

# Absolute and Relative Paths

## Hierarchical File System

A Hierarchical file system is a file system that organizes data and program files in a top-to-bottom structure. All modern operating systems use hierarchical file systems, wherein access to the data starts at the top and proceeds downward throughout the folders in the hierarchy. The windows hierarchical structure is conceptualized as an upside-down tree with the starting point at the top, which is the root.

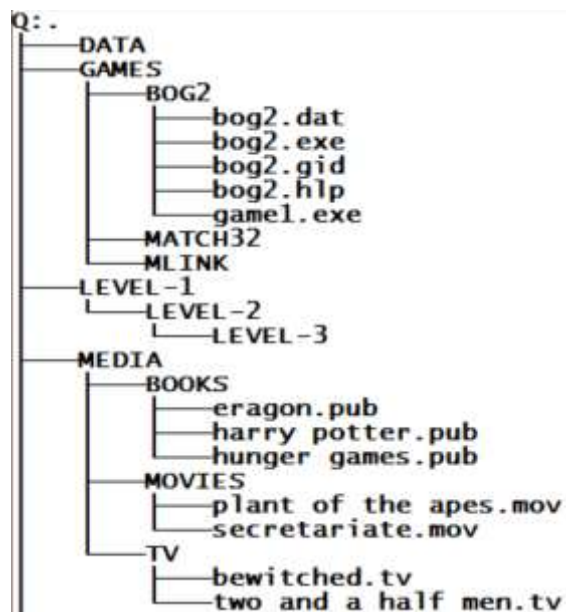
In a graphical interface, you'd access the root of a the drive through "Computer" or "This PC". You would start at the drive and then navigate through the folders using your mouse. At the command prompt you must type the path to the file or folder.

A *path* is the sequence of directories that must be traversed through to get from one file or directory to another. In order to work with files and folders effectively at the command prompt, it is important to know how to write a path in both absolute and relative form.

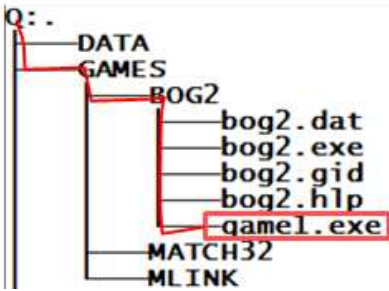
(Note: A folder in the graphical interface is often referred to as a directory at the command prompt. Just a heads up, I tend to use the terms interchangeably.)

**Absolute path** always exactly identifies the location of a file or folder in the file system. It *always* begins with a drive letter, a colon, and a backslash (\). If the path you're writing goes beyond the root you'd include, after the backslash, a list of the directories you must pass through to get to the file or folder in question. Backslashes are used to separate directory and file names in the path. No additional spaces can be included. If a space is part of a folder's name or a file's name, quotes must surround the name.

The below image is a command prompt view of the hierarchical folder structure of a drive that's been assigned the letter Q:. Anything presented in upper case is a folder. Anything presented in lower case is a file. I should note that this is not a standard or a requirement; it's simply how I created this structure to be able to easily identify a file or folder in the image. I will be using this folder structure in some of the examples within this document.



Examples of Absolute paths to files and folders in the

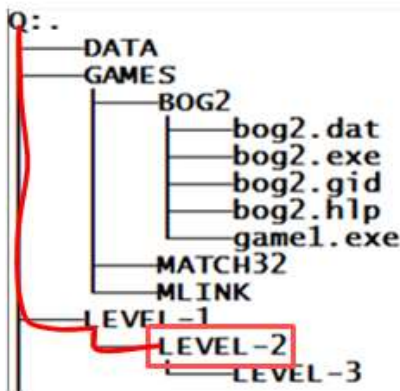


The absolute path to the game1.exe file is:

**Q:\GAMES\BOG2\game1.exe**

Starts with a drive letter, followed by a colon, followed by the backslash that represents the root of the drive, followed by a list of the folders you need to pass through to get to the file and, lastly the filename itself.

No extra spaces within the path.

