Unittest for `fstools`

May 17, 2026

Contents

1 Test Information **3**

1.1 Testobject Information . . . 3

1.2 Testdata Information . . . 3

1.3 Testsystem Information . . . 3

2 Statistic **3**

2.1 Test-Statistic for testrun: Library test . . . 3

2.2 Coverage Statistic . . . 4

3 Tested Requirements for testrun: Library test **5**

3.1 File and directory access . . . 5

3.1.1 Create a directory (REQ-0001) . . . 5

3.1.2 Create a directory which already exists (REQ-0002) . . . 5

3.1.3 Open file exclusive (non blocking) (REQ-0003) . . . 6

3.1.4 Open file exclusive (blocking) (REQ-0004) . . . 7

3.2 File and directory information . . . 7

3.2.1 Path is writable (REQ-0011) . . . 7

3.2.2 Determine a list of files (REQ-0012) . . . 8

3.2.3 Determine a list of files (REQ-0013) . . . 9

3.2.4 Determine a list of files (REQ-0014) . . . 10

3.2.5 Determine a list of files (REQ-0015) . . . 10

3.2.6 Determine a list of files (REQ-0016) . . . 11

3.2.7 Determine a list of directories (REQ-0021) . . . 12

3.2.8 Determine a list of directories (REQ-0022) . . . 12

3.2.9 Determine a list of directories (REQ-0023) . . . 13

3.2.10 Create a fast UID for a file /directory (REQ-0031) . . . 14

3.2.11 Create a UID for a file /directory (REQ-0032) . . . 14

3.3 File /Filelist UID generation . . . 15

A Trace for testrun: Library test **16**

A.1 Tests with status Info (15) . . . 16

A.1.1 Create a directory (REQ-0001) . . . 16

A.1.2 Create a directory which already exists (REQ-0002) . . . 17

A.1.3 Open file exclusive (non blocking) (REQ-0003) . . . 18

A.1.4 Open file exclusive (blocking) (REQ-0004) . . . 19

A.1.5 Path is writable (REQ-0011) . . . 20

A.1.6 Determine a list of files (REQ-0012) . . . 22

A.1.7 Determine a list of files (REQ-0013) . . . 23

A.1.8 Determine a list of files (REQ-0014) . . . 24

A.1.9 Determine a list of files (REQ-0015) . . . 25

A.1.10 Determine a list of files (REQ-0016) . . . 26

A.1.11 Determine a list of directories (REQ-0021) . . . 27

A.1.12 Determine a list of directories (REQ-0022) . . . 28

A.1.13 Determine a list of directories (REQ-0023) . . . 29

A.1.14 Create a fast UID for a file /directory (REQ-0031) . . . 30

A.1.15 Create a UID for a file /directory (REQ-0032) . . . 31

B Test-Coverage **32**

B.1 `fstools` . . . 32

B.1.1 `fstools.__init__.py` . . . 32

1 Test Information

1.1 Testobject Information

The Module fstools is designed to help on all issues with files and folders. For more Information read the documentation.

Information	
Name	fstools
State	Released
Version	0.4.8
Dependencies	
	Python standards

1.2 Testdata Information

Information	
Version	0.4.8
Testruns	Library test

1.3 Testsystem Information

System Information	
Architecture	64bit
Distribution	Debian GNU/Linux 13 trixie
Hostname	ahorn
Kernel	6.12.86+deb13-amd64 (#1 SMP PREEMPT_DYNAMIC Debian 6.12.86-1 (2026-05-08))
Machine	x86_64
Path	/home/dirk/work/unittest_collection/fstools/unittest/src
System	Linux
Username	dirk

2 Statistic

2.1 Test-Statistic for testrun: Library test

Number of tests	**15**
Number of successfull tests	**15**
Number of possibly failed tests	**0**
Number of failed tests	**0**
Executionlevel	Full Test (all defined tests)
Time consumption	0.592s

2.2 Coverage Statistic

Module- or Filename	Line-Coverage	Branch-Coverage
fstools	92.5%	91.7%
fstools.__init__.py	92.5%	

3 Tested Requirements for testrun: Library test

3.1 File and directory access

3.1.1 Create a directory (REQ-0001)

Description
There shall be a method, creating a directory.

Reason for the implementation
The standard python method will not create a directories recursively.

Fitcriterion
Create a directory where at least the last two subdirectories do not exist.

Testresult
This test was passed with the state: **Success**. See also full trace in section A.1.1!

Testrun:	
Caller:	/home/dirk/work/unittest_collection/fstools/unittest/src/tests/__init__.py (9)
Start-Time:	2026-05-17 16:12:04,218
Finished-Time:	2026-05-17 16:12:04,222
Time-Consumption	0.003s

Testsummary:	
Info	Files and folders for fstools unittest created under /home/dirk/work/unittest_collection/fstools/unittest/output_data/testfolder
Success	parentfolder /home/dirk/work/unittest_collection/fstools/unittest/output_data/testfolder/writable_folder/mkdir_test_folder does not exist is correct (Content False and Type is <class 'bool'>).
Info	Calling mkdir method to create /home/dirk/work/unittest_collection/fstools/unittest/output_data/testfolder/writable_folder/mkdir_test_folder/sub_folder
Success	testfolder /home/dirk/work/unittest_collection/fstools/unittest/output_data/testfolder/writable_folder/mkdir_test_folder/sub_folder exist is correct (Content True and Type is <class 'bool'>).
Success	Returnvalue of mkdir call is correct (Content True and Type is <class 'bool'>).

3.1.2 Create a directory which already exists (REQ-0002)

Description
There shall be a method, creating a directory. If the directory already exists, if shall not fail.

Reason for the implementation
Create a directory without need to determine, if the creation is needed.

Fitcriterion
Create a folder which already exists.

Testresult
This test was passed with the state: **Success**. See also full trace in section A.1.2!

Testrun:	
Caller:	/home/dirk/work/unittest_collection/fstools/unittest/src/tests/__init__.py (10)
Start-Time:	2026-05-17 16:12:04,222
Finished-Time:	2026-05-17 16:12:04,223
Time-Consumption	0.001s

Testsummary:

Success	testfolder /home/dirk/work/unittest_collection/fstools/unittest/output_data/testfolder already exist is correct (Content True and Type is <class 'bool'>).
Info	Calling mkdir method to create /home/dirk/work/unittest_collection/fstools/unittest/output_data/testfolder
Success	testfolder /home/dirk/work/unittest_collection/fstools/unittest/output_data/testfolder exist is correct (Content True and Type is <class 'bool'>).
Success	Returnvalue of mkdir call is correct (Content True and Type is <class 'bool'>).

3.1.3 Open file exclusive (non blocking) (REQ-0003)

Description
There shall be a method, opening a file exclusive and do not block.

Reason for the implementation
If you want to access a file from differnt threads, it might be needed to have exclusive file access.

Fitcriterion
Open a file and lock it and write text to the file. Try to open a file from a child process and write to it. Then write from the parent process additional text to the file. The Child process shall get a BlockingIOError exception. The file shall only contain the text written from the parent process.

Testresult
This test was passed with the state: **Success**. See also full trace in section A.1.3!

Testrun:	
Caller:	/home/dirk/work/unittest_collection/fstools/unittest/src/tests/__init__.py (11)
Start-Time:	2026-05-17 16:12:04,223
Finished-Time:	2026-05-17 16:12:04,229
Time-Consumption	0.006s

Testsummary:

Info	Files and folders for fstools unittest created under /home/dirk/work/unittest_collection/fstools/unittest/output_data/testfolder

Info	Opening a file with a non blocking lock, write to it, start child process and try writing to the file. Finally wite additional text from the parent process.
Success	Child process exception while open_locked_non_blocking is correct (Content 'BlockingIOError' and Type is <class 'str'>).
Success	Appended text to file is correct (Content 'parent_write_pattern_part_1::parent_write_pattern_part_2' and Type is <class 'str'>).

3.1.4 Open file exclusive (blocking) (REQ-0004)

Description

There shall be a method, opening a file exclusive and block till file is available again.

Reason for the implementation

If you want to access a file from differnt threads, it might be needed to have exclusive file access.

Fitcriterion

Open a file and lock it and write text to the file. Try to open a file from a child process and write to it. Then write from the parent process additional text to the file and close it. The Child process shall be blocked, till the parent process finished writing. Then the child process is able to write. The file shall contain the text written from the parent process followed by the text from the child process.

