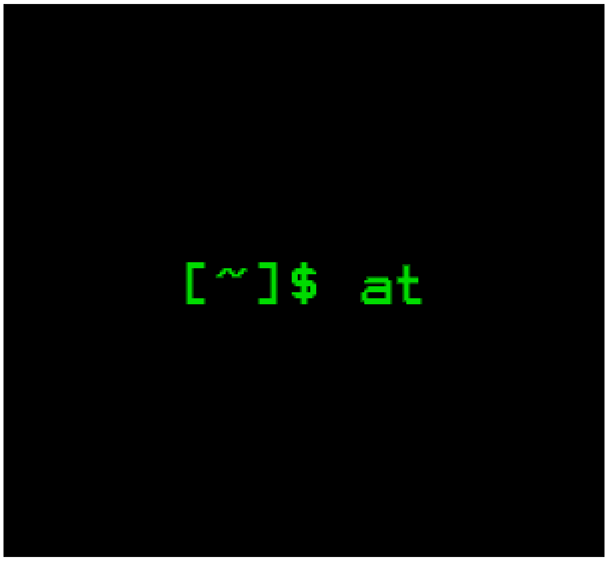
Home Help Linux and Unix

Linux at, batch, atq, and atrm commands

Updated: 12/29/2017 by Computer Hope



```
[~]$ at
```

ComputerHope.com

About at, batch, atq, atrm

The **at** command schedules a command to be run once at a particular time that you normally have permission to run. The at command can be anything from a simple reminder message, to a complex script. You start by running the **at** command at the command line, passing it the scheduled time as the option. It then places you at a special prompt, where you can type in the command (or series of commands) to be run at the scheduled time. When you're done, press **Control-D** on a new line, and your command will be placed in the queue.

A typical **at** command sequence looks like this (commands you type are shown here in the blue box, or in bold face below):

```
at 9:30 PM Tue
```

```
warning: commands will be executed using /bin/sh
at> echo "Well gosh golly, it's 9:30 PM on Tuesday."
at> ^D
job 1 at Tue Nov 16 09:30:00 2014
```

So let's take a look at what we see here. When we ran the command, the first thing **at** did was give us a "warning" telling us what command shell our commands will be run with: in this case, **/bin/sh**, the Bourne Shell. This shell is the traditional standard Unix shell.

It then places us at the **at>** prompt. Here we type in a simple **echo** command, which echoes a string of text. We press enter, and we're placed at a new **at>** prompt. We then press **Control-D**, telling **at** we're all done with our commands. It then tells us that our job is job number **1** and that it will run next Tuesday.

Note: The output of your specified command will be mailed to you. You can read this mail with the **mail** program, or a program like **pine** (or the modern version of pine, called **alpine**). You can download these programs if you don't have them, or install them with your package manager; for example, on Ubuntu, which uses the APT package management system, you can install them using the **apt-get**command, specifically: **sudo apt-get install mail** or **sudo apt-get install alpine**.

Specifying Time

at uses a very casual representation of time and date. It even knows some "commonly used" times you might not expect — it knows that "teatime" is traditionally at 4 PM, for instance.

Here are some examples of times you can pass to **at** to schedule a command. For instance, let's assume the current time is 10:00 AM, Tuesday, October 18, 2014. The following expressions would translate to the following times:

the expression:	would translate to:
noon	12:00 PM October 18 2014
midnight	12:00 AM October 19 2014

teatime	4:00 PM October 18 2014
tomorrow	10:00 AM October 19 2014
noon tomorrow	12:00 PM October 19 2014
next week	10:00 AM October 25 2014
next monday	10:00 AM October 24 2014
fri	10:00 AM October 21 2014
NOV	10:00 AM November 18 2014
9:00 AM	9:00 AM October 19 2014
2:30 PM	2:30 PM October 18 2014
1430	2:30 PM October 18 2014
2:30 PM tomorrow	2:30 PM October 19 2014

2:30 PM next month	2:30 PM November 18 2014
2:30 PM Fri	2:30 PM October 21 2014
2:30 PM 10/21	2:30 PM October 21 2014
2:30 PM Oct 21	2:30 PM October 21 2014
2:30 PM 10/21/2014	2:30 PM October 21 2014
2:30 PM 21.10.14	2:30 PM October 21 2014
now + 30 minutes	10:30 AM October 18 2014
now + 1 hour	11:00 AM October 18 2014
now + 2 days	10:00 AM October 20 2014
4 PM + 2 days	4:00 PM October 20 2014
now + 3 weeks	10:00 AM November 8 2014
now + 4 months	10:00 AM February 18 2015

now + 5 years	10:00 AM October 18 2019

...so if you run the command:

```
at now + 10 years
```

...and then enter a command at the **at>** prompt, press enter, and type **Control-D**, you will be mailed the results of your command ten years from now.

