MIE 1.1

Meta Information Encapsulation

File Format Specification

2007-01-21

(Rev. 6)

Table of Contents

What is MIE?

MIE stands for "Meta Information Encapsulation". The MIE format is an extensible, dedicated meta information format which supports storage of binary as well as textual meta information. MIE can be used to encapsulate meta information from many sources and bundle it together with any type of file.

Features

Below is very subjective score card comparing the features of a number of common file and meta information formats, and comparing them to MIE. The following features are rated for each format with a score of 0 to 10:

```
 1) Extensible (can incorporate user-defined information).
 2) Meaningful tag ID's (hint to meaning of unknown information).
 3) Sequential read/write ability (streamable).
 4) Hierarchical information structure.
 5) Easy to implement reader/writer/editor.
 6) Order of information well defined.
 7) Large data lengths supported: >64kB (+5) and >4GB (+5).
 8) Localized text strings.
 9) Multiple documents in a single file.
10) Compact format doesn't squander disk space or bandwidth.
11) Compressed meta information supported.
12) Relocatable data elements (ie. no fixed offsets).
13) Binary meta information (+7) with variable byte order (+3).
14) Mandatory tags not required (an unecessary complication).
15) Append information to end of file without editing.
```

	Feature number															Total
Format	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	Score
MIE	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	150
PDF	10	10	0	10	0	0	10	0	10	10	10	0	7	10	10	97
PNG	10	10	10	0	8	0	5	10	0	10	10	10	0	10	0	93
XMP	10	10	10	10	2	0	10	10	10	0	0	10	0	10	0	92
AIFF	0	5	10	10	10	0	5	0	0	10	0	10	7	10	0	77
RIFF	0	5	10	10	10	0	5	0	0	10	0	10	7	10	0	77
JPEG	10	0	10	0	10	0	0	0	0	10	0	10	7	10	0	67
EPS	10	10	10	0	0	0	10	0	10	0	0	5	0	10	0	65
CIFF	0	0	0	10	10	0	5	0	0	10	0	10	10	10	0	65
TIFF	0	0	0	10	5	10	5	0	10	10	0	0	10	0	0	60
EXIF	0	0	0	10	5	10	0	0	0	10	0	0	10	0	0	45
IPTC	0	0	10	0	8	0	0	0	0	10	0	10	7	0	0	45

By design, MIE ranks highest by a significant margin. Other formats with reasonable scores are PDF, PNG and XMP, but each has significant weak points. What may be surprising is that TIFF, EXIF and IPTC rank so low.

As well as scoring high in all these features, the MIE format has the unique ability to encapsulate any other type of file, and provides a non-invasive method of adding meta information to a file. The meta information is logically separated from the original file data, which is extremely important because meta information is routinely lost when files are edited.

Also, the MIE format supports multiple files by simple concatenation, enabling all kinds of wonderful features such as linear databases, edit histories or non-intrusive file updates. This ability can also be leveraged to allow MIE-format trailers to be added to some other file types.

MIE 1.1 Format Specification (2007-01-21)

File Structure

A MIE file consists of a series of MIE elements. A MIE element may contain either data or a group of MIE elements, providing a hierarchical format for storing data. Each MIE element is identified by a human-readable tag name, and may store data from zero to 2^64-1 bytes in length.

File Signature

The first element in the MIE file must be an uncompressed MIE group element with a tag name of "0MIE". This restriction allows the first 8 bytes of a MIE file to be used to identify a MIE format file. The following table lists the two possible initial byte sequences for a MIE-format file (the first for big-endian, and the second for little-endian byte ordering):

Byte Number:	0	1	2	3	4	5	6	7
C Characters:	~	\x10	\x04	?	0	M	I	E
or	~	\x18	\x04	?	0	M	I	E
Hexadecimal:	7e	10	04	?	30	4d	49	45
or	7e	18	04	?	30	4d	49	45
Decimal:	126	16	4	?	48	77	73	69
or	126	24	4	?	48	77	73	69

Note that byte 1 may have one of the two possible values (0x10 or 0x18), and byte 3 may have any value (0x00 to 0xff).

Element Structure

Offset	Bytes	Description
0	1	SyncByte = 0x7e (decimal 126, character '~')
1	1	FormatCode (see below)
2	1	TagLength (T)
3	1	DataLength (gives D if DataLength < 253)
4	T	TagName (T given by TagLength)
4+T	2	DataLength2 [exists only if DataLength == 255 (0xff)]
4+T	4	DataLength4 [exists only if DataLength == 254 (0xfe)]
4+T	8	DataLength8 [exists only if DataLength == 253 (0xfd)]
4+T+X	D	DataBlock (D given by DataLength; X is 0, 2, 4 or 8)

The minimum element length is 4 bytes (for a group terminator). The maximum DataBlock size is 2^64-1 bytes. TagLength and DataLength are unsigned integers, and the byte ordering for multi-byte DataLength fields is specified by the containing MIE group element. The SyncByte is byte aligned, so no padding is added to align on an N-byte boundary.

