Johns Hopkins University Center for Talented Youth

Windows Command Prompt Tutorial

Introduction

This document presents a brief tutorial describing how to use some basic features of the Windows operating system command prompt program. You can use these features to compile and run Java programs and manipulate the location of files on your computer's file system. The lectures in this course illustrate how to compile and run Java programs from the command prompt, but how to do this will be repeated here just so everything you need is contained in this tutorial.

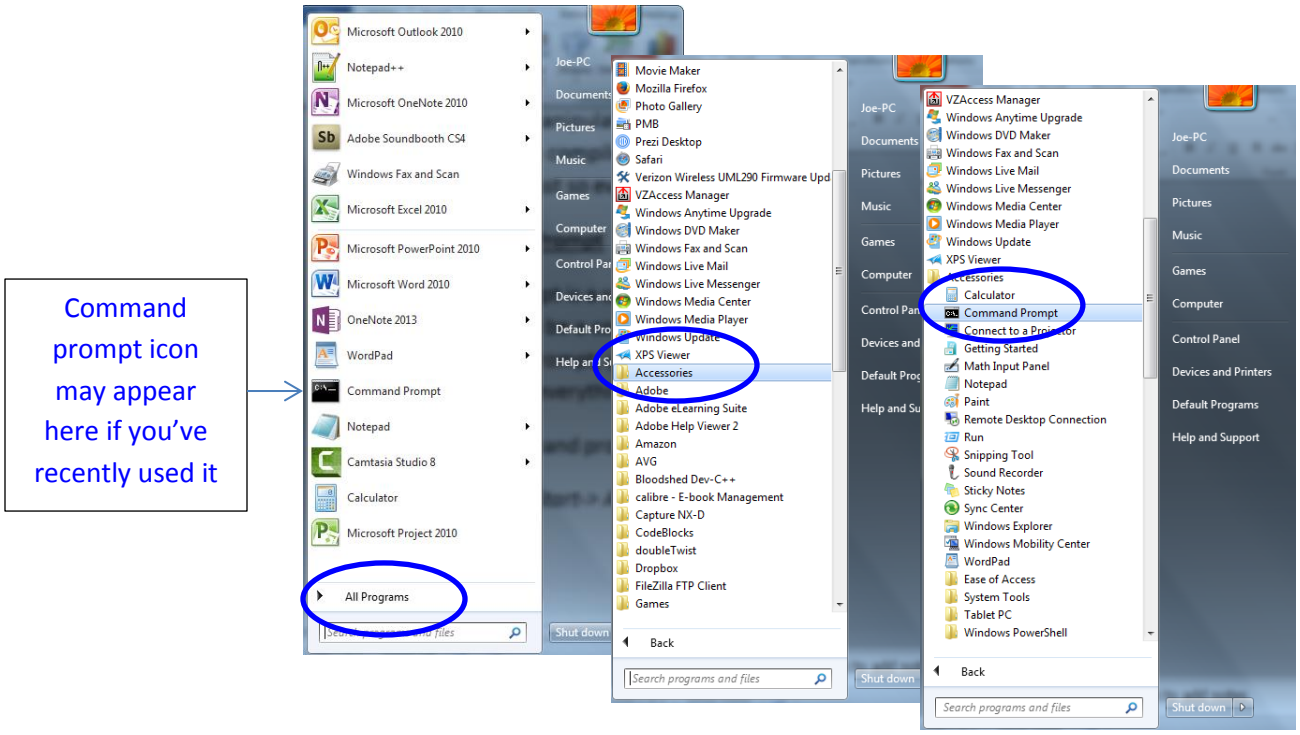
The Command Prompt

Command prompt is a special program that allows you to work on a Windows-based computer from what appears to be a command line instead of using the Windows graphical user interface. When using the command prompt, you enter a command, and the computer executes it. And…you use your keyboard to do everything rather than your mouse.

To launch command prompt, do the following:

Select  -> *All Programs* -> *Accessories* and click on the *Command Prompt* selection

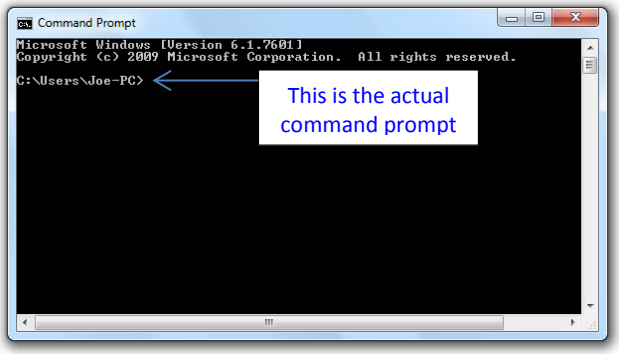
The  icon is located at the lower left corner of your video display screen. If you have recently used the command prompt, an icon will appear on the list of recent applications and you won't have to follow the more lengthy path of actions listed above.



