Meta
Journal des traducteurs
Translators' Journal

MicroMATER. A Proposed Standard Format for Exchanging Lexical / Terminological Data Files

Alan K. Melby

Volume 36, Number 1, mars 1991

La terminologie dans le monde : orientations et recherches

URI: https://id.erudit.org/iderudit/003620ar

See table of contents

Publisher(s)

Les Presses de l'Université de Montréal

ISSN

0026-0452 (print)

Explore this journal

Cite this article

Melby, A. K. (1991). MicroMATER. A Proposed Standard Format for Exchanging Lexical / Terminological Data Files. *Meta*, *36*(1), 135–160.

Article abstract

As more and more translators and terminologists use microcomputers, there will be an increased need to transfer terminology files electronically between requesters of translation and translators, between large terminology databases and users. Such exchanges could be greatly facilitated by widespread adoption of a standard intermediate format. Terminology files convened to this format would be accessible to all who use the standard, without their needing to know the details of the proprietary format from which the file was converted. MicroMATER, an application of the MATER standard, which can be used on microcomputers and mainframes alike, is proposed as a candidate to fill the need for an exchange format. It is flexible enough to accommodate a variety of record layouts and lexically-oriented files as well as concept-based terminological files. MicroMATER files can even be created in a text editor and manipulated by a shareware program available from the BYU-TRG.

MICROMATER. A PROPOSED STANDARD FORMAT FOR EXCHANGING LEXICAL/TERMINOLOGICAL DATA FILES

ALAN MELBY
Brigham Young University, Provo, USA

RÉSUMÉ

En raison de l'utilisation sans cesse croissante des micro-ordinateurs par les traducteurs et les terminologues, il y a une demande importante pour un échange électronique des fiches terminologiques. Cet échange serait grandement facilité par l'adoption générale d'un format intermédiaire standard. Les fiches terminologiques converties dans ce format seraient accessibles à tous sans qu'il y ait besoin de connaître les détails du format de départ. MicroMATER, une application du format de transfert MATER pouvant être employé tant sur les micro-ordinateurs que sur les gros appareils, est probablement le plus apte à remplir la fonction d'échange électronique de fiches. Les fiches du MicroMATER peuvent être intégrées à un programme de traitement de textes et peuvent également être manipulées par un programme dont les coûts d'exploitation seraient partagés par les utilisateurs.

ABSTRACT

As more and more translators and terminologists use microcomputers, there will be an increased need to transfer terminology files electronically between requesters of translation and translators, between large terminology databases and users. Such exchanges could be greatly facilitated by widespread adoption of a standard intermediate format. Terminology files converted to this format would be accessible to all who use the standard, without their needing to know the details of the proprietary format from which the file was converted. MicroMATER, an application of the MATER standard, which can be used on microcomputers and mainframes alike, is proposed as a candidate to fill the need for an exchange format. It is flexible enough to accommodate a variety of record layouts and lexically-oriented files as well as concept-based terminological files. MicroMATER files can even be created in a text editor and manipulated by a shareware program available from the BYU-TRG.

I. OVERVIEW OF MICROMATER

LEXICAL/TERMINOLOGICAL DATA

Lexical/terminological data (Lex/Term data for short, L/T data for even shorter) is a cover term for glossaries, dictionaries, terminologies, lexicons, and similar types of data, whether monolingual, bilingual, or multilingual. When L/T data is put into machine-readable form, it is called an L/T data file or an LTDF. (An LTDB would be a collection of LTDFs into a database.) The storage, retrieval, and manipulation of L/T data (esp. by computer software) is called L/T data management. The information in an L/T data file is divided into a header and a varying number of records. An L/T record consists of L/T Units (LTUs) and their associated information fields. Each record has a tree structure (implicit or explicit) organized around one or more L/T Units (an L/T Unit being a word, term, phrase, even boilerplate text or a bibliographic reference).

Glossaries, dictionaries, terminology files and other L/T data files have the following in common:

(1) They consist of records. That is, they are not continuous text. These records are often called entries or articles. Sometimes a very long entry or article is divided up into several records.

(2) Each record has a primary key. Although a record may be retrieved in various ways, its basic order in the file is determined by its primary key, which is sometimes called the headword or the record identifier. In some terminology files, the primary key will be a record number. In others, it will be based on the primary LTU, often called the main entry term. Whatever it is, each primary key is normally unique among all the primary keys in a file. Here the primary key will be called the record identifier (Record ID for short, RID for shorter).

(3) The data in the records is information about LTUs. It consists of definitions/descriptions, translation equivalents, sources, relationships to other LTUs, and administrative information about LTUs. A story divided up into paragraphs, with each paragraph in a record and a heading as a primary key, would not be an L/T data file. But an L/T data file is not like a traditional database either, since two records in the same L/T data file may have different fields and since some fields are repeated in the same record.

EXCHANGE VIA MATER

Until the 1980s and the rise of the microcomputer, L/T data was, of course, managed on mainframes and minicomputers. Developers of large-scale term banks such as Termium and Eurodicautom have for many years felt the need for a way to share L/T data files. The German Standards Institute initiated work on a standard format, which was developed by ISO Technical Committee 37 into a standard exchange format called MATER, which in 1986 was accepted as ISO Standard 6156. The basic motivation for MATER is to facilitate the exchange of L/T data among large scale lexicographical and terminological databases by avoiding the need for a proliferation of conversion programs. If just five databases with different formats wanted to be able to exchange directly with each other, it would require eleven separate two-way conversion programs, and the number of conversion programs goes up dramatically with more formats. If any number of databases all agree to write one program to convert to and from MATER, they can all exchange data by passing through the MATER format as a neutral intermediate format.

L/T DATA MANAGEMENT ON MICROCOMPUTERS

In the early 1980s, L/T data management software for microcomputers began to appear on the market. This trend has evolved until now L/T data management can be considered to be a new category of microcomputer software. There are five identifiable groups of professional users of L/T data management software: (1) terminologists and lexicographers, (2) translators, (3) technical writers, (4) students learning a second language, and (5) multilingual business persons who read or write in a second language as part of their daily work. Sometimes all five groups are included in a larger category of users called "wordworkers".

With the exception of some terminologists and dictating translators, all five groups often manage L/T data while writing a text and need the ability to quickly call up L/T data files without shutting down their word processing software.

With the exception of some terminologists and some translators and technical writers, all five groups want ready-made L/T data files to refer to, and they generally want the ability to annotate the ready-made files, and to add new records for items missing from the ready-made files, in addition to the ability to create their own L/T data files.

Some of the lexical data management software packages available (mostly in Europe, a few in North America) as of late 1989 are sold under the trademarks and corporate names TermTracer, Interdoc, Microcézeau, Term-PC, MTX (under various labels), SuperLex, ProfiLex, Inductel, Danterm, Phoenix/Aquila, Term-Lidas, and TermDok. Several more are just coming onto the market.

Many of these packages allow the user to create L/T data files. Wordworkers are becoming more open about sharing L/T data files they create. In the past, some translators treated their L/T data on card files as proprietary information. This was possible because requesters of translation generally did not provide, along with the document to be translated, a glossary of technical terms and translation equivalents used throughout the organization. The result was inconsistent terms in documents produced by different translators. In today's technological world, this is changing. A translator now protects his flow of work by using exactly the terminology specified by the requester and by creatively using general vocabulary to blend the technical terms into an excellent document.