FormatCode

The format code is a bitmask that defines the format of the data:

```
7654 3210
++++ ----  FormatType
---- +---  TypeModifier
---- -+--  Compressed
---- --++  FormatSize
```

FormatType (bitmask 0xf0):

```
0x00 - other (or unknown) data
0x10 - MIE group
0x20 - text string
0x30 - list of null-separated text strings
0x40 - integer
0x50 - rational
0x60 - fixed point
0x70 - floating point
0x80 - free space
```

TypeModifier (bitmask 0x08):

Modifies the meaning of certain FormatTypes (0x00-0x60):

```
0x08 - other data sensitive to MIE group byte order
0x18 - MIE group with little-endian byte ordering
0x28 - UTF encoded text string
0x38 - UTF encoded text string list
0x48 - signed integer
0x58 - signed rational (denominator is always unsigned)
0x68 - signed fixed-point
```

Compressed (bitmask 0x04):

If this bit is set, the data block is compressed using Zlib deflate. An entire MIE group may be compressed, with the exception of file-level groups.

FormatSize (bitmask 0x03):

Gives the byte size of each data element:

```
0x00 - 8 bits  (1 byte)
0x01 - 16 bits (2 bytes)
0x02 - 32 bits (4 bytes)
0x03 - 64 bits (8 bytes)
```

The number of bytes in a single value for this format is given by 2**FormatSize (or 1 << FormatSize). The number of values is the data length divided by this number of bytes. It is an error if the data length is not an even multiple of the format size in bytes.

The following is a list of all currently defined MIE FormatCode values for uncompressed data (add 0x04 to each value for compressed data):

```
0x00 - other data (insensitive to MIE group byte order) (1)
0x01 - other 16-bit data (may be byte swapped)
0x02 - other 32-bit data (may be byte swapped)
0x03 - other 64-bit data (may be byte swapped)
0x08 - other data (sensitive to MIE group byte order) (1)
0x10 - MIE group with big-endian values (1)
0x18 - MIE group with little-endian values (1)
0x20 - ASCII (ISO 8859-1) string (2,3)
0x28 - UTF-8 string (2,3,4)
0x29 - UTF-16 string (2,3,4)
0x2a - UTF-32 string (2,3,4)
0x30 - ASCII (ISO 8859-1) string list (3,5)
0x38 - UTF-8 string list (3,4,5)
0x39 - UTF-16 string list (3,4,5)
0x3a - UTF-32 string list (3,4,5)
0x40 - unsigned 8-bit integer
0x41 - unsigned 16-bit integer
0x42 - unsigned 32-bit integer
0x43 - unsigned 64-bit integer (6)
0x48 - signed 8-bit integer
0x49 - signed 16-bit integer
0x4a - signed 32-bit integer
0x4b - signed 64-bit integer (6)
0x52 - unsigned 32-bit rational (16-bit numerator then denominator) (7)
0x53 - unsigned 64-bit rational (32-bit numerator then denominator) (7)
0x5a - signed 32-bit rational (denominator is unsigned) (7)
0x5b - signed 64-bit rational (denominator is unsigned) (7)
0x61 - unsigned 16-bit fixed-point (high 8 bits is integer part) (8)
0x62 - unsigned 32-bit fixed-point (high 16 bits is integer part) (8)
0x69 - signed 16-bit fixed-point (high 8 bits is signed integer) (8)
0x6a - signed 32-bit fixed-point (high 16 bits is signed integer) (8)
0x72 - 32-bit IEEE float (not recommended for portability reasons)
0x73 - 64-bit IEEE double (not recommended for portability reasons) (6)
0x80 - free space (value data does not contain useful information)
```

Notes:

1. The byte ordering specified by the MIE group TypeModifier applies to the MIE group element as well as all elements within the group. Data for all FormatCodes except 0x08 (other data, sensitive to byte order) may be transferred between MIE groups with different byte order by byte swapping the uncompressed data according to the specified data format. The following list illustrates the byte-swapping pattern, based on FormatSize, for all format types except rational (FormatType 0x50).

FormatSize	Change in Byte Sequence	
0x00 (8 bits)	0 1 2 3 4 5 6 7 --> 0 1 2 3 4 5 6 7	(no change)
0x01 (16 bits)	0 1 2 3 4 5 6 7 --> 1 0 3 2 5 4 7 6	
0x02 (32 bits)	0 1 2 3 4 5 6 7 --> 3 2 1 0 7 6 5 4	
0x03 (64 bits)	0 1 2 3 4 5 6 7 --> 7 6 5 4 3 2 1 0	

Rational values consist of two integers, so they are swapped as the next lower FormatSize. For example, a 32-bit rational (FormatSize 0x02, and FormatCode 0x52 or 0x5a) is swapped as two 16-bit values (ie. as if it had FormatSize 0x01).

