NAME

eqn - eqn language reference for mandoc

DESCRIPTION

The **eqn** language is an equation-formatting language. It is used within mdoc(7) and man(7) UNIX manual pages. It describes the *structure* of an equation, not its mathematical meaning. This manual describes the **eqn** language accepted by the mandoc(1) utility, which corresponds to the Second Edition **eqn** specification (see *SEE ALSO* for references).

An equation starts with an input line containing exactly the characters ‘.EQ’, may contain multiple input lines, and ends with an input line containing exactly the characters ‘.EN’. Equivalently, an equation can be given in the middle of a single text input line by surrounding it with the equation delimiters defined with the **delim** statement.

The equation grammar is as follows, where quoted strings are case-sensitive literals in the input:

```
eqn     : box | eqn box
box     : text
        | "{" eqn "}"
        | "define" text text
        | "ndefine" text text
        | "tdefine" text text
        | "gfont" text
        | "gsize" text
        | "set" text text
        | "undef" text
        | "sqrt" box
        | box pos box
        | box mark
        | "matrix" "{" [col "{" list "}"]* "}"
        | pile "{" list "}"
        | font box
        | "size" text box
        | "left" text eqn ["right" text]
col     : "lcol" | "rcol" | "ccol" | "col"
text    : [^space\"]+ | \".*\"
pile    : "lpile" | "cpile" | "rpile" | "pile"
pos     : "over" | "sup" | "sub" | "to" | "from"
mark        : "dot" | "dotdot" | "hat" | "tilde" | "vec"
        | "dyad" | "bar" | "under"
```

```
font    : "roman" | "italic" | "bold" | "fat"
list    : eqn
        | list "above" eqn
space   : [\^~ \t]
```

White-space consists of the space, tab, circumflex, and tilde characters. It is required to delimit tokens consisting of alphabetic characters and it is ignored at other places. Braces and quotes also delimit tokens. If within a quoted string, these space characters are retained. Quoted strings are also not scanned for keywords, glyph names, and expansion of definitions. To print a literal quote character, it can be prepended with a backslash or expressed with the \(dq escape sequence.

Subequations can be enclosed in braces to pass them as arguments to operation keywords, overriding standard operation precedence. Braces can be nested. To set a brace verbatim, it needs to be enclosed in quotes.

The following text terms are translated into a rendered glyph, if available: alpha, beta, chi, delta, epsilon, eta, gamma, iota, kappa, lambda, mu, nu, omega, omicron, phi, pi, psi, rho, sigma, tau, theta, upsilon, xi, zeta, DELTA, GAMMA, LAMBDA, OMEGA, PHI, PI, PSI, SIGMA, THETA, UPSILON, XI, inter (intersection), union (union), prod (product), int (integral), sum (summation), grad (gradient), del (vector differential), times (multiply), cdot (center-dot), nothing (zero-width space), approx (approximately equals), prime (prime), half (one-half), partial (partial differential), inf (infinity), >> (much greater), << (much less), <- (left arrow), -> (right arrow), +- (plus-minus), != (not equal), == (equivalence), <= (less-than-equal), and >= (more-than-equal). The character escape sequences documented in mandoc_char(7) can be used, too.

The following control statements are available:

define Replace all occurrences of a key with a value. Its syntax is as follows:

define *key cvalc*

The first character of the value string, *c*, is used as the delimiter for the value *val*. This allows for arbitrary enclosure of terms (not just quotes), such as

define *foo 'bar baz'*
define *foo cbar bazc*

It is an error to have an empty *key* or *val*. Note that a quoted *key* causes errors in some **eqn** implementations and should not be considered portable. It is not expanded for replacements. Definitions may refer to other definitions; these are evaluated recursively when text

replacement occurs and not when the definition is created.

Definitions can create arbitrary strings, for example, the following is a legal construction.

```
define foo 'define'
foo bar 'baz'
```

Self-referencing definitions will raise an error. The **ndefine** statement is a synonym for **define**, while **tdefine** is discarded.

delim This statement takes a string argument consisting of two bytes, to be used as the opening and closing delimiters for equations in the middle of text input lines. Conventionally, the dollar sign is used for both delimiters, as follows:

```
.EQ
delim $$
.EN
An equation like $sin pi = 0$ can now be entered
in the middle of a text input line.
```

The special statement **delim off** temporarily disables previously declared delimiters and **delim on** reenables them.

gfont Set the default font of subsequent output. Its syntax is as follows:

gfont *font*

In mandoc, this value is discarded.

gsize Set the default size of subsequent output. Its syntax is as follows:

gsize [+|-]*size*

The *size* value should be an integer. If prepended by a sign, the font size is changed relative to the current size.

set Set an equation mode. In mandoc, both arguments are thrown away. Its syntax is as follows:

set *key val*

The *key* and *val* are not expanded for replacements. This statement is a GNU extension.

undef Unset a previously-defined key. Its syntax is as follows:

define *key*

Once invoked, the definition for *key* is discarded. The *key* is not expanded for replacements. This statement is a GNU extension.

