

# fileutil

## various file utilities

### Table of contents

1 Overview.....	2
2 Design.....	2
3 Constraints.....	2
4 Attributes.....	2
5 Commands.....	2
5.1 available.....	2
5.2 executable.....	3
5.3 writeable.....	3
5.4 readable.....	3
5.5 older.....	4
5.6 newer.....	4
5.7 ls.....	4
5.8 mkdir.....	5
5.9 rmdir.....	5
5.10 copy.....	5
5.11 move.....	6
5.12 remove.....	6
5.13 link.....	7
5.14 touch.....	7

## 1. Overview

**fileutil:** *various file utilities*

## 2. Design

**Super Type**  
Managed-Entity

Role	<b>Abstract.</b> (Objects cannot be created.)
Instance Names	<b>Unique</b>
Notification	<b>false</b>
Template Directory	
Data View	Children, proximity: <b>1</b>
Logger Name	fileutil

## 3. Constraints

None

## 4. Attributes

None

## 5. Commands

### Note:

Commandline options displayed in square brackets "[ ]" are optional. If an option expects arguments, then angle brackets are shown after the option "<>". Any default value is shown within the brackets.

### 5.1. available

*test if file available*

*static:* This command can be run outside of an object context.

#### Usage

```
available [-failonerror] -file <>
```

## Options

Option	Description
failonerror	<i>error the command if test fails</i>
file	<i>file name</i>

## 5.2. executable

*test if file executable*

*static:* This command can be run outside of an object context.

### Usage

```
executable [-failonerror] -file <>
```

## Options

Option	Description
failonerror	<i>error the command if test fails</i>
file	<i>file name</i>

## 5.3. writeable

*test if file writeable*

*static:* This command can be run outside of an object context.

### Usage

```
writeable [-failonerror] -file <>
```

## Options

Option	Description
failonerror	<i>error the command if test fails</i>
file	<i>file name</i>

## 5.4. readable

*test if file readable*

*static:* This command can be run outside of an object context.

### Usage

```
readable [-failonerror] -file <>
```

### Options

Option	Description
failonerror	<i>error the command if test fails</i>
file	<i>file name</i>

### 5.5. older

*test if file older than other file*

*static:* This command can be run outside of an object context.

#### Usage

```
older [-failonerror] -file <> -target <>
```

### Options

Option	Description
failonerror	<i>error the command if test fails</i>
file	<i>file name</i>
target	<i>target file name</i>

### 5.6. newer

*test if file newer than other file*

*static:* This command can be run outside of an object context.

#### Usage

```
newer [-failonerror] -file <> -target <>
```

### Options

Option	Description
failonerror	<i>error the command if test fails</i>
file	<i>file name</i>
target	<i>target file name</i>

### 5.7. ls

*fileutil*

*list directory*

*static:* This command can be run outside of an object context.

**Usage**

`ls -dir <> [-recursive]`

**Options**

Option	Description
<code>dir</code>	<i>directory name</i>
<code>recursive</code>	<i>list recursively</i>

## 5.8. mkdir

*make directory*

*static:* This command can be run outside of an object context.

**Usage**

`mkdir -dir <>`

**Options**

Option	Description
<code>dir</code>	<i>directory name</i>

## 5.9. rmdir

*remove directory*

*static:* This command can be run outside of an object context.

**Usage**

`rmdir -dir <> [-failonerror]`

**Options**

Option	Description
<code>dir</code>	<i>directory name</i>
<code>failonerror</code>	<i>if true fail on error</i>

## 5.10. copy

*copy file*

*static:* This command can be run outside of an object context.

**Usage**

```
copy [-backup] -file <> [-overwrite] [-suffix <.backup>]
-target <>
```

**Options**

Option	Description
backup	<i>if present make a backup if the target file exists and -overwrite is specified</i>
file	<i>file name</i>
overwrite	<i>if present overwrite any existing file</i>
suffix	<i>suffix of backup file</i>
target	<i>target file name</i>

**5.11. move***move file*

*static:* This command can be run outside of an object context.

**Usage**

```
move -file <> -target <>
```

**Options**

Option	Description
file	<i>file name</i>
target	<i>target file name</i>

**5.12. remove***remove file*

*static:* This command can be run outside of an object context.

**Usage**

```
remove [-failonerror] -file <>
```

**Options**

Option	Description
failonerror	<i>error the command if test fails</i>
file	<i>file name</i>

### 5.13. link

*make a link*

*static:* This command can be run outside of an object context.

#### Usage

```
link -file <> -target <>
```

#### Options

Option	Description
file	<i>link file to be created</i>
target	<i>file for link target</i>

### 5.14. touch

*touch a file*

*static:* This command can be run outside of an object context.

#### Usage

```
touch [-datetime <>] -file <>
```

#### Options

Option	Description
datetime	<i>specifies new mod time of the file</i>
file	<i>file name</i>