

NAME

tip - connect to a remote system

SYNOPSIS

tip [-**nv**] [-*speed*] [*system-name*]

DESCRIPTION

The **tip** utility establishes a full-duplex connection to another machine, giving the appearance of being logged in directly on the remote CPU. It goes without saying that you must have a login on the machine (or equivalent) to which you wish to connect.

The options are as follows:

-n No escape (disable tilde).

-v Set verbose mode.

If *speed* is specified, it will override any baudrate specified in the system description being used.

If neither *speed* nor *system-name* are specified, *system-name* will be set to the value of the `HOST` environment variable.

If *speed* is specified but *system-name* is not, *system-name* will be set to a value of "tip" with *speed* appended. For example, **tip -1200** will set *system-name* to "tip1200".

Typed characters are normally transmitted directly to the remote machine (which does the echoing as well). A tilde ('~') appearing as the first character of a line is an escape signal; the following are recognized:

~^D or **~.**

Drop the connection and exit. Only the connection is dropped - the login session is not terminated.

~c [*name*]

Change directory to *name* (no argument implies change to home directory).

~! Escape to a shell (exiting the shell will return to **tip**).

~> Copy file from local to remote. The **tip** utility prompts for the name of a local file to transmit.

- ~<** Copy file from remote to local. The **tip** utility prompts first for the name of the file to be sent, then for a command to be executed on the remote machine.

~p *from* [*to*]

Send a file to a remote UNIX host. This command causes the remote UNIX system to run the following command string, sending it the *from* file:

```
stty -echo; cat > 'to'; stty echo
```

If the *to* file is not specified, the *from* file name is used. This command is actually a UNIX specific version of the **~>** command.

~t *from* [*to*]

Take a file from a remote UNIX host. As in the **~p** command, the *to* file defaults to the *from* file name if it is not specified. The remote host executes the following command string to send the file to **tip**:

```
cat 'from'; echo '' | tr '\012' '\01'
```

- ~|** Pipe the output from a remote command to a local UNIX process. The command string sent to the local UNIX system is processed by the shell.

- ~\$** Pipe the output from a local UNIX process to the remote host. The command string sent to the local UNIX system is processed by the shell.

- ~C** Fork a child process on the local system to perform special protocols such as XMODEM. The child program will be run with the following arrangement of file descriptors:

```
0 <-> remote tty in
1 <-> remote tty out
2 <-> local tty stderr
```

- ~#** Send a **BREAK** to the remote system. For systems which do not support the necessary **ioctl()** call, the break is simulated by a sequence of line speed changes and DEL characters.

- ~s** Set a variable (see the discussion below).

- ~v** List all variables and their values (if set).

- ~^Z** Stop **tip** (only available with job control).

~^Y Stop only the "local side" of **tip** (only available with job control); the "remote side" of **tip**, the side that displays output from the remote host, is left running.

~? Get a summary of the tilde escapes.

To find the system description, and thus the operating characteristics of *system-name*, **tip** searches for a system description with a name identical to *system-name*. The search order is as follows:

1. If the environment variable REMOTE does not start with a '/' it is assumed to be a system description, and is considered first.
2. If the environment variable REMOTE begins with a '/' it is assumed to be a path to a remote(5) database, and the specified database is searched.
3. The default remote(5) database, */etc/remote*, is searched.

See remote(5) for full documentation on system descriptions.

The *br* capability is used in system descriptions to specify the baud rate with which to establish a connection. If the value specified is not suitable, the baud rate to be used may be given on the command line, e.g. "tip -300 mds".

When **tip** establishes a connection, it sends out the connection message specified in the *cm* capability of the system description being used.

When **tip** prompts for an argument, for example during setup of a file transfer, the line typed may be edited with the standard erase and kill characters. A null line in response to a prompt, or an interrupt, will abort the dialogue and return the user to the remote machine.

The **tip** utility guards against multiple users connecting to a remote system by opening modems and terminal lines with exclusive access, and by honoring the locking protocol used by uucico(8) (*ports/net/freebsd-uucp*).

During file transfers **tip** provides a running count of the number of lines transferred. When using the ~> and ~< commands, the *eofread* and *eofwrite* variables are used to recognize end-of-file when reading, and specify end-of-file when writing (see below). File transfers normally depend on hardwareflow or tandem mode for flow control. If the remote system does not support hardwareflow or tandem mode, *echocheck* may be set to indicate that **tip** should synchronize with the remote system on the echo of each transmitted character.

When **tip** must dial a phone number to connect to a system, it will print various messages indicating its actions. The **tip** utility supports a variety of auto-call units and modems with the *at* capability in system descriptions.

Support for Ventel 212+ (*ventel*), Hayes AT-style (*hayes*), USRobotics Courier (*courier*), Telebit T3000 (*t3000*) and Racal-Vadic 831 (*vadic*) units is enabled by default.

Support for Bizcomp 1031[fw] (*biz31[fw]*), Bizcomp 1022[fw] (*biz22[fw]*), DEC DF0[23]-AC (*df0[23]*), DEC DN-11 (*dn11*) and Racal-Vadic 3451 (*v3451*) units can be added by recompiling **tip** with the appropriate defines.

