation is used. The **GEOG** field may be subordinate to or embedded in any term field (1.1–1.6 and 5.6). The contents of the **GEOG** field will be restricted to a finite list. Country symbols are given in ISO 3166. Symbols used for other areas than countries should not be in conflict with the country symbols.
- 1.16 **TYPT** **type of term** : Indication of the type of term. The **TYPT** field may be subordinate to or embedded in any term field (1.1–1.6 and 5.6). Terms that differ from a “default type”, should be typed in the **TYPT** field. The contents of the **TYPT** field will be restricted to a finite list, e.g. “abbreviation” or “symbol”. Other additional information about terms may be given in the **ADD** field (6.21).
- 1.17 **HOGR** **homograph number** : A number indicating the position of the term within a sequence of homographs (terms with identical written form). The **HOGR** field may be subordinate to or embedded in any term field (1.1–1.6 and 5.6). The contents of the **HOGR** field will normally be restricted to Arabic numerals.

2 Field type tags for information relating to concepts

- 2.1 **DEF** **definition** : Text in this field should conform with the rules of proper definitions.

- 2.2 **EXPLAN** **explanation** : Explanatory text which does not conform with the rules of proper definitions.
- 2.3 **CX** **context** : Authentic context from running text.
- 2.4 **NOTE** **note** : Additional information about the concept.

3 Field type tags for information relating to concept relations

- 3.1 **BCON** **broader concept** : Term designating a broader concept. The **BCON** field may be embedded in the **DEF**, **EXPLAN** or **NOTE** field, or it may be an independent top level field. The contents of the **BCON** field will normally be a term. In certain applications the contents may be a pointer to a term.
- 3.2 **NCON** **narrower concept** : Term designating a narrower concept. The **NCON** field may be embedded in the **DEF**, **EXPLAN**, **NOTE** or **EXAMP** field, or it may be an independent top level field. The contents of the **NCON** field will normally be a term. In certain applications the contents may be a pointer to a term.
- 3.3 **SCON** **side concept** : Term designating a related concept on the same level in a concept system. The **SCON** field may be embedded in the **DEF**, **EXPLAN** or **NOTE** field, or it may be an independent top level field. The contents of the **SCON** field will normally be a term. In certain applications the contents may be a pointer to a term.
- 3.4 **RCON** **related concept** : Term designating a related concept. The **RCON** field may be embedded in the **DEF**, **EXPLAN** or **NOTE** field, or it may be an independent top level field. The contents of the **RCON** field will normally be a term. In certain applications the contents may be a pointer to a term.
- 3.5 **EXAMP** **example** : Example information connected to the concept.
- 3.6 **TYPR** **type of concept relation** : Indication of the type of concept relation. The **TYPR** field may be subordinate to or embedded in any concept relation field (3.1–3.4). The contents of the **TYPR** field will be restricted to a finite list, e.g. “generic”, “partitive”, “antonym”.
- 3.7 **EQUI** **equivalence** : Note on the equivalence between terms in different languages. The **EQUI** field may be subordinate to or embedded in any term field (1.1–1.6) or a **PHR** field (1.13). The contents of the **EQUI** field may be restricted to a finite list, but may also be given as free text.

4 Field type tags for information relating to classification

- 4.1 **SUBJ** **subject area** : Indication of the subject area of use. The **SUBJ** field may be subordinate to or embedded in any term field (1.1–1.6) or a **DEF** or **EXPLAN** field (2.1–2.2). The contents of the **SUBJ** field will be restricted to a finite list.
- 4.2 **SCOPE** **scope** : Indication of the specific area of use. The **SCOPE** field may be subordinate to or embedded in any term field (1.1–1.6) or a **DEF** or **EXPLAN** field (2.1–2.2). The contents of the **SCOPE** field may be restricted to a finite list, but may also be given as free text.
- 4.3 **CLAS** **classification** : Indication of the position within the main classification system of the terminology collection. The contents of the **CLAS** field will be restricted to a finite list.
- 4.4 **CLASPEC** **special classification** : Indication of the position within a classification system other than the main classification system of the terminology collection. The contents of the **CLASPEC** field will be restricted to a finite list.
- 4.5 **CLASYS** **classification system** : Name of classification system in the **CLAS** or **CLASPEC** field under which the **CLASYS** field is subordinate or into which it is embedded.

