IEEE P1003.2a Draft 8 – December 1991

IEEE Standards Department
Copyright and Permissions
445 Hoes Lane, P.O. Box 1331
Piscataway, NJ 08855-1331, USA
+1 (908) 562-3800
+1 (908) 562-1571 [FAX]

5.2 `at` — Execute commands at a later time

5.2.1 Synopsis

`at [−m][−f` *file* `][−q` *queuename* `] −t` *time*

`at [−m][−f` *file* `][−q` *queuename* `]` *timespec* . . .

`at −r` *at_job_id* . . .

`at −l −q` *queuename*

`at −l [`*at_job_id* . . . `]`

5.2.2 Description

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

8

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 8
error are described under the `−m` option. 8

5.2.3 Options

The `at` utility shall conform to the utility argument syntax guidelines described in 2.10.2.

The following options shall be supported by the implementation:

`−f` *file* Specify the pathname of a file to be used as the source of the at- 8
job, instead of standard input. 8

`−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 via an implementation-defined mechanism, 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 the `batch` utility in 5.3.) The meanings of all other *queuenames* are implementation defined.

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

`-t` *time* Submit the job to be run at the time specified by the *time* option-argument, which shall have the format as specified by the `touch` (see 4.63) utility.

5.2.4 Operands

The following operands shall be supported by the implementation:

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 shall be concatenated into the equivalent of a single string with `<space>` separation added between the operands and this string shall be parsed as described in the grammar at the end of this subclause. The date and time shall be interpreted as being in the time zone of the user (as determined by the **TZ** variable), unless a time-zone 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- and two-digit numbers shall be taken to be hours, four-digit numbers to be hours and minutes. The time can alternately 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 time-zone name can also follow to further qualify the time. The acceptable time-zone names are implementation defined, except that they shall be case-insensitive and the string `"utc"` shall be supported to indicate the time is in Coordinated Universal Time. The *time* field can also be one of the following tokens in the POSIX Locale:

- `midnight` Indicates the time 12:00 am (00:00).
- `noon` Indicates the time 12:00 pm.
- `now` Indicate the current day and time. Invoking `at` <*now*> shall submit an at-job for potentially immediate execution (i.e., 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). Two special ‘‘days’’ shall be recognized in the POSIX Locale:

- `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 be also 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
```
8
8

The following grammar describes the precise format of *timespec* in the POSIX 7
Locale. The general conventions for this style of grammar are described in 2.1.2. 7
Any discrepancies found between this grammar and the preceding description 7
shall be resolved in favor of the grammar. The longest possible token or delimiter 8
shall be recognized at a given point. White space shall also delimit tokens. 8

```
%token hr24clock_hr_min
%token hr24clock_hour
/*
  A hr24clock_hr_min is a one, two, or four digit number.
  A one or two digit number constitutes a hr24clock_hour.
  A hr24clock_hour may be any of the single digits '0' - '9',
  or may be double digits, ranging from "00" - "23".
  If a hr24clock_hr_min is a four digit number, the first
  two digits must 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 or four digit number.
  A one 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 must 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 values 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 time zone suffix to the time            8
  field, in an implementation-defined format.                     8
*/
%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                                           8
              ;
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"
                                                                  8
```

