NAME

at, batch – execute commands at a later time

SYNOPSIS

at [**–c** | **–k** | **–s**] [**–m**] [**–f** *file*] [**–q** *queuename*] **–t** *time*
at [**–c** | **–k** | **–s**] [**–m**] [**–f** *file*] [**–q** *queuename*] *timespec*. . .
at –l [**–q** *queuename*] [*at_job_id*. . .]
at –r *at_job_id*. . .

batch

AVAILABILITY

at SUNWcsu

batch SUNWesu

DESCRIPTION

at

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

The at-job will 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 (see **umask**(1)), and system resource limits (for **sh** and **ksh** only, see **ulimit**(1)) in effect when the **at** utility is executed will be retained and used when the at-job is executed.

When the at-job is submitted, the *at_job_id* and scheduled time are written to standard error. The *at_job_id* is an identifier that will be a string consisting solely of alphanumeric characters and the period character. The *at_job_id* is 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 are permitted to use **at** and **batch** (see below) if their name appears in the file **/usr/lib/cron/at.allow**. If that file does not exist, the file **/usr/lib/cron/at.deny** is checked to determine if the user should be denied access to **at**. If neither file exists, only a process with the super-user privileges is allowed to submit a job. If only **at.deny** exists and is empty, global usage is permitted. The **at.allow** and **at.deny** files consist of one user name per line.

batch

The **batch** utility reads commands to be executed at a later time. It is the equivalent of the command:

at -q b -m now

where queue **b** is a special **at** queue, specifically for batch jobs. Batch jobs will be submitted to the batch queue for immediate execution.

OPTIONS

The following options are supported. If the **–c**, **–k**, or **–s** options are not specified, the **SHELL** environment variable by default determines which shell to use.

–c C shell. **csh**(1) is used to execute the at-job.

–k Korn shell. **ksh**(1) is used to execute the at-job.

–s Bourne shell. **sh**(1) is used to execute the at-job.

–f *file* Specify the path 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.

–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 will be mailed to the user as well, unless redirected elsewhere. Mail will be sent even if the job produces no output.

If –**m** is not used, the job's standard output and standard error will be provided to the user by means of mail, unless they are redirected elsewhere; if there is no such output to provide, the user is not notified 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. Values for *queuename* are limited to the lower case letters **a** through **z**. By default, at-jobs will be scheduled in queue **a**. In contrast, queue **b** is reserved for batch jobs. Since queue **c** is reserved for cron jobs, it can not be used with the –**q** option.

–r *at_job_id* 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 must have the format as specified by the **touch**(1) utility.

OPERANDS

The following operands are 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. The date and time are 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 "C" locale, the following describes the three parts of the time specification string. All of the values from the **LC_TIME** categories in the "C" locale are recognized in a case-insensitive manner.

time The *time* can be specified as one, two or four digits. One- and two-digit numbers are 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 is understood. A timezone name can follow to further qualify the time; see **TZ** on the **environ**(5) manual page. The *time* field can also be one of the following tokens in the "C" 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** will 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). Two special days are recognized in the "C" locale:

today Indicates the current day.

tomorrow Indicates the day following the current day.

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

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

at 2pm + 1 week
at 2pm next week

USAGE

The format of the **at** command line shown here is guaranteed only for the "C" locale. Other locales are not supported for **midnight**, **noon**, **now**, **mon**, **abmon**, **day**, **abday**, **today**, **tomorrow**, **minutes**, **hours**, **days**, **weeks**, **months**, **years**, and **next**.

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.

EXAMPLES

at

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** will run every day (although **crontab** is a more appropriate vehicle for such work):
   # my.daily runs every day
   at now tomorrow < my.daily
   daily-processing
4. The spacing of the three portions of the "C" locale *timespec* is quite flexible as long as there are no ambiguities. Examples of various times and operand presentations include:
   at 0815am Jan 24
   at 8 :15amjan24
   at now "+ 1day"
   at 5 pm FRIday
   at '17
   utc+
   30minutes'

batch

1. This sequence can be used at a terminal:
   $ batch
   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):
   $ batch <<!
   diff file1 file2 2>&1 >outfile | mailx mygroup
   !

ENVIRONMENT

See **environ**(5) for descriptions of the following environment variables that affect the execution of **at** and **batch**: **LC_CTYPE**, **LC_MESSAGES**, **NLSPATH**, and **LC_TIME**.

SHELL Determine a name of a command interpreter to be used to invoke the at-job. If the variable is unset or **NULL**, **sh** will be used. If it is set to a value other than **sh**, the implementation will use that shell; a warning diagnostic will be printed telling which shell will be used.

TZ Determine the timezone. The job will 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 will override **TZ**. If *timespec* does not specify a timezone and **TZ** is unset or **NULL**, an unspecified default timezone will be used.

DATEMSK If the environment variable **DATEMSK** is set, **at** will use its value as the full path name of a template file containing format strings. The strings consist of format specifiers and text characters that are used to provide a richer set of allowable date formats in different languages by appropriate settings of the environment variable **LANG** or **LC_TIME**. The list of allowable format specifiers is located in the **getdate**(3C) manual page. The formats described in the **OPERANDS** section for the *time* and *date* arguments, the special names **noon**, **midnight**, **now**, **next**, **today**, **tomorrow**, and the *increment* argument are not recognized when **DATEMSK** is set.

EXIT STATUS

The following exit statuses are returned:

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

>0 An error occurred, and the job will not be scheduled.

FILES

/usr/lib/cron/at.allow names of users, one per line, who are authorized access to the **at** and **batch** utilities

/usr/lib/cron/at.deny names of users, one per line, who are denied access to the **at** and **batch** utilities

SEE ALSO

crontab(1), **csh**(1), **date**(1), **ksh**(1), **sh**(1), **touch**(1), **ulimit**(1), **umask**(1), **getdate**(3C), **environ**(5)

NOTES

Regardless of queue used, cron has a limit of 100 jobs in execution at any time.

There can be delays in cron at job execution. In some cases, these delays can compound to the point that cron job processing appears to be hung. All jobs will be executed eventually. When the delays are excessive, the only workaround is to kill and restart cron.