5 Field type tags for information relating to presentation

- 5.1 **HEAD** **heading** : Heading for use in a systematic presentation. Records containing **HEAD** field(s) will not contain term fields (1.1–1.5).

- 5.2 **SORT** **sorting form** : Text which will be used as a sorting criterion instead of the term field to which it is subordinate or into which it is embedded. The **SORT** field is used if the term has a form which will give wrong sorting sequence. The principles of alphabetical sorting of words and terms are being standardized internationally (cf ISO 12 199).
- 5.3 **ILLU** **illustration** : Reference (by pointer or otherwise) to a graphical illustration. Note that the illustration typically will be stored separately from the terminology record, and that one illustration may be referenced from several records.
- 5.4 **ILLT** **text to illustration** : Text to be printed/displayed in connection with a graphical illustration.
- 5.5 **OBJECT** **object** : Reference (by pointer or otherwise) to an object, e.g. sound, animation, etc. Note that the object typically will be stored separately from the terminology record, and that one object may be referenced from several records.
- 5.6 **STE** **search term, look-up term** : Term not designating the concept, but which may be useful for search or reference purposes.

6 Field type tags for administrative information

- 6.1 **NUMB** **running serial number** : A unique identification of the record. There may be only one **NUMB** field in each record.
- 6.2 **POSI** **position in systematic arrangement** : An identification of the record established within a project for the purpose of generating a systematic arrangement. The contents of the **POSI** field will be limited by format restrictions, e.g. chapter number plus a running number.
- 6.3 **EXNO** **excerpt number** : A unique identification of an excerpt. The **EXNO** field may be subordinate to or embedded in a **CX** field (2.3).
- 6.4 **TIT** **project title** : Identification of the project to which the record belongs.
- 6.5 **SOURC** **source of record** : Identification of the source from which the record (or larger parts of the record) has been derived. The contents of the **SOURC** field should normally be restricted to a finite list, but may in special cases be given as free text. It is essential to maintain a list of bibliographical information. This must be done externally to the NTRF file.
- 6.6 **SOURF** **source of field** : Identification of the source from which a particular field has been derived. The **SOURF** field may be subordinate to or embedded in any of the fields in clauses 1–5. The contents of the **SOURF** field should normally be restricted to a finite list, but may in special cases be given as free text.
- 6.7 **INST** **institution responsible** : Identification of the institution which is responsible for the record.
- 6.8 **CRDAT** **created date** : The date of creation of the record. The date should preferably be given in the yyyy-MM-dd format, e.g. 1996-09-28.
- 6.9 **CRBY** **created by** : The initials or name of the person creating the record. The **CRBY** field is grouped with the **CRDAT** field.
- 6.10 **CREA** **created date and by** : A combination of the information in the two fields above, the date and initials or name being given in one field.
- 6.11 **UPDAT** **updated date** : The date or dates when the record has been updated. The date should preferably be given in the yyyy-MM-dd format, e.g. 1996-09-28.
- 6.12 **UPBY** **updated by** : The initials or name of the person updating the record. The **UPBY** field is grouped with the **UPDAT** field.
- 6.13 **UPDA** **updated date and by** : A combination of the information in the two fields above, the date and initials or name being given in one field.
- 6.14 **CHDAT** **checked date** : The date or dates when the record has been checked. The date should preferably be given in the yyyy-MM-dd format, e.g. 1996-09-28.
- 6.15 **CHBY** **checked by** : The initials or name of the person or body responsible for checking the record. The **CHBY** field is grouped with the **CHDAT** field.

Nordic Terminological Record Format (NTRF)

- 6.16 **CHECK** **checked date and by** : A combination of the information in the two fields above, the date and initials or name being given in one field.
- 6.17 **APDAT** **approved date** : The date or dates when the record has been approved. The date should preferably be given in the yyyy-MM-dd format, e.g. 1996-09-28.
- 6.18 **APBY** **approved by** : The initials or name of the person or body responsible for the approval of the record. The **APBY** field is grouped with the **APDAT** field.
- 6.19 **APPR** **approved date and by** : A combination of the information in the two fields above, the date and initials or name being given in one field.
- 6.20 **STAT** **status** : An indication of the status of a field. The **STAT** field may be subordinate to or embedded in any of the fields in clauses 1–5. The contents of the **STAT** field will be restricted to a finite list, e.g. “standardized”, “unconfirmed”, etc.
- 6.21 **ADD** **additional information** : Any information that may be found useful in connection with some other field. The **ADD** field may be subordinate to or embedded in any of the other fields.
- 6.22 **REMK** **remark** : Any text which may be useful for internal purposes. The **REMK** field may be subordinate to or embedded in any other field, or it may be regarded as a top level field.
- 6.23 **FILEN** **file name** : File name. The **FILEN** field shall only be used when needed for technical reasons.

