COURSEWARE

Professional Skills

Agile Fundamentals

Jira

Git

Databases Introduction

Java Beginner

- What is Java?
- Installation
- Hello World Example
- Data Types
- Packages
- Naming Conventions Cheat Sheet
- Flow of Control
- Class Members
- Operators
- Conditionals
- Iteration
- Arrays
- ArrayList
- Enhanced For Loops
- String Manipulation
- Class Constructors
- Access Modifiers
- Installing Java & Maven To PATH
- Object-Oriented Programming Principles
- Encapsulation
- Inheritance
- Polymorphism
- Abstraction
- Interfaces
- Type Casting
- Static
- Final
- Garbage Collection
- Input With Scanner
- Pass by Value/Reference
- JUnit

Installing Java & Maven To PATH

Contents

- Overview
- Tutorial
  - Java
    - Windows
    - Ubuntu
  - Maven
    - Windows PATH
    - Ubuntu PATH
- Exercises

Overview

Environment variables hold values related to the current environment, like the Operating System or user sessions.

One of the most well-known is the `Path` on Windows, Linux and macOS.

It specifies the directories in which executable programs are located on the machine that can be started without knowing and typing the entire filepath on the command line.

That's why you can run `calc.exe` or `notepad.exe` from a Windows command line or `Run` dialog, but not `firefox.exe` - the paths to the first two applications are located in a file in the `Path` variable.

To properly utilise the full benefits of Java and Maven, we need to add some values to our environment variables.

Tutorial

Java

Windows

1. Press the Windows button and type "path" into the search field, then select "Edit the system environment variables"

- Test Driven Development
- UML Basics
- JavaDoc
- Peer Programming
- Code Reviews

Maven

Testing (Foundation)

Java Intermediate

HTML

CSS

Javascript

Spring Boot

Selenium

Sonarqube

Advanced Testing (Theory)

