at(1p) - Linux manual page

PROLOG top

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 top

at – execute commands at a later time

SYNOPSIS top

```
at [-m] [-f file] [-q queuename] -t time_arg

at [-m] [-f file] [-q queuename] timespec...

at -r at_job_id...

at -l -q queuename

at -l [at_job_id...]
```

DESCRIPTION top

The *at* utility shall read commands from standard input and group them together as an *at-job*, to be executed at a later time.

The at-job shall be executed in a separate invocation of the shell, running in a separate process group with no controlling terminal, except that the environment variables, current working directory, file creation mask, and other implementation-defined execution-time attributes in effect when the *at* utility is

executed shall be retained and used when the at-job is executed.

When the at-job is submitted, the *at_job_id* and scheduled time shall be written to standard error. The *at_job_id* is an identifier that shall be a string consisting solely of alphanumeric characters and the <period> character. The *at_job_id* shall be assigned by the system when the job is scheduled such that it uniquely identifies a particular job.

User notification and the processing of the job's standard output and standard error are described under the -m option.

Users shall be permitted to use *at* if their name appears in the file at.allow which is located in an implementation-defined directory. If that file does not exist, the file at.deny, which is located in an implementation-defined directory, shall be checked to determine whether the user shall be denied access to *at*. If neither file exists, only a process with appropriate privileges shall be allowed to submit a job. If only at.deny exists and is empty, global usage shall be permitted. The at.allow and at.deny files shall consist of one user name per line.

OPTIONS [top](#)

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

The following options shall be supported:

-f *file* Specify the pathname of a file to be used as the source of the at-job, instead of standard input.

-l (The letter ell.) Report all jobs scheduled for the invoking user if no *at_job_id* operands are specified. If *at_job_id*s are specified, report only information for these jobs. The output shall be written to standard output.

-m Send mail to the invoking user after the at-job has run, announcing its completion. Standard output and standard error produced by the at-job shall be mailed to the user as well, unless redirected elsewhere. Mail shall be sent even if the job produces no output.

If -m is not used, the job's standard output and standard error shall be provided to the user by means of

mail, unless they are redirected elsewhere; if there is no such output to provide, the implementation need not notify the user of the job's completion.

-q *queuename*
Specify in which queue to schedule a job for submission. When used with the -l option, limit the search to that particular queue. By default, at-jobs shall be scheduled in queue *a*. In contrast, queue *b* shall be reserved for batch jobs; see *batch*. The meanings of all other *queuename*s are implementation-defined. If -q is specified along with either of the -t *time_arg* or *timespec* arguments, the results are unspecified.

-r Remove the jobs with the specified *at_job_id* operands that were previously scheduled by the *at* utility.

-t *time_arg*
Submit the job to be run at the time specified by the *time* option-argument, which the application shall ensure has the format as specified by the *touch* -t *time* utility.

OPERANDS [top](#)

The following operands shall be supported:

at_job_id The name reported by a previous invocation of the *at* utility at the time the job was scheduled.

timespec Submit the job to be run at the date and time specified. All of the *timespec* operands are interpreted as if they were separated by <space> characters and concatenated, and shall be parsed as described in the grammar at the end of this section. The date and time shall be interpreted as being in the timezone of the user (as determined by the *TZ* variable), unless a timezone name appears as part of *time*, below.

In the POSIX locale, the following describes the three parts of the time specification string. All of the values from the *LC_TIME* categories in the POSIX locale shall be recognized in a case-insensitive manner.

time The time can be specified as one, two, or four digits. One-digit and two-digit numbers shall

be taken to be hours; four-digit numbers to be hours and minutes. The time can alternatively be specified as two numbers separated by a <colon>, meaning *hour*:*minute*. An AM/PM indication (one of the values from the am_pm keywords in the *LC_TIME* locale category) can follow the time; otherwise, a 24-hour clock time shall be understood. A timezone name can also follow to further qualify the time. The acceptable timezone names are implementation-defined, except that they shall be case-insensitive and the string utc is supported to indicate the time is in Coordinated Universal Time. In the POSIX locale, the *time* field can also be one of the following tokens:

midnight Indicates the time 12:00 am (00:00).

