

About The Shell

About The Shell

The shell (or "command line") allows you to interact with Athena without using the mouse. Most tasks on Athena can be accomplished either by using the mouse and the graphical environment, or by using the shell, but some advanced tasks require shell commands.

The Shell

The shell is a "command interpreter". This means that the shell serves as an intermediary between you and the operating system: you type a command name, and the shell interprets what you have entered and submits it to the operating system in a full form; when the program finishes, the shell comes back and asks for the next command (with another `joeuser@athena:~$` prompt).

Having a shell allows you to enter sophisticated commands using shorthand notation, and lets you edit commands and recall previous commands. (The operating system by itself just waits for the final version to be passed to it from the shell.)

On Athena, you can choose from among several shells with different abilities. The two primary types of shells are the [Bourne shell](#) and the [C shell](#). By default, Athena users automatically use a variant of the Bourne shell called **bash**. Athena accounts created prior to April 2009 will have a different shell called **tcsh** (a variant of the C shell) as their default shell.

The two main types of shells – Bourne and C – use very different syntax. In this document, Bourne shell syntax will be used by default, since the default shell is **bash**. Where Bourne and C shell syntax differ, it will be noted.

Identifying your shell

As noted above, if you registered for your Athena account in April 2009 or later, you likely have Bash as your default shell. If you want to check what your shell is, see: [Q: How can I figure out what my default Athena shell is?](#)

The Shell Prompt

The shell prompt is how the shell lets you know it's ready for input. The default shell prompt (for Bash users) looks like this:

```
| joeuser@w20-575-1:~$
```

The first part of the prompt (`joeuser@w20-575-1`) includes your username (`joeuser`) and the machine you're currently logged in to (`w20-575-1`). The part after the colon specifies the current *working directory* of the shell. The tilde (`~`) character is a shorthand notation for your home directory (`/mit/joeuser`). Finally, the dollar sign is the end of the prompt. The dollar sign historically signifies that you're using a variant of the Bourne shell.

Tcsh users will see a prompt like this:

```
| w20-575-1:~>
```

Unlike Bash, the default tcsh prompt does not contain your username, but it does contain the host name of the machine you're currently logged in to. It also contains your current working directory, but ends with a greater-than sign (`>`) instead of a dollar sign.

The "Athena Prompt"

In older documentation, you may see references to the "Athena prompt", or `joeuser@athena:~$`. The old shell prompts simply had the word "athena" followed by a dollar sign for Bash users or a percent sign (%) for Tcsh users. Although the prompts may look different, they should work like any other shell prompt.

Command History

The shell remembers each command you have typed so you can easily run previous commands. This can be particularly helpful if you're running the same (or similar) commands on a large number of files. You can easily access the history by pressing the "up" arrow key at your shell prompt. You will see the previous command displayed there. You can press "Enter" to run the command as is, or you can edit the command. To edit the command, use the left or right arrow keys to move the cursor to the part of the command you want to change, and delete and replace text as you would in any word processing program.

You can also refer to previous commands numerically. The **history** command will show the last 500 commands typed as well as their numbers (generated sequentially). As 500 is somewhat of an unwieldy number, you can tell the **history** command how many previous commands to

display by specifying a number. For example:

```
joeuser@athena$ history 5
510 ls
511 cp file 1 file2
512 cd $HOME
513 zlocate jruser
514 history 5
```

The left column contains the command number and the right column contains the command. You can run a previous command by typing an exclamation point (!) and the command number. Continuing from the example above, if you wanted to run command 514 again (**zlocate jruser**), you could type the following:

```
joeuser@athena$ !514
Hidden or not logged-in
joeuser@athena$
```

Additional history commands are available:

!! Run the previous command

!-n Run the nth most recent command

!str Run the most recent command that began with the letters *str*

^str1^str2 Run the previous command, but replace *str1* with *str2* (e.g., if the last command you typed was **ls Public** and you then typed

^Public^Private the shell would execute the command **ls Private**. See the [ps command](#).