2. The TagName of a string element may have an 6-character suffix to indicate a specific locale. (ie. “Title-en_US”, or “Keywords-de_DE”).
3. Text strings are not normally null terminated, however they may be padded with one or more null characters to the end of the data block to allow strings to be edited within fixed-length data blocks. Newlines in the text are indicated by a single LF (0x0a) character.
4. UTF strings must not begin with a byte order mark (BOM) since the byte order and byte size are specified by the MIE format. If a BOM is found, it should be treated as a zero-width non-breaking space.
5. A list of text strings separated by null characters. These lists must not be null padded or null terminated, since this would be interpreted as additional zero-length strings. For ASCII and UTF-8 strings, the null character is a single zero (0x00) byte. For UTF-16 or UTF-32 strings, the null character is 2 or 4 zero bytes respectively.
6. 64-bit integers and doubles are subject to the specified byte ordering for both 32-bit words and bytes within these words. For instance, the high order byte is always the first byte if big-endian, and the eighth byte if little-endian. This means that some swapping is always necessary for these values on systems where the byte order differs from the word order (ie. some ARM systems), regardless of the endian-ness of the stored values.
7. Rational values are treated as two separate integers. The numerator always comes first regardless of the byte ordering. In a signed rational value, only the numerator is signed. The denominator of all rational values is unsigned (ie. a signed 64-bit rational of 0x80000000/0x80000000 evaluates to -1, not +1).
8. 32-bit fixed point values are converted to floating point by treating them as an integer and dividing by an appropriate value. ie)

```
16-bit fixed value = 16-bit integer value / 256.0
32-bit fixed value = 32-bit integer value / 65536.0
```

TagLength

Gives the length of the TagName string. Any value between 0 and 255 is valid, but the TagLength of 0 is valid only for the MIE group terminator.

DataLength

DataLength is an unsigned byte that gives the number of bytes in the data block. A value between 0 and 252 gives the data length directly, and numbers from 253 to 255 are reserved for extended DataLength codes. Codes of 255, 254 and 253 indicate that the element contains an additional 2, 4 or 8 byte unsigned integer representing the data length.

```
0-252      - length of data block
255 (0xff) - use DataLength2
254 (0xfe) - use DataLength4
253 (0xfd) - use DataLength8
```

A DataLength of zero is valid for any element except a compressed MIE group. A zero DataLength for an uncompressed MIE group indicates that the group length is unknown. For other elements, a zero length indicates there is no associated data. A terminator element must have a DataLength of 0, 6 or 10, and may not use an extended DataLength.

TagName

The TagName string is 0 to 255 bytes long, and is composed of the ASCII characters A-Z, a-z, 0-9 and underline ('_'). Also, a dash ('-') is used to separate the language/country code in the TagName of a localized text string, and a units string (possibly containing other ASCII characters) may be appear in brackets at the end of the TagName (see Appendix A). The TagName string is NOT null terminated. A MIE element with a tag string of zero length is reserved for the group terminator.

MIE elements are sorted alphabetically by TagName within each group. Multiple elements with the same TagName are allowed, even within the same group.

TagNames should be meaningful. Case is significant. Words should be lowercase with an uppercase first character, and acronyms should be all upper case. The underline ("_") is provided to allow separation of two acronyms or two numbers, but it shouldn't be used unless necessary. No separation is necessary between an acronym and a word (ie. "ISOSetting").

All TagNames should start with an uppercase letter. An exception to this rule allows tags to begin with a digit (0-9) if they must come before other tags in the sort order, or a lowercase letter (a-z) if they must come after. For instance, the "0Type" element begins with a digit so it comes before, and the "data" element begins with a lowercase letter so that it comes after meta information tags in the main "0MIE" group.

Tag names for localized text strings have an 6-character suffix with the following format: The first character is a dash ('-'), followed by a 2-character lower case ISO 639-1 language code, then an underline ('_'), and ending with a 2-character upper case ISO 3166-1 alpha 2 country code. (ie. "-en_US", "-en_GB", "-de_DE" or "-fr_FR". Note that "GB", and not "UK" is the code for Great Britain, although "UK" should be recognized for compatibility reasons.) The suffix is included when sorting the tags alphabetically, so the default locale (with no tag-name suffix) always comes first. If the country is unknown or not applicable, a country code of "XX" should be used.

