Properties class in Java

The **properties** object contains key and value pair both as a string. The java.util.Properties class is the subclass of Hashtable.

It can be used to get property value based on the property key. The Properties class provides methods to get data from the properties file and store data into the properties file. Moreover, it can be used to get the properties of a system.

An Advantage of the properties file

**Recompilation is not required if the information is changed from a properties file:** If any information is changed from the properties file, you don't need to recompile the java class. It is used to store information which is to be changed frequently.

Constructors of Properties class

|  |  |
| --- | --- |
| **Method** | **Description** |
| Properties() | It creates an empty property list with no default values. |
| Properties(Properties defaults) | It creates an empty property list with the specified defaults. |

Methods of Properties class

The commonly used methods of Properties class are given below.

|  |  |
| --- | --- |
| **Method** | **Description** |
| public void load(Reader r) | It loads data from the Reader object. |
| public void load(InputStream is) | It loads data from the InputStream object |
| public void loadFromXML(InputStream in) | It is used to load all of the properties represented by the  XML document on the specified input stream into this  properties table. |
| public String getProperty(String key) | It returns value based on the key. |
| public String getProperty(String key, String defaultValue) | It searches for the property with the specified key. |
| public void setProperty(String key, String value) | It calls the put method of Hashtable. |
| public void list(PrintStream out) | It is used to print the property list out to the specified  output stream. |
| public void list(PrintWriter out)) | It is used to print the property list out to the specified  output stream. |
| public Enumeration<?> propertyNames()) | It returns an enumeration of all the keys from the property list. |
| public Set<String> stringPropertyNames() | It returns a set of keys in from property list where the key  and its corresponding value are strings. |
| public void store(Writer w, String comment) | It writes the properties in the writer object. |
| public void store(OutputStream os, String comment) | It writes the properties in the OutputStream object. |
| public void storeToXML(OutputStream os, String comment) | It writes the properties in the writer object for generating  XML document. |
| public void storeToXML(Writer w, String comment, String encoding) | It writes the properties in the writer object for generating  XML document with the specified encoding. |

Example of Properties class to get information from the properties file

To get information from the properties file, create the properties file first.

**db.properties**

1. user=system
2. password=oracle

Now, let's create the java class to read the data from the properties file.

**Test.java**

1. **import** java.util.\*;
2. **import** java.io.\*;
3. **public** **class** Test {
4. **public** **static** **void** main(String[] args)**throws** Exception{
5. FileReader reader=**new** FileReader("db.properties");
7. Properties p=**new** Properties();
8. p.load(reader);
10. System.out.println(p.getProperty("user"));
11. System.out.println(p.getProperty("password"));
12. }
13. }

Output:system

oracle

Now if you change the value of the properties file, you don't need to recompile the java class. That means no maintenance problem.

Example of Properties class to get all the system properties

By System.getProperties() method we can get all the properties of the system. Let's create the class that gets information from the system properties.

**Test.java**

1. **import** java.util.\*;
2. **import** java.io.\*;
3. **public** **class** Test {
4. **public** **static** **void** main(String[] args)**throws** Exception{
6. Properties p=System.getProperties();
7. Set set=p.entrySet();
9. Iterator itr=set.iterator();
10. **while**(itr.hasNext()){
11. Map.Entry entry=(Map.Entry)itr.next();
12. System.out.println(entry.getKey()+" = "+entry.getValue());
13. }
15. }
16. }

Output:

java.runtime.name = Java(TM) SE Runtime Environment

sun.boot.library.path = C:\Program Files\Java\jdk1.7.0\_01\jre\bin

java.vm.version = 21.1-b02

java.vm.vendor = Oracle Corporation

java.vendor.url = http://java.oracle.com/

path.separator = ;

java.vm.name = Java HotSpot(TM) Client VM

file.encoding.pkg = sun.io

user.country = US

user.script =

sun.java.launcher = SUN\_STANDARD

...........

Example of Properties class to create the properties file

Now let's write the code to create the properties file.

**Test.java**

1. **import** java.util.\*;
2. **import** java.io.\*;
3. **public** **class** Test {
4. **public** **static** **void** main(String[] args)**throws** Exception{
6. Properties p=**new** Properties();
7. p.setProperty("name","Sonoo Jaiswal");
8. p.setProperty("email","sonoojaiswal@javatpoint.com");
10. p.store(**new** FileWriter("info.properties"),"Javatpoint Properties Example");
12. }
13. }

Let's see the generated properties file.

**info.properties**

1. #Javatpoint Properties Example
2. #Thu Oct 03 22:35:53 IST 2013
3. email=sonoojaiswal@javatpoint.com
4. name=Sonoo Jaiswal