Monolingual technical writers also need to use the technical terms specified by the requesting organization.

The result is that many wordworkers and wordworking organizations need to exchange L/T data files, but they are not all using the same L/T data management package, and no one package or microcomputer is likely to be used exclusively. This means that there will be an increasing need to exchange L/T data files between incompatible packages, sometimes between incompatible hardware such as IBM-PS/2s and Apple Macintoshes.

THE NEED FOR MICROMATER

At a general level, the MATER standard, although designed for mainframe use, applies to the needs of a microcomputer user. A MATER file consists of a header (containing information that applies to the entire file) followed by any number of varying length records, followed by an end-of-file mark. Each record consists of any number of fields. Each field consists of a "tag" (a field name) and some data.

Here is an example of the kind of data that might be found in a very simple English-French terminology file on solar energy. The "source" consists of a coded name of a document and a page on which the term can be found in context.

Figure 1

HEADER:

Name of file: SOLAR
Languages: English and French

LEXICAL DATA RECORDS:
Record R00453:

English term: air conditioning
French term: climatisation (des locaux)
French source: Duse, page 164
Subject field classification (UDC): 620.9

Record R00877:

English term: backup system
French term: système d'appoint
French source: Bran, page 15
Subject field classification (UDC): 620.9

Record R01355:

English term: fiberglass
French term: laine de verre
French source: Ged, page 493
French term: fibre de verre
French source: PR, page 701
Subject field classification (UDC): 666

Record R01422:

English term: flat-plate collector
French term: capteur plan
French source: Nshci, page 3
Subject field classification (UDC): 620.9

Record R01593:

English term: heat loss
French term: déperdition de chaleur
French source: Duse, page 38
Subject field classification (UDC): 536

END of FILE

In this example, the tags (field names) in the records are: "Record", "English term", "French term", and "French source", and Subject field classification. The UDC (Universal Decimal Code) classification system is widely used in Europe. The UDC code 620.9 is "energy", 666 is "glass industry", and 536 is "thermodynamics". (MicroMATER does not require the use of the UDC system. Any system can be used, if it is used consistently.)

Of course, not all L/T data files are as simple and regular as this one. And a complete terminology file would contain many additional fields in each record, especially an indication of who is responsible for each record and when they last updated it.

Although the general content of a MATER file is quite universal, the specifics of the format are troublesome for a microcomputer user. An actual MATER file contains many binary pointers and length indicators. While this is not a problem when exchanging magnetic tapes which are created by one computer program and read by another computer program, it is a problem if one wants to edit a MATER file in a normal text editor, such as modifying an L/T data file into MATER format.

Keeping the spirit of a MATER file and adding the constraints that it contain only normal ASCII characters and that it be possible to bring it into a normal text editor and understand it and edit it, the Brigham Young University Translation Research Group (BYU-TRG), in close cooperation with other groups, has devised an exchange format called MicroMATER.

At an international symposium on microcomputers and terminology management software held in Vienna in November 1989, a recommendation was adopted which called for TermNet and cooperating groups to "develop a joint terminological exchange format and make it available to the general public". Since that symposium, several meetings have been held between the BYU-TRG and TermNet to combine efforts toward such an exchange format rather than wasting energy on several competing formats.

MICROMATER DESCRIPTION

Version 2 of MicroMATER, as presented below, is not intended to be the final version. It is hoped that feedback from the translation/terminology and lexicography communities will lead to many enhancements.

The general information in the sample L/T data file could be presented as follows in a MicroMATER file:

Figure 2

```
{MM} 2
{=} EN
{NAM} SOLAR
{TYP} NON-DIRECTIONAL
{LA} EN
{LB} FR
{CLS} UDC
———

*R00453
{EN:0LTU} air conditioning
{FR:1LTU} climatisation (des locaux)
{FR:1SRC} Duse, page 164
{RL:CLS} 620.9

*R00877
{EN:0LTU} backup system
{FR:1LTU} syst\eme d'appoint
{FR:1SRC} Bran, page 15
{RL:CLS} 620.9

*R01355
{EN:0LTU} fiberglass
{FR:1LTU} laine de verre
{FR:1SRC} Ged, page 493
{FR:2LTU} fibre de verre
{FR:2SRC} PR, page 701
{RL:CLS} 666

*R01422
{EN:0LTU} flat-plate collector
{FR:1LTU} capteur plan
{FR:1SRC} Nshci, page 3
{RL:CLS} 620.9
```

```
*R01593
{EN:0LTU} heat loss
{FR:1LTU} d/eperdition de chaleur
{FR:1SRC} Duse, page 38
{RL:CLS} 536
@!
```

A quick comparison between Figure 1 and Figure 2 shows that they contain essentially the same information. The sample file is designed to be readable and editable yet it is sufficiently formal to be processed by computer software. We will now examine some details of the sample file in Figure 2.

HEADER

The file header consists of all the lines down to the row of hyphens. The row of hyphens indicates that the L/T data records are beginning and the asterisk means that a new record is beginning and that the primary key will follow. The commercial-a (often called an at-sign) followed by an exclamation point marks the end of the L/T data file.

In both the header and the records, each field (except the primary key field) begins with a field name enclosed in curly brackets. (Pointed brackets can also be used throughout instead of curly brackets, if the computer being used does not support curly brackets.)

The first line of a header must always be a field with the name "MM" for MicroMATER. This field contains the version number. This document describes MicroMATER version 2. All other fields in the header are optional. The "=" field specifies the language of the codes for data categories, in this case ENglish. The ENglish codes are the default if there is no {=} field. The "NAM" field, which names the L/T data file, is optional, but each header should include fields (in this example, "LA" and "LB") which indicate the languages treated in the file. (A third language would, of course, be named in field "LC".) If the languages of the file are not specified in the header, they must be specified in each record. The two-letter abbreviations for languages are from ISO Standard 639 (en=English, fr=French, de=German, it=Italian, nl=Dutch, es=Spanish, ru=Russian, and ja=Japanese). The "CLS" field indicates that classification numbers in this file are UDC codes, unless otherwise specified. Any other information which applies to the whole file would also go into the header.

The asterisk is immediately followed by the data from the RID field (Record IDentifier). The RIDs in this example are arbitrary (consisting of R for record and a record number). The asterisk can be thought of as an abbreviation for "{RID}". Often, the RID will be a term in language A, possibly with a suffix (e.g. "heat pump — 1" or "heat pump — 2") to make it unique when the same term appears in two records.

The fields in the records have field names (sometimes called tags) enclosed in curly brackets. A field name in an L/T data record, other than the RID field, consists of a language code, an L/T Unit number, a data category, and various optional information for more complex cases. As we shall see, sometimes the language code, unit number, or data category can be left implicit for the convenience of the user, making the record more readable. Here the L/T Unit numbers are zero, one, and two.

DATA CATEGORIES

The data categories used in this example are "LTU" (L/T Unit), "SRC" (the source of the information), and "CLS" (Classification of the concept treated in the whole record with RL = Record Level and UDC codes), and where the sources are abbreviations for reference books. Somewhere, associated with the L/T data file (preferably in the same file so it will not inadvertently become separated), there should be a list of source abbreviations and their full bibliographic references.

