

## Java Notes

# java.io.File

`java.io.File` is the central class in working with files and directories. Files and directories are both represented by File objects. When a File object is created, the system doesn't test to see if a corresponding file/directory actually exists; you must call `exists()` to check. See [Example - FileTest.java](#).

## File Constructors and Methods

Assume

```
boolean b;
String s;
String path;    // Relative or absolute path.
String dirpath; // Relative or absolute path to a directory.
String fname;   // File name
File f, dir;    // Assume f is a file, dir is a directory.
long l;
File[] fa;     // Array of File objects.
String[] sa;   // Array of file or directory names.
```

### Constructors

```
f = new File(path);           Create File object for default directory (usually where program is located).
f = new File(dirpath, fname); Create File object for directory path given as string.
f = new File(dir, fname);     Create File object for directory.
```

### public static constants

```
s = File.separator;         Default path separator (eg, "/" in Unix, "\" in Windows).
```

### Getting Attributes

```
b = f.exists();             true if file exists.
b = f.isFile();             true if this is a normal file.
b = f.isDirectory();       true if f is a directory.
s = f.getName();           name of file or directory.
b = f.canRead();           true if can read file.
b = f.canWrite();          true if can write file.
b = f.isHidden();          true if file is hidden.
l = f.lastModified();      Time of last modification.
l = f.length();            Number of bytes in file.
```

### Setting Attributes

```
f.setLastModified(t);      Sets last modified time to long value t.
b = f.setReadOnly();        Make file read only. Returns true if successful.
```

### Paths

<code>s = f.getPath();</code>	path name.
<code>s = f.getAbsolutePath();</code>	path name (how is it different from above?).
<code>s = f.getCanonicalPath();</code>	path name. May throw IOException.
<code>s = f.toURL();</code>	path with "file:" prefix and /'s. Directory paths end with /.
<code>s = f.toURI();</code>	path with "file:" prefix and /'s. Directory paths end with /.

### Creating and deleting files and directories

<code>b = f.delete();</code>	Deletes the file.
<code>b = f.createNewFile();</code>	Create file, may throw IOException. true if OK; false if already exists.
<code>b = f.renameTo(f2);</code>	Renames <i>f</i> to File <i>f2</i> . Returns true if successful.
<code>b = f.mkdir();</code>	Creates a directory. Returns true if successful.
<code>b = f.mkdirs();</code>	Creates directory and all dirs in path. Returns true if successful.

### Parents and Children

<code>s = f.getParent();</code>	Name of parent directory.
<code>dir = f.getParentFile();</code>	File of parent.
<code>sa = dir.list();</code>	Array of file/directory names in <i>dir</i> .
<code>fa = dir.listFiles();</code>	Array of files/directories in <i>dir</i> .
<code>fa = dir.listFiles(ff);</code>	As above after applying <a href="#">java.io.FileFilter</a> <i>ff</i> .

---

Copyright 2005 [Fred Swartz MIT License](#)