Index of field tags

ACRO : acronym : 1.6
ADD : additional information : 6.21
APBY : approved by : 6.18
APDAT : approved date : 6.17
APPR : approved date and by : 6.19
BCON : broader concept : 3.1
CHBY : checked by : 6.15
CHDAT : checked date : 6.14
CHECK : checked date and by : 6.16
CLAS : classification : 4.3
CLASPEC : special classification : 4.4
CLASYS : classification system : 4.5
CRBY : created by : 6.9
CRDAT : created date : 6.8
CREA : created date and by : 6.10
CX : context : 2.3
DEF : definition : 2.1
DES : designation : 1.5
DTE : deprecated term : 1.3
EQUI : equivalence : 3.7
ETYM : etymology : 1.14
EXAMP : example : 3.5
EXNO : excerpt number : 6.3
EXPLAN : explanation : 2.2
EXTE : excerpted term : 1.4
FILEN : file name : 6.23
GEND : gender : 1.9
GEOG : geographical area of use : 1.15
GRAM : grammatical information : 1.7
HEAD : heading : 5.1

HOGR : homograph number : 1.17
ILLT : text to illustration : 5.4
ILLU : illustration : 5.3
INFL : inflection : 1.10
INST : institution responsible : 6.7
NCON : narrower concept : 3.2
NOTE : note : 2.4
NUMB : running serial number : 6.1
OBJECT : object : 5.5
PHR : phrase : 1.13
POS : part of speech : 1.8
POSI : position in systematic arrangement : 6.2
PRON : pronunciation : 1.12
RCON : related concept : 3.4
REMK : remark : 6.22
SCON : side concept : 3.3
SCOPE : scope : 4.2
SORT : sorting form : 5.2
SOURC : source of record : 6.5
SOURF : source of field : 6.6
STAT : status : 6.20
STE : search term, look-up term : 5.6
SUBJ : subject area : 4.1
SY : synonym term : 1.2
SYNT : syntactical information : 1.11
TE : primary term : 1.1
TIT : project title : 6.4
TYPR : type of concept relation : 3.6
TYPT : type of term : 1.16
UPBY : updated by : 6.12
UPDA : updated date and by : 6.13
UPDAT : updated date : 6.11

Index of field names

acronym : ACRO : 1.6
additional information : ADD : 6.21
approved by : APBY : 6.18
approved date : APDAT : 6.17
approved date and by : APPR : 6.19
broader concept : BCON : 3.1
checked by : CHBY : 6.15
checked date : CHDAT : 6.14
checked date and by : CHECK : 6.16
classification : CLAS : 4.3
classification system : CLASYS : 4.5
context : CX : 2.3
created by : CRBY : 6.9
created date : CRDAT : 6.8
created date and by : CREA : 6.10
definition : DEF : 2.1
deprecated term : DTE : 1.3
designation : DES : 1.5
equivalence : EQUI : 3.7
etymology : ETYM : 1.14
example : EXAMP : 3.5
excerpt number : EXNO : 6.3
excerpted term : EXTE : 1.4
explanation : EXPLAN : 2.2
file name : FILEN : 6.23
gender : GEND : 1.9
geographical area of use : GEOG : 1.15
grammatical information : GRAM : 1.7
heading : HEAD : 5.1
homograph number : HOGN : 1.17
illustration : ILLU : 5.3
inflection : INFL : 1.10
institution responsible : INST : 6.7
narrower concept : NCON : 3.2
note : NOTE : 2.4
object : OBJECT : 5.5
part of speech : POS : 1.8
phrase : PHR : 1.13
position in systematic arrangement : POSI : 6.2
primary term : TE : 1.1
project title : TIT : 6.4
pronunciation : PRON : 1.12
related concept : RCON : 3.4
remark : REMK : 6.22
running serial number : NUMB : 6.1
scope : SCOPE : 4.2
search term, look-up term : STE : 5.6
side concept : SCON : 3.3
sorting form : SORT : 5.2
source of field : SOURF : 6.6
source of record : SOURC : 6.5
special classification : CLASPEC : 4.4
status : STAT : 6.20
subject area : SUBJ : 4.1
synonym term : SY : 1.2
syntactical information : SYNT : 1.11
text to illustration : ILLT : 5.4
type of concept relation : TYPR : 3.6
type of term : TYPT : 1.16
updated by : UPBY : 6.12
updated date : UPDAT : 6.11
updated date and by : UPDA : 6.13