noon Indicates the time 12:00 pm.

now Indicates the current day and time. Invoking *at* <now> shall submit an at-job for potentially immediate execution (that is, subject only to unspecified scheduling delays).

date An optional *date* can be specified as either a month name (one of the values from the mon or abmon keywords in the *LC_TIME* locale category) followed by a day number (and possibly year number preceded by a comma), or a day of the week (one of the values from the day or abday keywords in the *LC_TIME* locale category). In the POSIX locale, two special days shall be recognized:

today Indicates the current day.

tomorrow Indicates the day following the current day.

If no *date* is given, today shall be assumed if the given time is greater than the current time, and tomorrow shall be assumed if it is less. If the given month is less than the current month (and no year is given), next year shall be assumed.

increment The optional *increment* shall be a number preceded by a <plus-sign> ('+') and suffixed by one of the following: minutes, hours, days, weeks, months, or years. (The singular forms shall also be accepted.) The keyword next shall be equivalent to an increment number of +1. For example, the following are equivalent commands:

```
at 2pm + 1 week
at 2pm next week
```

The following grammar describes the precise format of *timespec* in the POSIX locale. The general conventions for this style of grammar are described in *Section 1.3*, *Grammar Conventions*. This formal syntax shall take precedence over the preceding text syntax description. The longest possible token or delimiter shall be recognized at a given point. When used in a *timespec*, white space shall also delimit tokens.

```
%token hr24clock_hr_min
%token hr24clock_hour
/*
  An hr24clock_hr_min is a one, two, or four-digit number. A one-digit
  or two-digit number constitutes an hr24clock_hour. An hr24clock_hour
  may be any of the single digits [0,9], or may be double digits, ranging
  from [00,23]. If an hr24clock_hr_min is a four-digit number, the
  first two digits shall be a valid hr24clock_hour, while the last two
  represent the number of minutes, from [00,59].
*/

%token wallclock_hr_min
%token wallclock_hour
/*
  A wallclock_hr_min is a one, two-digit, or four-digit number.
  A one-digit or two-digit number constitutes a wallclock_hour.
  A wallclock_hour may be any of the single digits [1,9], or may
  be double digits, ranging from [01,12]. If a wallclock_hr_min
  is a four-digit number, the first two digits shall be a valid
  wallclock_hour, while the last two represent the number of
  minutes, from [00,59].
*/

%token minute
/*
  A minute is a one or two-digit number whose value can be [0,9]
```

```
  or [00,59].
*/

%token day_number
/*
  A day_number is a number in the range appropriate for the particular
  month and year specified by month_name and year_number, respectively.
  If no year_number is given, the current year is assumed if the given
  date and time are later this year. If no year_number is given and
  the date and time have already occurred this year and the month is
  not the current month, next year is the assumed year.
*/

%token year_number
/*
  A year_number is a four-digit number representing the year A.D., in
  which the at_job is to be run.
*/

%token inc_number
/*
  The inc_number is the number of times the succeeding increment
  period is to be added to the specified date and time.
*/

%token timezone_name
/*
  The name of an optional timezone suffix to the time field, in an
  implementation-defined format.
*/

%token month_name
/*
  One of the values from the mon or abmon keywords in the LC_TIME
  locale category.
*/

%token day_of_week
/*
  One of the values from the day or abday keywords in the LC_TIME
  locale category.
*/

%token am_pm
/*
  One of the values from the am_pm keyword in the LC_TIME locale
  category.
```

```
*/

%start timespec
%%
timespec    : time
            | time date
            | time increment
            | time date increment
            | nowspec
            ;

nowspec     : "now"
            | "now" increment
            ;

time        : hr24clock_hr_min
            | hr24clock_hr_min timezone_name
            | hr24clock_hour ":" minute
            | hr24clock_hour ":" minute timezone_name
            | wallclock_hr_min am_pm
            | wallclock_hr_min am_pm timezone_name
            | wallclock_hour ":" minute am_pm
            | wallclock_hour ":" minute am_pm timezone_name
            | "noon"
            | "midnight"
            ;

date        : month_name day_number
            | month_name day_number "," year_number
            | day_of_week
            | "today"
            | "tomorrow"
            ;

increment   : "+" inc_number inc_period
            | "next" inc_period
            ;

inc_period  : "minute" | "minutes"
            | "hour" | "hours"
            | "day" | "days"
            | "week" | "weeks"
            | "month" | "months"
            | "year" | "years"
            ;
```