Johns Hopkins University Center for Talented Youth

Windows Command Prompt Tutorial

A command prompt window with a black background will then open up, as illustrated below.



The **`C:\Users\Joe-PC>`** in the above window is called the *command prompt* or *prompt string*. The command prompt string on your computer will look something like the one above, depending upon the name of your computer. At the very minimum, the command string will be **`C:\>`**

The **`C:`** corresponds to the identifier for your computer's hard disk drive, and the backward slash, **`\`**, is a symbol for what is called the *root directory*. The root directory represents the very top of the file system on the hard drive.

The command prompt string (actually, the part to the left of the **>**) tells you where you are "sitting" with respect to the file system. Where you are "sitting" is called your *current directory*, or *current working directory*. The root directory is at the very top of the file system, and all other directories (called subdirectories) and files on the file system are located below it, creating a hierarchy. In the above command prompt window, the command prompt string is **`C:\Users\Joe-PC`**. Each backward slash after the root corresponds to a subdirectory.

By typing in various commands after the command prompt string you can do things like find out what files and subdirectories are located under the current working directory and you can logically "move around" the file system by changing the current working directory.

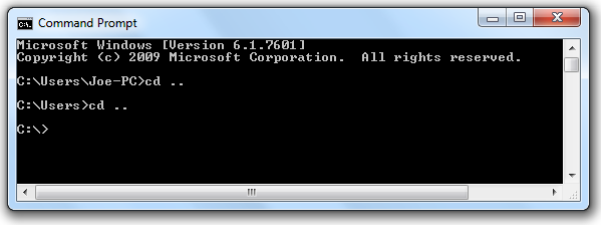
I've used the terms *file* and *subdirectory* above, but I didn't define what they mean, so let me do that now. A *file* is anything stored on the computer's file system. It can be a document created by a word processor; it can be a Java source file; or a Java compiled .class file; or…anything else. A *subdirectory*, also referred to as a *folder*, is a special type of file that can have other files or other subdirectories stored under it. When you view your computer's file system using the Windows visual features, subdirectories are shown as folder icons  . The conceptual idea is that a folder can be used to store stuff inside itself, namely, other folders and files.

Johns Hopkins University Center for Talented Youth

Windows Command Prompt Tutorial

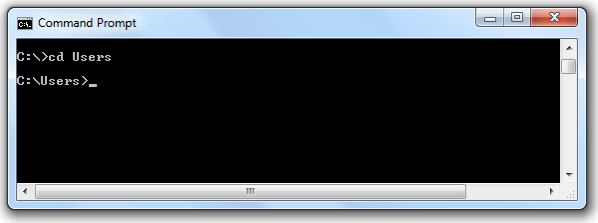
Changing the Current Directory

To change the current working directory and logically "move around" the file system, you use the **cd** command. The **cd** stands for *change directory*. You just type **cd** followed by the path name to another location on the file system or by a special character string (**..**) that is two periods. Let's look at an example.

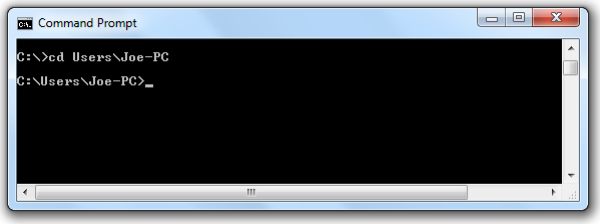


In the above example, I typed **cd**, a blank space, and then two periods (**..**). The two periods are special notation that means the *parent* of the current directory. The parent directory of the **Joe-PC** directory is the **Users** directory, so the first cd command changes my current directory to the **Users** directory. Note that the prompt string indicates the current directory is **C:\Users**. I then repeated the same command and changed the current directory to the root directory (****).

Now, let's suppose I want to change my current directory back to **C:\Users**. Since the **Users** directory is immediately under the root directory, I just use the name **Users** as the parameter of the **cd** command, as illustrated below.



As a different example, let's suppose my current directory was the root directory and I wanted to change it to **C:\Users\Joe-PC**. Then, I simply use the name **Users\Joe-PC** as the parameter of the **cd** command, as illustrated below.