Operation keywords have the following semantics:

above See **pile**.

bar Draw a line over the preceding box.

bold Set the following box using bold font.

ccol Like **cpile**, but for use in **matrix**.

cpile Like **pile**, but with slightly increased vertical spacing.

dot Set a single dot over the preceding box.

dotdot Set two dots (dieresis) over the preceding box.

dyad Set a dyad symbol (left-right arrow) over the preceding box.

fat A synonym for **bold**.

font Set the second argument using the font specified by the first argument; currently not recognized by the mandoc(1) **eqn** parser.

from Set the following box below the preceding box, using a slightly smaller font. Used for sums, integrals, limits, and the like.

hat Set a hat (circumflex) over the preceding box.

italic Set the following box using italic font.

lcol Like **lpile**, but for use in **matrix**.

left Set the first argument as a big left delimiter before the second argument. As an optional third argument, **right** can follow. In that case, the fourth argument is set as a big right delimiter after the second argument.

lpile Like **cpile**, but subequations are left-justified.

matrix Followed by a list of columns enclosed in braces. All columns need to have the same number of subequations. The columns are set as a matrix. The difference compared to multiple subsequent **pile** operators is that in a **matrix**, corresponding subequations in all columns line up horizontally, while each **pile** does vertical spacing independently.

over Set a fraction. The preceding box is the numerator, the following box is the denominator.

pile Followed by a list of subequations enclosed in braces, the subequations being separated by **above** keywords. Sets the subequations one above the other, each of them centered. Typically used to represent vectors in coordinate representation.

rcol Like **rpile**, but for use in **matrix**.

right See **left**; **right** cannot be used without **left**. To set a big right delimiter without a big left delimiter, the following construction can be used:

left "" *box* **right** *delimiter*

roman Set the following box using the default font.

rpile Like **cpile**, but subequations are right-justified.

size Set the second argument with the font size specified by the first argument; currently ignored by mandoc(1). By prepending a plus or minus sign to the first argument, the font size can be selected relative to the current size.

sqrt Set the square root of the following box.

sub Set the following box as a subscript to the preceding box.

sup Set the following box as a superscript to the preceding box. As a special case, if a **sup** clause immediately follows a **sub** clause as in

mainbox **sub** *subbox* **sup** *supbox*

both are set with respect to the same *mainbox*, that is, *supbox* is set above *subbox*.

tilde Set a tilde over the preceding box.

to Set the following box above the preceding box, using a slightly smaller font. Used for sums and integrals and the like. As a special case, if a **to** clause immediately follows a **from** clause as in

mainbox **from** *frombox* **to** *tobox*

both are set below and above the same *mainbox*.

under Underline the preceding box.

vec Set a vector symbol (right arrow) over the preceding box.

The binary operations **from**, **to**, **sub**, and **sup** group to the right, that is,

mainbox **sup** *supbox* **sub** *subbox*

is the same as

mainbox **sup** {*supbox* **sub** *subbox*}

and different from

{*mainbox* **sup** *supbox*} **sub** *subbox*.

By contrast, **over** groups to the left.

In the following list, earlier operations bind more tightly than later operations:

1. **dyad**, **vec**, **under**, **bar**, **tilde**, **hat**, **dot**, **dotdot**
2. **fat**, **roman**, **italic**, **bold**, **size**
3. **sub**, **sup**
4. **sqrt**
5. **over**
6. **from**, **to**

COMPATIBILITY

This section documents the compatibility of mandoc **eqn** and the troff **eqn** implementation (including GNU troff).

- The text string ‘\"’ is interpreted as a literal quote in troff. In mandoc, this is interpreted as a comment.
- In troff, The circumflex and tilde white-space symbols map to fixed-width spaces. In mandoc, these characters are synonyms for the space character.
- The troff implementation of **eqn** allows for equation alignment with the **mark** and **lineup** tokens. mandoc discards these tokens. The **back** *n*, **fwd** *n*, **up** *n*, and **down** *n* commands are also ignored.

SEE ALSO

mandoc(1), man(7), mandoc_char(7), mdoc(7), roff(7)

Brian W. Kernighan and Lorinda L. Cherry, "System for Typesetting Mathematics", *Communications of the ACM*, 18, pp. 151-157, March, 1975.

Brian W. Kernighan and Lorinda L. Cherry, *Typesetting Mathematics, User’s Guide*, 1976.

Brian W. Kernighan and Lorinda L. Cherry, *Typesetting Mathematics, User’s Guide (Second Edition)*, 1978.

HISTORY

The eqn utility, a preprocessor for troff, was originally written by Brian W. Kernighan and Lorinda L. Cherry in 1975. The GNU reimplementation of eqn, part of the GNU troff package, was released in 1989 by James Clark. The eqn component of mandoc(1) was added in 2011.

AUTHORS

This **eqn** reference was written by Kristaps Dzonsons <*kristaps@bsd.lv*>.