STDIN top

The standard input shall be a text file consisting of commands acceptable to the shell command language described in *Chapter 2*, *Shell Command Language*. The standard input shall only be used if no -f *file* option is specified.

INPUT FILES top

See the STDIN section.

The text files at.allow and at.deny, which are located in an implementation-defined directory, shall contain zero or more user names, one per line, of users who are, respectively, authorized or denied access to the *at* and *batch* utilities.

ENVIRONMENT VARIABLES top

The following environment variables shall affect the execution of *at*:

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).

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*.

LC_TIME Determine the format and contents for date and time strings written and accepted by *at*.

SHELL Determine a name of a command interpreter to be used to invoke the at-job. If the variable is unset or null, *sh* shall be used. If it is set to a value other than a name for *sh*, the implementation shall do one of the following: use that shell; use *sh*; use the login shell from the user database; or any of the preceding accompanied by a warning diagnostic about which was chosen.

TZ Determine the timezone. The job shall be submitted for execution at the time specified by *timespec* or -t *time* relative to the timezone specified by the *TZ* variable. If *timespec* specifies a timezone, it shall override *TZ*. If *timespec* does not specify a timezone and *TZ* is unset or null, an unspecified default timezone shall be used.

ASYNCHRONOUS EVENTS [top]

Default.

STDOUT [top]

When standard input is a terminal, prompts of unspecified format for each line of the user input described in the STDIN section may be written to standard output.

In the POSIX locale, the following shall be written to the standard output for each job when jobs are listed in response to the -l option:

"%s\t%s\n", *at_job_id*, <*date*>

where *date* shall be equivalent in format to the output of:

date +"%a %b %e %T %Y"

The date and time written shall be adjusted so that they appear in the timezone of the user (as determined by the *TZ* variable).

STDERR [top]

In the POSIX locale, the following shall be written to standard error when a job has been successfully submitted:

"job %s at %s\n", *at_job_id*, <*date*>

where *date* has the same format as that described in the STDOUT section. Neither this, nor warning messages concerning the selection of the command interpreter, shall be considered a diagnostic that changes the exit status.

Diagnostic messages, if any, shall be written to standard error.

OUTPUT FILES [top]

None.

EXTENDED DESCRIPTION [top]

None.

EXIT STATUS [top]

The following exit values shall be returned:

0 The *at* utility successfully submitted, removed, or listed a job or jobs.

>0 An error occurred.

CONSEQUENCES OF ERRORS [top]

The job shall not be scheduled, removed, or listed.

The following sections are informative.

APPLICATION USAGE [top]

The format of the *at* command line shown here is guaranteed only for the POSIX locale. Other cultures may be supported with substantially different interfaces, although implementations are encouraged to provide comparable levels of functionality.

Since the commands run in a separate shell invocation, running in a separate process group with no controlling terminal, open file descriptors, traps, and priority inherited from the invoking environment are lost.