Testresult

This test was passed with the state: **Success**. See also full trace in section A.1.4!

Testrun:	
Caller:	/home/dirk/work/unittest_collection/fstools/unittest/src/tests/__init__.py (12)
Start-Time:	2026-05-17 16:12:04,229
Finished-Time:	2026-05-17 16:12:04,736
Time-Consumption	0.507s

Testsummary:

Info	Files and folders for fstools unittest created under /home/dirk/work/unittest_collection/fstools/unittest/output_data/testfolder
Info	Opening a file with a blocking lock, writing to it, sleep a bit, trying to open and write to it from a child process
Success	Appended text to file is correct (Content 'parent_write_pattern_part_1::parent_write_pattern_part_2::... and Type is <class 'str'>).

3.2 File and directory information

3.2.1 Path is writable (REQ-0011)

Description

There shall be a method, to identify, if a path is writable.

Reason for the implementation
The standard python method is not as easy as expexted and this is a very common issue.

Fitcriterion
Test the method at least with a writable, nonwritable file and directory. In addition with a non existing path.

Testresult
This test was passed with the state: **Success**. See also full trace in section A.1.5!

Testrun:	
Caller:	/home/dirk/work/unittest_collection/fstools/unittest/src/tests/__init__.py (15)
Start-Time:	2026-05-17 16:12:04,736
Finished-Time:	2026-05-17 16:12:04,744
Time-Consumption	0.007s

Testsummary:	
Info	Files and folders for fstools unittest created under /home/dirk/work/unittest_collection/fstools/unittest/output_data/testfolder
Success	Returnvalue of fstools.is_writable('.../writable_file.rw') is correct (Content True and Type is <class 'bool'>).
Success	Returnvalue of fstools.is_writable('.../not_writable_file.ro') is correct (Content False and Type is <class 'bool'>).
Success	Returnvalue of fstools.is_writable('.../writable_folder') is correct (Content True and Type is <class 'bool'>).
Success	Returnvalue of fstools.is_writable('.../not_writable_folder') is correct (Content False and Type is <class 'bool'>).
Success	Returnvalue of fstools.is_writable('.../not_existing_folder') is correct (Content False and Type is <class 'bool'>).

3.2.2 Determine a list of files (REQ-0012)

Description
There shall be a method, to get a list of files below a specified folder. The user shall be able to choose *recursive* and *all available files*.

Reason for the implementation
The standard python method isn't very comftable to provide filelists.

Fitcriterion
Create a testfolder with some files. Run the method and check the feedback against the expected filelist.

Testresult
This test was passed with the state: **Success**. See also full trace in section A.1.6!

Testrun:	
Caller:	/home/dirk/work/unittest_collection/fstools/unittest/src/tests/__init__.py (17)

Start-Time:	2026-05-17 16:12:04,744
Finished-Time:	2026-05-17 16:12:04,754
Time-Consumption	0.010s

Testsummary:	
Info	Files and folders for fstools unittest created under /home/dirk/work/unittest_collection/fstools/unittest/output_data/testfolder
Info	Executing filelist to get list for /home/dirk/work/unittest_collection/fstools/unittest/output_data/testfolder
Success	Returnvalue of filelist is correct (Content ['/home/dirk/work/unittest_collection/fstools/unittest/output_data/testfolder/not_writable_file.ro', '/home/dirk/work/unittest_collection/fstools/unittest/output_data/testfolder/writable_file.rw', '/home/dirk/work/unittest_collection/fstools/unittest/output_data/testfolder/writable_folder/subfolder/subfile.rw'] and Type is <class 'list'>).

3.2.3 Determine a list of files (REQ-0013)

Description

There shall be a method, to get a list of files below a specified folder. The user shall be able to choose *recursive* and *files matching a pattern*.

Reason for the implementation

The standard python method isn't very comftable to provide filelists.

Fitcriterion

Create a testfolder with some files. Run the method and check the feedback against the expected filelist.

Testresult

This test was passed with the state: **Success**. See also full trace in section A.1.7!

Testrun:	
Caller:	/home/dirk/work/unittest_collection/fstools/unittest/src/tests/__init__.py (18)
Start-Time:	2026-05-17 16:12:04,755
Finished-Time:	2026-05-17 16:12:04,761
Time-Consumption	0.007s

Testsummary:	
Info	Files and folders for fstools unittest created under /home/dirk/work/unittest_collection/fstools/unittest/output_data/testfolder
Info	Executing filelist to get list for /home/dirk/work/unittest_collection/fstools/unittest/output_data/testfolder
Success	Returnvalue of filelist is correct (Content ['/home/dirk/work/unittest_collection/fstools/unittest/output_data/testfolder/not_writable_file.ro'] and Type is <class 'list'>).

3.2.4 Determine a list of files (REQ-0014)

Description

There shall be a method, to get a list of files below a specified folder. The user shall be able to choose *non recursive* and *files matching a pattern*.

Reason for the implementation

The standard python method isn't very comftable to provide filelists.

Fitcriterion

Create a testfolder with some files. Run the method and check the feedback against the expected filelist.

Testresult

This test was passed with the state: **Success**. See also full trace in section A.1.8!

Testrun:	
Caller:	/home/dirk/work/unittest_collection/fstools/unittest/src/tests/__init__.py (19)
Start-Time:	2026-05-17 16:12:04,762
Finished-Time:	2026-05-17 16:12:04,768
Time-Consumption	0.006s

Testsummary:	
Info	Files and folders for fstools unittest created under /home/dirk/work/unittest_collection/fstools/unittest/output_data/testfolder
Info	Executing filelist to get list for /home/dirk/work/unittest_collection/fstools/unittest/output_data/testfolder
Success	Returnvalue of filelist is correct (Content ['/home/dirk/work/unittest_collection/fstools/unittest/output_data/testfolder/not_writable_file.ro', '/home/dirk/work/unittest_collection/fstools/unittest/output_data/testfolder/writable_file.rw'] and Type is <class 'list'>).

3.2.5 Determine a list of files (REQ-0015)

Description

There shall be a method, to get a list of files below a specified folder. The user shall be able to choose *non recursive* and *all available files*.

Reason for the implementation

The standard python method isn't very comftable to provide filelists.

Fitcriterion

Create a testfolder with some files. Run the method and check the feedback against the expected filelist.

Testresult

This test was passed with the state: **Success**. See also full trace in section A.1.9!

Testrun:	
Caller:	/home/dirk/work/unittest_collection/fstools/unittest/src/tests/__init__.py (20)
Start-Time:	2026-05-17 16:12:04,768
Finished-Time:	2026-05-17 16:12:04,775
Time-Consumption	0.007s

Testsummary:	
Info	Files and folders for fstools unittest created under /home/dirk/work/unittest_collection/fstools/unittest/output_data/testfolder
Info	Executing filelist to get list for /home/dirk/work/unittest_collection/fstools/unittest/output_data/testfolder
Success	Returnvalue of filelist is correct (Content ['/home/dirk/work/unittest_collection/fstools/unittest/output_data/testfolder/writable_file.rw'] and Type is <class 'list'>).

3.2.6 Determine a list of files (REQ-0016)

Description

There shall be a method, to get a list of files below a specified folder. If the folder does not exist, a empty list shall be returned

Reason for the implementation

The standard python method isn't very comftable to provide filelists.

Fitcriterion

Run the method, give a folder which does not exist as input and compare the feedback against a empty list.

Testresult

This test was passed with the state: **Success**. See also full trace in section A.1.10!

Testrun:	
Caller:	/home/dirk/work/unittest_collection/fstools/unittest/src/tests/__init__.py (21)
Start-Time:	2026-05-17 16:12:04,775
Finished-Time:	2026-05-17 16:12:04,781
Time-Consumption	0.006s

Testsummary:	
Info	Files and folders for fstools unittest created under /home/dirk/work/unittest_collection/fstools/unittest/output_data/testfolder
Info	Executing filelist to get list for /home/dirk/work/unittest_collection/fstools/unittest/output_data/testfolder/not_existing_folder
Success	Returnvalue of filelist is correct (Content [] and Type is <class 'list'>).