```
                ;
date            : month_name day_number
                | month_name day_number "," year_number
                | day_of_week
                | "today"
                | "tomorrow"
                ;
increment       : "+" inc_number inc_period
                : "next" inc_period                          8
                ;
inc_period      : "minute" | "minutes"
                | "hour" | "hours"
                | "day" | "days"
                | "week" | "weeks"
                | "month" | "months"
                | "year" | "years"
                ;
```

5.2.5 External Influences

5.2.5.1 Standard Input

The standard input shall be a text file consisting of commands acceptable to the shell command language described in Section 3. The standard input shall only be 8 used if no `-f` *file* option is specified.

5.2.5.2 Input Files

See Standard Input.

5.2.5.3 Environment Variables

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

LANG This variable shall determine the locale to use for the locale categories when both **LC_ALL** and the corresponding environment variable (beginning with **LC_**) do not specify a locale. See 2.6.

LC_ALL This variable shall determine the locale to be used to override any values for locale categories specified by the settings of **LANG** or any environment variables beginning with **LC_**.

LC_CTYPE This variable shall determine the interpretation of sequences of bytes of text data as characters (e.g., single- versus multibyte characters in arguments and input files).

LC_MESSAGES This variable shall determine the language in which messages should be written.

LC_TIME This variable shall determine the the format and contents for date and time strings written and accepted by `at`.

SHELL This variable shall be interpreted as a name of a com- 7
mand interpreter to be used to invoke the at-job. If the 7
variable is unset or null, `sh` (see 4.56) shall be used. If it 7
is set to a value other than a name for `sh`, the implemen- 7
tation shall do one of the following: use that shell; use `sh`; 7
use the login shell from the user database; any of the 7
preceding accompanied by a warning diagnostic about 7
which was chosen. 7

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

5.2.5.4 Asynchronous Events

Default.

5.2.6 External Effects

5.2.6.1 Standard Output

When standard input is a terminal, prompts of unspecified format for each line of 8
the user input described in 5.2.5.1 may be written to standard output. 8

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"` 7

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

5.2.6.2 Standard Error

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>* shall have the same format as is described in Standard Output. 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.

5.2.6.3 Output Files

None.

5.2.7 Extended Description

None.

5.2.8 Exit Status

The `at` utility shall exit with one of the following values:

- 0 The `at` utility successfully submitted, removed, or listed a job or jobs.
- >0 An error occurred.

5.2.9 Consequences of Errors

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

5.2.10 Rationale. *(This subclause is not a part of P1003.2a)*

Usage, Examples

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 7
specified commands. 7

The format of the `at` command line shown here is guaranteed only for the POSIX 8
Locale. Other cultures may be supported with substantially different interfaces, 8
although implementations are encouraged to provide comparable levels of func- 8
tionality. 8

This sequence can be used at a terminal:

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

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
!
```

To have a job reschedule itself, `at` can be invoked from within the at-job.

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'
```

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 reliably select another command interpreter, the user must include it as part of the script, such as:

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

History of Decisions Made

Due to various implementations, and since the POSIX system administration group is dealing with administrative issues, the location or use of the files `at.allow` and `at.deny` were omitted. The following, if using these files, is a suggested implementation:

> Users shall be permitted to use `at` if their name appears in the file `at.allow`. If that file does not exist, the file `at.deny` shall be checked to determine if the user shall be denied access to `at`. If neither file exists, only root shall be allowed to submit a job. If only `at.deny` exists and is empty, global usage shall be permitted. The allow/deny files shall consist of one user name per line.

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 drafts 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. It is anticipated that considerably more sophisticated ways of controlling background or batch work will be the subject of future revisions to this standard.

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. This information is mailed to the user when the `−m` option is provided. When `−m` is not given, most systems mail the standard output and standard error to the user, but this is not required; some other method, such as a log file, could be used.

The behavior of `at` <*now*> was changed in an earlier draft 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 backward 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 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 could not be easily parsed by shell scripts.

The `at` utility varies between System V and BSD systems in the way time zones 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 time zone default to that of the system, she merely needs to issue the `at` command immediately following an unsetting or null assignment to **TZ**, For example,

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

will give the desired BSD result.

A previous draft indicated that various signals would be ignored. What it meant to say was that the at-job would not be affected by the various keyboard-related signals the user could initiate. The current wording about no controlling terminal should now suffice.

While the `yacc`-like grammar specified in 5.2.4 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`, etc., based on the context.

5.3 `batch` — Execute commands when the system load permits

5.3.1 Synopsis

```
batch
```

5.3.2 Description

The `batch` utility shall read commands to be executed at a later time. It shall be the equivalent of the command:

```
at -q b -m now
```
7

where queue `b` is a special `at` queue, specifically for batch jobs. Batch jobs shall be submitted to the batch queue with no time constraints and shall be run by the system using algorithms, based on unspecified factors, that may vary with each 7
invocation of `batch`.

5.3.3 Options

None.

5.3.4 Operands

None.

5.3.5 External Influences

5.3.5.1 Standard Input

See 5.2.

5.3.5.2 Input Files

None.

5.3.5.3 Environment Variables

See 5.2.