Cucumber

MongoDB

Express

NodeJS

React

Express-Testing

Networking

Security

Cloud Fundamentals

AWS Foundations

AWS Intermediate

Linux

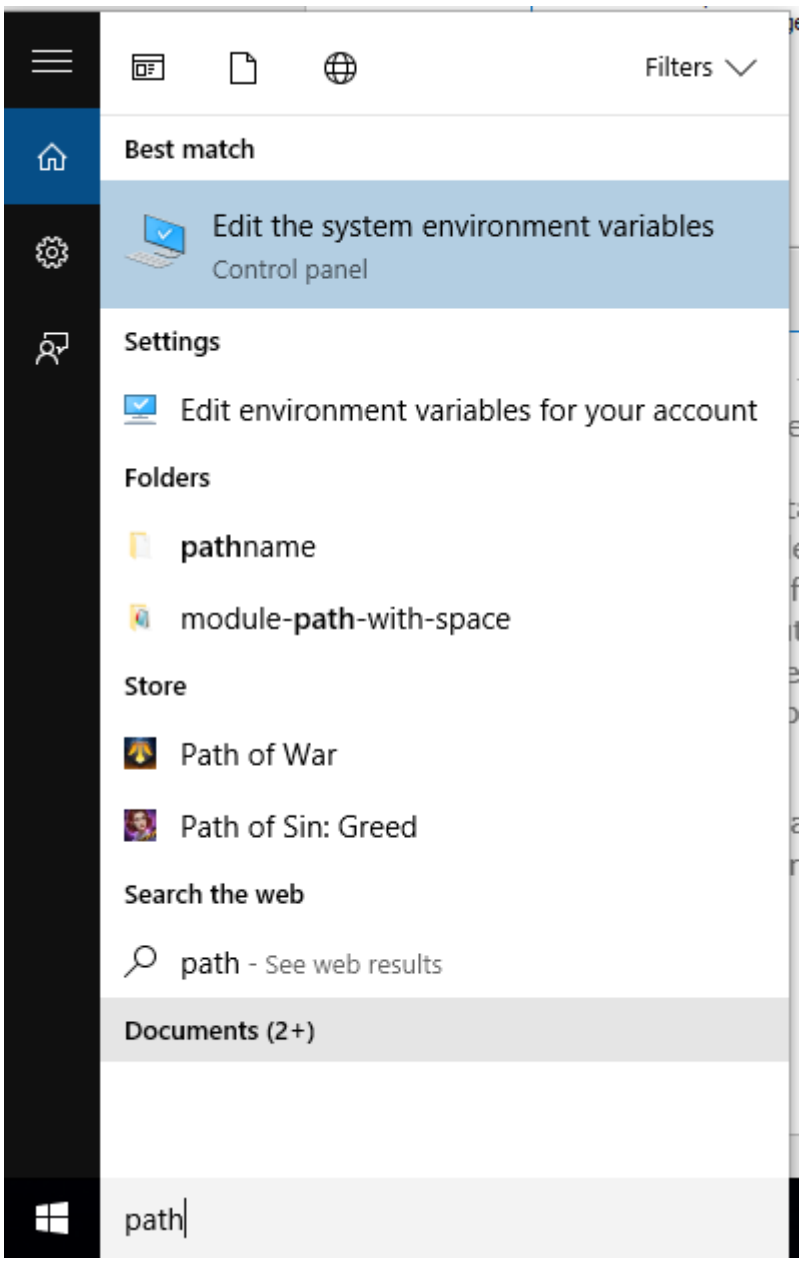
DevOps

Jenkins Introduction

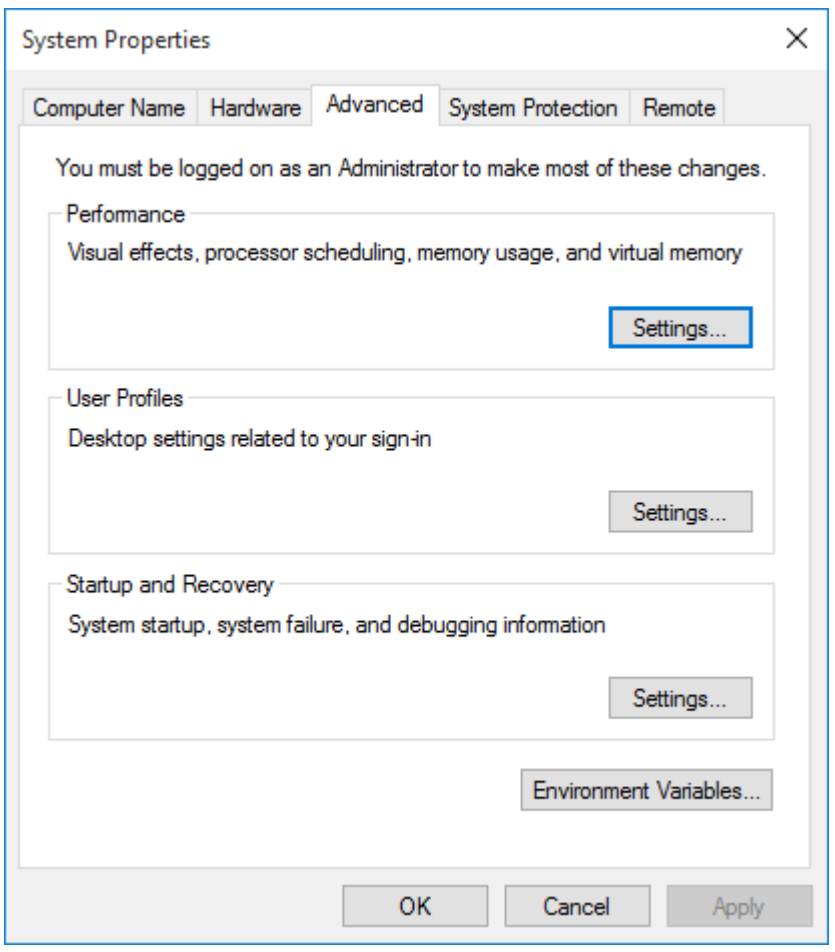
Jenkins Pipeline

Markdown

IDE Cheatsheet

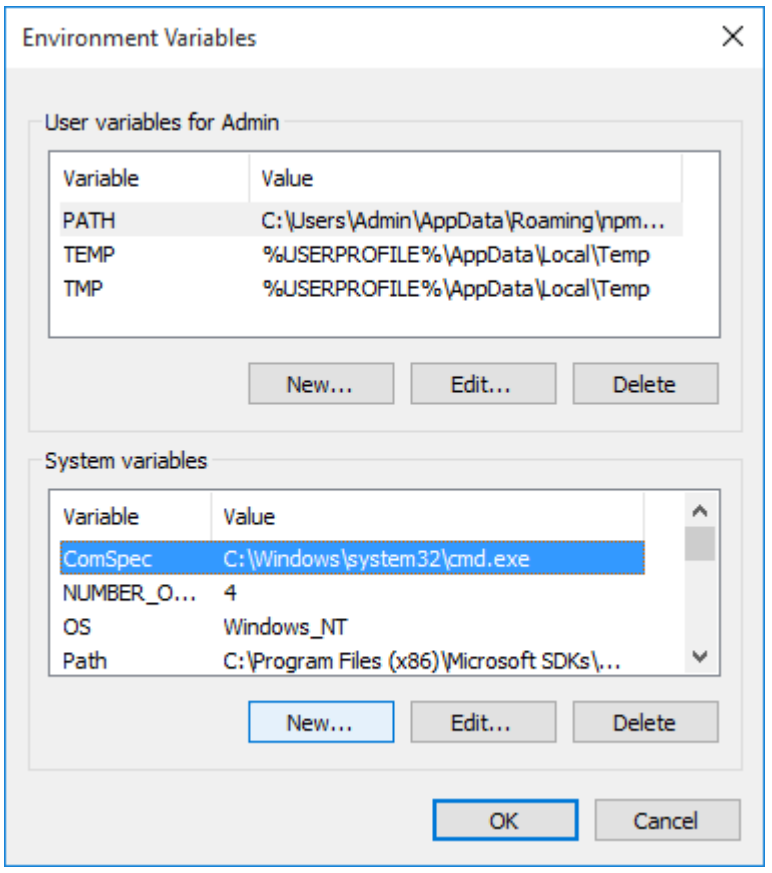