Tags with numerical values may allow units of measurement to be specified. The units string is stored in brackets at the end of the tag name, and is composed of zero or more ASCII characters in the range 0x21 to 0x7d, excluding the bracket characters 0x28 and 0x29. (ie. "Resolution(/cm)" or "SpecificHeat(J/kg.K)".) See the Image::ExifTool::MIEUnits manpage for details. Unit strings are not localized, and may not be used in combination with localized text strings. See Appendix A for a description of MIE Units of Measure.

Sets of tags which would require a common prefix should be added in a separate MIE group instead of adding the prefix to all tag names. For example, instead of these TagName's:

```
ExternalFlashType
ExternalFlashSerialNumber
ExternalFlashFired
```

one would instead designate a separate “ExternalFlash” MIE group to contain the following elements:

```
Type
SerialNumber
Fired
```

DataLength2/4/8

These extended DataLength fields exist only if DataLength is 255, 254 or 253, and are respectively 2, 4 or 8 byte unsigned integers giving the data block length. One of these values must be used if the data block is larger than 252 bytes, but they may be used if desired for smaller blocks too (although this may add a few unnecessary bytes to the MIE element).

DataBlock

The data value for the MIE element. The format of the data is given by the FormatCode. For MIE group elements, the data includes all contained elements and the group terminator.

MIE groups

All MIE data elements must be contained within a group. A group begins with a MIE group element, and ends with a group terminator. Groups may be nested in a hierarchy to arbitrary depth.

A MIE group element is identified by a format code of 0x10 (big endian byte ordering) or 0x18 (little endian). The group terminator is distinguished by a zero TagLength (it is the only element allowed to have a zero TagLength), and has a FormatCode of 0x00.

The MIE group element is permitted to have a zero DataLength only if the data is uncompressed. This special value indicates that the group length is unknown (otherwise the minimum value for DataLength is 4, corresponding the the minimum group size which includes a terminator of at least 4 bytes). If DataLength is zero, all elements in the group must be parsed until the group terminator is found. If non-zero, DataLength includes the length of all elements contained within the group, including the group terminator. Use of a non-zero DataLength is encouraged because it allows readers quickly skip over entire MIE groups. For compressed groups DataLength must be non-zero, and is the length of the compressed group data (which includes the compressed group terminator).

Group Terminator

The group terminator has a FormatCode and TagLength of zero. The terminator DataLength must be 0, 6 or 10 bytes, and extended DataLength codes may not be used. With a zero

DataLength, the byte sequence for a terminator is “7e 00 00 00” (hex). With a DataLength of 6 or 10 bytes, the terminator data block contains information about the length and byte ordering of the preceding group. This additional information is recommended for file-level groups, and is used in multi-document MIE files and MIE trailers to allow the file to be scanned backwards from the end. (This may also allow some documents to be recovered if part of the file is corrupted.) The structure of this optional terminator data block is as follows:

```
4 or 8 bytes  GroupLength (unsigned integer)
1 byte        ByteOrder (0x10 or 0x18, same as MIE group)
1 byte        GroupLengthSize (0x04 or 0x08)
```

The ByteOrder and GroupLengthSize values give the byte ordering and size of the GroupLength integer. The GroupLength value is the total length of the entire MIE group ending with this terminator, including the opening MIE group element and the terminator itself.

File-level MIE groups

File-level MIE groups may NOT be compressed.

All elements in a MIE file are contained within a special group with a TagName of “0MIE”. The purpose of the “OMIE” group is to provide a unique signature at the start of the file, and to encapsulate information allowing files to be easily combined. The “0MIE” group must be terminated like any other group, but it is recommended that the terminator of a file-level group include the optional data block (defined above) to provide information about the group length and byte order.

It is valid to have more than one “0MIE” group at the file level, allowing multiple documents in a single MIE file. Furthermore, the MIE structure enables multi-document files to be generated by simply concatenating two or more MIE files.

Scanning Backwards through a MIE File

The steps below give an algorithm to quickly locate the last document in a MIE file:

1. Read the last 10 bytes of the file. (Note that a valid MIE file may be as short as 12 bytes long, but a file this length contains only an an empty MIE group.)
2. If the last byte of the file is zero, then it is not possible to scan backward through the file, so the file must be scanned from the beginning. Otherwise, proceed to the next step.
3. If the last byte is 4 or 8, the terminator contains information about the byte ordering and length of the group. Otherwise, stop here because this isn't a valid MIE file.
4. The next-to-last byte must be either 0x10 indicating big-endian byte ordering or 0x18 for little-endian ordering, otherwise this isn't a valid MIE file.
5. The value of the preceding 4 or 8 bytes gives the length of the complete file-level MIE group (GroupLength). This length includes both the leading MIE group element and the terminator element itself. The value is an unsigned integer with a byte length given in step 3, and a byte order from step 4. From the current file position (at the end of the data

read in step 1, seek backward by this number of bytes to find the start of the MIE group element for this document.