Note that if support for both the Racal-Vadic 831 and 3451 is enabled, they are referred to as the *v831* and *v3451*, respectively. If only one of the two is supported, it is referred to as *vadic*.

Variables

The **tip** utility maintains a set of variables which control its operation. Some of these variables are read-only to normal users (root is allowed to change anything of interest). Variables may be displayed and set through the *~s* escape. The syntax for variables is patterned after *vi*(1) and *Mail*(1). Supplying "all" as an argument to the set command displays all variables readable by the user. Alternatively, the user may request display of a particular variable by attaching a '?' to the end. For example, "escape?" displays the current escape character.

Variables are numeric, string, character, or boolean values. Boolean variables are set merely by specifying their name; they may be reset by prepending a '!' to the name. Other variable types are set by concatenating an '=' and the value. The entire assignment must not have any blanks in it. A single set command may be used to interrogate as well as set a number of variables. Variables may be initialized at run time by placing set commands (without the *~s* prefix) in the initialization file *~/.tiprc*; the **-v** option additionally causes **tip** to display the sets as they are made. Certain common variables have abbreviations. The following is a list of common variables, their abbreviations, and their default values:

baudrate

(*num*) The baud rate at which the connection was established; abbreviated *ba*.

beautify

(*bool*) Discard unprintable characters when a session is being scripted; abbreviated *be*.

dialtimeout

(*num*) When dialing a phone number, the time (in seconds) to wait for a connection to be established; abbreviated *dial*.

echocheck

(*bool*) Synchronize with the remote host during file transfer by waiting for the echo of the last character transmitted; default is **off**.

eofread

(*str*) The set of characters which signify an end-of-transmission during a ~< file transfer command; abbreviated *eofr*.

eofwrite

(*str*) The string sent to indicate end-of-transmission during a ~> file transfer command; abbreviated *eofw*.

eol

(*str*) The set of characters which indicate an end-of-line. The **tip** utility will recognize escape characters only after an end-of-line.

escape

(*char*) The command prefix (escape) character; abbreviated *es*; default value is '~'.

exceptions

(*str*) The set of characters which should not be discarded due to the beautification switch; abbreviated *ex*; default value is "\t\n\f\b".

force (*char*) The character used to force literal data transmission; abbreviated *fo*; default value is '^P'.

framesize

(*num*) The amount of data (in bytes) to buffer between file system writes when receiving files; abbreviated *fr*.

hardwareflow

(*bool*) Whether hardware flow control (CRTSCTS) is enabled for the connection; abbreviated *hf*; default value is **off**.

host

(*str*) The name of the host to which you are connected; abbreviated *ho*.

linedisc

(*num*) The line discipline to use; abbreviated *ld*.

prompt

(*char*) The character which indicates an end-of-line on the remote host; abbreviated *pr*; default value is '\n'. This value is used to synchronize during data transfers. The count of lines

transferred during a file transfer command is based on receipt of this character.

raise (*bool*) Upper case mapping mode; abbreviated *ra*; default value is **off**. When this mode is enabled, all lowercase letters will be mapped to uppercase by **tip** for transmission to the remote machine.

raisechar

(*char*) The input character used to toggle uppercase mapping mode; abbreviated *rc*; not set by default.

record

(*str*) The name of the file in which a session script is recorded; abbreviated *rec*; default value is *tip.record*.

script (*bool*) Session scripting mode; abbreviated *sc*; default is **off**. When *script* is **true**, **tip** will record everything transmitted by the remote machine in the script record file specified in *record*. If the *beautify* switch is on, only printable ASCII characters will be included in the script file (those characters between 040 and 0177). The variable *exceptions* is used to indicate characters which are an exception to the normal beautification rules.

tabexpand

(*bool*) Expand tabs to spaces during file transfers; abbreviated *tab*; default value is **false**. Each tab is expanded to 8 spaces.

tandem

(*bool*) Use XON/XOFF flow control to throttle data from the remote host; abbreviated *ta*. The default value is **true** unless the *nt* capability has been specified in */etc/remote*, in which case the default value is **false**.

verbose

(*bool*) Verbose mode; abbreviated *verb*; default is **true**. When verbose mode is enabled, **tip** prints messages while dialing, shows the current number of lines transferred during a file transfer operations, and more.

ENVIRONMENT

HOME

The home directory to use for the **~c** command.

HOST

The default value for *system-name* if none is specified via the command line.

PHONES

A path to a phones(5) database.

REMOTE

A system description, or an absolute path to a remote(5) system description database.

SHELL

The name of the shell to use for the `~!` command; default value is `"/bin/sh"`.

FILES

<code>~/.tiprc</code>	initialization file
<code>tip.record</code>	record file
<code>/etc/phones</code>	default phones(5) file
<code>/etc/remote</code>	global remote(5) database
<code>/var/log/aculog</code>	line access log
<code>/var/spool/lock/LCK.*</code>	lock file to avoid conflicts with uucp(1) (<i>ports/net/freebsd-uucp</i>)

EXAMPLES

Connect to the first USB serial port at the speed of 115200 baud:

```
tip ucom1 -115200
```

SEE ALSO

`cu(1)`, `phones(5)`, `remote(5)`

HISTORY

The **tip** command appeared in 4.2BSD.

BUGS

The full set of variables is undocumented and should, probably, be pared down.