Note: If you don't specify a time at the command line, **at**will return the following error message:

```
Garbled time
```

...and no job will be added to the queue. So, always specify your time at the command line.

Using atq To View Your at Queue

You can use the program **atq** to view your currently-queued **at** jobs. Type **atq** to display the queue.

```
atq
```

```
1      Fri Oct 22 09:48:00 2014 a hope
```

This information is, from left to right: **job number**, **date**, **hour**, **year**, **queue**, and **username**.

atq will only list jobs that belong to you — unless you are the super user, in which case it will list the jobs of all users. So to list all **at** jobs currently queued on the system, type this command (if you have superuser privileges):

```
sudo atq
```

...and type your password, when prompted.

at, batch, atq, and atrm syntax

```
at [-V] [-q queue] [-f file] [-mMlv] timespec...
```

```
at [-V] [-q queue] [-f file] [-mMkv] [-t time]
```

```
at -c job [job...]
```

```
atq [-V] [-q queue]
```

```
at [-rd] job [job...]
```

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

```
batch
```

```
at -b
```

Technical Description

at and **batch** read commands from standard input or a specified file that are to be executed at a later time, using sh.

at executes commands at a specified time.

atq lists the user's pending jobs, unless the user is the superuser; in that case, everybody's jobs are listed. The format of the output lines (one for each job) is: **job number**, **date**, **hour**, **year**, **queue**, and **username**.

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 1.5, or the value specified in the invocation of **atd**.

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 specific time of day. (If that time is already past, the next day is assumed.) You may also specify **midnight**, **noon**, or **teatime** (4pm) 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 **MMDD[CC]YY**, **MM/DD/[CC]YY**, **DD.MM.[CC]YY** or **[CC]YY-MM-DD**. The specification 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 4pm three days from now, you would do at **4pm + 3 days**, to run a job at 10:00am on July 31, you would do at **10am Jul 31** and to run a job at 1am tomorrow, you would do at **1am tomorrow**.

The definition of the time specification can be found in **/usr/share/doc/at/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 **BASH_VERSINFO**, **DISPLAY**, **EUID**, **GROUPS**, **SHELLOPTS**, **TERM**, **UID**, and **_**) and the umask are retained from the time of invocation.

As at is currently implemented as a setuid program, other environment variables (e.g., **LD_LIBRARY_PATH** or **LD_PRELOAD**) are also not exported. This may change in the future. As a workaround, set these variables explicitly in your job.

An **at** or **batch** command run from a **su** shell will retain the current userid. The user will be mailed standard error and standard output from his commands, if any. Mail will be sent using the command **/usr/sbin/sendmail**. If at is executed from a **su** shell, the owner of the login shell will receive the mail.

The superuser may always use these commands. For other users, permission to use **at** is determined by the files **/etc/at.allow** and **/etc/at.deny**. See **at.allow** for details.

Options

-V	Prints the version number to standard error and exits successfully.
-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 that are currently running.

If a job is submitted to a queue designated with an uppercase letter, the job is treated as if it were submitted to batch at the time of the job. Once the time is reached, the batch processing rules with respect to load average apply. 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.

-M Never send mail to the user. In other words, execute the command, but do not notify the user of its output.

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

-t*time* Run the job at *time*, given in the format [[*CC*]*YY*]*MMDDhhmm*[**.***ss*].

-l Running **at -l** is the same as running **atq**; it displays all queued **at** jobs.

-r Is the same as running **atrm**. It removes a job from the **at** queue.

-d Is also an alias for **atrm**.

-b Is an alias for **batch**.

-v Shows the time the job will be executed before reading the job. Times displayed will be in the format "Thu Feb 20 14:50:00 1997".

-c The **cats** the jobs listed on the command line to standard output.

Files

/var/spool/cron/atjobs
/var/spool/cron/atspool
/proc/loadavg
/var/run/utmp
/etc/at.allow
/etc/at.deny

at, batch, atq, and atrm examples

```
at -m 01:35 < my-at-jobs.txt
```

Run the commands listed in the '**my-at-jobs.txt**' file at **1:35**AM. All output from the job will be mailed to the user running the task. When this command has been successfully entered you should receive a prompt similar to the example below:

```
commands will be executed using /bin/sh
job 1 at Wed Dec 24 00:22:00 2014
```

```
at -l
```

This command will list each of the scheduled jobs in a format like the following:

```
1          Wed Dec 24 00:22:00 2003
```

...this is the same as running the command **atq**.

```
at -r 1
```

Deletes job **1**. This command is the same as running the command **atrm 1**.

```
atrm 23
```

Deletes job 23. This command is the same as running the command **at -r 23**.

Related commands

crontab — View and edit a list of jobs for the system to run at regular intervals.