

Interacting with the Environment, using `java.lang.System` class

By Dr. Subrat Kumar Nayak

Associate Professor,

Department of CSE, FET, S'O'A (Deemed to be) University.

Introduction

- This chapter describes how your Java program can deal with its immediate surroundings, with what we call the runtime environment.
- In other sense, everything we do in a Java program using almost any Java API involves the environment.
- Here, the focus more narrowly on things that directly surround your program.
- This also talks about the **System** class that will help us to know a lot about the system.
- Many operating systems use *environment variables* to pass configuration information to applications. Like properties in the Java platform, environment variables are **key/value pairs**, where both the key and the value are strings.
- The conventions for setting and using environment variables vary between operating systems, and also between command line interpreters.

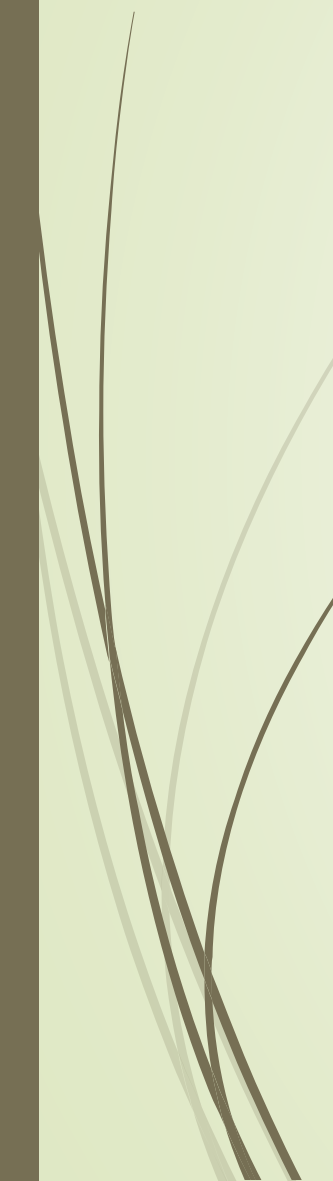

Getting Environment Variables

- Here the intention is to get the value of “environment variables” from within your Java program.
- Environment variables are commonly used for customizing an individual computer user's runtime environment.
- How to read environment variable using Java API?
 - System class provides two methods
 - ✓ `System.getenv(String name)` - which returns specific variable value
 - ✓ `System.getenv()` which returns all environment variables values.

Getting Information from System Properties

- Get information from the system properties.
- A property is just a name and value pair stored in a `java.util.Properties` object.
- In [Properties](#), we examined **the way an application can use Properties objects to maintain its configuration**. The Java platform itself uses a Properties object to maintain its own configuration. The **System** class maintains a **Properties object** that describes the **configuration of the current working environment**.
- System properties include **information about the current user, the current version of the Java runtime, and the character used to separate components of a file path name**.
- The following table describes some of the most important system properties

Key	Meaning
<code>"file.separator"</code>	Character that separates components of a file path. This is "/" on UNIX and "\" on Windows.
<code>"java.class.path"</code>	Path used to find directories and JAR archives containing class files. Elements of the class path are separated by a platform-specific character specified in the path.separator property.
<code>"java.home"</code>	Installation directory for Java Runtime Environment (JRE)
<code>"java.vendor"</code>	JRE vendor name



Key	Meaning
<code>"java.vendor.url"</code>	JRE vendor URL
<code>"java.version"</code>	JRE version number
<code>"line.separator"</code>	Sequence used by operating system to separate lines in text files
<code>"os.arch"</code>	Operating system architecture
<code>"os.name"</code>	Operating system name
<code>"os.version"</code>	Operating system version
<code>"path.separator"</code>	Path separator character used in <code>java.class.path</code>
<code>"user.dir"</code>	User working directory
<code>"user.home"</code>	User home directory
<code>"user.name"</code>	User account name

Continue...(Reading System Properties)

- The **System** class has two methods used to read system properties:
`getProperty` and `getProperties`
- To retrieve one system-provided property, use `System.getProperty()` . If you want them all, use `System.getProperties()` .
- Example: `System.getProperty("path.separator");`

Learning About the Current JDK Release

- You need to write code that looks at the current JDK release
- Use `System.getProperty()` with an argument of `java.specification.version`.
- Solution:

```
System.out.println(System.getProperty("java.specification.version"))
```


Dealing with Operating System–Dependent Variations

- To write code that adapts to the underlying operating system.
- Though Java is designed to be portable, some things aren't.
- Such as file separator (/) for unix and (\) for dos or windows.
- You can use System.Properties to find out the operating system, and various features in the File class to find out some platform-dependent features.

- Example:

```
use System . getProperty (" file . separator ")
```

or

```
use String ret = java .io. File . separator ;
```




End of Chapter