5

**Special
Characters
and
Functions**

Special Characters and Functions

This chapter gives details about the coding of special characters and functions in NTRF. The coding documented here may be applied to any set of data regardless of format.

Terminology

Three concepts are important:

- symbol** A sequence of letters (a–z, A–Z) and numerals (0–9) starting with a lowercase letter (a–z) and delimited by **<\$** and **>**.
Examples of symbols are **<\$ast>** and **<\$acute>** representing “asterisk” and “acute accent” respectively.
Some of the symbols have a simplified short form consisting of the character **#** followed by one character. The short forms are all given in the table of symbols below.
An example of a short representation is **#'a** which is identical with **<\$acute>a** representing **á**.
- box** A sequence of characters delimited by **<\$** and **>**.
Examples of boxes are **<\$text>** and **<\$<\$ast>>** representing **text** and ***** respectively.
The contents of a box is treated as a unit. The use of boxes is limited to certain functions.
- function** A sequence identified by a function name consisting of one or more uppercase letters (A–Z) and numerals (0–9) starting with an uppercase letter, and followed by one or more space characters (or followed by carriage return and one or more space or tab characters) and delimited by **<\$** and **>**.
Examples of functions are **<\$GREEK aps>** and **<\$SUPER aps>** representing $\alpha\pi\sigma$ and **aps** respectively.
The sequence following the function name is the contents of the function. The contents may be text, symbols or boxes.

Character set

This documentation is based on the seven-bit ASCII character set. Most computer systems support extended character sets including a number of special characters and accented letters. To avoid confusion only extra alphabetical characters (such as æ ä á â) should be used from the extended characters sets. Conversion programs both ways between extended character sets and the set documented here should be written.

1 Symbols

1.1 Diacritical signs

- <\$acute>** or **#'** : acute accent to the following character
Usage: **<\$acute>a** and **#'a** both will be printed/displayed as **á**
- <\$breve>** : breve to the following character
Usage: **<\$breve>a** will be printed/displayed as **ă**
- <\$caron>** or **#*** : caron (or háček) above the following character
Usage: **<\$caron>s** and **#*s** both will be printed/displayed as **š**
- <\$cedil>** or **#,** : cedilla to the following character
Usage: **<\$cedil>c** and **#,c** both will be printed/displayed as **ç**
- <\$circum>** or **#^** : circumflex accent to the following character
Usage: **<\$circum>a** and **#^a** both will be printed/displayed as **â**

- <\$diaer>** or **#"** : diaeresis (or umlaut) to the following character
Usage: **<\$diaer>a** and **#"a** both will be printed/displayed as ä
- <\$dobacute>** : double acute accent to the following character (Hungarian “long ö” and “long ü”)
Usage: **<\$dobacute>u** will be printed/displayed as ú
- <\$dotabove>** : dot above the following character
Usage: **<\$dotabove>z** will be printed/displayed as ź (Polish)
- <\$dotbelow>** : dot below the following character
Usage: **<\$dotbelow>a** will be printed/displayed as ą
- <\$grave>** or **#`** : grave accent to the following character
Usage: **<\$grave>a** and **#`a** both will be printed/displayed as à
- <\$hook>** : hook or caudate under the following character
Usage: **<\$hook>o** will be printed/displayed as q
- <\$macron>** : macron (horizontal bar above) to the following character
Usage: **<\$macron>a** will be printed/displayed as ā
- <\$ogonek>** : ogonek (Polish nasal mark) to the following character
Usage: **<\$ogonek>a** will be printed/displayed as ą
- <\$ring>** : ring above the following character
Usage: **<\$ring>a** will be printed/displayed as â
- <\$stroke>** : stroke through the following character
Usage: **<\$stroke>n** will be printed/displayed as ñ
- <\$tilde>** or **#~** : tilde to the following character
Usage: **<\$tilde>n** and **#~n** both will be printed/displayed as ñ