3.2.7 Determine a list of directories (REQ-0021)

Description

There shall be a method, to get a list of directories below a specified folder. The user shall be able to choose *recursive* search.

Reason for the implementation

The standard python method isn't very comftable to provide filelists.

Fitcriterion

Create a testfolder with some files. Run the method and check the feedback against the expected dirlist.

Testresult

This test was passed with the state: **Success**. See also full trace in section A.1.11!

Testrun:	
Caller:	/home/dirk/work/unittest_collection/fstools/unittest/src/tests/__init__.py (23)
Start-Time:	2026-05-17 16:12:04,781
Finished-Time:	2026-05-17 16:12:04,788
Time-Consumption	0.006s

Testsummary:	
Info	Files and folders for fstools unittest created under /home/dirk/work/unittest_collection/fstools/unittest/output_data/testfolder
Info	Executing dirlist to get list for /home/dirk/work/unittest_collection/fstools/unittest/output_data/testfolder
Success	Returnvalue of dirlist is correct (Content ['/home/dirk/work/unittest_collection/fstools/unittest/output_data/testfolder/writable_folder', '/home/dirk/work/unittest_collection/fstools/unittest/output_data/testfolder/writable_folder/subfolder', '/home/dirk/work/unittest_collection/fstools/unittest/output_data/testfolder/not_writable_folder'] and Type is <class 'list'>).

3.2.8 Determine a list of directories (REQ-0022)

Description

There shall be a method, to get a list of directories below a specified folder. The user shall be able to choose *non recursive* search.

Reason for the implementation

The standard python method isn't very comftable to provide filelists.

Fitcriterion

Create a testfolder with some files. Run the method and check the feedback against the expected dirlist.

Testresult

This test was passed with the state: **Success**. See also full trace in section A.1.12!

Testrun:	
Caller:	/home/dirk/work/unittest_collection/fstools/unittest/src/tests/__init__.py (24)
Start-Time:	2026-05-17 16:12:04,788
Finished-Time:	2026-05-17 16:12:04,796
Time-Consumption	0.008s

Testsummary:	
Info	Files and folders for fstools unittest created under /home/dirk/work/unittest_collection/fstools/unittest/output_data/testfolder
Info	Executing dirlist to get list for /home/dirk/work/unittest_collection/fstools/unittest/output_data/testfolder
Success	Returnvalue of dirlist is correct (Content ['/home/dirk/work/unittest_collection/fstools/unittest/output_data/testfolder/writable_folder', '/home/dirk/work/unittest_collection/fstools/unittest/output_data/testfolder/not_writable_folder'] and Type is <class 'list'>).

3.2.9 Determine a list of directories (REQ-0023)

Description

There shall be a method, to get a list of directories below a specified folder. If the folder does not exist, a empty list shall be returned

Reason for the implementation

The standard python method isn't very comftable to provide filelists.

Fitcriterion

Run the method, give a folder which does not exist as input and compare the feedback against a empty list.

Testresult

This test was passed with the state: **Success**. See also full trace in section A.1.13!

Testrun:	
Caller:	/home/dirk/work/unittest_collection/fstools/unittest/src/tests/__init__.py (25)
Start-Time:	2026-05-17 16:12:04,797
Finished-Time:	2026-05-17 16:12:04,802
Time-Consumption	0.006s

Testsummary:	
Info	Files and folders for fstools unittest created under /home/dirk/work/unittest_collection/fstools/unittest/output_data/testfolder
Info	Executing dirlist to get list for /home/dirk/work/unittest_collection/fstools/unittest/output_data/testfolder/not_existing_folder
Success	Returnvalue of dirlist is correct (Content [] and Type is <class 'list'>).

3.2.10 Create a fast UID for a file /directory (REQ-0031)

Description
There shall be a method, to create a UID with very low effort. The UID shall be generated from the file / directory metadata.

Reason for the implementation
Identify changes of a file / below a directory.

Fitcriterion
Create a foler and filestructure. Create the UID and compare it with the expected UID.

Testresult
This test was passed with the state: **Success**. See also full trace in section A.1.14!

Testrun:	
Caller:	/home/dirk/work/unittest_collection/fstools/unittest/src/tests/__init__.py (28)
Start-Time:	2026-05-17 16:12:04,803
Finished-Time:	2026-05-17 16:12:04,810
Time-Consumption	0.007s

Testsummary:	
Info	Files and folders for fstools unittest created under /home/dirk/work/unittest_collection/fstools/unittest/output_data/testfolder
Success	UID for a file is correct (Content '1894aa51f9f9267a9feb47e0cb0ae6d3f0abb8be' and Type is <class 'str'>).
Success	UID for a directory is correct (Content '813ef1c4d30ec81a3ef8f058bdc00fa605a9e8f8' and Type is <class 'str'>).

3.2.11 Create a UID for a file /directory (REQ-0032)

Description
There shall be a method, to create a UID. The UID shall be generated from the content in a file / specified files.

Reason for the implementation
Identify changes of a file / below a directory.

Fitcriterion
Create a foler and filestructure. Create the UID and compare it with the expected UID.

Testresult
This test was passed with the state: **Success**. See also full trace in section A.1.15!

Testrun:	
Caller:	/home/dirk/work/unittest_collection/fstools/unittest/src/tests/__init__.py (29)

Start-Time:	2026-05-17 16:12:04,810
Finished-Time:	2026-05-17 16:12:04,816
Time-Consumption	0.006s

Testsummary:

Info	Files and folders for fstools unittest created under /home/dirk/work/unittest_collection/fstools/unittest/output_data/testfolder
Success	Returnvalue of uid_filelist is correct (Content 'e8b354439db7dfdbdb4cbe4e8f9392b6' and Type is <class 'str'>).

3.3 File /Filelist UID generation

A Trace for testrun: Library test

A.1 Tests with status Info (15)

A.1.1 Create a directory (REQ-0001)

Description
There shall be a method, creating a directory.

Reason for the implementation
The standard python method will not create a directories recursively.

Fitcriterion
Create a directory where at least the last two subdirectories do not exist.

Testresult
This test was passed with the state: Success.

Info	Files and folders for fstools unittest created under /home/dirk/work/unittest_collection/fstools/unittest/output_data/testfolder

```
Creating general Basefolder
↪ /home/dirk/work/unittest_collection/fstools/unittest/output_data/testfolder
Creating folder in general Basefolder: .../writable_folder
Creating folder in general Basefolder: .../not_writable_folder
Creating file in general Basefolder: .../writable_file.rw
Creating file in general Basefolder: .../not_writable_file.ro
Creating folder in general Basefolder: .../writable_folder/subfolder
Creating file in general Basefolder: .../writable_folder/subfolder/subfile.rw
Creating UID-Basefolder
↪ /home/dirk/work/unittest_collection/fstools/unittest/output_data/uid_testfolder
Creating folder in UID-Basefolder: ...
Creating file in UID-Basefolder: .../uid_test_file
Creating folder in UID-Basefolder: .../uid_test_folder
```

Success	parentfolder /home/dirk/work/unittest_collection/fstools/unittest/output_data/testfolder/writable_folder/mkdir_test_folder does not exist is correct (Content False and Type is <class 'bool'>).

```
Result (parentfolder /home/dirk/work/unittest_collection/fstools/unittest/output_data/testfol
↪ der/writable_folder/mkdir_test_folder does not exist): False (<class 'bool'>)
Expectation (parentfolder /home/dirk/work/unittest_collection/fstools/unittest/output_data/te
↪ stfolder/writable_folder/mkdir_test_folder does not exist): result = False (<class
↪ 'bool'>)
```

Info	Calling mkdir method to create /home/dirk/work/unittest_collection/fstools/unittest/output_data/testfolder/writable_folder/mkdir_test_folder/sub_folder

Success	testfolder /home/dirk/work/unittest_collection/fstools/unittest/output_data/testfolder/writable_folder/mkdir_test_folder/sub_folder exist is correct (Content True and Type is <class 'bool'>).

```
Result (testfolder /home/dirk/work/unittest_collection/fstools/unittest/output_data/testfolde
↪ r/writable_folder/mkdir_test_folder/sub_folder exist): True (<class 'bool'>)
```

```
Expectation (testfolder /home/dirk/work/unittest_collection/fstools/unittest/output_data/test
↪ folder/writable_folder/mkdir_test_folder/sub_folder exist): result = True (<class 'bool'>)
```

Success	Returnvalue of mkdir call is correct (Content True and Type is <class 'bool'>).

```
Result (Returnvalue of mkdir call): True (<class 'bool'>)
```

```
Expectation (Returnvalue of mkdir call): result = True (<class 'bool'>)
```

A.1.2 Create a directory which already exists (REQ-0002)

Description

There shall be a method, creating a directory. If the directory already exists, if shall not fail.