There are several ways to abbreviate names of sources, and the abbreviations in the SRC fields of the sample data in this paper are arbitrary. It would be nice to have a systematic method, and fortunately there is an international standard for abbreviating the names of sources currently being drafted by ISO/TC 37, the same technical committee which prepared the original MATER standard. The abbreviation system will be included in a standard which will be called, approximately, "Documentation for Terminology Work and Terminography" and will be based on ISO 690, a more general purpose documentation standard.

MESC (MICROMATER ESCAPE)

The commercial-a (at-sign) is a used as an "escape" character which modifies the meaning of the next character. It was chosen since it is used relatively seldom and since its literal use can be avoided by expanding it to the word "at". In those few cases where it is necessary to represent a literal commercial-a, double it ("@@"). Two commercial-a symbols together are interpreted, in a MicroMATER file, as a single, literal commercial-a rather than as an escape character. We will call the commercial-a in its special, non-literal, usage a MESC character (for MicroMATER ESCape), since it lets the character which follows it "escape" from one usage into another. We chose not to use a normal ESC (decimal 27) because not all text editors can edit files containing ESC characters and many printers use the ESC character for commands.

The asterisk also has a special meaning. It marks the beginning of a new lexical data record and is immediately followed on the same line by the primary key (which is the data from the implicit RID field). If a literal asterisk is needed in a MicroMATER file, it is preceded by a MESC ("@*"). An asterisk, in its special usage, is also called a BOR (beginning of record).

EOF (END-OF-FILE)

The exclamation point, used by itself, is not a special character in MicroMATER. However, any character, when preceded by a MESC, receives an alternative meaning just for that instance. For example, an exclamation point preceded by a MESC ("@!") is used to mark the end of an L/T data file and is called an EOF. This mark is optional if the end of the L/T data file is also the end of the file so far as the operating system is concerned.

EOL (END-OF-LINE)

Related to the end-of-file marker are the end-of-line marks. These marks are also optional, since the operating system conventions for marking end-of-line in a text file can also be used. In the sample file in Figure 2, each line is assumed to be terminated by an end-of-line (EOL) mark, which is defined by the operating system being used and is normally invisible. If the user wishes to make the end of line explicit, there are two MicroMATER marks for doing this: HARD-EOL ("@.") and SOFT-EOL ("@;").

The difference between hard and soft EOL can be explained by considering a text consisting of a series of paragraphs. The division of one paragraph into lines is normally arbitrary and the margins can be changed, resulting in more or fewer lines. These lines are terminated by SOFT-EOLs. However, the division of the text into paragraphs is not as arbitrary, and each paragraph is terminated by a HARD-EOL. In a MicroMATER file, a field can be longer than one line. If a user wants to create an arbitrary EOL that can be removed when reformatting, the user may insert a SOFT-EOL mark immediately before the operating system's EOL.

In a MicroMATER file, a field can span several lines. So a field is terminated not by an EOL but rather by the beginning of a new field. This is consistent with another international standard which has been used in formulating MicroMATER, namely SGML (Standard Generalized Markup Language), which is ISO Standard 8879. (Editor's note: please footnote/endnote the following section of text.)

[[[In fact a MicroMATER file can be viewed as an SGML file with field names as commands enclosed in curly brackets instead of pointed brackets (i.e. a "pure" SGML file would use "<MM>" instead of "{MM}"). One change between MicroMATER version 1 and version 2 has been to stop using the pointed brackets as special characters so that they could be used to enclose field names if desired. The special field name "*" can be viewed as an abbreviation for "<RID>", and the begin and end lexical-data-file marks ("———" and "@!") can be viewed as equivalent to "<ltfile>" and "</ltfile>" in pure SGML. The TEI (Text Encoding Initiative) committee, an international panel which is working on an application of SGML for the exchange of scholarly texts, even considered including L/T data files in their standard, but stopped short of that and agreed to cooperate with the Translation Research Group during parallel development.]]]

(Editor's note: end of footnote)

CHARACTER SETS

The base character set in a MicroMATER file is ISO 646 (which is essentially the same as seven-bit ASCII). Both the IBM and Apple Macintosh character sets share these common seven bits, but they differ on accented characters, which use an eighth bit. Therefore a truly universal system for representing accented characters must be seven-bit, not eight-bit. In order to represent accented characters with seven bits, MicroMATER uses one character as a universal escape character, namely the commercial-a. Other characters may have a dual meaning, depending on whether or not they are preceded by the MicroMATER escape character.

Accented characters are not found as single characters in ISO 646, and MicroMATER uses character sequences to represent them. In the sample file (Figure 2), the "e" with a grave accent in "système" is represented by preceding the "e" with a backslash ("\e"), and the "e" with an acute accent in "déperdition" is represented by preceding the "e" with a normal slash ("/e"). This method was chosen as one option because it is relatively compact and mnemonic. It does mean, however, that all literal slashes (actually all accent characters, namely, "/\%^~#") must be preceded by a MESC (e.g. "@/" and "@\"). The "/" (slash) is used for the acute accent, the "\" (backslash) for the grave accent, the "%" (percent) for the diarisis or umlaut, the "^" (caret) and "~" (tilde), naturally, for the circumflex and tilde accents respectively, and the "#" (crosshatch/number sign) for miscellaneous accents not covered above (e.g. #s for German sharp-s).

An alternative method of representing accented characters is the one suggested by the SGML standard, in which the accented character is spelled out between an ampersand ("&") and a semicolon (";") as follows:

"système" becomes "syst&egrave;me" and

"déperdition" becomes "d&eacute;perdition".

This method allows an unlimited variety of extended characters to be represented, but it is not nearly as readable nor as compact. See Appendix C.

A third method is to switch into an alternative character set. A MESC followed by a left parenthesis indicates a shift to another character set. For example, @(ISO649,64) specifies a shift to the base character set and the use of the commercial-a (decimal 64) in that set as the MESC character, which will be needed to shift back out of the character set. Until explicitly changed, a MicroMATER file may use only characters from ISO 649.

FIELD NAMES

The field names in the example do not reveal the total power of MicroMATER field names in records. The full form of the field name {FR:1SRC} is actually {FR:1SRC1EN}. The five parts of a full field name are:

(1) Section Language (*FR*:1SRC1EN)
(2) Unit Number (FR:*1*SRC1EN)
(3) Data Category Name (FR:1*SRC*1EN)
(4) Data Category Iteration Number (FR:1SRC*1*EN)
(5) Field Data Language (if different from Section Language)
(FR:1SRC1*EN*)

The field name is then followed by the field data. The field data is also called the "value" of the field.

In terminology files one record deals with only one concept, so there should be only one lexical unit per language, but on occasion, one might find two or more L/T Units that are practically synonymous. On the other hand, in dictionaries of general vocabulary, one record deals with the range of senses of one headword, so there will often be several L/T Units, such as several definitions or translation equivalents. For example a dictionary might include various senses of the word "match" as a device to light fires, an athletic competition, etc. Hence, in both lexical and terminological files, there may be several LTUs in one record, and in multilingual files they will fall into subsets or sections according to language. Thus, in both lexical and terminological files, the fields of a record are related to each other in a tree structure, with the RID field at the root (level 1), the language sections at the second level, the L/T Unit numbers with associated data categories at the next level, the field data language at the fourth level, and the field data attached as terminal nodes at the fifth level.