1.2 Special graphical symbols

- <\$align>** or **#&** : table alignment. See function MATRIX (clause 2.5)
- <\$emdash>** or **#=** : em-dash: —. Dash with length equal to the width of the letter m. Used as a punctuation dash
- <\$endash>** or **#-** : en-dash: –. Dash with length equal to the width of the letter n. Used as minus sign and to express “from – to”: 2–3
- <\$lincont>** or **#+** : line continues. The symbol is used at the end of a line if it is not possible to fit a “word” or for instance a complicated formula into one line of data
- <\$linshift>** or **#!** : line-shift. The symbol is used at the end of a line of data if there is to be a line shift at that position in print or other presentation formats
- <\$para>** or **##** : new printed paragraph. The symbol is used at the beginning of a line if it is necessary to indicate new paragraph explicitly
- <\$space>** or **#:** : non-breakable space. Space of ordinary width, where line break is not permitted
- <\$thinspace>** or **#.** : non-breakable thin space. Narrow space where line break is not permitted (for instance in large numbers: 100 000 000)

1.3 Special alphabetical characters

- <\$aelig>** : letter ae: æ (Danish, Norwegian)
- <\$aeligcap>** : letter AE: Æ (Danish, Norwegian)
- <\$aring>** : a with ring above: å (Danish, Norwegian, Swedish)
- <\$aringcap>** : A with ring above: Å (Danish, Norwegian, Swedish)
- <\$eng>** : eng: ŋ (Sami)

<\$engcap> : uppercase eng: Ŋ (Sami)¹
<\$eth> : eth: ð (Icelandic)
<\$ethcap> : uppercase eth: Ð (Icelandic)
<\$hardl> : l with stroke: ł (Polish)
<\$hardlcap> : L with stroke: Ł (Polish)
<\$ijlig> : digraph ij (Dutch)
<\$ijligcap> : digraph IJ (Dutch)
<\$inodot> : dotless i: ı (Turkish)
<\$oelig> : ligature oe: œ (French)
<\$oeligcap> : ligature OE: Œ (French)
<\$oeslash> : o with stroke: ø (Danish, Norwegian)
<\$oeslashcap> : O with stroke: Ø (Danish, Norwegian)
<\$szlig> : sharp s: ſ (German)
<\$thorn> : thorn: þ (Icelandic)
<\$thorncap> : uppercase thorn: Þ (Icelandic)

1.4 Mathematical symbols etc.

<\$approx> : approximately equal to sign (double snake): ≈
<\$arrowdoub> : double arrow, one arrow above the other, each direction
<\$arrowdown> : arrow down: ↓
<\$arrowleft> : arrow to the left: ←
<\$arrowright> : arrow to the right: →
<\$arrowup> : arrow up: ↑
<\$backsl> : backslash: \
<\$brackl> : left square bracket: [
<\$brackr> : right square bracket:]
<\$copyr> : copyright sign, ©
<\$deg> : degree symbol: °
<\$doubx> : double cross: #
<\$gt> : greater than sign: >
<\$gteq> : greater than or equal sign: ≥
<\$gtgt> : much greater than sign: >>
<\$ident> : identical symbol: ≡
<\$infin> : infinite symbol: ∞
<\$linecol> : line-colon minus/division sign: ÷
<\$lt> : less than sign: <

¹ The proper shape of uppercase eng is an uppercase N with a hook under the right leg. Several existing fonts display this character incorrectly.