This algorithm may be repeated again beginning at this point in the file to locate the next-to-last document, etc.

The table below lists all 5 valid patterns for the last 14 bytes of a file-level MIE group, with all numbers in hex. The comments indicate the length and byte ordering of GroupLength (xx) if available:

```
?? ?? ?? ?? ?? ?? ?? ?? ?? ?? 7e 00 00 00  - (no GroupLength)
?? ?? ?? ?? 7e 00 00 06 xx xx xx xx 10 04  - 4 bytes, big endian
?? ?? ?? ?? 7e 00 00 06 xx xx xx xx 18 04  - 4 bytes, little endian
7e 00 00 0a xx xx xx xx xx xx xx xx 10 08  - 8 bytes, big endian
7e 00 00 0a xx xx xx xx xx xx xx xx 18 08  - 8 bytes, little endian
```

Trailer Signature

The MIE format may be used for trailer information appended to other types of files. When this is done, a signature must appear at the end of the main MIE group to uniquely identify the MIE format trailer. To achieve this, a “zmie” trailer signature is written as the last element in the main “0MIE” group. This element has a FormatCode of 0, a TagLength of 4, a DataLength of 0, and a TagName of “zmie”. With this signature, the hex byte sequence “7e 00 04 00 7a 6d 69 65” appears immediately before the final group terminator, and the last 22 bytes of the trailer correspond to one of the following 4 patterns (where the trailer length is given by “xx”, as above):

```
?? ?? ?? ?? 7e 00 04 00 7a 6d 69 65 7e 00 00 06 xx xx xx xx 10 04
?? ?? ?? ?? 7e 00 04 00 7a 6d 69 65 7e 00 00 06 xx xx xx xx 18 04
7e 00 04 00 7a 6d 69 65 7e 00 00 0a xx xx xx xx xx xx xx xx 10 08
7e 00 04 00 7a 6d 69 65 7e 00 00 0a xx xx xx xx xx xx xx xx 18 08
```

Note that the zero-DataLength terminator may not be used here because the trailer length must be known for seeking backwards from the end of the file.

Multiple trailers may be appended to the same file using this technique.

MIE Data Values

MIE data values for a given tag are usually not restricted to a specific FormatCode. Any value may be represented in any appropriate format, including numbers represented in string (ASCII or UTF) form.

It is preferred that closely related values with the same format are written to a single tag instead of using multiple tags. This improves localization of like values and decreases MIE element overhead. For instance, instead of separate ImageWidth and ImageHeight tags, a single ImageSize tag is defined.

Tags which may take on a discrete set of values should have meaningful values if possible. This improves the extensibility of the format and allows a more reasonable interpretation of unrecognized values.

Numerical Representation

Integer and floating point numbers may be represented in binary or string form. In string form, integers are a series of digits with an optional leading sign (ie. "[+|-]DDDDDD"), and multiple values are separated by a single space character (ie. "23 128 -32"). Floating point numbers are similar but may also contain a decimal point and/or a signed exponent with a leading 'e' character (ie. "[+|-]DD[.DDDDDD][e(+|-)EEE]"). The string "inf" is used to represent infinity. One advantage of numerical strings is that they can have an arbitrarily high precision because the possible number of significant digits is virtually unlimited.

Note that numerical values may have associated units of measurement which are specified in the TagName string (see Appendix A).

Date/Time Format

All MIE dates are strings in the form "YYYY:mm:dd HH:MM:SS.ss+HH:MM". The fractional seconds (".ss") are optional, and if included may contain any number of significant digits (unlike all other fields which are a fixed number of digits and must be padded with leading zeros if necessary). The timezone ("+HH:MM" or "-HH:MM") is recommended but not required. If not given, the local system timezone is assumed.

MIME Type

The basic MIME type for a MIE file is "application/x-mie", however the specific MIME type depends on the type of subfile, and is obtained by adding "x-mie-" to the MIME type of the subfile. For example, with a subfile of type "image/jpeg", the MIE file MIME type is "image/x-mie-jpeg". But note that the "x-" is not duplicated if the subfile MIME type already starts with "x-". So a subfile with MIME type "image/x-raw" is contained within a MIE file of type "image/x-mie-raw", not "image/x-mie-x-raw". In the case of multiple documents in a MIE file, the MIME type is taken from the first document. Regardless of the subfile type, all MIE-format files should have a filename extension of ".MIE".

Levels of Support

Basic MIE reader/writer applications may choose not to provide support for some advanced features of the MIE format. Features which may not be supported by all software are:

Compression

Software not supporting compression must ignore compressed elements and groups, but should be able to process the remaining information.

