



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

\$ man join.1p

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

join ? relational database operator

SYNOPSIS

```
join [-a file_number|-v file_number] [-e string] [-o list] [-t char]
     [-1 field] [-2 field] file1 file2
```

DESCRIPTION

The join utility shall perform an equality join on the files file1 and file2. The joined files shall be written to the standard output.

The join field is a field in each file on which the files are compared.

The join utility shall write one line in the output for each pair of lines in file1 and file2 that have join fields that collate equally.

The output line by default shall consist of the join field, then the remaining fields from file1, then the remaining fields from file2.

This format can be changed by using the -o option (see below). The -a option can be used to add unmatched lines to the output. The -v option can be used to output only unmatched lines.

The files file1 and file2 shall be ordered in the collating sequence of

sort -b on the fields on which they shall be joined, by default the first in each line. All selected output shall be written in the same collating sequence.

The default input field separators shall be <blank> characters. In this case, multiple separators shall count as one field separator, and leading separators shall be ignored. The default output field separator shall be a <space>.

The field separator and collating sequence can be changed by using the -t option (see below).

If the same key appears more than once in either file, all combinations of the set of remaining fields in file1 and the set of remaining fields in file2 are output in the order of the lines encountered.

If the input files are not in the appropriate collating sequence, the results are unspecified.

OPTIONS

The join utility shall conform to the Base Definitions volume of POSIX.1?2017, Section 12.2, Utility Syntax Guidelines.

The following options shall be supported:

-a file_number

Produce a line for each unpairable line in file file_number, where file_number is 1 or 2, in addition to the default output. If both -a1 and -a2 are specified, all unpairable lines shall be output.

-e string Replace empty output fields in the list selected by -o with the string string.

-o list Construct the output line to comprise the fields specified in list, each element of which shall have one of the following two forms:

1. file_number.field, where file_number is a file number and field is a decimal integer field number
2. 0 (zero), representing the join field

The elements of list shall be either <comma>-separated or <blank>-separated, as specified in Guideline 8 of the Base

Definitions volume of POSIX.1?2017, Section 12.2, Utility Syntax Guidelines. The fields specified by list shall be written for all selected output lines. Fields selected by list that do not appear in the input shall be treated as empty output fields. (See the -e option.) Only specifically requested fields shall be written. The application shall ensure that list is a single command line argument.

-t char Use character char as a separator, for both input and output.

Every appearance of char in a line shall be significant. When this option is specified, the collating sequence shall be the same as sort without the -b option.

-v file_number

Instead of the default output, produce a line only for each unpairable line in file_number, where file_number is 1 or 2.

If both -v1 and -v2 are specified, all unpairable lines shall be output.

-1 field Join on the fieldth field of file 1. Fields are decimal integers starting with 1.

-2 field Join on the fieldth field of file 2. Fields are decimal integers starting with 1.

OPERANDS

The following operands shall be supported:

file1, file2

A pathname of a file to be joined. If either of the file1 or file2 operands is '-', the standard input shall be used in its place.

STDIN

The standard input shall be used only if the file1 or file2 operand is '-'. See the INPUT FILES section.

INPUT FILES

The input files shall be text files.

ENVIRONMENT VARIABLES

The following environment variables shall affect the execution of join:

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_COLLATE

Determine the locale of the collating sequence `join` expects to have been used when the input files were sorted.

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).

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 `join` utility output shall be a concatenation of selected character fields. When the `-o` option is not specified, the output shall be:

```
"%s%s%s\n", <join field>, <other file1 fields>,  
<other file2 fields>
```

If the `join` field is not the first field in a file, the `<other file fields>` for that file shall be:

```
<fields preceding join field>, <fields following join field>
```

When the `-o` option is specified, the output format shall be:

```
"%s\n", <concatenation of fields>
```

where the concatenation of fields is described by the `-o` option, above.

For either format, each field (except the last) shall be written with its trailing separator character. If the separator is the default (<blank> characters), a single <space> shall be written after each field (except the last).

STDERR

The standard error shall be used only for diagnostic messages.

