**Linux at Command and crontab**

**NAME**

at, batch, atq, atrm - queue, examine or delete jobs for later execu-

tion

**SYNOPSIS**

at [-V] [-q queue] [-f file] [-mldbv] TIME

at [-V] [-q queue] [-f file] [-mldbv] -t time\_arg

at -c job [job...]

atq [-V] [-q queue]

atrm [-V] job [job...]

batch [-V] [-q queue] [-f file] [-mv] [TIME]

**DESCRIPTION**

at and batch read commands from standard input or a specified file

which are to be executed at a later time.

at executes commands at a specified time.

atq lists the user’s pending jobs, unless the user is the supe-

ruser; in that case, everybody’s jobs are listed. The format

of the output lines (one for each job) is: Job number, date,

hour, job class.

atrm deletes jobs, identified by their job number.

batch executes commands when system load levels permit; in other

words, when the load average drops below 0.8, or the value

specified in the invocation of atrun.

At allows fairly complex time specifications, extending the POSIX.2

standard. It accepts times of the form HH:MM to run a job at a spe

cific time of day. (If that time is already past, the next day is

assumed.) You may also specify **midnight**, **noon**, or **teatime** (4 p.m.) and

you can have a time-of-day suffixed with AM or PM for running in the

morning or the evening. You can also say what day the job will be run,

by giving a date in the form month-name day with an optional year, or

giving a date of the form MMDDYY or MM/DD/YY or DD.MM.YY. The specifi-

cation of a date must follow the specification of the time of day. You

can also give times like now + count time-units, where the time-units

can be minutes, hours, days, or weeks and you can tell at to run the

job today by suffixing the time with today and to run the job tomorrow

by suffixing the time with tomorrow.

For example, to run a job at 4 p.m. three days from now, you would do at

4pm + 3 days. To run a job at 10 a.m. on July 31, you would do at 10am

Jul 31. And to run a job at 1 a.m. tomorrow, you would do at 1am tomorrow.

The exact definition of the time specification can be found in

/usr/share/doc/at-3.1.8/timespec.

For both at and batch, commands are read from standard input or the

file specified with the -f option and executed. The working directory,

the environment (except for the variables TERM, DISPLAY and \_) and the

umask are retained from the time of invocation. An at - or batch -

command invoked from a su(1) shell will retain the current userid. The

user will be mailed standard error and standard output from his com-

mands, if any. Mail will be sent using the command /usr/sbin/sendmail.

If at is executed from a su(1) shell, the owner of the login shell will

receive the mail.

The superuser may use these commands in any case. For other users,

permission to use at is determined by the files /etc/at.allow and

/etc/at.deny.

If the file /etc/at.allow exists, only usernames mentioned in it are

allowed to use at.

If /etc/at.allow does not exist, /etc/at.deny is checked, every user-

name not mentioned in it is then allowed to use at.

If neither exists, only the superuser is allowed use of at.

An empty /etc/at.deny means that every user is allowed use these com-

mands, this is the default configuration.

**OPTIONS**

-V prints the version number to standard error.

-q queue

uses the specified queue. A queue designation consists of a

single letter; valid queue designations range from a to z. and

A to Z. The a queue is the default for at and the b queue for

batch. Queues with higher letters run with increased niceness.

The special queue "=" is reserved for jobs which are currently

running.

If a job is submitted to a queue designated with an uppercase letter,

it is treated as if it had been submitted to batch at that time. If

atq is given a specific queue, it will only show jobs pending in that

queue.

-m Send mail to the user when the job has completed even if there

was no output.

-f file Reads the job from file rather than standard input.

-l Is an alias for atq.

-d Is an alias for atrm.

-v Shows the time the job will be executed.

Times displayed will be in the format "1997-02-20 14:50" unless

the environment variable POSIXLY\_CORRECT is set; then, it will

be "Thu Feb 20 14:50:00 1997".

-c cats the jobs listed on the command line to standard out-

put.

-t time\_arg

Submit the job to be run at the time specified by the

time\_arg option argument, which must have the same format

as specified for the touch(1) utility’s -t time option

argument ([[CC]YY]MMDDhhmm).

#### Examples

**at -m 01:35 < atjob =** Run the commands listed in the 'atjob' file at 1:35AM, in addition all output that is generated from job mail to the user running the task. When this command has been successfully enter you should receive a prompt similar to the below example.

commands will be executed using /bin/csh  
job 1072250520.a at Wed Dec 24 00:22:00 2003

**at -l =** This command will list each of the scheduled jobs as seen below.

1072250520.a Wed Dec 24 00:22:00 2003

**at -r 1072250520.a** = Deletes the job just created.

or

**atrm 23** = Deletes job 23.

If you wish to create a job that is repeated, you could modify the file that executes the commands with another command that recreates the job or, better yet, use the [crontab command](http://www.computerhope.com/unix/ucrontab.htm).

**Note:** Performing just the **at** command at the prompt will give you an error "Garbled Time"; this is a standard error message if no switch or time setting is given.