2. Click "Environment Variables...".

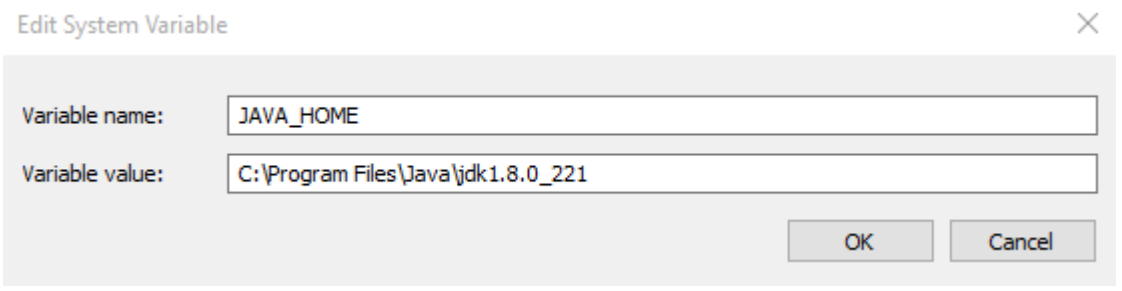


3. From this screen you can edit both the "User" and the "System" variables, you should change the "System" variables (bottom half of the window) as we want our changes to affect ALL users.

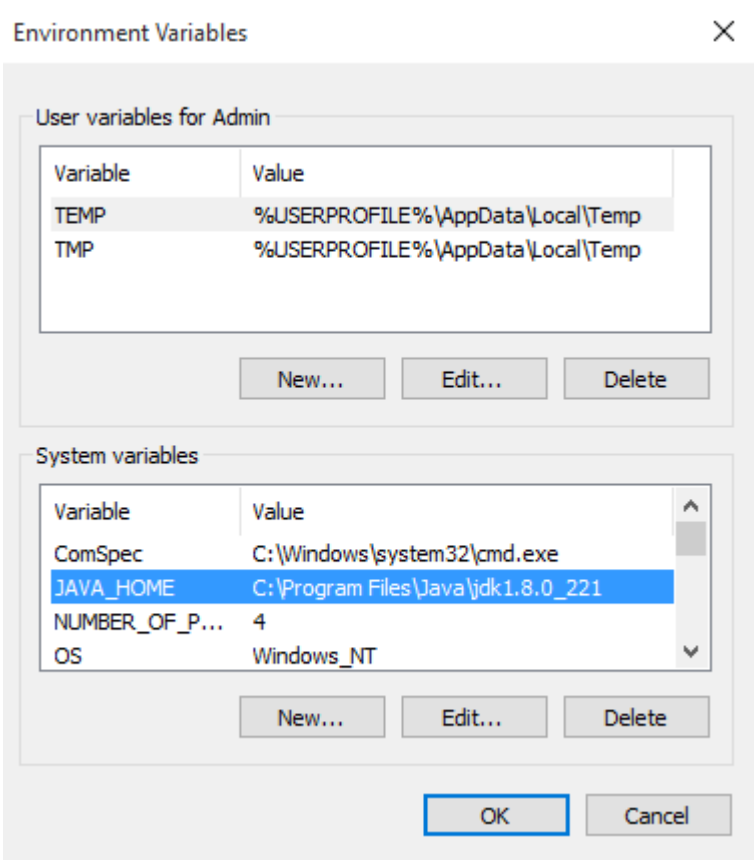
   Click "New..." to add a new system variable:



4. In the new popup window, we will create our new `JAVA_HOME` variable and give it a value corresponding to our JDK folder:

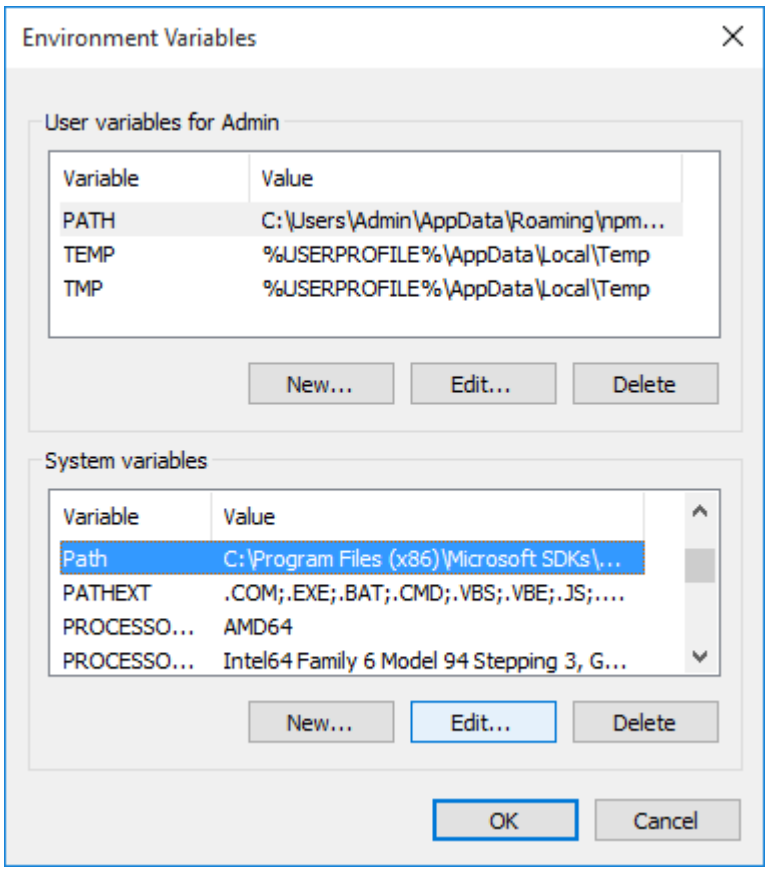


5. Click "OK" and we should now see our new `JAVA_HOME` system variable in the "System variables" box:



6. Now we need to edit the `Path` variable to include a reference to the `/bin` (binary files) folder in our `JAVA_HOME`.