Often an element of the field name can be omitted if that element can be reconstructed according to the rules for defaults. There are several rules for reconstructing implicit elements of a field name:

RULES FOR DEFAULTS

The elements of a field name must always appear in the standard order to avoid ambiguity and allow machine processing of MicroMATER files.

When there is no data category specified, it is assumed to be "LTU". So, {FR:1} can be expanded to {FR:1LTU}.

In a directional bilingual L/T file, several defaults are used to simplify the appearance of the field names. The source language is language A and the target language is language B as defined in the MicroMATER file header. Then, by convention, L/T Unit number zero is assumed to be in the source language section and L/T Units greater than or equal to the number "1" are assumed to be in the target language section. This allows the section language to be left implicit. For example, if language B in the file header is French, the simple field name {1} can be expanded to {FR:1LTU}, {2} becomes {FR:2LTU}, etc.

An LTU number can also be given an explicit language. For example, in a trilingual file from English into French and German one might find two LTUs into French and one into German. The question that arises is whether to number the German LTU {DE:3} or {DE:1}. MicroMATER allows both options, but the options have different implications for defaults. Suppose one has three translation equivalent fields in a row ({FR:1}, {FR:2}, {DE:1}) followed by a definition for LTU one ({1DEF}). Does this refer to the French LTU "1" or the German LTU "1"? A general MicroMATER rule is to choose the language of the closest preceding field with the same LTU number, so {1DEF} would be interpreted as {DE:1DEF}. But to avoid confusion, it might be best to number the LTUs consecutively ({FR:1}, {FR:2}, {DE:3}).

Often, especially in a carefully made terminology file or general dictionary, there will be several pieces of information associated with each L/T unit, such as a definition, a contextual example, links to related records, and the status of the unit (such as standard, deprecated, or preferred). If a field lacks an L/T unit number, it inherits the L/T unit number of the preceding field. So, "{1} climatisation {SRC} Duse, page 164" can be expanded to "{FR:1LTU} climatisation {FR:1SRC} Duse, page 164". Of course, the L/T unit number should not be left off a field whose data category is LTU (since LTU is normally the data category of the first field of a new unit).

The above defaults allow the field names in Figure 2 to be simplified as follows:

Sometimes an entire field can be implicit (e.g. a grammar or definition field). If a field must be generated for an L/T unit, it is assumed to have the same data as the same data category field on L/T unit "0" or "1", depending on the language.

Figure 3 shows a MicroMATER equivalent to the one in Figure 2, except that it has been simplified by taking advantage of defaults in field names.

Figure 3

```
{MM} 2
{=} EN
{NAM} SOLAR
{TYP} NON-DIRECTIONAL
{LA} EN
{LB} FR
{CLS} UDC
————

*R00453
{0} air conditioning
{1} climatisation (des locaux)
{SRC} Duse, page 164
{RL:CLS} 620.9
*R00877
{0} backup system
{1} syst\eme d'appoint
{SRC} Bran, page 15
{RL:CLS} 620.9
*R01355
{0} fiberglass
{1} laine de verre
{SRC} Ged, page 493
{2} fibre de verre
{SRC} PR, page 701
{RL:CLS} 666
```

```
*R01422
{0} flat-plate collector
{1} capteur plan
{SRC} Nshci, page 3
{RL:CLS} 620.9
*R01593
{0} heat loss
{1} d/eperdition de chaleur
{SRC} Duse, page 38
{RL:CLS} 536
@!
```

In the case of a simple, directional bilingual L/T data file, one can even simplify further by placing the source language term in the RID field. In this case, the {EN:0LTU} field data is assumed to be RID field data.

The first part of the sample file might be reformatted as in Figure 4.

Figure 4

```
{MM} 2
{=}EN
{NAM} SOLAR
{TYP} DIRECTIONAL
{LA} EN
{LB} FR
{CLS} UDC
————
*air conditioning
{1} climatisation (des locaux) {SRC} Duse, 164
{RL:CLS} 620.9
*backup system
{1} syst\eme d'appoint {SRC} Bran, page 15
{RL:CLS} 620.9
*fiberglass
{1} laine de verre {SRC} Ged, page 493
{2} fibre de verre {SRC} PR, page 701
{RL:CLS} 666
```

MicroMATER is a powerful application of MATER because it is highly flexible and can represent the information in many different proprietary formats for purposes of exchange. Yet for simple cases, defaults make it look simple, and existing text files can be hand edited into MicroMATER format without writing a computer program.

ANCILLARY INFORMATION

There is a more detailed definition of MicroMATER which follows in the second part of this paper. It describes five conventions. A file fully conforms to the MicroMATER standard if it conforms to all five conventions. However, it may still be useful for a file to conform to a subset of the conventions. The five MicroMATER conventions are:

(1) the General Convention — the use of the commercial-a and how to mark end-of-line and end-of-file

(2) the Record Format Convention — the file header, the record delimiters, and the field names

(3) Record Identifier Convention — RID uniqueness

(4) he Field Name Convention — suggested data category names and allowable combination field names

(5) the Character Code convention — mnemonic codes for accented characters

The Translation Research Group also distributes at cost of diskette, shipping and handling, a utility program (called MMUTS) for PC-DOS computers that manipulates MicroMATER files in various ways. It is intended to encourage the use of MicroMATER.

A long-range objective of the Translation Research Group concerning MicroMATER is to convince every vendor of an L/T data manager to supply with the software package a utility which converts both ways between the vendor's proprietary format and MicroMATER format. This would allow end-users to exchange L/T data files between many different hardware and software systems without having to write any computer programs.

II. MICROMATER DEFINITION

The MicroMATER standard consists of five conventions, which are described below in somewhat more detail than in the first part of this document. The conventions being described here apply only to version 2 of MicroMATER.

1. GENERAL CONVENTION

This convention guarantees that a file will be editable in any standard text editor and printable on any standard printer on any ASCII-based computer. It should also allow easy conversion to and from files on EBCDIC-based computers (typically IBM mainframes and compatibles). It uses the commercial-a (@) as the MicroMATER escape character (also called the MESC character) to permit soft (@;) and hard (@.) end-of-line marks and an end-of-file mark (@!) which are independent of any particular operating system conventions. But it also allows the use of operating system conventions in place of these marks to avoid visual clutter in the file. Typically, operating system conventions are used for hard-end-of-line and end-of-file marks while the MicroMATER convention (@;) is used for soft-end-of-line marks. A soft-end-of-line mark is useful to preserve the distinction between an arbitrary end-of-line which could change in reformatting (soft) and an end-of-paragraph (hard) which should remain unchanged in reformatting.

The General Convention is useful for exchanging text files even if they are not L/T data files because it allows paragraphs to be kept as units. In this case, lines could be assumed to be terminated by soft-end-of-line marks unless there is an explicit hard end-of-line mark (@.).