Large data lengths

Some software may limit the maximum size of a MIE group or element. Historically, a limit of 2GB may be imposed by some systems. However, 8-byte data lengths should be supported by all applications provided the value doesn't exceed the system limit. (ie. For systems with a 2GB limit, 8-byte data lengths should be supported if the upper 17 bits are all zero.) If a data length above the system limit is encountered, it may be necessary for the application to stop processing if it can not seek to the next element in the file.

Revisions

```
2012-08-21 – Added “Offset” column to Element Structure table
2010-04-05 - Fixed “Format Size” Note 7 to give the correct number of bits
             in the example rational value
2007-01-21 - Specified LF character (0x0a) for text newline sequence
2007-01-19 - Specified ISO 8859-1 character set for extended ASCII codes
2007-01-01 - Improved wording of Step 5 for scanning backwards in MIE file
2006-12-30 - Added EXAMPLES section and note about UTF BOM
2006-12-20 - MIE 1.1:  Changed meaning of TypeModifier bit (0x08) for
             unknown data (FormatType 0x00), and documented byte swapping
2006-12-14 - MIE 1.0:  Added Data Values and Numerical Representations
             sections, and added ability to specify units in tag names
2006-11-09 - Added Levels of Support section
2006-11-03 - Added Trailer Signature
2005-11-18 - Original specification created
```

License

Appendix A – Units of Measure

Description

The MIE format allows units of measurement to be specified in brackets at the end of a MIE tag name (ie. “Volume(m3)”). This appendix describes the standard MIE units abbreviations.

Syntax

The units string may contain any ASCII characters in the range 0x21 ('!') to 0x7d ('}'), excepting the bracket characters (0x28 and 0x29). An empty string is allowed, and indicates a dimensionless value. Standard units should be used where possible. In the standard units, an underline ('_') is used to indicate a subscript, and multiple words may be separated with a hyphen ('-').

Exponents should be positive, and require no separator (ie. “m2” for square meters). Prefixes may be added to the standard units (ie. “cm”) except when the resulting name conflicts with another standard unit.

Multiplication is indicated by '.', and division by '/'. Reciprocal units (ie. the multiplicative inverse) are obtained through division rather than the use of negative exponents (ie. “/in”, not “in-1”). In MIE units, multiplication has precedence over division, so everything to the right of a '/' is in the denominator. (See Examples below for a few examples.)

Below is a table summarizing the special characters used in MIE units strings.

```
.           multiplication
/           division (used for negative exponents)
-           word separator
_           subscript
{}          annotation
[]          used to avoid name conflicts if necessary
0-9         exponentiation
```

