

- `\*[string [arg ...]]`  
Interpolate *string*, passing it *arg ...* as arguments.
- `\/`  
Apply an *italic correction*: modify the spacing of the preceding glyph so that the distance between it and the following glyph is correct if the latter is of upright shape. For example, if an italic “f” is followed immediately by a roman right parenthesis, then in many fonts the top right portion of the “f” overlaps the top of the right parenthesis, producing *f)*, which is ugly. Inserting `\/` between them produces *f)* and avoids this problem. Use this escape sequence whenever an oblique glyph is immediately followed by an upright glyph without any intervening space.
- `\,`  
Apply a *left italic correction*: modify the spacing of the following glyph so that the distance between it and the preceding glyph is correct if the latter is of upright shape. For example, if a roman left parenthesis is immediately followed by an italic “f”, then in many fonts the bottom left portion of the “f” overlaps the bottom of the left parenthesis, producing *(f*, which is ugly. Inserting `\,` between them produces *(f* and avoids this problem. Use this escape sequence whenever an upright glyph is followed immediately by an oblique glyph without any intervening space.
- `\:`  
Insert a non-printing break point. That is, a word can break there, but the soft hyphen character does not mark the break point if it does (in contrast to “\%”). The remainder of the word is subject to hyphenation as normal.
- `\?anything\?`  
Suppress formatting of *anything*. This feature has two applications.  
Use it with the output comparison operator to compare its operands by character rather than as formatted output. Since *troff* reads comparands protected with `\?` in copy mode, they need not even be valid *groff* syntax. The escape character is still lexically recognized, however, and consumes the next character.  
When used in a diversion, `\?` transparently embeds input, read in copy mode, until its own next occurrence on the input line. Use `\!` if you want to embed newlines in a diversion. Unlike `\!`, `\?` is interpreted even in copy mode, and *anything* in the top-level diversion is not sent to device-independent output.
- `\[char]`  
Typeset the special character *char*. See [groff\\_char\(7\)](#).
- `\[base-char combining-component ...]`  
Typeset a composite glyph consisting of *base-char* overlaid with one or more *combining-components*. For example, “ `\[A ho]`” is a capital letter “A” with a “hook accent” (ogonek). See the **composite** request below; *Groff: The GNU Implementation of troff*, the *groff* Texinfo manual, for details of composite glyph name construction; and [groff\\_char\(7\)](#) for a list of components used in composite glyph names.
- `\~`  
Insert an unbreakable space that is adjustable like an ordinary space. It is discarded from the end of an output line if a break is forced.

### Restricted requests

To mitigate risks from untrusted input documents, the **cf**, **pi**, and **sy** requests are disabled by default. [troff\(1\)](#)’s `-U` option enables the formatter’s “unsafe mode”, restoring their function (and enabling additional *groff* extension requests, **open**, **opena**, and **ps0**).

### New requests

Several GNU *troff* requests work like AT&T *troff*’s “**as**” and **ds** requests, accepting an optional leading neutral double-quote, notated “[”], in an argument that the formatter reads in copy mode to the end of the input line, permitting inclusion of leading spaces.

#### **.aln** *new-register existing-register*

Create alias (additional name) *new-register* of *existing-register*. If *existing-register* is undefined, GNU *troff* produces a warning in category “**reg**” and ignores the request. See section “Warnings”