Reason for the implementation

Create a directory without need to determine, if the creation is needed.

Fitcriterion

Create a folder which already exists.

Testresult

This test was passed with the state: **Success**.

Success	testfolder /home/dirk/work/unittest_collection/fstools/unittest/output_data/testfolder already exist is correct (Content True and Type is <class 'bool'>).

```
Result (testfolder /home/dirk/work/unittest_collection/fstools/unittest/output_data/testfolder
↪ already exist): True (<class 'bool'>)
```

```
Expectation (testfolder
↪ /home/dirk/work/unittest_collection/fstools/unittest/output_data/testfolder already
↪ exist): result = True (<class 'bool'>)
```

Info	Calling mkdir method to create /home/dirk/work/unittest_collection/fstools/unittest/output_data/testfolder

Success	testfolder /home/dirk/work/unittest_collection/fstools/unittest/output_data/testfolder exist is correct (Content True and Type is <class 'bool'>).

```
Result (testfolder /home/dirk/work/unittest_collection/fstools/unittest/output_data/testfolder
↪  exist): True (<class 'bool'>)
```

```
Expectation (testfolder
↪  /home/dirk/work/unittest_collection/fstools/unittest/output_data/testfolder exist): result
↪  = True (<class 'bool'>)
```

Success	Returnvalue of mkdir call is correct (Content True and Type is <class 'bool'>).

```
Result (Returnvalue of mkdir call): True (<class 'bool'>)
```

```
Expectation (Returnvalue of mkdir call): result = True (<class 'bool'>)
```

A.1.3 Open file exclusive (non blocking) (REQ-0003)

Description

There shall be a method, opening a file exclusive and do not block.

Reason for the implementation

If you want to access a file from differnt threads, it might be needed to have exclusive file access.

Fitcriterion

Open a file and lock it and write text to the file. Try to open a file from a child process and write to it. Then write from the parent process additional text to the file. The Child process shall get a BlockingIOError exception. The file shall only contain the text written from the parent process.

Testresult

This test was passed with the state: **Success**.

Info	Files and folders for fstools unittest created under /home/dirk/work/unittest_collection/fstools/unittest/output_data/testfolder

```
Creating general Basefolder
↪  /home/dirk/work/unittest_collection/fstools/unittest/output_data/testfolder
```

```
Creating folder in general Basefolder: .../writable_folder
```

```
Creating folder in general Basefolder: .../not_writable_folder
```

```
Creating file in general Basefolder: .../writable_file.rw
```

```
Creating file in general Basefolder: .../not_writable_file.ro
```

```
Creating folder in general Basefolder: .../writable_folder/subfolder
Creating file in general Basefolder: .../writable_folder/subfolder/subfile.rw
Creating UID-Basefolder
↪ /home/dirk/work/unittest_collection/fstools/unittest/output_data/uid_testfolder
Creating folder in UID-Basefolder: ...
Creating file in UID-Basefolder: .../uid_test_file
Creating folder in UID-Basefolder: .../uid_test_folder
```

Info	Opening a file with a non blocking lock, write to it, start child process and try writing to the file. Finally wite additional text from the parent process.

Success	Child process exception while open_locked_non_blocking is correct (Content 'BlockingIOError' and Type is <class 'str'>).

```
Result (Child process exception while open_locked_non_blocking): 'BlockingIOError' (<class
↪ 'str'>)
Expectation (Child process exception while open_locked_non_blocking): result =
↪ 'BlockingIOError' (<class 'str'>)
```

Success	Appended text to file is correct (Content 'parent_write_pattern_part_1::parent_write_pattern_part_2' and Type is <class 'str'>).

```
Result (Appended text to file): 'parent_write_pattern_part_1::parent_write_pattern_part_2'
↪ (<class 'str'>)
Expectation (Appended text to file): result =
↪ 'parent_write_pattern_part_1::parent_write_pattern_part_2' (<class 'str'>)
```

A.1.4 Open file exclusive (blocking) (REQ-0004)

Description

There shall be a method, opening a file exclusive and block till file is available again.

Reason for the implementation

If you want to access a file from differnt threads, it might be needed to have exclusive file access.

Fitcriterion

Open a file and lock it and write text to the file. Try to open a file from a child process and write to it. Then write from the parent process additional text to the file and close it. The Child process shall be blocked, till the parent process finished writing. Then the child process is able to write. The file shall contain the text written from the parent process followed by the text from the child process.

Testresult
This test was passed with the state: **Success**.

Info	Files and folders for fstools unittest created under /home/dirk/work/unittest_collection/fstools/unittest/output_data/testfolder

```
Creating general Basefolder
↪ /home/dirk/work/unittest_collection/fstools/unittest/output_data/testfolder
Creating folder in general Basefolder: .../writable_folder
Creating folder in general Basefolder: .../not_writable_folder
Creating file in general Basefolder: .../writable_file.rw
Creating file in general Basefolder: .../not_writable_file.ro
Creating folder in general Basefolder: .../writable_folder/subfolder
Creating file in general Basefolder: .../writable_folder/subfolder/subfile.rw
Creating UID-Basefolder
↪ /home/dirk/work/unittest_collection/fstools/unittest/output_data/uid_testfolder
Creating folder in UID-Basefolder: ...
Creating file in UID-Basefolder: .../uid_test_file
Creating folder in UID-Basefolder: .../uid_test_folder
```

Info	Opening a file with a blocking lock, writing to it, sleep a bit, trying to open and write to it from a child process

Success	Appended text to file is correct (Content 'parent_write_pattern_part_1::parent_write_pattern_part_2::child_writ and Type is <class 'str'>).

```
Result (Appended text to file):
↪ 'parent_write_pattern_part_1::parent_write_pattern_part_2::child_write_pattern' (<class
↪ 'str'>)
Expectation (Appended text to file): result =
↪ 'parent_write_pattern_part_1::parent_write_pattern_part_2::child_write_pattern' (<class
↪ 'str'>)
```

A.1.5 Path is writable (REQ-0011)

Description
There shall be a method, to identify, if a path is writable.

Reason for the implementation
The standard python method is not as easy as expexted and this is a very common issue.

Fitcriterion
Test the method at least with a writable, nonwritable file and directory. In addition with a non existing path.

Testresult

This test was passed with the state: **Success**.

Info	Files and folders for fstools unittest created under /home/dirk/work/unittest_collection/fstools/unittest/output_data/testfolder

```
Creating general Basefolder
↪ /home/dirk/work/unittest_collection/fstools/unittest/output_data/testfolder
Creating folder in general Basefolder: .../writable_folder
Creating folder in general Basefolder: .../not_writable_folder
Creating file in general Basefolder: .../writable_file.rw
Creating file in general Basefolder: .../not_writable_file.ro
Creating folder in general Basefolder: .../writable_folder/subfolder
Creating file in general Basefolder: .../writable_folder/subfolder/subfile.rw
Creating UID-Basefolder
↪ /home/dirk/work/unittest_collection/fstools/unittest/output_data/uid_testfolder
Creating folder in UID-Basefolder: ...
Creating file in UID-Basefolder: .../uid_test_file
Creating folder in UID-Basefolder: .../uid_test_folder
```

Success	Returnvalue of fstools.is_writable('.../writable_file.rw') is correct (Content True and Type is <class 'bool'>).

```
Result (Returnvalue of fstools.is_writable(".../writable_file.rw")): True (<class 'bool'>)
Expectation (Returnvalue of fstools.is_writable(".../writable_file.rw")): result = True
↪ (<class 'bool'>)
```

Success	Returnvalue of fstools.is_writable('.../not_writable_file.ro') is correct (Content False and Type is <class 'bool'>).

```
Result (Returnvalue of fstools.is_writable(".../not_writable_file.ro")): False (<class
↪ 'bool'>)
Expectation (Returnvalue of fstools.is_writable(".../not_writable_file.ro")): result = False
↪ (<class 'bool'>)
```