Example (using explicit General Convention marks for text):

```
      Radu says that events in Hungary probably @;
had the most significant impact because of the @;
proximity and the number of ethnic Hungarians @;
in Romania. @.
      Events in Bulgaria were yet another source @;
of inspiration for the Romanians. @.
@!
```

2. RECORD FORMAT CONVENTION

This convention defines the structure of an L/T data file as a file header consisting of one or more fields, followed by a row of hyphens, followed by one or more records.

Each record must begin with the Record ID (RID) preceded by a Beginning-of-Record mark (the asterisk). Within a record, fields are identified by enclosing the field name in Field-Name marks (usually the opening and closing curly brackets). In a traditional database record, the field names are implicit, rather than explicit as in a MicroMATER record. Implicit field names force every record to have the same layout; explicit field names allow enormous flexibility.

The Record Format convention assumes that the file conforms to the General Convention, and the MicroMATER escape character (@) is used to indicate a literal asterisk (@*) or curly bracket (@{) when the user does not want the asterisk (*) or curly bracket ({) interpreted as Record Format marks.

The file header must always begin with the field name {MM} followed by the version of MicroMATER.

When the first line of a file is "{MM} 2", it is assumed that the file conforms to the MicroMATER version 2 Record Format convention.

Some computer systems do not have curly brackets in their basic character set. In this case pointed brackets may be used. Then the first line of the file would be "<MM> 2" and pointed brackets would be used throughout the file as field name marks. And literal pointed brackets would be represented as "@<" and "@>".

The Record Format Convention requires only the {MM} field name. All other field names are detailed in the Field Name Convention. So even if a file uses nonstandard field names, it can still conform to the Record Format Convention. Of course, exchange of files between different systems will be greatly facilitated if the files follow the Field Name Conventions.

Example (for a monolingual dictionary entry): [Note that the field name DES is used for the DEScription of modern words and phrases based on the item of mythology.]

{MM} 2
{NAM} Mythology

*Morpheus
{1DEF} A Roman god of sleep and dreams.
{DES} Someone who is "in the arms of Morpheus" is @;
asleep. The narcotic morphine was named after Morpheus.
{SRC} The Dictionary of Cultural Literacy

3. RECORD IDENTIFIER CONVENTION

The Record Identifier (RID) is the data following the Beginning-of-Record mark (the asterisk) on the first line of a record. The RID must be confined to a single line (a maximum of 50 characters is recommended) and must be unique throughout the file. That is, no two RIDs in the same file may be the same.

If a MicroMATER file uses the Primary LTU (the PLT) as the RID (instead of an arbitrary number), it is usually much easier to read the file, but certain difficulties can arise. For example the PLT can be too long for an RID if it is a very complex technical term. When this happens, a shortened but unique form of the PLT must be used as the RID and the full PLT must appear in a regular field (usually field zero, i.e., {0}).

If two records have the same PLT, then some "tie breaker" (such as a number) must be placed after the PLT in each record to make them unique (for example, "bank — 1" and "bank — 2").

4. FIELD NAME CONVENTION

There are two types of fields, a field in the file header and a field in the body of a record.

File header fields

The first line of a file header must consist of a field ({MM}) that identifies the file as a MicroMATER file. This field was discussed in the convention on Record Format. The other field names that can appear in the file header are:

{=} This field is used to specify the set of data categories that are used in the file. If this field is omitted, it is assumed to have the value "EN", meaning the standard three-character ENglish data categories.

{NAM} The value of this optional field is the NAMe of the file.

{TYP} The value of this optional field is either DIRectional or NON-directional (written out in full or abbreviated to three characters. A non-directional file is a concept-based terminology file which is fully symmetrical, that is, reversible. A directional file is a lexical file of more general vocabulary or a terminology file which is not fully reversible.

{LA} {LB} These fields should contain standard two-character codes of the first two languages in the file. If the file is directional, they are assumed to be the source and target languages. If there are units for more than two languages, those languages should be listed as {LC}, {LD}, etc.

{CLS} This field, when it appears in the header, shows which classification scheme is being used throughout the file. For example, "{CLS} UDC" means that in any record a number following a {CLS} field number will be assumed to be a UDC code.

In the future there will be a field to specify collating sequence.

Record level fields

The fields in the file header apply to all the records in the file. Within a record, the record level fields apply to all the fields in a record. One record level field, the record identifier, receives special treatment and has already been discussed in Convention 3, above. Another record level field is {RL:RTY} (Record TYpe), whose values can be:

LTR	Lexical/Terminological Record
PHR	PHRaseology record
BPL	Boiler PLate record
BIB	BIBligraphy record
ARC	ARChive term record
XRR	X[cross]-Reference Record

This MicroMATER document discusses only LTRs, and if the RTY field is omitted, it is assumed to have the value LTR.

Within an LTR, some data categories can apply to either a single unit or to the entire record. To indicate that a field applies to the entire record, the code RL is used in the position where a language code would normally appear. Since no language in ISO 639 uses RL as its code, this causes no ambiguity.

Unit level fields

Most fields contain either an LTU or information about a single LTU. These are called Unit level fields.

A fully explicit field name consists of five parts (using the sample field name {FR:1SRC1EN}):

(1) Section Language (*FR*:1SRC1EN)
(2) Unit Number (FR:*1*SRC1EN)

(3) Data Category Name (FR:1*SRC*1EN)
(4) Data Category Iteration Number (FR:1SRC*1*EN)
(5) Field Data Language (if different from Section Language) (FR:1SRC1*EN*)

Defaults

Field names are seldom written out with all five parts explicit. Humans prefer to leave some of the information implicit, but computers prefer explicit information. So, MicroMATER includes a set of defaults for deciding the value of the implicit parts of a field name. This allows humans to use shorter field names and computers to expand them to their full form.

We will work from right to left on the five parts of a field name:

The Field Data Language is almost always left implicit because it is almost always the same as the Section Language. For example, German LTUs will probably have German definitions. So, if the Field Data Language is left off, it is assumed to be the Section Language, even if the Section Language is implicit and is inferred using some other default.

The Data Category Iteration number is assumed to be "1" if not mentioned explicitly. An example of using an iteration number would be a single LTU that has two SYNonyms. The field names would be {SYN1} and {SYN2}.

The Data Category Name is usually explicit. If it is omitted, and the Unit Number is explicit, it is assumed to be LTU.

The Unit Number is usually explicit when the Data Category Name is LTU, which is normally the first field of information about a Unit. Other fields of information about a Unit, which tell about the Unit's source, subject matter field, etc., often omit the Unit Number. When a field has a Data Category but not Unit Number, the Unit Number is assumed to be the same as the Unit Number of the previous field, even if that field's Unit Number was also implicit.

The Section Language, if explicit, must be followed by a colon so that it will not be confused with a Data Category Name. If a file is directional and the source and target languages are specified in the header, the Section Language can be left out of all field names if the Unit Numbers conform to the following rule: Any field associated with the source language uses Unit Number zero, and any field associated with the target language uses Unit Number one or more. If a field name has no Unit Number, the implicit Unit Number is calculated and the implicit Section Language is assumed to be the language associated with the calculated Unit Number. If a field name does have a Unit Number but no explicit Section Language, the closest previous field with the same Unit Number is used to infer the Section Language.