Some implementations do not allow substitution of different shells using *SHELL*. System V systems, for example, have used the login shell value for the user in /etc/passwd. To select reliably another command interpreter, the user must include it as part of the script, such as:

```
$ at 1800
myshell myscript
EOT
job ... at ...
$
```

EXAMPLES [top](#)

1. This sequence can be used at a terminal:

   ```
   at -m 0730 tomorrow
   sort < file >outfile
   EOT
   ```

2. This sequence, which demonstrates redirecting standard error to a pipe, is useful in a command procedure (the sequence of output redirection specifications is significant):

   ```
   at now + 1 hour <<!
   diff file1 file2 2>&1 >outfile | mailx mygroup
   !
   ```

3. To have a job reschedule itself, *at* can be invoked from within the at-job. For example, this daily processing script named my.daily runs every day (although *crontab* is a more appropriate vehicle for such work):

   ```
   # my.daily runs every day
   daily processing
   at now tomorrow < my.daily
   ```

4. The spacing of the three portions of the POSIX locale *timespec* is quite flexible as long as there are no ambiguities. Examples of various times and operand presentation include:

```
at 0815am Jan 24
at 8 :15amjan24
at now "+ 1day"
at 5 pm FRIday
at '17
    utc+
    30minutes'
```

RATIONALE [top](#)

The *at* utility reads from standard input the commands to be executed at a later time. It may be useful to redirect standard output and standard error within the specified commands.

The -t *time* option was added as a new capability to support an internationalized way of specifying a time for execution of the submitted job.

Early proposals added a ``jobname'' concept as a way of giving submitted jobs names that are meaningful to the user submitting them. The historical, system-specified *at_job_id* gives no indication of what the job is. Upon further reflection, it was decided that the benefit of this was not worth the change in historical interface. The *at* functionality is useful in simple environments, but in large or complex situations, the functionality provided by the Batch Services option is more suitable.

The -q option historically has been an undocumented option, used mainly by the *batch* utility.

The System V -m option was added to provide a method for informing users that an at-job had completed. Otherwise, users are only informed when output to standard error or standard output are not redirected.

The behavior of *at* <now> was changed in an early proposal from being unspecified to submitting a job for potentially immediate execution. Historical BSD *at* implementations support this. Historical System V implementations give an error in that case, but a change to the System V versions should have no backwards-compatibility ramifications.

On BSD-based systems, a -u *user* option has allowed those with appropriate privileges to access the work of other users. Since

this is primarily a system administration feature and is not universally implemented, it has been omitted. Similarly, a specification for the output format for a user with appropriate privileges viewing the queues of other users has been omitted.

The -f *file* option from System V is used instead of the BSD method of using the last operand as the pathname. The BSD method is ambiguous—does:

```
at 1200 friday
```

mean the same thing if there is a file named friday in the current directory?

The *at_job_id* is composed of a limited character set in historical practice, and it is mandated here to invalidate systems that might try using characters that require shell quoting or that could not be easily parsed by shell scripts.

The *at* utility varies between System V and BSD systems in the way timezones are used. On System V systems, the *TZ* variable affects the at-job submission times and the times displayed for the user. On BSD systems, *TZ* is not taken into account. The BSD behavior is easily achieved with the current specification. If the user wishes to have the timezone default to that of the system, they merely need to issue the *at* command immediately following an unsetting or null assignment to *TZ*. For example:

```
TZ= at noon ...
```

gives the desired BSD result.

While the *yacc*-like grammar specified in the OPERANDS section is lexically unambiguous with respect to the digit strings, a lexical analyzer would probably be written to look for and return digit strings in those cases. The parser could then check whether the digit string returned is a valid *day_number*, *year_number*, and so on, based on the context.

FUTURE DIRECTIONS top

None.

SEE ALSO top

batch(1p), crontab(1p)

The Base Definitions volume of POSIX.1-2017, *Chapter 8*, *Environment Variables*, *Section 12.2*, *Utility Syntax Guidelines*

COPYRIGHT top

Pages that refer to this page: batch(1p), crontab(1p)