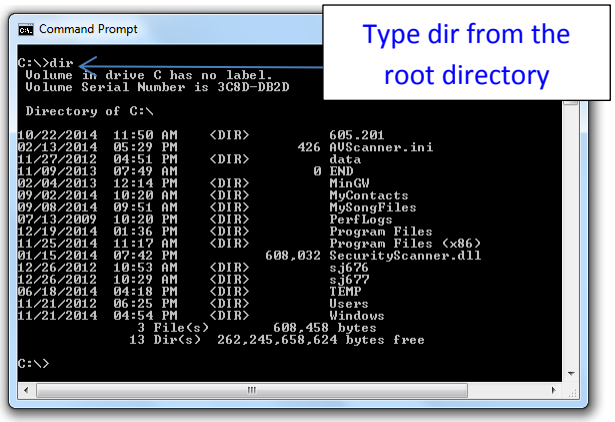


Johns Hopkins University Center for Talented Youth

Windows Command Prompt Tutorial

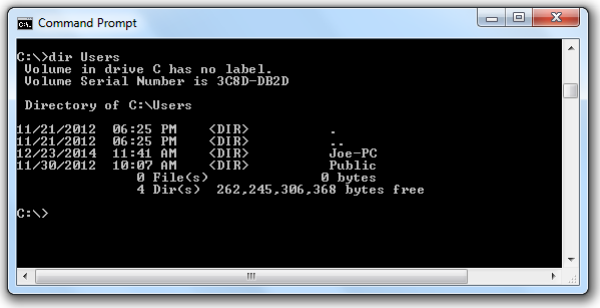
Viewing the Contents of a Directory

When we use the command prompt window to navigate the file system, we often need to know what files are stored in a particular directory. We can use the **dir** command to do that. Using dir by itself will display the contents of the current directory. Using **dir** with the name of a subdirectory will display the contents of that subdirectory. Let's see what kinds of things are stored directly under my root directory.



Using the **cd** command, I made the root directory my current directory. Then, I simply typed **dir** followed by the enter key to display the contents of the root directory. As you can see, there are 13 directories directly under my root directory and three ordinary files. The **dir** command also provides date and time information, and for ordinary files provides file sizes.

Now, let's suppose we want to find out what was stored in one of the subdirectories. We can do that by using the **dir** command with the name of the directory we are interested in, as illustrated below.



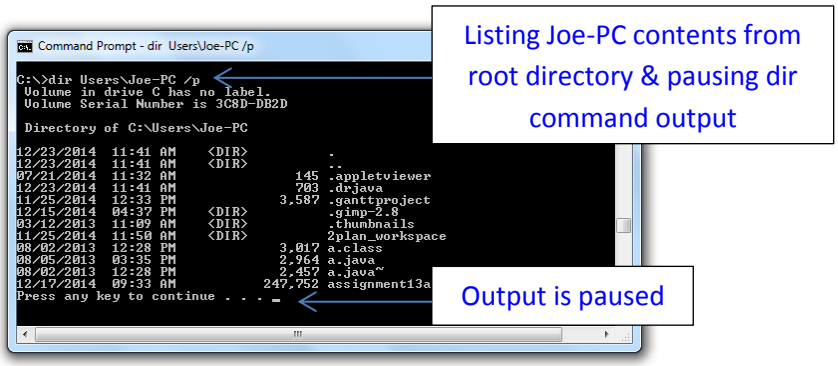
In the above example, I was interested in the contents of the **Users** subdirectory. Note that there are two subdirectories, **Joe-PC**, and **Public**, in the **Users** subdirectory. The **dir** command also shows the **Users** subdirectory itself, represented by a single period, and the parent directory, represented by two periods…but, in practice we never really care about these…they're just how the **dir** command was designed.

Johns Hopkins University Center for Talented Youth

Windows Command Prompt Tutorial

Now, let's suppose I want to see the contents of the **Joe-PC** subdirectory of **Users**. I have several ways of doing that. One way would be to use the **cd** command and make **Joe-PC** the current directory, and then use the **dir** command without any parameters. Another way would be to make **Users** the current directory and use the **dir** command with **Joe-PC** as its parameter. Yet another way is to keep the current directory the root directory, use the **dir** command, and just supply it with the path name to that subdirectory as a parameter, which is **Users\Joe-PC**.

There's another useful parameter associated with the **dir** command. The **/p** parameter causes the output to pause after displaying about a dozen file or directory names, so that you can look at them. This can be helpful if there are lots and lots of files in a particular directory. The example below illustrates listing the contents of the **Joe-PC** directory while the current directory is the root directory, and it also illustrates using the **/p** parameter to pause the **dir** command output.