- <\$lteq> : less than or equal sign: ≤
- <\$ltlt> : much less than sign: ≪
- <\$negat> : negation sign: ¬
- <\$parall> : parallel sign: ||
- <\$plusminus> : plus minus sign: ±
- <\$permil> : per thousand: ‰
- <\$regt> : registered trade mark sign: ®
- <\$sect> : section sign: §
- <\$timesdot> : multiplication symbol centred dot: ·
- <\$timesx> : multiplication symbol x: ×
- <\$stridown> : open triangle with corner down
- <\$striup> : open triangle with corner up
- <\$suneq> : unequal sign: ≠
- <\$vert> : vertical line: |

1.5 Free symbols

#0 #1 #2 #3 #4 #5 #6 #7 #8 #9 may in each application be defined to mean any other symbol. This must be documented in each case, and the free symbols should be substituted for ordinary symbols for data transfer.

2 Functions

2.1 Typefaces

Information about typefaces are rarely needed in primary data in NTRF. Since this part of the NTRF documentation may be applied to other sets of data, typeface information may be needed. For the purpose of transfer of data to typesetting typeface information may be needed even in NTRF data.

ROMAN : roman typeface

Usage: <\$ROMAN text> where text is printed/displayed in Roman (“ordinary”) typeface

ITALIC : italic typeface

Usage: <\$ITALIC text> where text is printed/displayed in *Italic* (“slanted”) typeface

BOLD : bold typeface

Usage: <\$BOLD text> where text is printed/displayed in **bold typeface**; <\$BOLD <\$ITALIC text>> where text is printed/displayed in **bold Italic typeface**

TYPEFACE1 – TYPEFACE9 : free typefaces, may be specified for each application

Usage: <\$TYPEFACE3 text> where text is printed/displayed in “typeface 3” as defined in the application

2.2 Alphabets

Default alphabet is the Latin alphabet unless otherwise specified for individual fields.

GREEK : Greek text. The transliterated Greek text is printed/displayed in Greek characters

Usage: <\$GREEK aps> will be printed/displayed as απσ

The transliteration of Greek text is as follows:

Roman	Greek	Character names
-------	-------	-----------------

Nordic Terminological Record Format (NTRF)

Roman		Greek		Character names
a	A	α	A	alpha
b	B	β	B	beta
g	G	γ	Γ	gamma
d	D	δ	Δ	delta
e	E	ε	E	epsilon
z	Z	ζ	Z	zeta
h	H	η	H	eta
u	U	θ	Θ	theta
i	I	ι	I	iota
k	K	κ	K	kappa
l	L	λ	Λ	lambda
m	M	μ	M	mu
n	N	ν	N	nu
o	O	ο	O	omicron
p	P	π	Π	pi
r	R	ρ	P	rho
s	S	σ	Σ	sigma
c		ς		final sigma
t	T	τ	T	tau
y	Y	υ	Y	upsilon
f	F	φ	Φ	phi
x	X	χ	X	chi
v	V	ψ	Ψ	psi
w	W	ω	Ω	omega
	C		Ϝ	capital omega shaped as small

CYRILLIC : text in Cyrillic alphabet. The transliterated Cyrillic text is printed/displayed in Cyrillic characters

Usage: <\$CYRILLIC SSSR> is printed/displayed as CCCP

The transliteration of Cyrillic text is as follows:

Roman		Cyrillic		Notes
a	A	а	А	
b	B	б	Б	
v	V	в	В	
g	G	г	Г	
d	D	д	Д	
e	E	е	Е	
#"e	#"E	ё	Ё	
#*z	#*Z	ж	Ж	
z	Z	з	З	
i	I	и	И	
j	J	й	Й	
k	K	к	К	
l	L	л	Л	
m	M	м	М	
n	N	н	Н	
o	O	о	О	
p	P	п	П	

Roman		Cyrillic		Notes
r	R	р	Р	
s	S	с	С	
t	T	т	Т	
u	U	у	У	
f	F	ф	Ф	
h	H	х	Х	
c	C	ц	Ц	
#*c	#*C	ч	Ч	
#*s	#*S	ш	Ш	
#*s#*c	#*S#*C	щ	Щ	
"		ъ	Ъ	hard sign
y	Y	ы	Ы	
'		ь	Ь	soft sign
<\$.dotabove>e	<\$.dotabove>E	э	Э	
ju	JU	ю	Ю	
ja	JA	я	Я	

LATIN : text in Latin alphabet. The text is printed/displayed in the Latin alphabet. This function is useful only within fields where a non-Latin alphabet is specified as the default.
Usage: <\$LATIN text> is printed/displayed as text

