Class java.io.File

```
java.lang.Object
   |
   +----java.io.File
```

---

public class **File**
extends Object

This class represents a file name of the host file system. The file name can be relative or absolute. It must use the file name conventions of the host platform.

The intention is to provide an abstraction that deals with most of the system−dependent file name features such as the separator character, root, device name, etc. Not all features are currently fully implemented.

Note that whenever a file name or path is used it is assumed that the host's file name conventions are used.

Version:
1.34, 10/22/95
Author:
Jonathan Payne, Arthur van Hoff

Variable Index

o **pathSeparator**
The system dependent path separator string.
o **pathSeparatorChar**
The system dependent path separator character.
o **separator**
The system dependent file separator String.
o **separatorChar**
The system dependent file separator character.

Constructor Index

o **File**(String)
    Creates a File object.
o **File**(String, String)
    Creates a File object from the specified directory.
o **File**(File, String)
    Creates a File object (given a directory File object).

Method Index

o **canRead**()
    Returns a boolean indicating whether or not a readable file exists.
o **canWrite**()
    Returns a boolean indicating whether or not a writable file exists.
o **equals**(Object)
    Compares this object against the specified object.
o **exists**()
    Returns a boolean indicating whether or not a file exists.
o **getAbsolutePath**()
    Gets the absolute path of the file.
o **getName**()
    Gets the name of the file.
o **getParent**()
    Gets the name of the parent directory.
o **getPath**()
    Gets the path of the file.
o **hashCode**()
    Computes a hashcode for the file.
o **isAbsolute**()
    Returns a boolean indicating if the file name is absolute.
o **isDirectory**()
    Returns a boolean indicating whether or not a directory file exists.
o **isFile**()
    Returns a boolean indicating whether or not a normal file exists.
o **lastModified**()
    Returns the last modification time.
o **length**()
    Returns the length of the file.
o **list**()
    Lists the files in a directory.
o **list**(FilenameFilter)
    Uses the specified filter to list files in a directory.
o **mkdir**()
    Creates a directory and returns a boolean indicating the success of the creation.
o **mkdirs**()
    Creates all directories in this path.
o **renameTo**(File)

Renames a file and returns a boolean indicating whether or not this method was successful.

- **toString**()

  Returns a String object representing this file's path.

Variables

- **separator**

```
public final static String separator
```

  The system dependent file separator String.

- **separatorChar**

```
public final static char separatorChar
```

  The system dependent file separator character.

- **pathSeparator**

```
public final static String pathSeparator
```

  The system dependent path separator string.

- **pathSeparatorChar**

```
public final static char pathSeparatorChar
```

  The system dependent path separator character.

Constructors

- **File**

```
public File(String path)
```

  Creates a File object.
  Parameters:
  path – the file path
  Throws: NullPointerException
  If the file path is equal to null.

- **File**

```
public File(String path,
            String name)
```

Creates a File object from the specified directory.
Parameters:
path – the directory path
name – the file name

o **File**

```
public File(File dir,
            String name)
```

Creates a File object (given a directory File object).
Parameters:
dir – the directory
name – the file name

Methods

o **getName**

```
public String getName()
```

Gets the name of the file. This method does not include the directory.
Returns:
the file name.

o **getPath**

```
public String getPath()
```

Gets the path of the file.
Returns:
the file path.

o **getAbsolutePath**

```
public String getAbsolutePath()
```

Gets the absolute path of the file.
Returns:
the absolute file path.

o **getParent**

```
public String getParent()
```

Gets the name of the parent directory.
Returns:
the parent directory, or null if one is not found.

o **exists**

```
public boolean exists()
```

Returns a boolean indicating whether or not a file exists.

o **canWrite**

```
public boolean canWrite()
```

Returns a boolean indicating whether or not a writable file exists.

o **canRead**

```
public boolean canRead()
```

Returns a boolean indicating whether or not a readable file exists.

o **isFile**

```
public boolean isFile()
```

Returns a boolean indicating whether or not a normal file exists.

o **isDirectory**

```
public boolean isDirectory()
```

Returns a boolean indicating whether or not a directory file exists.

o **isAbsolute**

```
public boolean isAbsolute()
```

Returns a boolean indicating if the file name is absolute.

o **lastModified**

```
public long lastModified() throws IOException
```

Returns the last modification time. The return value should only be used to compare modification dates. It is meaningless as an absolute time.

o **length**

```
public long length() throws IOException
```

Returns the length of the file.

o **mkdir**

```
public boolean mkdir() throws IOException
```

Creates a directory and returns a boolean indicating the success of the creation.

o **renameTo**

```
public boolean renameTo(File dest) throws IOException
```

Renames a file and returns a boolean indicating whether or not this method was successful.
Parameters:
dest – the new file name

o **mkdirs**

```
public boolean mkdirs() throws IOException
```

Creates all directories in this path. This method returns true if all directories in this path are created.

o **list**

```
public String[] list() throws IOException
```

Lists the files in a directory. Works only on directories.
Returns:
an array of file names. This list will include all files in the directory except the equivalent of "." and ".." .

o **list**

```
public String[] list(FilenameFilter filter) throws IOException
```

Uses the specified filter to list files in a directory.
Parameters:
filter – the filter used to select file names
Returns:
the filter selected files in this directory.
See Also:
FilenameFilter

o **hashCode**

```
public int hashCode()
```

Computes a hashcode for the file.
Overrides:
hashCode in class Object

o **equals**

```
public boolean equals(Object obj)
```

Compares this object against the specified object.

Parameters:

obj – the object to compare with

Returns:

true if the objects are the same; false otherwise.

Overrides:

equals in class Object

o **toString**

```
public String toString()
```

Returns a String object representing this file's path.

Overrides:

toString in class Object