Success	Returnvalue of fstools.is_writable('.../writable_folder') is correct (Content True and Type is <class 'bool'>).

```
Result (Returnvalue of fstools.is_writable(".../writable_folder")): True (<class 'bool'>)
Expectation (Returnvalue of fstools.is_writable(".../writable_folder")): result = True (<class
↪ 'bool'>)
```

Success	Returnvalue of fstools.is_writable('.../not_writable_folder') is correct (Content False and Type is <class 'bool'>).

```
Result (Returnvalue of fstools.is_writable(".../not_writable_folder")): False (<class 'bool'>)
```

```
Expectation (Returnvalue of fstools.is_writable(".../not_writable_folder")): result = False
↪  (<class 'bool'>)
```

Success	Returnvalue of fstools.is_writable('.../not_existing_folder') is correct (Content False and Type is <class 'bool'>).

```
Result (Returnvalue of fstools.is_writable(".../not_existing_folder")): False (<class 'bool'>)
```

```
Expectation (Returnvalue of fstools.is_writable(".../not_existing_folder")): result = False
↪  (<class 'bool'>)
```

A.1.6 Determine a list of files (REQ-0012)

Description

There shall be a method, to get a list of files below a specified folder. The user shall be able to choose *recursive* and *all available files*.

Reason for the implementation

The standard python method isn't very comftable to provide filelists.

Fitcriterion

Create a testfolder with some files. Run the method and check the feedback against the expected filelist.

Testresult

This test was passed with the state: **Success**.

Info	Files and folders for fstools unittest created under /home/dirk/work/unittest_collection/fstools/unittest/output_data/testfolder

```
Creating general Basefolder
↪  /home/dirk/work/unittest_collection/fstools/unittest/output_data/testfolder
Creating folder in general Basefolder: .../writable_folder
Creating folder in general Basefolder: .../not_writable_folder
Creating file in general Basefolder: .../writable_file.rw
Creating file in general Basefolder: .../not_writable_file.ro
Creating folder in general Basefolder: .../writable_folder/subfolder
Creating file in general Basefolder: .../writable_folder/subfolder/subfile.rw
Creating UID-Basefolder
↪  /home/dirk/work/unittest_collection/fstools/unittest/output_data/uid_testfolder
Creating folder in UID-Basefolder: ...
Creating file in UID-Basefolder: .../uid_test_file
Creating folder in UID-Basefolder: .../uid_test_folder
```

Info	Executing filelist to get list for /home/dirk/work/unittest_collection/fstools/unittest/output_data/testfolder

Success	Returnvalue of filelist is correct (Content ['/home/dirk/work/unittest_collection/fstools/unittest/output_data/testfolder/not_writable_file.ro', '/home/dirk/work/unittest_collection/fstools/unittest/output_data/testfolder/writable_file.rw', '/home/dirk/work/unittest_collection/fstools/unittest/output_data/testfolder/writable_folder/subfolder/subfile.rw'] and Type is <class 'list'>).

```
Result (Returnvalue of filelist): [ '/home/dirk/work/unittest_collection/fstools/unittest/out
↪ put_data/testfolder/not_writable_file.ro',
↪ '/home/dirk/work/unittest_collection/fstools/unittest/output_data/testfolder/writable_fil
↪ e.rw',
↪ '/home/dirk/work/unittest_collection/fstools/unittest/output_data/testfolder/writable_fol
↪ der/subfolder/subfile.rw' ] (<class 'list'>)
```

```
Expectation (Returnvalue of filelist): result = [ '/home/dirk/work/unittest_collection/fstool
↪ s/unittest/output_data/testfolder/writable_file.rw',
↪ '/home/dirk/work/unittest_collection/fstools/unittest/output_data/testfolder/not_writable
↪ _file.ro',
↪ '/home/dirk/work/unittest_collection/fstools/unittest/output_data/testfolder/writable_fol
↪ der/subfolder/subfile.rw' ] (<class 'list'>)
```

```
Content '/home/dirk/work/unittest_collection/fstools/unittest/output_data/testfolder/not_writ
↪ able_file.ro' is incorrect for test_variable.
```

A.1.7 Determine a list of files (REQ-0013)

Description

There shall be a method, to get a list of files below a specified folder. The user shall be able to choose *recursive* and *files matching a pattern*.

Reason for the implementation

The standard python method isn't very comftable to provide filelists.

Fitcriterion

Create a testfolder with some files. Run the method and check the feedback against the expected filelist.

Testresult

This test was passed with the state: **Success**.

Info	Files and folders for fstools unittest created under /home/dirk/work/unittest_collection/fstools/unittest/output_data/testfolder

```
Creating general Basefolder
↪ /home/dirk/work/unittest_collection/fstools/unittest/output_data/testfolder
```

```
Creating folder in general Basefolder: .../writable_folder
```

```
Creating folder in general Basefolder: .../not_writable_folder
Creating file in general Basefolder: .../writable_file.rw
Creating file in general Basefolder: .../not_writable_file.ro
Creating folder in general Basefolder: .../writable_folder/subfolder
Creating file in general Basefolder: .../writable_folder/subfolder/subfile.rw
Creating UID-Basefolder
↪  /home/dirk/work/unittest_collection/fstools/unittest/output_data/uid_testfolder
Creating folder in UID-Basefolder: ...
Creating file in UID-Basefolder: .../uid_test_file
Creating folder in UID-Basefolder: .../uid_test_folder
```

Info	Executing filelist to get list for /home/dirk/work/unittest_collection/fstools/unittest/output_data/testfolder

Success	Returnvalue of filelist is correct (Content ['/home/dirk/work/unittest_collection/fstools/unittest/output_data/testfolder/not_writable_file.ro'] and Type is <class 'list'>).

```
Result (Returnvalue of filelist): [ '/home/dirk/work/unittest_collection/fstools/unittest/out
↪  put_data/testfolder/not_writable_file.ro' ] (<class 'list'>)
Expectation (Returnvalue of filelist): result = [ '/home/dirk/work/unittest_collection/fstool
↪  s/unittest/output_data/testfolder/not_writable_file.ro' ] (<class 'list'>)
```

A.1.8 Determine a list of files (REQ-0014)

Description

There shall be a method, to get a list of files below a specified folder. The user shall be able to choose *non recursive* and *files matching a pattern*.

Reason for the implementation

The standard python method isn't very comftable to provide filelists.

Fitcriterion

Create a testfolder with some files. Run the method and check the feedback against the expected filelist.

Testresult

This test was passed with the state: **Success**.

Info	Files and folders for fstools unittest created under /home/dirk/work/unittest_collection/fstools/unittest/output_data/testfolder

```
Creating general Basefolder
↪  /home/dirk/work/unittest_collection/fstools/unittest/output_data/testfolder
Creating folder in general Basefolder: .../writable_folder
Creating folder in general Basefolder: .../not_writable_folder
Creating file in general Basefolder: .../writable_file.rw
```

```
Creating file in general Basefolder: .../not_writable_file.ro
Creating folder in general Basefolder: .../writable_folder/subfolder
Creating file in general Basefolder: .../writable_folder/subfolder/subfile.rw
Creating UID-Basefolder
↪ /home/dirk/work/unittest_collection/fstools/unittest/output_data/uid_testfolder
Creating folder in UID-Basefolder: ...
Creating file in UID-Basefolder: .../uid_test_file
Creating folder in UID-Basefolder: .../uid_test_folder
```

Info	Executing filelist to get list for /home/dirk/work/unittest_collection/fstools/unittest/output_data/testfolder

Success	Returnvalue of filelist is correct (Content ['/home/dirk/work/unittest_collection/fstools/unittest/output_data/testfolder/not_writable_file.ro', '/home/dirk/work/unittest_collection/fstools/unittest/output_data/testfolder/writable_file.rw'] and Type is <class 'list'>).

```
Result (Returnvalue of filelist): [ '/home/dirk/work/unittest_collection/fstools/unittest/out
↪ put_data/testfolder/not_writable_file.ro',
↪ '/home/dirk/work/unittest_collection/fstools/unittest/output_data/testfolder/writable_fil
↪ e.rw' ] (<class 'list'>)
Expectation (Returnvalue of filelist): result = [ '/home/dirk/work/unittest_collection/fstool
↪ s/unittest/output_data/testfolder/writable_file.rw',
↪ '/home/dirk/work/unittest_collection/fstools/unittest/output_data/testfolder/not_writable
↪ _file.ro' ] (<class 'list'>)
Content '/home/dirk/work/unittest_collection/fstools/unittest/output_data/testfolder/not_writ
↪ able_file.ro' is incorrect for test_variable.
```

A.1.9 Determine a list of files (REQ-0015)

Description

There shall be a method, to get a list of files below a specified folder. The user shall be able to choose *non recursive* and *all available files*.