2.3 Superscript, subscript, etc.

SUPER : superscript text

Usage: <\$SUPER text> where text is printed/displayed as superscript: text

SUBSC : subscript text

Usage: <\$SUBSC text> where text is printed/displayed as subscript: text

CHEMF : chemical formula. All numerals are printed/displayed as subscript

Usage: <\$CHEMF H2O> which is printed/displayed as H₂O

2.4 Mathematical functions

INTEGR : integral. The function needs two boxes containing the text above and below the integral sign respectively

Usage: <\$INTEGR <\$overtext><\$undertext>> where overtext is printed/displayed above the integral sign, and undertext is printed/displayed below the integral sign

SUM : summation. The function needs two boxes containing the text above and below the summation sign respectively

Usage: <\$SUM <\$overtext><\$undertext>> where overtext is printed/displayed above the summation sign, and undertext is printed/displayed below the summation sign

SQROOT : square root

Usage: <\$SQROOT text> where text is printed/displayed under a square root sign

ROOT : root. The function needs two boxes, the first one containing the power of the root and the second one containing the text to be printed/displayed under the root sign

Usage: <\$ROOT <\$3><\$text>> where text is printed/displayed under a root sign and 3 is printed/displayed above the root sign indicating the power

FRAC : fraction. The function needs two boxes containing the text above and below the line respectively
 Usage: **<\$FRAC <\$\$abc><\$\$def>>** where abc is printed/displayed above the line and def is printed/displayed below the line

2.5 Tables etc.

TEXTSTACK : text stack. The function needs two to four boxes containing texts which is to be printed/displayed one on top of the other
 Usage: **<\$TEXTSTACK <\$\$text1><\$\$text2><\$\$text3>>** where text1 will be printed/displayed on top, text2 below text1, and text3 below text2. In this case text2 normally will be printed/displayed on the baseline of the running text

MATRIX : matrix. Text is printed/displayed in matrix form, possibly delimited by parentheses. Columns may be aligned using the special symbol **<\$align>** or **#&**. The function needs at least three boxes, but may have any number above that. The first two boxes contain the delimiting parentheses. These boxes are empty if there is no delimiting parentheses. From the third box each box contains one line of matrix text

Usage:
<\$MATRIX <\$\$><\$\$>
<\$\$text1A#&text1B#&text1C>
<\$\$t2A#&t2B#&t2C>
<\$\$longertext3A#&3B#&longertext3C>>

This is printed/displayed as follows:
 text1A text1B text1c
 t2A t2B t2C
 longertext3A 3B longertext3C