This concludes the rules for calculating implicit parts of a field name. Now we will discuss what values the parts of a field name can take.

Values for Section Language

The Section Language is always a two-character code for the name of the language being treated in the section. A section is all the units of a given language in a given record. The two-character code for language conforms to the ISO standard 639 (1988). Some codes for major languages are:

de=German; el=Greek; en=English; es=Spanish; fr=French; it=Italian; ja=Japanese; ru=Russian; zh=Chinese. (ISO standard 639 gives the abbreviations in lower case, while MicroMATER is case-insensitive.)

Additional standard language codes are given in Appendix A.

Values for Unit Number

Unit Numbers are usually whole numbers, but in very complex records, they can contain decimal points to form implicit sub- branches.

Values for Data Category

The Data Categories used in this document are three-letter mnemonics for English data category names. They are based on years of work in careful terminography by Infoterm and the Terminology Committee of the American Translators Association and should be capable of adequately representing most of the information in terminology files that will be converted to MicroMATER format. If an alternative set of data categories is used, this should be indicated in a field in the file header. This special field "{=}" should be followed by the name of the set of data categories being used in the file.

The standard MicroMATER Data Categories are described in Appendix B.

The five-part field name structure allows considerable "combinability". For example, there can be a SRC field for each Unit in each Section. But sometimes, additional pinpointed fields are needed, such as a SRC for a definition within a Unit or a Date and/or Responsibility code for one part of a Unit. To accomplish this, a fourth letter (D=Date, R=Responsible person, S=Source) is added to the appropriate three-letter Data Category, forming a new, additionally combined Data Category.

Values for Iteration Number

Iteration Numbers are usually whole numbers, like Unit Numbers.

Values for Field Data Language are the same as for Section Language.

5. CHARACTER CODE CONVENTION

To maximize the ability to share a file among different computers, the individual characters in a file should be standard ASCII characters between 32 and 126. The only control characters allowed are 10 and 13 (to mark end of line). Even though 127 is an ASCII character, it is to be avoided since it does not have a standard symbol associated with it, and its effect on editors and printers is unpredictable. The dollar sign ($) should also be avoided since it may differ from country to country.

Not counting the methods of extending the character set, the following characters have a special meaning and must be preceded by a MESC (commercial-a) to have their literal meaning:

@ (commercial-a)
* (asterisk)
{ (left curly bracket)
} (right curly bracket)

There are three methods of extending the character set in a MicroMATER file.

(1) Character signals

The following characters are used by default as signals:

% (percent) [for umlaut]
/ (slash) [for acute accent]
\ (back-slash) [for grave accent]
^ (caret) [for circumflex accent]
^ (tilde) [for tilde accent]
(cross-hatch) [for miscellaneous special characters]
& (ampersand) [to signal an SGML character name]

#a = a-circle

#a = A-circle
#c = c-cedilla
#C = C-cedilla
#S = German sharp-S
#? = Spanish open question mark
#! = Spanish open exclamation mark

When any one of the seven character signals is needed in its literal meaning, it must be preceded by a MESC. Just as the Record Format marks.

Sometimes, when a file is being exchanged with another software package on the same computer, eight-bit codes can be used, even though this will not work in general (since the same eight-bit code may mean something different on an IBM-PC and an Apple- Macintosh). When it is safe to use eight-bit codes, the character signals can be used as literal characters, but the file no longer conforms to the Character Code convention.

The two-character MicroMATER codes, formed by a signal character followed by the character the signal applies to, are convenient, relatively short, and mnemonic. But they do not include all the symbols one might wish to include in a MicroMATER file. There are two additional methods available in MicroMATER. One method is borrowed from the conventions associated with SGML (ISO standard 8879).

There are a number of accented characters and other symbols defined in SGML using an ampersand to begin the code and a semi-colon to end it. The semi-colon is not a character signal because it has a special meaning only in the context of an SGML character code. Some of the common SGML character codes are listed in Appendix C.

The third method of obtaining additional characters is to switch to another ISO character set temporarily. A MicroMATER file is normally in ISO character set 649 (which differs from ASCII only in that the local currency symbol is not necessarily a dollar sign). Suppose one wants to create an English-Russian bilingual terminology file. Then one might switch to ISO character set 8859-5:1988, which includes both Latin and Cyrillic characters. In this character set, the commercial-a (here called the commercial-at) is in position decimal 64 (which is equivalent to ISO position 04/00), just as it is in ISO 649. So the switch might look like this: @(ISO8859-5,64). After that character set code, all characters will be interpreted as being in ISO character set 8859-5 until the designated escape character (@=64) is encountered. Once the escape character is encountered, the next character is assumed to be a left paren or some other character from ISO 649 and a new character set can be designated. If the character set being switched into does not have a commercial-a, some other character can be designated as the escape character. If one is going to be switching back and forth between character sets regularly, single letters can be set up in the file header to designate character sets. For example, the fields "{CS1} R=(ISO8859-5,64)" and "{CS2} E=(ISO649,64)" define two character sets (R for Russian and E for English). Then a simple @R will switch to Cyrillic characters and @E will switch back to English characters.

When in an alternate character set, the character signals are not treated in any special way, as they are when in ISO 649.

Acknowledgements: Of the many colleagues who have willingly contributed to the definition of MicroMATER over the past several years, I would like to name three (Dr. Sue Ellen Wright, chair of the Terminology Committee of the American Translators Association; Christian Galinski and his staff at Infoterm in Vienna; and Don Chapman, my research assistant at the BYU-TRG) and would like to thank them all very much.

APPENDIX A — STANDARD TWO-CHARACTER NAMES FOR LANGUAGES

For a complete list of standard two-character codes, see ISO 639: 1988. ISO standards are available from national standards organizations, such as ANSI in the United States or from the ISO office in Switzerland, case postale 56; 1, rue de Varembé; CH-1211 Geneva 20, Switzerland.

ISO standards are also available from Global Engineering Documents, (800) 854-7179 (from within the USA) / (714) 261-1455 in the United States 2805 McGaw Ave, 92714 Irvine, California USA.

af=Afrikaans;
ar=Arabic;
bg=Bulgarian;
bo=Tibetan;
ca=Catalan;
co=Corsican;
cs=Czech;
cy=Welsh;
da=Danish;
de=German;
el=Greek;
en=English;
eo=Esperanto;
es=Spanish;
et=Estonia;
eu=Basque;
fa=Persian;
fi=Finnish;
fj=Fiji;
fr=French;
ga=Irish;
hi=Hindi;
hu=Hungarian;
hy=Armenian;
in=Indonesian;
is=Icelandic;
it=Italian;
iw=Hebrew;
ja=Japanese;
ji=Yiddish;
ka=Georgian;
km=Cambodian;
ko=Korean;
la=Latin;
lt=Lithuanian;
mo=Moldavian;
nl=Dutch;
no=Norwegian;
pl=Polish;
pt=Portuguese;
qu=Quechua;
ro=Romanian;
ru=Russian;
sa=Sanskrit;
sm=Samoan;
sq=Albanian;
sv=Swedish;
sw=Swahili;
th=Thai;
tl=Tagalog;
to=Tonga;
tr=Turkish;
uk=Ukrainian;
ur=Urdu;
vi=Vietnamese;
zh=Chinese;
zu=Zulu

APPENDIX B — DATA CATEGORIES FOR L/T RECORDS

There are two types of data categories: basic and qualifying. In this appendix the basic data category is placed on the left margin, while the qualifying data category is indented. Remember that a record is always organized around an LTU. Any further information about an LTU such as its type or status must be given in another field associated with the LTU. That associated field consists of a basic data category in the field name and a qualifying data category AS A VALUE. Perhaps an example will clarify:

{LTU} Harmonica
{LTY} PLT

Here the LTU is "Harmonica" and its type (LTY) is PLT, or Primary LTU as indicated in this appendix. Obviously this format is verbose and can be compressed into a composite field as follows:

{PLT} Harmonica

Here the qualifying data category (PLT) is used as if it were a basic data category.

