## File Handling in Java

Some of the common file handling operations are;

1. Create file
2. Delete file
3. Read file
4. Write file
5. Change file permissions

### Create File

We can use File class createNewFile() method to create new file. This method returns true if file is successfully created, otherwise it returns false. Below is a simple program showing how to create a new file in java.

package com.netparam.files;

import java.io.File;

import java.io.IOException;

public class FileHandling {

public static void main(String[] args) {

File file = new File("data.txt");

try {

boolean createNewFile = file.createNewFile();

System.out.println("File Created = "+createNewFile);

} catch (IOException e) {

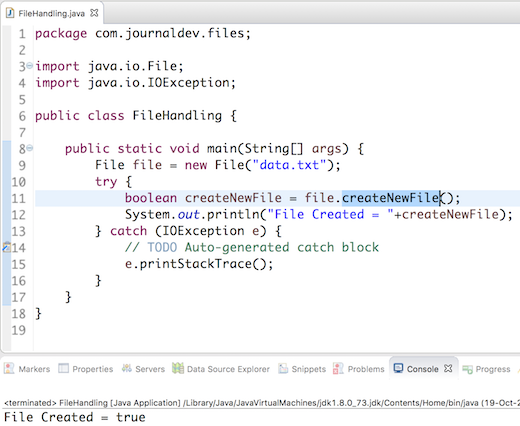
// TODO Auto-generated catch block

e.printStackTrace();

}

}

}

[](http://cdn.journaldev.com/wp-content/uploads/2016/10/file-handling-in-java-create.png)

Above image shows the output produced in first run, in subsequent execution the file will be present so the createNewFile will return false.

There are some rules associated with absolute path and relative path, read about them at java create new file.

### Delete File

File class delete method is used to delete a file or an empty directory. Below is a simple example to delete a file.

package com.netparam.files;

import java.io.File;

public class FileHandling {

public static void main(String[] args) {

File file = new File("data.txt");

boolean delete = file.delete();

System.out.println("File deleted = " + delete);

}

}

File delete method returns true if file is deleted successfully or else it returns false.

### Read File

There are many ways to read a file in java. We can use BufferedReader, FileReader or Files class. Below code snippet is to read file line by line.

File file = new File("data.txt");

FileInputStream fis = new FileInputStream(file);

InputStreamReader isr = new InputStreamReader(fis, cs);

BufferedReader br = new BufferedReader(isr);

String line;

while((line = br.readLine()) != null){

//process the line

System.out.println(line);

}

br.close();

For all other ways to read file, go to java read file.

### Write File

We can use FileWriter, BufferedWriter, Files or FileOutputStream to write file in java. Below code snippet use Stream to write data to file.

OutputStream os = null;

try {

os = new FileOutputStream(new File("/Users/pankaj/os.txt"));

os.write(data.getBytes(), 0, data.length());

} catch (IOException e) {

e.printStackTrace();

}finally{

try {

os.close();

} catch (IOException e) {

e.printStackTrace();

}

}

For a complete example with other classes, refer java write to file.

### Change File Permissions

File class provide methods to get file permission details as well as change them. Below code snippet shows you how to read file permissions and change them.

File file = new File("/Users/pankaj/run.sh");

//check file permissions for application user

System.out.println("File is readable? "+file.canRead());

System.out.println("File is writable? "+file.canWrite());

System.out.println("File is executable? "+file.canExecute());

//change file permissions for application user only

file.setReadable(false);

file.setWritable(false);

file.setExecutable(false);

//change file permissions for other users also

file.setReadable(true, false);

file.setWritable(true, false);

file.setExecutable(true, true);

However if you are using Java 7 or higher version, you should use PosixFilePermission that provides more options.

//using PosixFilePermission to set file permissions 777

Set<PosixFilePermission> perms = new HashSet<PosixFilePermission>();

//add owners permission

perms.add(PosixFilePermission.OWNER\_READ);

perms.add(PosixFilePermission.OWNER\_WRITE);

perms.add(PosixFilePermission.OWNER\_EXECUTE);

//add group permissions

perms.add(PosixFilePermission.GROUP\_READ);

perms.add(PosixFilePermission.GROUP\_WRITE);

perms.add(PosixFilePermission.GROUP\_EXECUTE);

//add others permissions

perms.add(PosixFilePermission.OTHERS\_READ);

perms.add(PosixFilePermission.OTHERS\_WRITE);

perms.add(PosixFilePermission.OTHERS\_EXECUTE);

Files.setPosixFilePermissions(Paths.get("/Users/pankaj/run.sh"), perms);