|  |  |
| --- | --- |
| **PATH** | **CLASSPATH** |
| PATH is an environment variable. | CLASSPATH is also an environment variable. |
| It is used by the operating system to find the executable files (.exe). | It is used by Application ClassLoader to locate the .class file. |
| You are required to include the directory which contains .exe files. | You are required to include all the directories which contain .class and JAR files. |
| PATH environment variable once set, cannot be overridden. | The CLASSPATH environment variable can be overridden by using the command line option -cp or -CLASSPATH to both javac and java command. |

## 1. What is a Java Classpath?

This is a system variable that we set. It is used by the JVM or more precisely the Application class loader to load user-defined classes and packages

1.1 So, PATH and CLASSPATH are different?

Yes, the PATH variable and the CLASSPATH variable are different.

The PATH is an environmental variable. This variable is the path to Java executables like “java” or “javac” and is required by the Java compiler and runtime environment to run and compile Java programs from anywhere.

This is also an environment variable. This variable is used by the JVM or the Application class loader to locate and load user-defined classes that are not a part of the Java JDK installed on the system.

2. How to set the CLASSPATH variable?

There are multiple ways of setting the path variable.

2.1 Using the Command line

Using the command line i.e. cmd we can set a classpath variable in 2 ways.

2.1.1 Using the set command

We can set the path for the entire time the command prompt is open*. This is discouraged and is not the preferred way to set a path.*To set the classpath using the set command:

1. Open the cmd.
2. Write the command:

|  |  |
| --- | --- |
| 1 | set classpath1;classpath2.. |

example: set classpath=D:TestProject/bin

set command

2.1.2 Using the -cp or -classpath command

This is the preferred way of setting the path variable. According, to the Oracle docs

*The -classpath option is preferred because you can set it individually for each application without affecting other applications and without other applications modifying its value.*

*Java Docs*

To set the classpath using the -classpath command:

1. Open the command prompt.
2. Run the required command(SDK tool eg: java, javac) with -classpath added

Example: 1. Javac command

 Assume that our HelloWorld.java file requires a third-paty jar called “mailer.jar” to compile it. We can use the command

|  |  |
| --- | --- |
| 1 | javac -classpath mailer.jar HelloWorld.java |

2. Java command

Assume that our java files are in the src folder and the classes are in the bin folder, then while executing the java programs we can do-

|  |  |
| --- | --- |
| 1 | java -classpath D:/TestProject/bin MainClass |

 All the rules related to classpath setting are available on the [Oracle Java docs](https://docs.oracle.com/javase/8/docs/technotes/tools/windows/classpath.html#BEHJBHCD).

2.2 Environment Variable

One other way to set the path variable is through the Environment Variables. This method is **highly discouraged** since the classpath required by different programs is different. Also setting path variables this way may hinder the execution of other programs.

To set the classpath through the environment variables:

1. Type “environment variables” in the “Type here to search” box.
2. Click on the first option which opens the System properties dialog box.
3. In the User variables for Admin variables, click on the “New” button.
4. In the dialog box that open set the classpath variable.

environment variable

classpath set in the environment variables

3. Understanding classpath in action with an example

To understand classpath, lets take a brief look at how java loads classes.

3.1 Why is the classpath needed?

* Java allows users to load external jars and packages that are not a part of the Java JDK and use them in programs.
* However, these classes are loaded in the Java runtime environment only when they are required.
* To do this, the Application classloader is used.
* For the application classloader to find the packages/classes/jars that it requires, we need to set the classpath variable. More details about the class loader are available [here](https://docs.oracle.com/cd/E19501-01/819-3659/beadf/index.html).

3.2 Example

We will see the path in action with a small project. The name of the Project is TestProject.

The structure of the project is as follows:

Project structure

* The class files are in the “bin” folder and the source files are in the src folder.
* We have a MainClass and an Employees class inside the com.examples.javacodegeeks packages.
* The MainClass uses the Employees package and so has an import for it.

*MainClass.java*

|  |  |
| --- | --- |
| 01  02  03  04  05  06  07  08  09  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27 | import java.util.ArrayList;  import com.examples.javacodegeeks.Employees;  public class MainClass {      public static void main(String[] args) {          //Make an array of employees          ArrayList emp\_list= new ArrayList();          emp\_list.add(new Employees("Thorin Oakenshield",1,"King under the Mountain"));          emp\_list.add(new Employees("Balin",2,"Second-in-command"));          emp\_list.add(new Employees("Dwalin",3,"Erebor dwarf"));          emp\_list.add(new Employees("Dori",4,"Erebor dwarf"));          emp\_list.add(new Employees("Nori",5,"Erebor dwarf"));          emp\_list.add(new Employees("Ori",6,"Erebor dwarf"));          emp\_list.add(new Employees("Oin",7,"Erebor dwarf"));          emp\_list.add(new Employees("Gloin",8,"Erebor dwarf"));          emp\_list.add(new Employees("Bifur",9,"Erebor dwarf"));          emp\_list.add(new Employees("Bofur",10,"Erebor dwarf"));          emp\_list.add(new Employees("Bombur",11,"Erebor dwarf"));          emp\_list.add(new Employees("Fili",12,"Erebor dwarf"));          emp\_list.add(new Employees("Kili",13,"Erebor dwarf"));            for(Employees e:emp\_list) {              System.out.println(e.getEmployeeDescription(e));          }        }    } |

*Employees.java*

|  |  |
| --- | --- |
| 01  02  03  04  05  06  07  08  09  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39 | package com.examples.javacodegeeks;    public class Employees {        private String fullname;      private int empid;      private String designation;        public Employees(String fullname, int empid, String designation) {          super();          this.fullname = fullname;          this.empid = empid;          this.designation = designation;      }        public String getFullname() {          return fullname;      }      public void setFullname(String fullname) {          this.fullname = fullname;      }      public int getEmpid() {          return empid;      }      public void setEmpid(int empid) {          this.empid = empid;      }      public String getDesignation() {          return designation;      }      public void setDesignation(String designation) {          this.designation = designation;      }          public String getEmployeeDescription(Employees emp) {          return "Employee name: " + emp.getFullname() + " with employee id: " + emp.getEmpid() + " and designation: " +emp.getDesignation()  ;      }  } |

* Assume that we are in an environment where we need to execute the Java program from the C drive and the project is available in the D drive.
* To check if the path is set or not, we use the command below. IF the output is %classpath% that means that the path is not set.

|  |  |
| --- | --- |
| 1 | echo %classpath% |

No classpath variable set

* If we try to execute the java command from the C drive without setting the path, it will result in the following error.

ClassNotFound exception

* To fix the error we need to set the path variable using any of the commands mentioned above and then run the program again. The outputs are given below.

*Method 1: set command*

|  |  |
| --- | --- |
| 1 | set classpath=D:/TestProject/bin |

set classpath output

Preferred method: java -classpath option

Java -classpath option

4. Clearing out the classpath variable

To clear the path variable of it s value we use the set path command

|  |  |
| --- | --- |
| 1 | set classpath= |

This would clear the value of the path variable.