Therefore the qualifying data category can be used either as a value of the field in the long form of the field name or as shorthand form for the data category of a composite field.

THE LEXICAL/TERMINOLOGICAL UNIT

LTU Lexical/Terminological Unit

LTY	Lex/Term TYpe	
	PLT	Primary Lexical/Terminological unit
	SYN	SYNonym
	SCI	SCI international SCIentific term
	VAR	VARiant/alternate spelling or form
	LEG	LEGal term
	STT	STandardized Term
	LTE	Lex/Term Equivalent
	SFO	Short FOrm
	FFO	Full FOrm
	ELE	term ELEment(s)
	SYM	SYMbolic representation

NOTES ABOUT THE LTU, INTERNAL AND EXTERNAL

LTS	Lexical/Terminological Status	
	PRE	PREferred term
	ADM	ADMitted
	DEP	DEPrecated
	SPS	SuPerSeded
	NEO	NEOlogism
GRM	GRaMmatical reference	
NOT	NOTe [usage, comment, discussion]	
NTY	Note TYpe	
	REM	REMark
	TAB	TABle
	FIG	FIGure
	FOR	FORmula
XRF	X[cross] ReFerence	
XRY	X[cross] Reference tYpe	
	ANT	ANTonym
	SAS	See AlSo

DESCRIPTION OF THE LTU

DES	DEScription	
DTY	Description TYpe	
	DEF	DEFinition
	EXP	EXPlanation
	CTX	ConTeXt
DEM	DEscription Mode	
	GEN	GENeric/Specific
	PAR	PARtitive

RESTRICTIONS IN THE USE OF THE LTU

RSR	ReStrictions	
TYR	TYpe of Restriction	
	GEO	GEOgraphical restriction (country, etc.)
	INH	IN-House usage
	BEN	BENch-level terminology
	TRD	TRaDe name/TRaDemark

ADMINISTRATIVE DATA

SRC	SouRCe text/document (in which LTU is used)	
DAT	DATe	
RES	RESponsible person/group	
RST	Record STatus	
	STR	STaRting record
	WOR	WORking term record
	CON	CONsolidated record
FLD	subject matter FieLD	
FTY	Field TYpe	
	CLS	Classification system (e.g. UDC, Dewey Decimal)
	THS	THesaurus descriptor
	IND	INDexing term
	KYW	KeYWord (other)

One common value for a CLS field is a UDC code. Here are some common UDC codes:

Sample Universal Decimal Classification (UDC) codes

Information Sciences 007

Organizations; congresses 06

Journalism (newspapers) 07

Philosophy 1

Logic 16

Psychology 159.9

Religion 2

Arts 7

Architecture 72

Plastic arts 73

Drawing; painting; graphic arts 74/76

Photography; cinema 77

Music 78

Theatre; dance 79

Linguistics 80

Literature 82

History 93/99

Archaeology 90

Social sciences (general) 3

Sociology 30

Politics 32

Economics 33

Labour 331

Finance 336

Trade 339

Insurance 368

Social welfare 362

Leisure 379.8

Sports 796/799

Law 34

International law 341

Public law 342

  - Criminal law 343
  - Civil law 347
- Public administration 35
  - Public health 614
- Education 37
- Ethnology; ethnography 39
- Natural sciences (general) 5
- Environmental protection 502
- Mathematics 51
- Astronomy 52
- Physics 53
- Chemistry 54
  - Analytical chemistry 543
  - Inorganic chemistry 546
  - Organic chemistry 547
  - Biochemistry 577
- Geosciences (general)
  - Geology 55
  - Geophysics 550
  - Hydrology 556
  - Geography 91
- Paleontology 56
- Biology 57
  - Genetics 575
  - Microbiology 579
- Botany 58
- Zoology 59
- Medicine 61
  - Anatomy 611
  - Physiology 612
  - Hygiene 613
  - Pathology 616
  - Veterinary medicine 619
- Pharmacology 615
  - Pharmaceutics 615.1
- Engineering (general) 62
- Materials testing 620.1
- Energy 620.9
  - Nuclear engineering 621.039
- Mechanical engineering 621
  - Automotive engineering 621.43; 629.113
  - Hydraulics 621.22
  - Pneumatics; refrigeration 621.5
  - Fluids; pumps 621.6
  - Welding 621.791
  - Packing 621.798
  - Power transmission 621.8
  - Machine tools 621.9

- Electrical engineering 621.3
  - Electronics 621.38
  - Telecommunications 621.39; 654
- Mining 622
- Military 623; 355/359
- Civil engineering 624
  - Building construction 69
- Agriculture 63
  - Forestry 630; Plant cultivation 631/635
  - Animal husbandry; hunting; fishing 636/639
- Home economics 64
  - Cooking 641
  - Dwelling 643
  - Installations 644
  - Laundry 648
- Management; business administration 65
  - Advertising 659
  - Accountancy 657
- Printing and publishing 655
- Transport 656
  - Transport vehicles 629
- Industrial products (general)
  - Chemical products 661
  - Explosives; fuels 662
  - Beverage industry 663
  - Food industry 664
  - Oils and fats 665
  - Glass and ceramics 666.1/.7
  - Concrete and cement industry 666.9
  - Paints. Varnishes 667.6
  - Metallurgy 669
  - Timber and wood 674
  - Leather industry 675
  - Paper industry 676
  - Textile industry 677; Clothing industry 687
  - Instruments 681.1/.2
  - Data processing 681.3
  - Automatic control 681.5
  - Optics 535; 681.7
  - Acoustics 534; 681.8
    - (Musical instruments 681.81/.83)
    - (Sound recording and reproduction 681.84)
  - Smithery 682
  - Ironmongery 683
  - Sports and camping equipment 685
  - Toys 688

One important set of Data Categories not yet mentioned is the concept position Data Categories. These categories are used to represent the position a concept occupies in a hierarchal system of concepts. Tentatively, these will all begin with an equals sign or a hyphen in the first position (= for a human-entered link, hyphen for a computer-generated link), a direction letter in the second position (U=Up to superordinate, D=Down to subordinate, C=Coordinate concept at same level, B=Broader term up two or more levels), and a link type in the third position (G=Generic/Specific, P=Part/Whole). The description of a network of concepts in a domain using these Data Categories deserves a separate document.