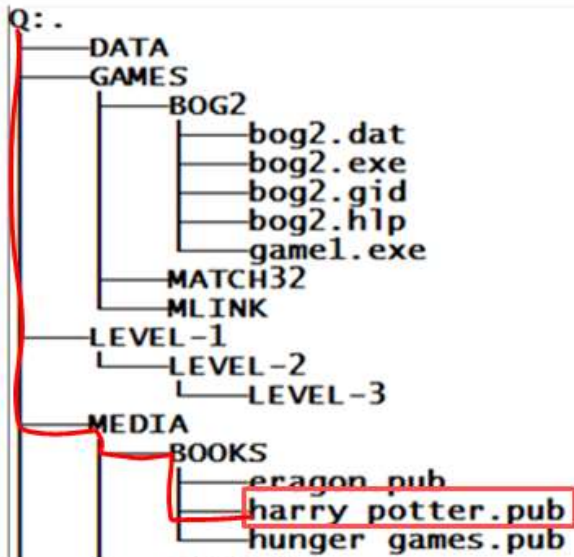


The absolute path to the LEVEL-2 folder is:

**Q:\LEVEL-1\LEVEL-2**

Starts with a drive letter, followed by a colon, followed by the backslash that represents the root of the drive, followed by a list of the folders you need to pass through to get to the folder.

No extra spaces within the path.



The absolute path to the harry potter.pub file is:

**Q:\MEDIA\BOOKS\“harry potter.pub”**  
Or  
**“Q:\MEDIA\BOOKS\harry potter.pub”**

Starts with a drive letter, followed by a colon, followed by the backslash that represents the root of the drive, followed by a list of the folders you need to pass through to get to the file and lastly the filename itself.

A key difference in this example are the quotes included in the path. When a space is part of a folder's name or a file's name you must enclose the name (or the entire path) in quotes.

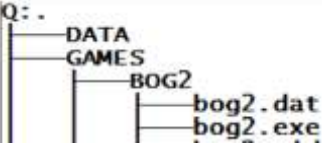
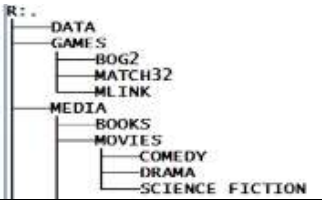

A **Relative Path** refers to a file or folder's location based on your current location. You could never write a relative path to a file or folder without knowing where you're starting from. It would be like trying to provide driving directions for someone to your house without knowing where the person is starting from.

A relative path will NEVER begin with a drive letter colon and a backslash.

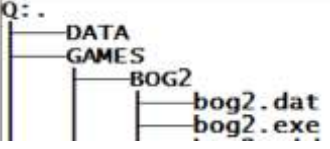
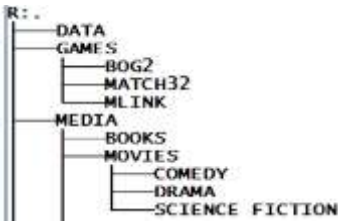
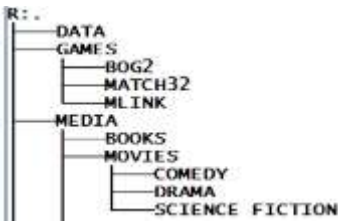
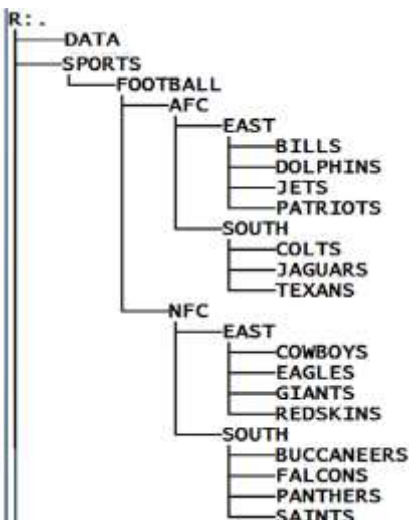
The existence of a leading backslash (with no drive or colon) in a path 'tells' the operating system to begin 'searching' for the file or folder at the root of the current drive. The absence of a leading backslash (with no drive or colon), 'tells' the operating system to begin 'searching' for the file or folder from your current location.