OUTPUT FILES

None.

EXTENDED DESCRIPTION

None.

EXIT STATUS

The following exit values shall be returned:

- 0 All input files were output successfully.
- >0 An error occurred.

CONSEQUENCES OF ERRORS

Default.

The following sections are informative.

APPLICATION USAGE

Pathnames consisting of numeric digits or of the form string.string should not be specified directly following the -o list.

If the collating sequence of the current locale does not have a total ordering of all characters (see the Base Definitions volume of POSIX.1?2017, Section 7.3.2, LC_COLLATE), join treats fields that collate equally but are not identical as being the same. If this behavior is not desired, it can be avoided by forcing the use of the POSIX locale (although this means re-sorting the input files into the POSIX locale collating sequence.)

When using join 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 -o 0 field essentially selects the union of the join fields. For example, given file phone:

!Name	Phone Number
Don	+1 123-456-7890
Hal	+1 234-567-8901
Yasushi	+2 345-678-9012

and file fax:

!Name	Fax Number
Don	+1 123-456-7899
Keith	+1 456-789-0122
Yasushi	+2 345-678-9011

(where the large expanses of white space are meant to each represent a single <tab>), the command:

```
join -t "<tab>" -a 1 -a 2 -e '(unknown)' -o 0,1.2,2.2 phone fax
```

(where <tab> is a literal <tab> character) would produce:

!Name	Phone Number	Fax Number
Don	+1 123-456-7890	+1 123-456-7899
Hal	+1 234-567-8901	(unknown)
Keith	(unknown)	+1 456-789-0122
Yasushi	+2 345-678-9012	+2 345-678-9011

Multiple instances of the same key will produce combinatorial results.

The following:

fa:

a x

a y

a z

fb:

a p

will produce:

a x p

a y p

a z p

And the following:

fa:

a b c

a d e

fb:

a w x

a y z

a o p

will produce:

a b c w x

a b c y z

a b c o p

a d e w x

a d e y z

a d e o p

RATIONALE

The `-e` option is only effective when used with `-o` because, unless specific fields are identified using `-o`, `join` is not aware of what fields might be empty. The exception to this is the join field, but identifying an empty join field with the `-e` string is not historical practice and some scripts might break if this were changed.

The `0` field in the `-o` list was adopted from the Tenth Edition version of `join` to satisfy international objections that the join in the base documents for IEEE Std 1003.2-1992 did not support the "full join" or "outer join" described in relational database literature. Although it has been possible to include a join field in the output (by default, or by field number using `-o`), the join field could not be included for an unpaired line selected by `-a`. The `-o 0` field essentially selects the union of the join fields.

This sort of outer join was not possible with the join commands in the base documents for IEEE Std 1003.2-1992. The `-o 0` field was chosen because it is an upwards-compatible change for applications. An alternative was considered: have the join field represent the union of the

fields in the files (where they are identical for matched lines, and one or both are null for unmatched lines). This was not adopted because it would break some historical applications.

The ability to specify file2 as - is not historical practice; it was added for completeness.

The -v option is not historical practice, but was considered necessary because it permitted the writing of only those lines that do not match on the join field, as opposed to the -a option, which prints both lines that do and do not match. This additional facility is parallel with the -v option of grep.

Some historical implementations have been encountered where a blank line in one of the input files was considered to be the end of the file; the description in this volume of POSIX.1?2017 does not cite this as an allowable case.

Earlier versions of this standard allowed -j, -j1, -j2 options, and a form of the -o option that allowed the list option-argument to be multiple arguments. These forms are no longer specified by POSIX.1?2008 but may be present in some implementations.

FUTURE DIRECTIONS

None.

SEE ALSO

awk, comm, sort, uniq

The Base Definitions volume of POSIX.1?2017, Section 7.3.2, LC_COLLATE, Chapter 8, Environment Variables, Section 12.2, Utility Syntax Guide? lines

COPYRIGHT

Portions of this text are reprinted and reproduced in electronic form from IEEE Std 1003.1-2017, Standard for Information Technology -- Portable 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

JOIN(1P)