APPENDIX C — SOME COMMON SGML CHARACTER CODES

&aacute;	Small a with acute accent
&Aacute;	Capital A with acute accent
&abreve;	Small a with breve accent
&Abreve;	Capital A with breve accent
&acirc;	Small a with circumflex accent
&Acirc;	Capital A with circumflex accent
&aelig;	Small ae dipthong (ligature)
&AElig;	Capital AE dipthong (ligature)
&agrave;	Small a with grave accent
&Agrave;	Capital A with grave accent
&amacr;	Small a with macron accent
&Amacr;	Capital A with macron accent
&aring;	Small a with ring accent
&Aring;	Capital A with ring accent (Angstrom)
&atilde;	Small a with tilde accent
&Atilde;	Capital A with tilde accent
&auml;	Small a with umlaut (diaeresis) accent
&Auml;	Capital A with umlaut (diaeresis) accent
&ccaron;	Small c with caron accent
&Ccaron;	Capital C with caron accent
&ccedil;	Small c with cedilla accent
&Ccedil;	Capital C with cedilla accent
&ccirc;	Small c with circumflex accent
&Ccirc;	Capital C with circumflex accent
&dcaron;	Small d with caron accent
&Dcaron;	Capital D with caron accent
&eth;	Small eth (Icelandic character)
&eacute;	Small e with acute accent
&Eacute;	Capital E with acute accent
&ecaron;	Small e with caron accent
&Ecaron;	Capital E with caron accent
&ecirc;	Small e with circumflex accent
&Ecirc;	Capital E with circumflex accent
&egrave;	Small e with grave accent
&Egrave;	Capital E with grave accent
&emacr;	Small e with macron accent
&Emacr;	Capital E with macron accent
&euml;	Small e with umlaut (diaeresis) accent
&Euml;	Capital E with umlaut (diaeresis) accent
&hcirc;	Small h with circumflex accent
&Hcirc;	Capital H with circumflex accent
&iacute;	Small i with acute accent
&Iacute;	Capital I with acute accent

&icirc;	Small i with circumflex accent
&Icirc;	Capital I with circumlex accent
&igrave;	Small i with grave accent
&Igrave;	Capital I with grave accent
&imacr;	Small i with macron accent
&Imacr;	Capital I with macron accent
&inodot;	Small i without dot
&itilde;	Small i with tilde accent
&Itilde;	Capital I with tilde accent
&iuml;	Small i with umlaut (diaeresis) accent
&Iuml;	Capital I with umlaut (diaeresis) accent
&jcirc;	Small j with circumflex accent
&Jcirc;	Capital J with circumflex accent
&lacute;	Small l with acute accent
&Lacute;	Capital L with acute accent
&lcaron;	Small l with caron accent
&Lcaron;	Capital L with caron accent
&nacute;	Small n with acute accent
&Nacute;	Capital N with acute accent
&ncaron;	Small n with caron accent
&Ncaron;	Capital N with caron accent
&ntilde;	Small n with tilde accent
&Ntilde;	Capital N with tilde accent
&oacute;	Small o with acute accent
&Oacute;	Capital O with acute accent
&ocirc;	Small o with circumflex accent
&Ocirc;	Capital O with circumflex accent
&oelig;	Small oe ligature
&OElig;	Capital OE ligature
&ograve;	Small o with grave accent
&Ograve;	Capital O with grave accent
&omacr;	Small o with macron accent
&Omacr;	Capital O with macron accent
&oslash;	Small slashed o
&Oslash;	Capital slashed O
&otilde;	Small o with tilde accent
&Otilde;	Capital O with tilde accent
&ouml;	Small o with umlaut (diaeresis) accent
&Ouml;	Capital O with umlaut (diaeresis) accent
&racute;	Small r with acute accent
&Racute;	Capital R with acute accent
&rcaron;	Small r with caron accent
&Rcaron;	Capital R with caron accent
&sacute;	Small s with acute accent
&Sacute;	Capital S with acute accent
&scaron;	Small s with caron accent
&Scaron;	Capital S with caron accent
&scirc;	Small s with circumflex accent
&Scirc;	Capital S with circumflex accent
&szlig;	German sz ligature (sharp s)

&tcaron;	Small t with caron accent
&Tcaron;	Capital T with caron accent
&thorn;	Small thorn (Icelandic character)
&THORN;	Capital thorn (Icelandic character)
&uacute;	Small u with acute accent
&Uacute;	Capital U with acute accent
&ubreve;	Small u with breve accent
&Ubreve;	Capital U with breve accent
&ucirc;	Small u with circumflex accent
&Ucirc;	Capital U with circumflex accent
&ugrave;	Small u with grave accent
&Ugrave;	Capital U with grave accent
&umacr;	Small u with macron accent
&Umacr;	Capital U with macron accent
&utilde;	Small u with tilde accent
&Utilde;	Capital U with tilde accent
&uuml;	Small u with umlaut (diaeresis) accent
&Uuml;	Capital U with umlaut (diaeresis) accent
&yacute;	Small y with acute accent
&Yacute;	Capital Y with acute accent
&ycirc;	Small y with circumflex accent
&Ycirc;	Capital Y with circumflex accent
&yuml;	Small y with umlaut (diaeresis) accent
&Yuml;	Capital Y with umlaut (diaeresis) accent
&zacute;	Small z with acute accent
&Zacute;	Capital Z with acute accent
&zcaron;	Small z with caron accent
&Zcaron;	Capital Z with caron accent

&acute;	Acute accent
&breve;	Breve
&caron;	Caron
&cedil;	Cedilla
&circ;	Circumflex accent
&die;	Diaeresis
&grave;	Grave accent
&macr;	Macron
&tilde;	Tilde
&uml;	Umlaut mark

&alpha;	Small alpha, Greek
&Agr;	A, Greek capital alpha
&beta;	Small beta, Greek
&Bgr;	B, Greek capital beta
&gamma;	Small gamma, Greek
&Gamma;	Capital gamma, Greek
&delta;	Small delta, Greek
&Delta;	Capital delta, Greek
&epsi;	Small epsilon, Greek
&epsiv;	Small epsilon variant, Greek
&Egr;	E, Greek capital epsilon

&zeta;	Small zeta, Greek
&Zgr;	Z, Greek capital zeta
&half;	Fraction, one-half
&frac12;	Fraction, one-half, alternative notation
&frac14;	Fraction, one-quarter
&frac34;	Fraction, three-quarters
&frac18;	Fraction, one-eigth
&frac38;	Fraction, three-eigths
&hyphen;	Hyphen
&minus;	Minus sign
&plus;	Plus sign
&plusmn;	Plus-or-minus sign
&copy;	Copyright sign
&reg;	Registered sign
&trade;	Trademark symbol
&laquo;	Left angle quotation mark (French opening guillemet)
&raquo;	Right angle quotation mark (French closing guillemet)
&equals;	Equals sign
&sube;	Subset of, or equal to, sign
&supe;	Superset of, or equal to, sign
&lt;	Less-than sign
&gt;	Greater-than sign
&le;	Less-than-or-equals sign
&ge;	Greater-than-or-equals sign
&infin;	Infinity sign
&forall;	For all sign
&exist;	At least one exists sign
&bottom;	Perpendicular sign
&sum;	Summation operator