The relative path to a file or folder that exists beneath your current location is specified by listing, in order, the directories (omit the one that you are in) that you must pass through to get to the file or folder. Use backslashes to separate directory and file names in the path.

Examples of valid relative paths.

<p>C:\users\jarruda&gt;DIR \temp</p>	<p>The current location is C:\users\jarruda The leading backslash in the path after DIR 'tells' the system to start searching at the root of the current drive. In this example, it would look for temp at the root of drive C:</p>
<p>C:\users\jarruda&gt;dir temp</p>	<p>The current location is C:\users\jarruda The absence of a leading backslash in the path after DIR 'tells' the system to start searching in your current location. In this example, it would look for temp in the c:\users\jarruda directory.</p>
<p>Q:\Games&gt;dir bog2\bog2.exe</p> 	<p>The current location is Q:\Games The absence of a leading backslash in the path after DIR 'tells' the system to start searching in your current location. In this example, it would look for bog2 in your current location (Games). It would find it. It would then look in BOG2 for bog2.exe.</p>
<p>R:\MEDIA&gt;dir \games\bog2</p> 	<p>The current location is R:\Media The leading backslash in the path after DIR 'tells' the system to start searching at the root of the current drive. In this example, it would look for Games at the root of the drive. It would find it. It would then look in games for BOG2.</p>
<p>R:\media\movies&gt;dir comedy</p> 	<p>The current location is R:\media\movies. The absence of a leading backslash in the path after DIR 'tells' the system to start searching in the current locaiont. In this example it will look in Movies for Comedy. It will find it.</p>

The relative path to a file or folder that exists either above your current location or in a different branch of the folder hierarchy can be written using **leading periods**. Two dots (..) refers to the directory above the current directory (parent directory). ..\.. references the directory two directories up, ..\..\.. references the directory three directories up, and so on.

<pre>Q:\Games\bog2&gt;dir ..</pre> 	<p>The current location is Q:\Games\bog2</p> <p>The two periods used as a parameter to DIR ‘tells’ the system to reference the parent directory. In this case, the parent directory to your current location is Games.</p> <p>This command would display a directory listing of the Q:\games directory.</p>
<pre>R:\media\movies\drama&gt;dir ..\..</pre> 	<p>The current location is R:\Media\movies\drama</p> <p>The two sets of leading periods used as a parameter to DIR ‘tells’ the system to reference two directories up from your current location. In this case, Movies is one directory up from drama and Media is two directories up from Drama.</p> <p>The command would display a directory listing of the r:\media</p>
<pre>R:\media\movies\drama&gt;dir ..\Comedy</pre> 	<p>The current location is R:\Media\movies\drama</p> <p>This command “tells” the system to reference the parent directory and to look in that location for a directory named comedy.</p> <p>This command will display a directory listing of the files in the R:\media\movies\comedy directory.</p>
<pre>R:\sports\football\afc\east&gt;dir ..\..\nfc\east</pre> 	<p>The current location is R:\sports\football\afc\east.</p> <p>The two sets of leading periods used as a parameter to DIR ‘tells’ the system to reference two directories up from your current location. In this case, that would be FOOTBALL and, within that directory, look for nfc and then within nfc look for east.</p> <p>The command would display a directory listing of the r:\sports\football\nfc\east folder&gt;</p>

Two questions that can help you when trying to determine if a path is written correctly are:  
Where is it going to start looking? What is it going to look for?

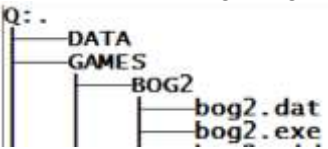
If it's an absolute path it will always start looking at the root of the drive specified in the path. It will 'look' for the item that appears at the left-most end of the path.

