

NAME

uniq — merge or filter adjacent identical lines

SYNOPSIS

```
uniq [-c] [-u|-d] [-iz] [-f fields] [-s skip] [-w limit] [from [to]]
uniq [-D|-G|--{all-repeated|group}=none|separate|prepend|append|both] [-iz]
    [-f fields] [-s skip] [-w limit] [from [to]]
```

DESCRIPTION

Copies consecutive lines from *from* (standard input stream if "-", the default) to *to* (standard output stream if "-", the default; otherwise created as `rw - umask` and truncated – equivalent to shell `>`):

- by default writing only the first line of each equal sequence,
- with **-u** writing only locally-unique lines,
- with **-d** writing only the first of each sequence of duplicates,
- with **-D** writing each duplicate value in a sequence, potentially separated by empty lines,
- with **-G** separating equal sequences with empty lines.

By default, the entire line is compared; **-f** slices off *fields* leading fields (defined as a maximal series of blanks (spaces or tabs in the **C** locale) followed by a maximal series of nonblanks), then **-s** slices off *skip* leading characters, then **-w** yields a maximum of *limit* characters.

The *entire* line is always written.

Unless **-i**, comparisons are byte-wise; otherwise, they're case-insensitive across characters in the current locale (invalid sequences are assumed to have a length of **1** byte and yield the maximum character).

The last of **-udDG** specified, if any, applies.

OPTIONS

- c, --count** Prepend each written line with the number of lines it had coalesced.
- u, --unique** Only write lines that are non-equal to their neighbours, i.e. are the sole members of a sequence of length **1**.
- d, --repeated** Write only the first line of each equal sequence longer than **1**.
- D, --all-repeated, --all-repeated=none** Write all lines of each equal sequence longer than **1**.
- all-repeated=separate** Likewise, but separate sequences with an empty line.
- all-repeated=prepend** Likewise, but prefix each sequence with an empty line.
- all-repeated=append** Likewise, but suffix each sequence with an empty line.
- all-repeated=both** Likewise, but prefix *and* suffix the first such sequence, suffixing the subsequent ones.
- G, --group, --group=separate** Write all lines of all sequences, separating sequences with an empty line.
- group=none** Likewise, but don't insert empty lines. This is mostly equivalent to **cat**.
- group=prepend, --group=append, --group=both** Analogous to **--all-repeated=**. All **--all-repeated** and **--group** values are prefix-matched (**--group=b** is equivalent to **--group=both**, &c.).
- i, --ignore-case** Compare lines case-insensitively according to the current locale.
- z, --zero-terminated** Line separator is NUL instead of newline.
- f, --skip-fields=*fields*** Skip the first *fields* maximal series of blanks then nonblanks for comparison.
- s, --skip-chars=*skip*** Skip the first *skip* characters for comparison.

-w, --check-chars=limit Compare up to *limit* characters.

EXIT STATUS

1 if *from* or *to* couldn't be opened.

EXAMPLES

Exercise all slicing/comparison options:

```
$ printf '%s\n' 'a 0aQ' ' b 1Awo' | uniq -ci -f1 -s2 -w1
2 a 0aQ
```

SEE ALSO

`sort(1)` to make equivalent lines adjacent, or its **-u** flag, which can uniquify lines based on collation sequence instead of equality.

`iswblank(3)`, `mbrtowc(3)`

STANDARDS

Conforms to IEEE Std 1003.1-2024 (“POSIX.1”), except **0** is allowed for **-fs**; the standard allows any (or no) number alignment for the **-c** column — this implementation matches the GNU system at **7** columns and a space, deviating from the AT&T UNIX of **4** and a space. The input file is specified to be a text file, which must not contain NULs: most other implementations terminate the line at the first NUL.

-Dizw, --group are extensions, originating from the GNU system; the **-G** spelling is an extension; the GNU system forbids **--all-repeated=append**, **--all-repeated=both**, and **--group=none**.

Because **-fsw** operate on characters, they are not suitable for slicing arbitrary data: set `LC_ALL=C` (`LC_CTYPE`, **POSIX**) to slice by byte (this also replicates the (broken) behaviour of the GNU system; the same applies to **-i**, questionable though its usefulness in that domain may be).

