
*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 'talk.1p' command*

***$ man talk.1p***

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

talk ? talk to another user

SYNOPSIS

talk address [terminal]

DESCRIPTION

The talk utility is a two-way, screen-oriented communication program.

When first invoked, talk shall send a message similar to:

Message from <unspecified string>

talk: connection requested by your_address

talk: respond with: talk your_address

to the specified address. At this point, the recipient of the message can reply by typing:

talk your_address

Once communication is established, the two parties can type simultane? ously, with their output displayed in separate regions of the screen.

Characters shall be processed as follows:

* Typing the <alert> character shall alert the recipient's terminal.

* Typing <control>?L shall cause the sender's screen regions to be refreshed.
* Typing the erase and kill characters shall affect the sender's ter? minal in the manner described by the termios interface in the Base Definitions volume of POSIX.1?2017, Chapter 11, General Terminal Interface.
* Typing the interrupt or end-of-file characters shall terminate the local talk utility. Once the talk session has been terminated on one side, the other side of the talk session shall be notified that the talk session has been terminated and shall be able to do noth? ing except exit.
* Typing characters from LC_CTYPE classifications print or space shall cause those characters to be sent to the recipient's termi? nal.
* When and only when the stty iexten local mode is enabled, the exis? tence and processing of additional special control characters and multi-byte or single-byte functions shall be implementation-de? fined.
* Typing other non-printable characters shall cause implementation-defined sequences of printable characters to be sent to the recipi? ent's terminal.

Permission to be a recipient of a talk message can be denied or granted by use of the mesg utility. However, a user's privilege may further constrain the domain of accessibility of other users' terminals. The talk utility shall fail when the user lacks appropriate privileges to perform the requested action.

Certain block-mode terminals do not have all the capabilities necessary to support the simultaneous exchange of messages required for talk. When this type of exchange cannot be supported on such terminals, the implementation may support an exchange with reduced levels of simulta? neous interaction or it may report an error describing the terminal-re? lated deficiency.

OPTIONS

None.

OPERANDS

The following operands shall be supported:

address The recipient of the talk session. One form of address is the <user name>, as returned by the who utility. Other address formats and how they are handled are unspecified.

terminal If the recipient is logged in more than once, the terminal argument can be used to indicate the appropriate terminal name. If terminal is not specified, the talk message shall be displayed on one or more accessible terminals in use by the recipient. The format of terminal shall be the same as that returned by the who utility.

STDIN

Characters read from standard input shall be copied to the recipient's terminal in an unspecified manner. If standard input is not a terminal, talk shall write a diagnostic message and exit with a non-zero status.

INPUT FILES

None.

ENVIRONMENT VARIABLES

The following environment variables shall affect the execution of talk:

LANG Provide a default value for the internationalization vari? ables that are unset or null. (See the Base Definitions vol? ume of POSIX.1?2017, Section 8.2, Internationalization Vari? ables 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). If the recipient's locale does not use an LC_CTYPE equivalent to the sender's, the results are undefined.

LC_MESSAGES

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

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

TERM Determine the name of the invoker's terminal type. If this variable is unset or null, an unspecified default terminal type shall be used.

ASYNCHRONOUS EVENTS

When the talk utility receives a SIGINT signal, the utility shall ter? minate and exit with a zero status. It shall take the standard action for all other signals.

STDOUT

If standard output is a terminal, characters copied from the recipi? ent's standard input may be written to standard output. Standard output also may be used for diagnostic messages. If standard output is not a terminal, talk shall exit with a non-zero status.

STDERR

None.

OUTPUT FILES

None.

EXTENDED DESCRIPTION

None.

EXIT STATUS

The following exit values shall be returned:

0 Successful completion.

>0 An error occurred or talk was invoked on a terminal incapable of supporting it.

CONSEQUENCES OF ERRORS

Default.

The following sections are informative.

APPLICATION USAGE

Because the handling of non-printable, non-<space> characters is tied

to the stty description of iexten, implementation extensions within the terminal driver can be accessed. For example, some implementations provide line editing functions with certain control character se? quences.

EXAMPLES

None.

RATIONALE

The write utility was included in this volume of POSIX.1?2017 since it can be implemented on all terminal types. The talk utility, which can? not be implemented on certain terminals, was considered to be a ``bet? ter'' communications interface. Both of these programs are in wide? spread use on historical implementations. Therefore, both utilities have been specified.

All references to networking abilities (talking to a user on another system) were removed as being outside the scope of this volume of POSIX.1?2017.

Historical BSD and System V versions of talk terminate both of the con? versations when either user breaks out of the session. This can lead to adverse consequences if a user unwittingly continues to enter text that is interpreted by the shell when the other terminates the session. Therefore, the version of talk specified by this volume of POSIX.1?2017 requires both users to terminate their end of the session explicitly.

Only messages sent to the terminal of the invoking user can be interna? tionalized in any way:

* The original ``Message from <unspecified string> ...'' message sent to the terminal of the recipient cannot be internationalized because the environment of the recipient is as yet inaccessible to the talk utility. The environment of the invoking party is irrele? vant.
* Subsequent communication between the two parties cannot be interna? tionalized because the two parties may specify different languages in their environment (and non-portable characters cannot be mapped from one language to another).

* Neither party can be required to communicate in a language other than C and/or the one specified by their environment because un? available terminal hardware support (for example, fonts) may be re? quired.

The text in the STDOUT section reflects the usage of the verb ``dis? play'' in this section; some talk implementations actually use standard output to write to the terminal, but this volume of POSIX.1?2017 does not require that to be the case.

The format of the terminal name is unspecified, but the descriptions of ps, talk, who, and write require that they all use or accept the same format.

The handling of non-printable characters is partially implementation-defined because the details of mapping them to printable sequences is not needed by the user. Historical implementations, for security rea? sons, disallow the transmission of non-printable characters that may send commands to the other terminal.

FUTURE DIRECTIONS

None.

SEE ALSO

mesg, stty, who, write

The Base Definitions volume of POSIX.1?2017, Chapter 8, Environment Variables, Chapter 11, General Terminal Interface

COPYRIGHT

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 .