C:\windows\system32 -- would look for Windows at the root of drive C:

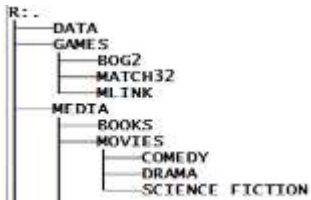
G:\mystuff\data\homework -- would look for mystuff at the root of drive G:

If it's a relative path it will always start in your current location. If leading periods are used in the path then it will start looking in the directory referenced by the periods (based on your current location). If there's a leading backslash (no drive), it will start looking at the root of the current drive. If there's no leading dots or drive specifications, it will start looking in your current location. It will always look for the file or folder at the left-most end of the path.

The below table includes examples of **INCORRECT** relative paths along with an explanation as to why it's wrong.

<pre>C:\users\jarruda&gt;dir temp</pre>	<p>Objective: Display the contents of the C:\temp directory using a relative path.</p> <p>The current location is C:\users\jarruda The absence of a leading backslash in the path after DIR 'tells' the system to start searching in your current location. In this example, it would look for temp in the c:\users\jarruda directory. The objective is to display the contents of the C:\temp directory.</p> <p>To 'fix' this and keep it a relative path you could use: C:\users\jarruda&gt;dir \temp The leading backslash tells the system to start looking at the root of the drive</p>
<pre>Q:\Games&gt;dir \bog2\bog2.exe</pre> 	<p>Objective: Display the q:\games\bog2\bog2.exe file in a directory listing using a relative path.</p> <p>The leading backslash in this path 'tells' the system to look for bog2 at the root of the current drive. It is not there.</p> <p>To fix this and keep it a relative path you could use: Q:\Games&gt;dir bog2\bog2.exe</p> <p>The absence of the leading backslash 'tells' the system to start searching in your current location for bog2. It is there.</p>

R:\MEDIA>dir \books



Objective: Display the contents of the R:\media\books directory using a relative path.

The leading backslash in the path after DIR ‘tells’ the system to start searching at the root of the current drive.

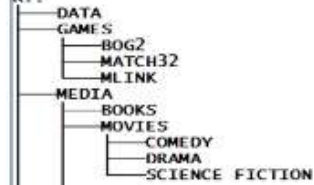
In this example, it would look for Books at the root of drive R: It would not find it.

To fix this and keep it a relative path you could use:

R:\MEDIA>dir books

The absence of the leading backslash ‘tells’ the system to start searching in your current location for books. It is there.

R:\media\movies>dir ..\media\books



Objective: Display the contents of the R:\media\books directory using a relative path.

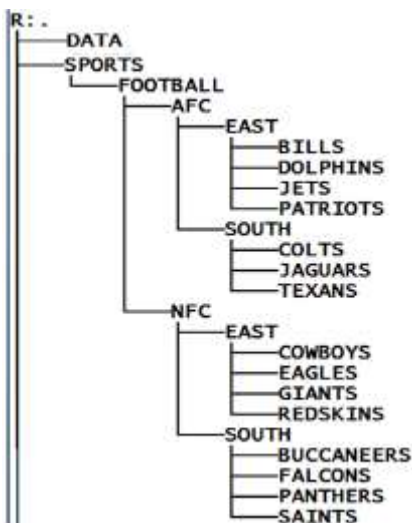
..\media at the beginning of this path tells the system to start looking in the parent directory for Media. The parent directory to your current location is media therefore it would look for media in media. It won’t find it.

To fix this and keep it a relative path you could use:

R:\media\movies>dir ..\books

This will look for books in the parent directory. The parent directory to your current location s media. Books is in media. The absence of a

R:\sports\football\afc\east\jets>dir ..\..\..\.



Objective: display the contents of the sports directory using a relative path.

The given command will work. This one isn’t wrong but it’s not the best approach to completing this task. Sometimes using leading dots is not the best or efficient approach.

In this example: DIR \SPORTS is a relative path and from the JETS directory would accomplish the desired task.

When you start spending too much time counting how many dot dot sequences you need to reference a folder, re-evaluate. It might be faster to start at the root. A good rule of thumb is if the folder you’re referencing is closer to the root than to your current location, reference it from the root.