HISTORY

Appeared in Version 3 AT&T UNIX as `uniq(I)`:

```
NAME          uniq -- report repeated lines in a file
SYNOPSIS      uniq [ ud ] [ input [ output ] ]
```

With the default case and both flags described as present-day.

Version 4 AT&T UNIX sees a **SYNOPSIS** of

```
uniq [ -udc [ +n ] [ -n ] ] [ input [ output ] ]
```

with **-c** always applying the default filter, overriding **-ud** (if specified), the count aligned to **4** columns, followed by a space, **-n** is equivalent to present-day **-f n**, and **+n** to **-s n** (though, expectedly, byte-wise). The maximal line size is **1000** bytes, unprotected against overflows, and terminating at a NUL.

Version 7 AT&T UNIX exits **1** on failure to open either file and writes the error to the standard error stream.

4.4BSD sees a rewrite, citing IEEE Std 1003.2 (“POSIX.2”), with a **SYNOPSIS** of

```
uniq [-c | -d | -u] [-f fields] [-s chars] [input_file] [output_file]
```

but a much more representative usage string of

```
usage: uniq [-c | -du] [-f fields] [-s chars] [input] [output]
```

insofar as **-c** excludes either of **-du**, and specifying both **-du** is equivalent to the default output (curiously, this matches all prior manuals, which read

Note that the normal mode output is the union of the **-u** and **-d** mode outputs. but is unnoted in the rewritten one). The line sizes are now **8** KiB and protected, and the “historic” **-n** and **+n** options are undocumented beyond a **COMPATIBILITY** mention, but recognised for compatibility. Fields are separated not by blanks (`isblank()`: space (**0x20**), tab (**0x09**)) but by whitespace (`isspace()`: also the vertical tab (**0x0B**), form-feed (**0x0C**), and carriage return (**0x0D**)).

X/Open Portability Guide Issue 2 (“XPG2”) includes Version 4 AT&T UNIX **uniq** verbatim.

X/Open Portability Guide Issue 3 (“XPG3”) adds **APPLICATION USAGE**, entirely shaded IN (“Internationalised functionality”, defined as optional), of:

In an internationalised environment, the value of the `LC_COLLATE` environment variable must be equal to the value it had when the input files were sorted.

If **uniq** does not support selection of collating sequences via `LC_COLLATE`, the input files must be sorted according to the collating sequence of the “C” locale (see **Volume 3, XSI Supplementary Definitions, Chapter 7, C Program Locale**).

— indeed, specifying the comparison as maybe current collation, maybe not, and limiting the domain to 7 bits if not, and also weirdly discounting all uses of **uniq** that aren’t in consort with **sort**. Unsurprisingly, no implementation does this.

IEEE Std 1003.2-1992 (“POSIX.2”) sees largely-present-day **uniq** with **-cdufs**, the **-n +m** syntax marked obsolete, **-f** defined in terms of blanks from the current locale and **-s** in terms of characters, likewise, and “-”-as-standard-input-stream is allowed for *from*, but not for *to*. *from* must be a text file — no embedded NULs, lines of up to `LINE_MAX` bytes, and must end in a newline. No mention is made of collation.

Version 3 of the Single UNIX Specification (“SUSv3”) removes the obsolescent syntax and requires, in **ENVIRONMENT VARIABLES**:

`LC_COLLATE` Determine the locale for ordering rules.

For no apparent reason, considering that the wording remains “repeated”, which implies equality, not equivalence, and no mention of ordering is made in the rest of the **uniq** section.

IEEE Std 1003.1-2008 (“POSIX.1”) allows the obsolete syntax by allowing the option delimiter to be **+**, allows *to* being “-” to mean the standard output stream, explicitly discards newlines for comparison (matching existing practice), removes the `LC_COLLATE` mention and clarifies in **EXAMPLES** the current guidance that

To remove duplicate lines based on whether they collate equally instead of whether they are identical, applications should use:

```
sort -u
```

instead of:

```
sort | uniq
```