Standard Units

```
            dimensionless
[G]         Newtonian constant of gravitation (unclassified)
[g]         standard acceleration of free fall (acceleration)
[h]         Planck constant (action)
[k]         Boltzmann constant (unclassified)
{cfu}       colony forming units (number)
{rbc}       red blood cell count (number)
{tbl}       tablets (number)
{tot}       particles total count (number)
%           percent (fraction)
10^N        the number ten for arbitrary powers (number)
A           Ampere (electric current)
a           year (time)
```

```
a_g         mean Gregorian year (time)
a_j         mean Julian year (time)
a_t         tropical year (time)
acr         acre, U.S. (area)
acr_br      acre, British (area)
Ao          Angstrom (length)
APL-U       APL unit (biologic activity of anticardiolipin IgA)
ar          are (area)
arb-U       arbitrary unit (arbitrary)
arcmin      minute of arc (plane angle)
arcsec      second of arc (plane angle)
atm         standard atmosphere (pressure)
att         technical atmosphere (pressure)
AU          astronomic unit (length)
b           barn (action area)
B           bel (level)
B-kW        bel kilowatt (power level)
B-mV        bel millivolt (electric potential level)
B-SPL       bel sound pressure (pressure level)
B-uV        bel microvolt (electric potential level)
B-V         bel volt (electric potential level)
B-W         bel watt (power level)
bar         bar (pressure)
bbl         barrel (fluid volume)
Bd          baud (signal transmission rate)
bdsk-U      Bodansky unit (biologic activity of phosphatase)
beth-U      Bethesda unit (biologic activity of factor VIII
            inhibitor)
bf          board foot (volume)
Bi          Biot (electric current)
bit         bit (amount of information)
Bq          Becquerel (radioactivity)
Btu         British thermal unit (energy)
Btu_39      British thermal unit at 39 degF (energy)
Btu_59      British thermal unit at 59 degF (energy)
Btu_60      British thermal unit at 60 degF (energy)
Btu_IT      international table British thermal unit (energy)
Btu_m       mean British thermal unit (energy)
Btu_th      thermochemical British thermal unit (energy)
bu          bushel, U.S. (dry volume)
bu_br       bushel, British (volume)
By          byte (amount of information)
C           Coulomb (electric charge)
c           velocity of light (velocity)
cal         calorie (energy)
Cal         nutrition label Calories (energy)
cal_15      calorie at 15 degC (energy)
cal_20      calorie at 20 degC (energy)
cal_IT      international table calorie (energy)
cal_m       mean calorie (energy)
cal_th      thermochemical calorie (energy)
car_Au      carat of gold alloys (mass fraction)
car_m       metric carat (mass)
cd          candela (luminous intensity)
```

```
Cel         degree Celsius (temperature)
Ch          Charriere (gauge of catheters)
ch          Gunter's chain, U.S. (length)
ch_br       Gunter's chain, British (length)
Ci          Curie (radioactivity)
cicero      cicero (length)
circ        circle (plane angle)
cml         circular mil, international (area)
cr          cord, international (volume)
crd_us      cord, U.S. (fluid volume)
cup_us      cup (volume)
d           day (time)
deg         degree (plane angle)
deg{mag}    degree from magnetic north (plane angle)
degF        degree Fahrenheit (temperature)
didot       didot (length)
diop        diopter (refraction of a lens)
dpt         dry pint, U.S. (dry volume)
dqt         dry quart, U.S. (dry volume)
dr          dram (mass)
dr_ap       dram, apothecary (mass)
drp         drop (volume)
dye-U       Dye unit (biologic activity of amylase)
dyn         dyne (force)
e           elementary charge (electric charge)
eps_0       permittivity of vacuum (electric permittivity)
eq          equivalents (amount of substance)
erg         erg (energy)
eV          electronvolt (energy)
F           Farad (electric capacitance)
fdr         fluid dram, U.S. (fluid volume)
fdr_br      fluid dram, British (volume)
foz         fluid ounce, U.S. (fluid volume)
foz_br      fluid ounce, British (volume)
ft          foot, international (length)
ft_br       foot, British (length)
ft_us       foot, U.S. (length)
fth         fathom, international (length)
fth_br      fathom, British (length)
fth_us      fathom, U.S. (length)
fur         furlong, U.S. (length)
G           Gauss (magnetic flux density)
g           gram (mass)
g.m/{H-B}   gram meter per heartbeat (prop. to ventricular stroke
            work)
g%          gram percent (mass fraction)
Gal         Gal (acceleration)
gal         gallon, U.S. (fluid volume)
gal_br      gallon, British (volume)
gal_wi      historical winchester gallon (dry volume)
Gb          Gilbert (magnetic tension)
gf          gram-force (force)
gil         gill, U.S. (fluid volume)
gil_br      gill, British (volume)
```

```
gon          gon (plane angle)
GPL-U        GPL unit (biologic activity of anticardiolipin IgG)
gr           grain (mass)
Gy           Gray (energy dose)
H            Henry (inductance)
h            hour (time)
hd           hand, international (height of horses)
hnsf-U       Hounsfield unit (x-ray attenuation)
HP           horsepower (power)
hp_C         homeopathic potency of centesimal series (homeopathic
             potency)
hp_X         homeopathic potency of decimal series (homeopathic
             potency)
HPF          high power field (view area in microscope)
Hz           Herz (frequency)
in           inch, international (length)
in_br        inch, British (length)
in_us        inch, U.S. (length)
in-H2O       inch of water column (pressure)
in-Hg        inch of mercury column (pressure)
iU           international unit (arbitrary)
J            Joule (energy)
K            Kelvin (temperature)
ka-U         King-Armstrong unit (biologic activity of phosphatase)
kat          katal (catalytic activity)
kg{wet-tis}  kilogram of wet tissue (mass)
kn           knot, international (velocity)
kn_br        knot, British (velocity)
knk-U        Kunkel unit (arbitrary biologic activity)
Ky           Kayser (lineic number)
l            liter (volume)
L            liter (volume)
lb           pound (mass)
lb_ap        pound, apothecary (mass)
lb_tr        pound, troy (mass)
lbf          pound force (force)
lcwt         long hunderdweight (mass)
ligne        ligne (length)
lk           link for Gunter's chain, U.S. (length)
lk_br        link for Gunter's chain, British (length)
lm           lumen (luminous flux)
Lmb          Lambert (brightness)
lne          line (length)
LPF          low power field (view area in microscope)
lton         long ton (mass)
lx           lux (illuminance)
ly           light-year (length)
m            meter (length)
m_e          electron mass (mass)
m_p          proton mass (mass)
m-H2O        meter of water column (pressure)
m-Hg         meter of mercury column (pressure)
mclg-U       Mac Lagan unit (arbitrary biologic activity)
mesh         mesh, international (lineic number)
```