Reason for the implementation

The standard python method isn't very comftable to provide filelists.

Fitcriterion

Create a testfolder with some files. Run the method and check the feedback against the expected filelist.

Testresult

This test was passed with the state: **Success**.

Info	Files and folders for fstools unittest created under /home/dirk/work/unittest_collection/fstools/unittest/output_data/testfolder

```
Creating general Basefolder
↪ /home/dirk/work/unittest_collection/fstools/unittest/output_data/testfolder
```

```
Creating folder in general Basefolder: .../writable_folder
Creating folder in general Basefolder: .../not_writable_folder
Creating file in general Basefolder: .../writable_file.rw
Creating file in general Basefolder: .../not_writable_file.ro
Creating folder in general Basefolder: .../writable_folder/subfolder
Creating file in general Basefolder: .../writable_folder/subfolder/subfile.rw
Creating UID-Basefolder
↪ /home/dirk/work/unittest_collection/fstools/unittest/output_data/uid_testfolder
Creating folder in UID-Basefolder: ...
Creating file in UID-Basefolder: .../uid_test_file
Creating folder in UID-Basefolder: .../uid_test_folder
```

Info	Executing filelist to get list for /home/dirk/work/unittest_collection/fstools/unittest/output_data/testfolder

Success	Returnvalue of filelist is correct (Content ['/home/dirk/work/unittest_collection/fstools/unittest/output_data/testfolder/writable_file.rw'] and Type is <class 'list'>).

```
Result (Returnvalue of filelist): [ '/home/dirk/work/unittest_collection/fstools/unittest/out
↪ put_data/testfolder/writable_file.rw' ] (<class 'list'>)
Expectation (Returnvalue of filelist): result = [ '/home/dirk/work/unittest_collection/fstool
↪ s/unittest/output_data/testfolder/writable_file.rw' ] (<class 'list'>)
```

A.1.10 Determine a list of files (REQ-0016)

Description

There shall be a method, to get a list of files below a specified folder. If the folder does not exist, a empty list shall be returned

Reason for the implementation

The standard python method isn't very comftable to provide filelists.

Fitcriterion

Run the method, give a folder which does not exist as input and compare the feedback against a empty list.

Testresult

This test was passed with the state: **Success**.

Info	Files and folders for fstools unittest created under /home/dirk/work/unittest_collection/fstools/unittest/output_data/testfolder

```
Creating general Basefolder
↪ /home/dirk/work/unittest_collection/fstools/unittest/output_data/testfolder
Creating folder in general Basefolder: .../writable_folder
Creating folder in general Basefolder: .../not_writable_folder
```

```
Creating file in general Basefolder: .../writable_file.rw
Creating file in general Basefolder: .../not_writable_file.ro
Creating folder in general Basefolder: .../writable_folder/subfolder
Creating file in general Basefolder: .../writable_folder/subfolder/subfile.rw
Creating UID-Basefolder
↪ /home/dirk/work/unittest_collection/fstools/unittest/output_data/uid_testfolder
Creating folder in UID-Basefolder: ...
Creating file in UID-Basefolder: .../uid_test_file
Creating folder in UID-Basefolder: .../uid_test_folder
```

Info	Executing filelist to get list for /home/dirk/work/unittest_collection/fstools/unittest/output_data/testfolder/not_existing_folder

Success	Returnvalue of filelist is correct (Content [] and Type is <class 'list'>).

```
Result (Returnvalue of filelist): [  ] (<class 'list'>)
Expectation (Returnvalue of filelist): result = [  ] (<class 'list'>)
```

A.1.11 Determine a list of directories (REQ-0021)

Description

There shall be a method, to get a list of directories below a specified folder. The user shall be able to choose *recursive* search.

Reason for the implementation

The standard python method isn't very comftable to provide filelists.

Fitcriterion

Create a testfolder with some files. Run the method and check the feedback against the expected dirlist.

Testresult

This test was passed with the state: **Success**.

Info	Files and folders for fstools unittest created under /home/dirk/work/unittest_collection/fstools/unittest/output_data/testfolder

```
Creating general Basefolder
↪ /home/dirk/work/unittest_collection/fstools/unittest/output_data/testfolder
Creating folder in general Basefolder: .../writable_folder
Creating folder in general Basefolder: .../not_writable_folder
Creating file in general Basefolder: .../writable_file.rw
Creating file in general Basefolder: .../not_writable_file.ro
Creating folder in general Basefolder: .../writable_folder/subfolder
```

```
Creating file in general Basefolder: .../writable_folder/subfolder/subfile.rw
Creating UID-Basefolder
↪ /home/dirk/work/unittest_collection/fstools/unittest/output_data/uid_testfolder
Creating folder in UID-Basefolder: ...
Creating file in UID-Basefolder: .../uid_test_file
Creating folder in UID-Basefolder: .../uid_test_folder
```

Info	Executing dirlist to get list for /home/dirk/work/unittest_collection/fstools/unittest/output_data/testfolder

Success	Returnvalue of dirlist is correct (Content ['/home/dirk/work/unittest_collection/fstools/unittest/output_data/testfolder/writable_folder', '/home/dirk/work/unittest_collection/fstools/unittest/output_data/testfolder/writable_folder/subfolder', '/home/dirk/work/unittest_collection/fstools/unittest/output_data/testfolder/not_writable_folder'] and Type is <class 'list'>).

```
Result (Returnvalue of dirlist): [ '/home/dirk/work/unittest_collection/fstools/unittest/outp
↪ ut_data/testfolder/writable_folder',
↪ '/home/dirk/work/unittest_collection/fstools/unittest/output_data/testfolder/writable_fol
↪ der/subfolder',
↪ '/home/dirk/work/unittest_collection/fstools/unittest/output_data/testfolder/not_writable
↪ _folder' ] (<class 'list'>)
Expectation (Returnvalue of dirlist): result = [ '/home/dirk/work/unittest_collection/fstools
↪ /unittest/output_data/testfolder/writable_folder',
↪ '/home/dirk/work/unittest_collection/fstools/unittest/output_data/testfolder/not_writable
↪ _folder',
↪ '/home/dirk/work/unittest_collection/fstools/unittest/output_data/testfolder/writable_fol
↪ der/subfolder' ] (<class 'list'>)
Content '/home/dirk/work/unittest_collection/fstools/unittest/output_data/testfolder/writable
↪ _folder/subfolder' is incorrect for test_variable.
```

A.1.12 Determine a list of directories (REQ-0022)

Description

There shall be a method, to get a list of directories below a specified folder. The user shall be able to choose *non recursive* search.

Reason for the implementation

The standard python method isn't very comftable to provide filelists.

Fitcriterion

Create a testfolder with some files. Run the method and check the feedback against the expected dirlist.

Testresult

This test was passed with the state: **Success**.

