



Full credit is given to the above companies including the OS that this PDF file was generated!

Red Hat Enterprise Linux Release 9.2 Manual Pages on 'uniq.1p' command

\$ man uniq.1p

UNIQ(1P) POSIX Programmer's Manual UNIQ(1P)

PROLOG

This manual page is part of the POSIX Programmer's Manual. The Linux implementation of this interface may differ (consult the corresponding Linux manual page for details of Linux behavior), or the interface may not be implemented on Linux.

NAME

uniq ? report or filter out repeated lines in a file

SYNOPSIS

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

DESCRIPTION

The uniq utility shall read an input file comparing adjacent lines, and write one copy of each input line on the output. The second and succeeding copies of repeated adjacent input lines shall not be written. The trailing <newline> of each line in the input shall be ignored when doing comparisons. Repeated lines in the input shall not be detected if they are not adjacent.

OPTIONS

The uniq utility shall conform to the Base Definitions volume of POSIX.1?2017, Section 12.2, Utility Syntax Guidelines, except that '+' may be recognized as an option delimiter as well as '-'.

The following options shall be supported:

- c Precede each output line with a count of the number of times the line occurred in the input.
- d Suppress the writing of lines that are not repeated in the input.
- f fields Ignore the first fields fields on each input line when doing comparisons, where fields is a positive decimal integer. A field is the maximal string matched by the basic regular expression:


```
[[[:blank:]]*[^[:blank:]]*
```

 If the fields option-argument specifies more fields than appear on an input line, a null string shall be used for comparison.
- s chars Ignore the first chars characters when doing comparisons, where chars shall be a positive decimal integer. If specified in conjunction with the -f option, the first chars characters after the first fields fields shall be ignored. If the chars option-argument specifies more characters than remain on an input line, a null string shall be used for comparison.
- u Suppress the writing of lines that are repeated in the input.

OPERANDS

The following operands shall be supported:

input_file

A pathname of the input file. If the input_file operand is not specified, or if the input_file is '-', the standard input shall be used.

output_file

A pathname of the output file. If the output_file operand is not specified, the standard output shall be used. The results are unspecified if the file named by output_file is the file named by input_file.

STDIN

The standard input shall be used only if no input_file operand is specified or if input_file is '-'. See the INPUT FILES section.

INPUT FILES

The input file shall be a text file.

ENVIRONMENT VARIABLES

The following environment variables shall affect the execution of `uniq`:

LANG Provide a default value for the internationalization variables that are unset or null. (See the Base Definitions volume of POSIX.1?2017, Section 8.2, Internationalization Variables for the precedence of internationalization variables used to determine the values of locale categories.)

LC_ALL If set to a non-empty string value, override the values of all the other internationalization variables.

LC_CTYPE Determine the locale for the interpretation of sequences of bytes of text data as characters (for example, single-byte as opposed to multi-byte characters in arguments and input files) and which characters constitute a <blank> in the current locale.

LC_MESSAGES

Determine the locale that should be used to affect the format and contents of diagnostic messages written to standard error.

NLSPATH Determine the location of message catalogs for the processing of `LC_MESSAGES`.

ASYNCHRONOUS EVENTS

Default.

STDOUT

The standard output shall be used if no `output_file` operand is specified, and shall be used if the `output_file` operand is '-' and the implementation treats the '-' as meaning standard output. Otherwise, the standard output shall not be used. See the `OUTPUT FILES` section.

STDERR

The standard error shall be used only for diagnostic messages.

OUTPUT FILES

If the `-c` option is specified, the output file shall be empty or each

line shall be of the form:

```
"%d %s", <number of duplicates>, <line>
```

otherwise, the output file shall be empty or each line shall be of the form:

```
"%s", <line>
```

EXTENDED DESCRIPTION

None.

EXIT STATUS

The following exit values shall be returned:

0 The utility executed successfully.

>0 An error occurred.

CONSEQUENCES OF ERRORS

Default.

The following sections are informative.

APPLICATION USAGE

If the collating sequence of the current locale has a total ordering of all characters, the sort utility can be used to cause repeated lines to be adjacent in the input file. If the collating sequence does not have a total ordering of all characters, the sort utility should still do this but it might not. To ensure that all duplicate lines are eliminated, and have the output sorted according the collating sequence of the current locale, applications should use:

```
LC_ALL=C sort -u | sort
```

instead of:

```
sort | uniq
```

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
```

When using `uniq` to process pathnames, it is recommended that `LC_ALL`, or at least `LC_CTYPE` and `LC_COLLATE`, are set to `POSIX` or `C` in the environment, since pathnames can contain byte sequences that do not form valid

characters in some locales, in which case the utility's behavior would be undefined. In the POSIX locale each byte is a valid single-byte character, and therefore this problem is avoided.

EXAMPLES

The following input file data (but flushed left) was used for a test series on `uniq`:

```
#01 foo0 bar0 foo1 bar1
#02 bar0 foo1 bar1 foo1
#03 foo0 bar0 foo1 bar1
#04
#05 foo0 bar0 foo1 bar1
#06 foo0 bar0 foo1 bar1
#07 bar0 foo1 bar1 foo0
```

What follows is a series of test invocations of the `uniq` utility that use a mixture of `uniq` options against the input file data. These tests verify the meaning of adjacent. The `uniq` utility views the input data as a sequence of strings delimited by `'\n'`. Accordingly, for the `fieldsth` member of the sequence, `uniq` interprets unique or repeated adjacent lines strictly relative to the `fields+1`th member.

1. This first example tests the line counting option, comparing each line of the input file data starting from the second field:

```
uniq -c -f 1 uniq_01.t
 1 #01 foo0 bar0 foo1 bar1
 1 #02 bar0 foo1 bar1 foo1
 1 #03 foo0 bar0 foo1 bar1
 1 #04
 2 #05 foo0 bar0 foo1 bar1
 1 #07 bar0 foo1 bar1 foo0
```

The number '2', prefixing the fifth line of output, signifies that the `uniq` utility detected a pair of repeated lines. Given the input data, this can only be true when `uniq` is run using the `-f 1` option (which shall cause `uniq` to ignore the first field on each input line).

table Operating System Interface (POSIX), The Open Group Base Specifications Issue 7, 2018 Edition, Copyright (C) 2018 by the Institute of Electrical and Electronics Engineers, Inc and The Open Group. In the event of any discrepancy between this version and the original IEEE and The Open Group Standard, the original IEEE and The Open Group Standard is the referee document. The original Standard can be obtained online at <http://www.opengroup.org/unix/online.html> .

Any typographical or formatting errors that appear in this page are most likely to have been introduced during the conversion of the source files to man page format. To report such errors, see https://www.kernel.org/doc/man-pages/reporting_bugs.html .

IEEE/The Open Group

2017

UNIQ(1P)