MET	metabolic equivalent (metabolic cost of physical activity)
mg{creat}	milligram of creatinine (mass)
mho	mho (electric conductance)
mi	mile, international (statute mile) (length)
mi_br	mile, British (length)
mi_us	mile, U.S. (length)
mil	mil, international (length)
mil_us	mil, U.S. (length)
min	minute (time)
min_br	minim, British (volume)
min_us	minim, U.S. (fluid volume)
mo	month (time)
mo_g	mean Gregorian month (time)
mo_j	mean Julian month (time)
mo_s	synodal month (time)
mol	mole (amount of substance)
MPL-U	MPL unit (biologic activity of anticardiolipin IgM)
mu_0	permeability of vacuum (magnetic permeability)
Mx	Maxwell (flux of magnetic induction)
N	Newton (force)
nmi	nautical mile, international (length)
nmi_br	nautical mile, British (length)
Np	neper (level)
Oe	Oersted (magnetic field intensity)
Ohm	Ohm (electric resistance)
osm	osmole of dissolved particles (amount of substance)
oz	ounce (mass)
oz_ap	ounce, apothecary (mass)
oz_tr	ounce, troy (mass)
P	Poise (dynamic viscosity)
Pa	Pascal (pressure)
pc	parsec (length)
pc_br	pace (length)
pca	pica (length)
pca_pr	Printer's pica (length)
pH	pH (acidity)
ph	phot (illuminance)
pi	the number pi (number)
pied	pied (length)
pk	peck, U.S. (dry volume)
pk_br	peck, British (volume)
pnt	point (length)
pnt_pr	Printer's point (length)
pouce	pouce (length)
ppb	parts per billion (fraction)
ppm	parts per million (fraction)
ppth	parts per thousand (fraction)
pptr	parts per trillion (fraction)
PRU	peripheral vascular resistance unit (fluid resistance)
psi	pound per square inch (pressure)
pt	pint, U.S. (fluid volume)
pt_br	pint, British (volume)
pwt_tr	pennyweight (mass)

```
qt          quart, U.S. (fluid volume)
qt_br       quart, British (volume)
R           Roentgen (ion dose)
rad         radian (plane angle)
RAD         radiation absorbed dose (energy dose)
rch         Ramden's chain, U.S. (length)
rd          rod, U.S. (length)
rd_br       rod, British (length)
REM         radiation equivalent man (dose equivalent)
rlk_us      link for Ramden's chain (length)
s           second (time)
S           Siemens (electric conductance)
sb          stilb (lum. intensity density)
sc_ap       scruple, apothecary (mass)
sct         section (area)
scwt        short hundredweight (mass)
smgy-U      Somogyi unit (biologic activity of amylase)
sph         spere (solid angle)
sr          streadian (solid angle)
st          stere (volume)
St          Stokes (kinematic viscosity)
ston        short ton (mass)
stone       stone (mass)
Sv          Sievert (dose equivalent)
Sv-U        Svedberg unit (sedimentation coefficient)
T           Tesla (magnetic flux density)
t           tonne (mass)
tb-U        tuberculin unit (biologic activity of tuberculin)
tbs         tablespoon, U.S. (volume)
todd-U      Todd unit (biologic activity antistreptolysin O)
tsp         teaspoon, U.S. (volume)
twp         township (area)
u           unified atomic mass unit (mass)
U           Unit (catalytic activity)
USP-U       U.S. Pharmacopeia unit (arbitrary)
V           Volt (electric potential)
W           Watt (power)
Wb          Weber (magnetic flux)
wk          week (time)
yd          yard, international (length)
yd_br       yard, British (length)
yd_us       yard, U.S. (length)
```

Prefixes

Standard SI prefixes:

```
y           yocto (10^-24)
z           zepto (10^-21)
a           atto  (10^-18)
f           femto (10^-15)
p           pico  (10^-12)
n           nano  (10^-9)
```

```
u           micro (10^-6)
m           milli (10^-3)
c           centi (10^-2)
d           deci  (10^-1)
da          deka  (10)
h           hecto (10^2)
k           kilo  (10^3)
G           giga  (10^9)
M           mega  (10^6)
T           tera  (10^12)
P           peta  (10^15)
E           exa   (10^18)
Z           zetta (10^21)
Y           yotta (10^24)
```

Binary power prefixes:

```
Ki          kibi  (2^10)
Mi          mebi  (2^20)
Gi          gibi  (2^30)
Ti          tebi  (2^40)
```

Examples

A few examples of combined units strings:

```
/cm3        per cubic centimeter
in2         square inches
kg.m/s2     Newtons (N)
m3/kg.s2    units of [G]
10^100      googols
```