Info	Files and folders for fstools unittest created under /home/dirk/work/unittest_collection/fstools/unittest/output_data/testfolder

```
Creating general Basefolder
↪ /home/dirk/work/unittest_collection/fstools/unittest/output_data/testfolder
Creating folder in general Basefolder: .../writable_folder
Creating folder in general Basefolder: .../not_writable_folder
Creating file in general Basefolder: .../writable_file.rw
Creating file in general Basefolder: .../not_writable_file.ro
Creating folder in general Basefolder: .../writable_folder/subfolder
Creating file in general Basefolder: .../writable_folder/subfolder/subfile.rw
Creating UID-Basefolder
↪ /home/dirk/work/unittest_collection/fstools/unittest/output_data/uid_testfolder
Creating folder in UID-Basefolder: ...
Creating file in UID-Basefolder: .../uid_test_file
Creating folder in UID-Basefolder: .../uid_test_folder
```

Info	Executing dirlist to get list for /home/dirk/work/unittest_collection/fstools/unittest/output_data/testfolder

Success	Returnvalue of dirlist is correct (Content ['/home/dirk/work/unittest_collection/fstools/unittest/output_data/testfolder/writable_folder', '/home/dirk/work/unittest_collection/fstools/unittest/output_data/testfolder/not_writable_folder'] and Type is <class 'list'>).

```
Result (Returnvalue of dirlist): [ '/home/dirk/work/unittest_collection/fstools/unittest/outp⌋
↪ ut_data/testfolder/writable_folder',
↪ '/home/dirk/work/unittest_collection/fstools/unittest/output_data/testfolder/not_writable⌋
↪ _folder' ] (<class 'list'>)
Expectation (Returnvalue of dirlist): result = [ '/home/dirk/work/unittest_collection/fstools⌋
↪ /unittest/output_data/testfolder/writable_folder',
↪ '/home/dirk/work/unittest_collection/fstools/unittest/output_data/testfolder/not_writable⌋
↪ _folder' ] (<class 'list'>)
```

A.1.13 Determine a list of directories (REQ-0023)

Description

There shall be a method, to get a list of directories below a specified folder. If the folder does not exist, a empty list shall be returned

Reason for the implementation

The standard python method isn't very comftable to provide filelists.

Fitcriterion
Run the method, give a folder which does not exist as input and compare the feedback against a empty list.

Testresult
This test was passed with the state: **Success**.

Info	Files and folders for fstools unittest created under /home/dirk/work/unittest_collection/fstools/unittest/output_data/testfolder

```
Creating general Basefolder
↪ /home/dirk/work/unittest_collection/fstools/unittest/output_data/testfolder
Creating folder in general Basefolder: .../writable_folder
Creating folder in general Basefolder: .../not_writable_folder
Creating file in general Basefolder: .../writable_file.rw
Creating file in general Basefolder: .../not_writable_file.ro
Creating folder in general Basefolder: .../writable_folder/subfolder
Creating file in general Basefolder: .../writable_folder/subfolder/subfile.rw
Creating UID-Basefolder
↪ /home/dirk/work/unittest_collection/fstools/unittest/output_data/uid_testfolder
Creating folder in UID-Basefolder: ...
Creating file in UID-Basefolder: .../uid_test_file
Creating folder in UID-Basefolder: .../uid_test_folder
```

Info	Executing dirlist to get list for /home/dirk/work/unittest_collection/fstools/unittest/output_data/testfolder/not_existing_folder

Success	Returnvalue of dirlist is correct (Content [] and Type is <class 'list'>).

```
Result (Returnvalue of dirlist): [  ] (<class 'list'>)
Expectation (Returnvalue of dirlist): result = [  ] (<class 'list'>)
```

A.1.14 Create a fast UID for a file /directory (REQ-0031)

Description
There shall be a method, to create a UID with very low effort. The UID shall be generated from the file / directory metadata.

Reason for the implementation
Identify changes of a file / below a directory.

Fitcriterion
Create a foler and filestructure. Create the UID and compare it with the expected UID.

Testresult
This test was passed with the state: **Success**.

Info	Files and folders for fstools unittest created under /home/dirk/work/unittest_collection/fstools/unittest/output_data/testfolder

```
Creating general Basefolder
↪ /home/dirk/work/unittest_collection/fstools/unittest/output_data/testfolder
Creating folder in general Basefolder: .../writable_folder
Creating folder in general Basefolder: .../not_writable_folder
Creating file in general Basefolder: .../writable_file.rw
Creating file in general Basefolder: .../not_writable_file.ro
Creating folder in general Basefolder: .../writable_folder/subfolder
Creating file in general Basefolder: .../writable_folder/subfolder/subfile.rw
Creating UID-Basefolder
↪ /home/dirk/work/unittest_collection/fstools/unittest/output_data/uid_testfolder
Creating folder in UID-Basefolder: ...
Creating file in UID-Basefolder: .../uid_test_file
Creating folder in UID-Basefolder: .../uid_test_folder
```

Success	UID for a file is correct (Content '1894aa51f9f9267a9feb47e0cb0ae6d3f0abb8be' and Type is <class 'str'>).

```
Result (UID for a file): '1894aa51f9f9267a9feb47e0cb0ae6d3f0abb8be' (<class 'str'>)
Expectation (UID for a file): result = '1894aa51f9f9267a9feb47e0cb0ae6d3f0abb8be' (<class
↪ 'str'>)
```

Success	UID for a directory is correct (Content '813ef1c4d30ec81a3ef8f058bdc00fa605a9e8f8' and Type is <class 'str'>).

```
Result (UID for a directory): '813ef1c4d30ec81a3ef8f058bdc00fa605a9e8f8' (<class 'str'>)
Expectation (UID for a directory): result = '813ef1c4d30ec81a3ef8f058bdc00fa605a9e8f8' (<class
↪ 'str'>)
```

A.1.15 Create a UID for a file /directory (REQ-0032)

Description
There shall be a method, to create a UID. The UID shall be generated from the content in a file / specified files.

Reason for the implementation
Identify changes of a file / below a directory.

Fitcriterion
Create a foler and filestructure. Create the UID and compare it with the expected UID.

Testresult

This test was passed with the state: **Success**.

Info	Files and folders for fstools unittest created under /home/dirk/work/unittest_collection/fstools/unittest/output_data/testfolder

```
Creating general Basefolder
↪ /home/dirk/work/unittest_collection/fstools/unittest/output_data/testfolder
Creating folder in general Basefolder: .../writable_folder
Creating folder in general Basefolder: .../not_writable_folder
Creating file in general Basefolder: .../writable_file.rw
Creating file in general Basefolder: .../not_writable_file.ro
Creating folder in general Basefolder: .../writable_folder/subfolder
Creating file in general Basefolder: .../writable_folder/subfolder/subfile.rw
Creating UID-Basefolder
↪ /home/dirk/work/unittest_collection/fstools/unittest/output_data/uid_testfolder
Creating folder in UID-Basefolder: ...
Creating file in UID-Basefolder: .../uid_test_file
Creating folder in UID-Basefolder: .../uid_test_folder
```

Success	Returnvalue of uid_filelist is correct (Content 'e8b354439db7dfdbdb4cbe4e8f9392b6' and Type is <class 'str'>).

```
Result (Returnvalue of uid_filelist): 'e8b354439db7dfdbdb4cbe4e8f9392b6' (<class 'str'>)
Expectation (Returnvalue of uid_filelist): result = 'e8b354439db7dfdbdb4cbe4e8f9392b6' (<class
↪ 'str'>)
```

B Test-Coverage

B.1 fstools

The line coverage for fstools was 92.5%
The branch coverage for fstools was 91.7%

B.1.1 fstools.__init__.py

The line coverage for fstools.__init__.py was 92.5%
The branch coverage for fstools.__init__.py was 91.7%

```
"""
fstools (Filesystem Tools)
==========================

**Author:**

* Dirk Alders <sudo-dirk@mount-mockery.de>