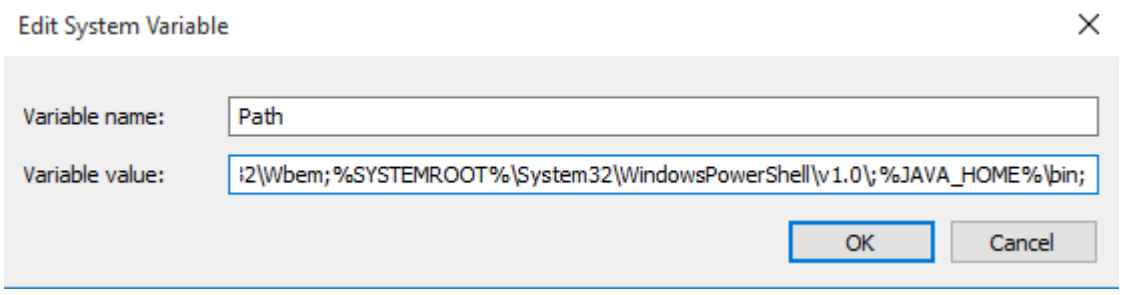
   To do this select the `Path` variable under "System variables" and then click "Edit...":



7. You need to append the path of the Java `/bin` folder to this `Path` variable - to do this, we will wildcard the location of the JAVA_HOME variable:

```
%JAVA_HOME%\bin
```

8. If this is successful, the `Path` variable should look like the image below (make sure that there is a semi-colon before and after `%JAVA_HOME%\bin`):



9. To test that this has saved, open a command line, type `java`, and hit Enter.

   If the `Path` variable has been configured correctly then the OS will run `java.exe` in your `JAVA_HOME` location's `/bin` folder:

```
C:\Users\Admin>java
Usage: java [-options] class [args...]
           (to execute a class)
   or  java [-options] -jar jarfile [args...]
           (to execute a jar file)
where options include:
    -d32          use a 32-bit data model if available
    -d64          use a 64-bit data model if available
    -server       to select the "server" VM
                  The default VM is server.

    -cp <class search path of directories and zip/jar files>
    -classpath <class search path of directories and zip/jar files>
                  A ; separated list of directories, JAR archives,
                  and ZIP archives to search for class files.
    -D<name>=<value>
                  set a system property
    -verbose:[class|gc|jni]
                  enable verbose output
    -version      print product version and exit
    -version:<value>
                  Warning: this feature is deprecated and will be removed
                  in a future release.
                  require the specified version to run
    -showversion  print product version and continue
    -jre-restrict-search | -no-jre-restrict-search
                  Warning: this feature is deprecated and will be removed
                  in a future release.
                  include/exclude user private JREs in the version search
    -? -help      print this help message
    -X            print help on non-standard options
    -ea[:<packagename>...|:<classname>]
    -enableassertions[:<packagename>...|:<classname>]
                  enable assertions with specified granularity
    -da[:<packagename>...|:<classname>]
    -disableassertions[:<packagename>...|:<classname>]
                  disable assertions with specified granularity
    -esa | -enablesystemassertions
                  enable system assertions
    -dsa | -disablesystemassertions
                  disable system assertions
    -agentlib:<libname>[=<options>]
                  load native agent library <libname>, e.g. -agentlib:hprof
                  see also, -agentlib:jdwp=help and -agentlib:hprof=help
    -agentpath:<pathname>[=<options>]
                  load native agent library by full pathname
    -javaagent:<jarpath>[=<options>]
                  load Java programming language agent, see java.lang.instrument
    -splash:<imagepath>
                  show splash screen with specified image
See http://www.oracle.com/technetwork/java/javase/documentation/index.html for more details.
```

Ubuntu

Ubuntu helpfully sorts out its `PATH` variables for us.

1. Open the terminal.
2. Run the following commands:

```
sudo apt-get update
```

```
sudo apt install openjdk-8-jdk -y
```

3. To verify that java is installed run the following command:

```
java -version
```

Maven

Windows PATH

We're going to be using the same method described above to set the Maven variable.

1. Navigate to the Environment Variables menu
2. Set two variables; `M2_HOME` and `MAVEN_HOME` - both of which should point to the path of your Maven install folder. (We need both, as certain programs reference Maven in different ways.)
3. Now we'll edit the `Path` variable in the same way we did for `JAVA_HOME`:

```
%MAVEN_HOME%\bin;
```

4. To verify that Maven is installed correctly, open a command line and enter the following command:

```
mvn -version
```

Ubuntu PATH

Ubuntu again helpfully sorts out the `PATH` variables for us.

1. Open the terminal.
2. Run the following commands:

```
sudo apt-get update
```

```
sudo apt install maven -y
```

3. To verify that Maven is installed run the following command:

```
mvn -version
```

Exercises

There is no exercise for this module