Index of symbols and functions

#! or <\$linshift> : line-shift [1.2]	<\$gteq> : greater than or equal sign [1.4]
#" or <\$diaer> : diaeresis (or umlaut) [1.1]	<\$gtgt> : much greater than sign [1.4]
#& or <\$align> : table alignment [1.2]	<\$hardl> : l with stroke: ł [1.3]
#' or <\$acute> : acute accent [1.1]	<\$hardlcap> : L with stroke: Ł [1.3]
#* or <\$caron> : caron (or háček) [1.1]	<\$hook> : hook or caudate [1.1]
#+ or <\$lincont> : line continues [1.2]	<\$ident> : identical symbol [1.4]
#, or <\$cedil> : cedilla [1.1]	<\$ijlig> : digraph ij [1.3]
#- or <\$endash> : en-dash [1.2]	<\$ijligcap> : digraph IJ [1.3]
#. or <\$thinspace> : non-breakable thin space [1.2]	<\$infin> : infinite symbol [1.4]
#/ or <\$para> : new printed paragraph [1.2]	<\$inodot> : dotless i: i [1.3]
#: or <\$space> : non-breakable space [1.2]	<\$INTEGR <\$\$><\$\$>> : integral [2.4]
#= or <\$emdash> : em-dash [1.2]	<\$ITALIC > : italic typeface [2.1]
#^ or <\$circum> : circumflex accent [1.1]	<\$LATIN > : text in Latin alphabet [2.2]
#` or <\$grave> : grave accent [1.1]	<\$lincont> or #+ : line continues [1.2]
#~ or <\$tilde> : tilde [1.1]	<\$linecol> : line-colon minus/division sign [1.4]
#0–#9 : free symbols [1.5]	<\$linshift> or #! : line-shift [1.2]
<\$acute> or #' : acute accent [1.1]	<\$lt> : less than sign [1.4]
<\$aelig> : letter ae: æ [1.3]	<\$lteq> : less than or equal sign [1.4]
<\$aeligcap> : letter AE: Æ [1.3]	<\$ltlt> : much less than sign [1.4]
<\$align> or #& : table alignment [1.2]	<\$macron> : macron [1.1]
<\$approx> : approximately equal to sign [1.4]	<\$MATRIX <\$\$><\$\$> <\$\$> <\$\$>> : matrix [2.5]
<\$aring> : a with ring above: å [1.3]	<\$negat> : negation sign [1.4]
<\$aringcap> : A with ring above: Å [1.3]	<\$oelig> : ligature oe: ø [1.3]
<\$arrowdoub> : double arrow [1.4]	<\$oeligcap> : ligature OE: Œ [1.3]
<\$arrowdown> : arrow down [1.4]	<\$oeslash> : o with stroke: ø [1.3]
<\$arrowleft> : arrow to the left [1.4]	<\$oeslashcap> : O with stroke: Ø [1.3]
<\$arrowright> : arrow to the right [1.4]	<\$ogonek> : ogonek [1.1]
<\$arrowup> : arrow up [1.4]	<\$para> or #/ : new printed paragraph [1.2]
<\$backsl> : backslash [1.4]	<\$parall> : parallel sign [1.4]
<\$BOLD > : bold typeface [2.1]	<\$plusminus> : plus minus sign [1.4]
<\$brackl> : left square bracket [1.4]	<\$prmil> : per thousand [1.4]
<\$brackr> : right square bracket [1.4]	<\$regt> : registered trade mark sign [1.4]
<\$breve> : breve [1.1]	<\$ring> : ring above [1.1]
<\$caron> or #* : check (or háček) [1.1]	<\$ROMAN > : roman typeface [2.1]
<\$cedil> or #, : cedilla [1.1]	<\$ROOT <\$\$><\$\$>> : root [2.4]
<\$CHEMF > : chemical formula [2.3]	<\$sect> : section sign: § [1.4]
<\$circum> or #^ : circumflex accent [1.1]	<\$space> or #: : non-breakable space [1.2]
<\$copyr> : copyright sign: © [1.4]	<\$SQROOT > : square root [2.4]
<\$CYRILLIC > : text in Cyrillic alphabet [2.2]	<\$stroke> : stroke through [1.1]
<\$deg> : degree symbol [1.4]	<\$SUBSC > : subscript text [2.3]
<\$diaer> or #" : diaeresis (or umlaut) [1.1]	<\$SUM <\$\$><\$\$>> : summation [2.4]
<\$dobacute> : double acute accent [1.1]	<\$SUPER > : superscript text [2.3]
<\$dotabove> : dot above [1.1]	<\$szlig> : sharp s: ß [1.3]
<\$dotbelow> : dot below [1.1]	<\$TEXTSTACK <\$\$><\$\$><\$\$>> : text stack [2.5]
<\$doubx> : double cross [1.4]	<\$thinspace> or #. : non-breakable thin space [1.2]
<\$emdash> or #= : em-dash [1.2]	<\$thorn> : thorn: þ [1.3]
<\$endash> or #- : en-dash [1.2]	<\$thorncap> : uppercase thorn: Þ [1.3]
<\$eng> : eng: ŋ [1.3]	<\$tilde> or #~ : tilde [1.1]
<\$engcap> : uppercase eng: Ŋ [1.3]	<\$timesdot> : multiplication symbol centred dot [1.4]
<\$eth> : eth: ð [1.3]	<\$timesx> : multiplication symbol x [1.4]
<\$ethcap> : uppercase eth: Æ [1.3]	<\$tridown> : open triangle with corner down [1.4]
<\$FRAC <\$\$><\$\$>> : fraction [2.4]	<\$triup> : open triangle with corner up [1.4]
<\$grave> or #` : grave accent [1.1]	<\$TYPEFACE1 > – <\$TYPEFACE9 > : free typefaces [2.1]
<\$GREEK > : Greek text [2.2]	<\$uneq> : unequal sign [1.4]
<\$gt> : greater than sign [1.4]	<\$vert> : vertical line [1.4]