

Core Java 8 and Development Tools

Lesson 15 : Property Files





Lesson Objectives

After completing this lesson, participants will be able to

- Define property files and use them
- Use properties and its methods
- Define and use user specific properties



15.1: What are Property Files? Property Files

Property files

- have .properties extension
- are used to store the configuration parameters
- each parameter is stored as key/value pair

The java.util. Properties class

- represents a persistent set of key/value properties
- are subclasses of Hashtables
- provides methods to store and retrieve values from **properties files**. Example ->

```
#Properties File to the Test Application  
password=tiger  
username=scott
```



15.2: Types of Property files

Categories of Property Files

User Specific Properties

System Properties



15.3: User defined Properties

The `java.util.Property` Class

To manage properties, create instances of `java.util.Properties` class.

This class provides methods for the following:

- Loading key/value pairs into a **Properties** object from a stream
- Retrieving a value from its key
- Listing the keys and their values
- Saving the properties to a stream



15.3: user defined Properties

Setting Properties

`setProperty(String key, String value)`

- Puts the key/value pair in the Properties object.

`remove(Object key)`

- Removes the key/value pair associated with key.



15.3: User defined Properties

Getting Property Information

contains(Object value)
containsKey(Object key)
getProperty(String key)
getProperty(String key, String default)
list(PrintStream s)
list(PrintWriter w)
elements()
keys()
propertyNames()
stringPropertyNames()
size()



15.3: Demo: User defined Properties

Demo: User Specific Properties

```
private static void saveProperties(Properties p) {  
    try { OutputStream propsFile = new FileOutputStream(fileName);  
        p.store(propsFile, "Properties File to the Test Application");  
        propsFile.close();  
    } catch (IOException ioe) {... }  
}  
private static Properties loadProperties(String fileName) {  
    Properties tempProp = new Properties();  
    try { InputStream propsFile = new FileInputStream(fileName);  
        tempProp.load(propsFile);  
        propsFile.close();  
    } catch (IOException ioe) {... }  
    return tempProp;    }
```




15.3: Demo: User defined Properties

Demo: User Specific Properties

```
private static Properties createDefaultProperties() {
    Properties tempProp = new Properties();
    /* Database connection parameter properties are set */
    tempProp.setProperty("url",
        "jdbc:oracle:thin:@192.168.12.16:1521:oracle8i");
    tempProp.setProperty("driver","oracle.jdbc.driver.OracleDriver");
    tempProp.setProperty("username", "trg1");
    tempProp.setProperty("password", "tiger");
    return tempProp;
}
private static void printProperties(Properties p, String s) {
    p.list(System.out);
}
```



15.3: Demo: User defined Properties

Demo : Concept of properties

Execute the DatasourcePropertyfiles.java program



Summary

In this lesson, you have learnt the following:

- What are Property files and their usage?
- User specific properties



Review Question

Question 1: load(_____) throws IOException

- **Option 1** : InputStream
- **Option 2** : OutputStream

Question 2: Is this a valid key value pair?

fruit apple

- True/False.



Review Question

Question 3: If the property file contains

```
fruits=apple,\mango
```

What will be the output of

```
System.out.println(p.getProperty("fruits"));
```

- Where p is a properties object
- **Option 1** : apple, mango
- **Option 2** : apple,\mango
- **Option 3** : null