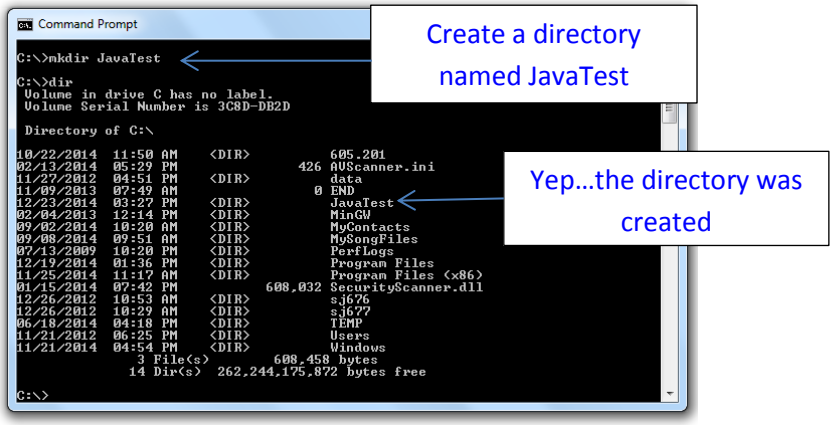


Creating a Directory

Another thing you can do with command prompt is to create a directory. To do this we use the **mkdir** command. In the example below, I create a directory named **JavaTest** under the root directory, and then I use the **dir** command to verify that the directory has, in fact, been created.

Johns Hopkins University Center for Talented Youth

Windows Command Prompt Tutorial



Copying and Moving Files

Two more handy commands are the **`copy`** command and the **`move`** command. The **`copy`** command makes a *copy* of an existing file, and the **`move`** command *moves* an existing file to another location on the file system.

The syntax of the **`copy`** command is : **`copy source_file destination_file`**

The **`destination_file`** in the above syntax can be an ordinary file name or the name of a directory. If it is an ordinary file name then the copy of **`source_file`** will be named **`destination_file`**. If it is the name of a directory, then a copy of **`source_file`** will be made and stored under the **`destination_file`** directory.

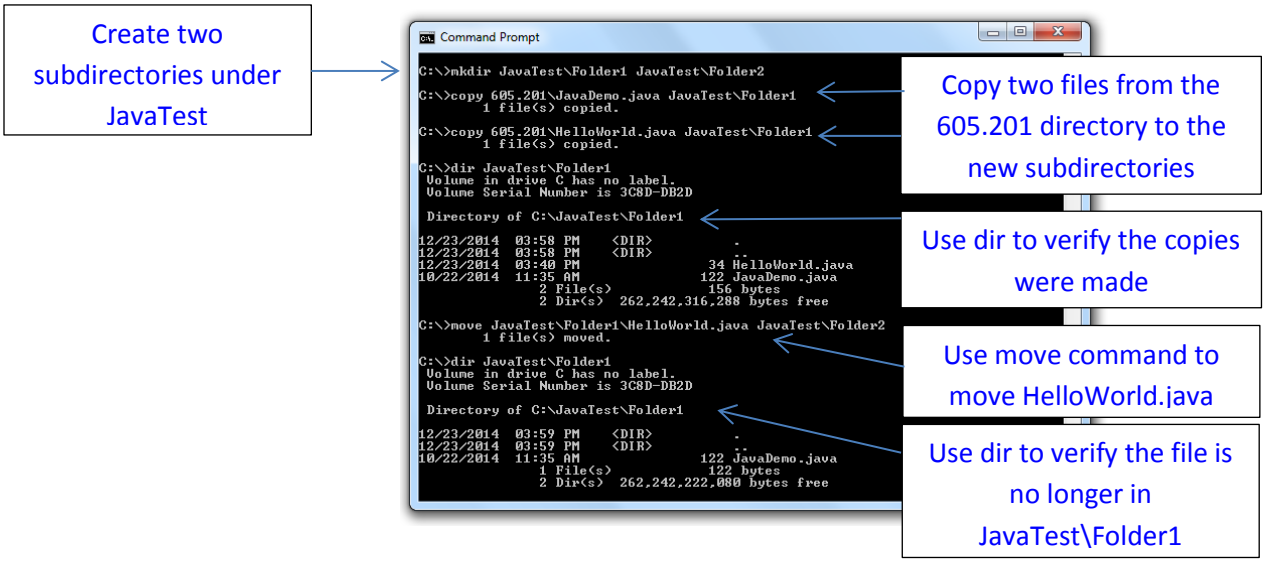
The syntax of the **`move`** command is: **`move source_file destination_directory`**

The **`source_file`** will be moved to the **`destination_directory`**.

In the example below, I demonstrate several things. First, I will create two subdirectories under the **`JavaTest`** directory. Then, I will copy two files into one of those subdirectories. Finally, I will move one file from one subdirectory to the other.

Johns Hopkins University Center for Talented Youth

Windows Command Prompt Tutorial



Closing the Command Prompt Window

To close the Command Prompt window, you can type the **`exit`** command or just click on the 'X' in the upper right-hand corner of the window.