Java IO

#### 创建一个文件：使用File.createNewFile()方法创建。

#### 构建文件路径：

we will show you three Java examples to construct a file path :

1. File.separator or System.getProperty(“file.separator”) (Recommended)
2. File file = new File(workingDir, filename); (Recommended)
3. Create the file separator manually. (Not recommend, just for fun)

#### 如何在Java中设置文件权限（set the file permission ）

**Check if the file permission allow** :

1. file.canExecute(); – return true, file is executable; false is not.
2. file.canWrite(); – return true, file is writable; false is not.
3. file.canRead(); – return true, file is readable; false is not.

**Set the file permission** :

1. file.setExecutable(boolean); – true, allow execute operations; false to disallow it.
2. file.setReadable(boolean); – true, allow read operations; false to disallow it.
3. file.setWritable(boolean); – true, allow write operations; false to disallow it.

#### 读取文件——BufferedInputStream（字节）、FileInputStream、DataInputStream

The readLine() from the type DataInputStream is deprecated. Sun officially announced this method can not convert property from bytes to characters. It’s advised to use BufferedReader.

#### 读取文件——BufferedReader（字符）、FileReader

**Note**  
There are many ways to read a file, but this BufferedReader is the simplest and most common-used method.

#### 写入文件——File、FileOutputStream

In Java, [FileOutputStream](http://docs.oracle.com/javase/1.4.2/docs/api/java/io/FileOutputStream.html) is a bytes stream class that’s used to handle raw binary data. To write the data to file, you have to convert the data into bytes and save it to file

#### 写入文件——FileWriter、BufferedWriter

**Note**  
The BufferedWriter is a character stream class to handle the character data. Unlike byte stream (convert data into bytes), you can just write the strings, arrays or character data directly to a file.

#### 追加文件内容

In Java, you can use FileWriter(file,true) to append new content to the end of a file.

#### 删除文件——File.delete()

No nonsense, just issue the **File.delete()** to delete a file, it will return a boolean value to indicate the delete operation status; true if the file is deleted; false if failed.

#### 删除特定后缀的文件——FilenameFilter

In Java, you can implements the [FilenameFilter](http://java.sun.com/j2se/1.4.2/docs/api/java/io/FilenameFilter.html), override the accept(File dir, String name) method, to perform the file filtering function.

In this example, we show you how to use FilenameFilter to list out all files that are end with “**.txt**” extension in folder “**c:\\folder**“, and then delete it.

#### 查找特定后缀的文件——FilenameFilter

#### 使用递归方法删除文件夹及文件夹下的文件

public static void deleteAllFilesOfDir(File path) {

    if (!path.exists())

        return;

    if (path.isFile()) {

        path.delete();

        return;

    }

    File[] files = path.listFiles();  //列出文件夹中的所以文件，并逐一删除

    for (int i = 0; i < files.length; i++) {

        deleteAllFilesOfDir(files[i]);

    }

    path.delete();

}

#### 文件重命名——renameTo()

#### 复制文件

Java didn’t comes with any ready make file copy function, you have to manual create the file copy process. To copy file, just convert the file into a bytes stream with **FileInputStream** and write the bytes into another file with **FileOutputStream**.

#### 移动文件到另一个目录

**Java.io.File** does not contains any ready make move file method, but you can workaround with the following two alternatives :

File.renameTo().

Copy to new file and delete the original file.

public static void main(String[] args) {

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

if(file.renameTo(new File("test\\"+file.getName()))){

System.out.println("文件移动成功");

}else{

System.out.println("移动文件失败");

}

}

#### 获取文件的创建日期

There are no official way to get the file creation date in Java. However, you can use the following workaround to get the file creation date in Windows platform.

**How it work**

In Windows command prompt, type the command to list the file creation date.

C:\>cmd /c dir c:\logfile.log /tc

Volume in drive C has no label.

Volume Serial Number is 0410-1EC3

Directory of c:\

31/05/2010 08:05 14 logfile.log

1 File(s) 14 bytes

0 Dir(s) 35,389,460,480 bytes free

The “**31/05/2010 08:05**” is what you need. The idea is use the Java “**Runtime.getRuntime().exec**” to execute the above command, hold the output, and parse it by lines until you get the date and time.

#### 获取文件的最后修改时间——File.lastModified()

In Java, you can use the **File.lastModified()** to get the file’s last modified timestamps. This method will returns the time in milliseconds (long value), you may to format it with **SimpleDateFormat** to make it a human readable format.

#### 获取文件大小

In Java, you can use the **File.length()** method to get the file size in bytes.

#### 获取一个文件的完整路径（绝对路径，包括文件名）——File.getAbsolutePath()

The **File.getAbsolutePath()** will give you the full complete path name (filepath + filename) of a file.

For example：

File file = File("C:\\abcfolder\\textfile.txt");

System.out.println("Path : " + file.getAbsolutePath());