```

```
**Description:**

    This module supports functions and classes to handle files and paths

**Submodules:**

* :func:`fstools.dirlist`
* :func:`fstools.filelist`
* :func:`fstools.is_writeable`
* :func:`fstools.mkdir`
* :func:`fstools.open_locked_blocking`
* :func:`fstools.open_locked_non_blocking`
* :func:`fstools.uid`

**Unittest:**

    See also the :download:`unittest <fstools/_testresults_/unittest.pdf>` documentation.

**Module Documentation:**

"""

__VERSION__ = "0.4.8"
__DEPENDENCIES__ = ["Python standards"]

import glob
import hashlib
import hmac
import logging
import os
import time
from functools import partial

try:
    from config import APP_NAME as ROOT_LOGGER_NAME
except ImportError:
    ROOT_LOGGER_NAME = "root"
logger = logging.getLogger(ROOT_LOGGER_NAME).getChild(__name__)


try:
    import fcntl
except ImportError:
    logger.warning('Importing "fcntl" was not possible. Only i limited functionality of fstools
    is available.')


__DESCRIPTION__ = """The Module {\\tt %s} is designed to help on all issues with files and
    folders.
For more Information read the documentation.""" % __name__.replace("_", "\\_")
"""The Module Description"""


def dirlist(path=".", expression="*", rekursive=True):
    """
    Function returning a list of directories below a given path.

    :param str path: folder which is the basepath for searching files.
    :param str expression: expression to fit including shell-style wildcards. It is only used for
     the first directory level, if rekursive is set to True.
    :param bool rekursive: search all subfolders if True.
    :returns: list of filenames including the pathe
```

```
68      :rtype: list
69
70      .. note:: The returned filenames could be relative pathes depending on argument path.
71
72      **Example:**
73
74      .. literalinclude:: fstools/_examples_/dirlist.py
75
76      .. literalinclude:: fstools/_examples_/dirlist.log
77      """
78      li = list()
79      if os.path.exists(path):
80          logger.debug("DIRLIST: path (%s) exists - looking for directories to append", path)
81          for dirname in glob.glob(os.path.join(path, expression)):
82              if os.path.isdir(dirname):
83                  li.append(dirname)
84                  if rekursive:
85                      li.extend(dirlist(dirname))
86      else:
87          logger.warning("DIRLIST: path (%s) does not exist - empty filelist will be returned",
        path)
88      return li
89
90
91  def filelist(path=".", expression="*", rekursive=True):
92      """
93      Function returning a list of files below a given path.
94
95      :param str path: folder which is the basepath for searching files.
96      :param str expression: expression to fit including shell-style wildcards.
97      :param bool rekursive: search all subfolders if True.
98      :returns: list of filenames including the pathe
99      :rtype: list
100
101     .. note:: The returned filenames could be relative pathes depending on argument path.
102
103     **Example:**
104
105     .. literalinclude:: fstools/_examples_/filelist.py
106
107     .. literalinclude:: fstools/_examples_/filelist.log
108     """
109     li = list()
110     if os.path.exists(path):
111         logger.debug("FILELIST: path (%s) exists - looking for files to append", path)
112         for filename in glob.glob(os.path.join(path, expression)):
113             if os.path.isfile(filename):
114                 li.append(filename)
115         for directory in os.listdir(path):
116             directory = os.path.join(path, directory)
117             if os.path.isdir(directory) and rekursive and not os.path.islink(directory):
118                 li.extend(filelist(directory, expression))
119     else:
120         logger.warning("FILELIST: path (%s) does not exist - empty filelist will be returned",
        path)
121     return li
122
123
124 def is_writeable(path):
```

```
125     """
126     Method to get the Information, if a file or folder is writable.
127 
128     :param str path: file or folder to check.
129     :returns: Whether path is writable or not.
130     :rtype: bool
131 
132     .. note:: If path does not exist, the return Value is :const:`False`.
133 
134     **Example:**
135 
136     .. literalinclude:: fstools/_examples_/is_writeable.py
137 
138     .. literalinclude:: fstools/_examples_/is_writeable.log
139     """
140     if os.access(path, os.W_OK):
141         # path is writable whatever it is, file or directory
142         return True
143     else:
144         # path is not writable whatever it is, file or directory
145         return False
146 
147 
148 def mkdir(path):
149     """
150     Method to create a folder.
151 
152     .. note:: All needed subfoilders will also be created (rekursive mkdir).
153 
154     :param str path: folder to be created.
155     :returns: True, if folder exists after creation commands, otherwise False.
156     :rtype: bool
157     """
158     path = os.path.abspath(path)
159     if not os.path.exists(os.path.dirname(path)):
160         mkdir(os.path.dirname(path))
161     if not os.path.exists(path):
162         os.mkdir(path)
163     return os.path.isdir(path)
164 
165 
166 def open_locked_blocking(*args, **kwargs):
167     """
168     Method to get exclusive access to a file.
169 
170     :param args: Arguments for a standard file open call.
171     :param kwargs: Keyword arguments for a standard file open call.
172     :returns: A file descriptor.
173     :rtype: file handle
174 
175     .. note:: The call blocks until file is able to be used. This can cause a deadlock, if the
    file release es done after trying to open the file!
176     """
177     locked_file_descriptor = open(*args, **kwargs)
178     fcntl.lockf(locked_file_descriptor, fcntl.LOCK_EX)
179     return locked_file_descriptor
180 
181 
182 def open_locked_non_blocking(*args, **kwargs):
```

```
183     """
184     Method to get exclusive access to a file.
185
186     :param args: Arguments for a standard file open call.
187     :param kwargs: Keyword arguments for a standard file open call.
188     :raises: OSError, if the file is already blocked.
189     :returns: A file descriptor.
190     :rtype: file handle
191
192     .. note:: The call blocks until file is able to be used. This can cause a deadlock, if the
        file release es done after trying to open the file!
193     """
194     locked_file_descriptor = open(*args, **kwargs)
195     fcntl.lockf(locked_file_descriptor, fcntl.LOCK_EX | fcntl.LOCK_NB)
196     return locked_file_descriptor
197
198
199 def uid(path, max_staleness=3600):
200     """
201     Function returning a "unique" id for a given file or path.
202
203     :param str path: File or folder to generate a uid for.
204     :param int max_staleness: If a file or folder is older than that, we may consider
205                               it stale and return a different uid - this is a
206                               dirty trick to work around changes never being
207                               detected. Default is 3600 seconds, use None to
208                               disable this trickery. See below for more details.
209     :returns:  An object that changes value if the file changed,
210                None is returned if there were problems accessing the file
211                or folder.
212     :rtype: str
213
214     .. warning:: Depending on the operating system capabilities and the way the
215                  file update is done, this function might return the same value
216                  even if the file has changed. It should be better than just
217                  using file's mtime though.
218                  max_staleness tries to avoid the worst for these cases.
219
220     .. note:: If this function is used for a path, it will stat all pathes and files rekursively.
221
222     **Example:**
223
224     .. literalinclude:: fstools/_examples_/uid.py
225
226     .. literalinclude:: fstools/_examples_/uid.log
227
228     Using just the file's mtime to determine if the file has changed is
229     not reliable - if file updates happen faster than the file system's
230     mtime granularity, then the modification is not detectable because
231     the mtime is still the same.
232
233     This function tries to improve by using not only the mtime, but also
234     other metadata values like file size and inode to improve reliability.
235
236     For the calculation of this value, we of course only want to use data
237     that we can get rather fast, thus we use file metadata, not file data
238     (file content).
239     """
240     if os.path.isdir(path):
241         pathlist = dirlist(path) + filelist(path)
242         pathlist.sort()
```

```
243     else:
244         pathlist = [path]
245     uid = []
246     for element in pathlist:
247         try:
248             st = os.stat(element)
249         except (IOError, OSError):
250             uid.append(None)  # for permanent errors on stat() this does not change, but
251             #                   having a changing value would be pointless because if we
252             #                   can't even stat the file, it is unlikely we can read it.
253         else:
254             fake_mtime = int(st.st_mtime)
255             if not st.st_ino and max_staleness:
256                 # st_ino being 0 likely means that we run on a platform not
257                 # supporting it (e.g. win32) - thus we likely need this dirty
258                 # trick
259                 now = int(time.time())
260                 if now >= st.st_mtime + max_staleness:
261                     # keep same fake_mtime for each max_staleness interval
262                     fake_mtime = int(now / max_staleness) * max_staleness
263             uid.append(
264                 (
265                     st.st_mtime,  # might have a rather rough granularity, e.g. 2s
266                     # on FAT, 1s on ext3 and might not change on fast
267                     # updates
268                     st.st_ino,  # inode number (will change if the update is done
269                     # by e.g. renaming a temp file to the real file).
270                     # not supported on win32 (0 ever)
271                     st.st_size,  # likely to change on many updates, but not
272                     # sufficient alone
273                     fake_mtime,
274                 )  # trick to workaround file system / platform
275                 # limitations causing permanent trouble
276             )
277     secret = b""
278     return hmac.new(secret, bytes(repr(uid), "latin-1"), hashlib.sha1).hexdigest()
279
280
281 def uid_filelist(path=".", expression="*", rekursive=True):
282     """
283     Function returning a unique id for a given file or path.
284
285     :param str path: folder which is the basepath for searching files.
286     :param str expression: expression to fit including shell-style wildcards.
287     :param bool rekursive: search all subfolders if True.
288     :returns:  An object that changes value if one of the files change.
289     :rtype: str
290
291     .. note:: This UID is created out of the file content. Therefore it is more
292               reliable then :func:`fstools.uid`, but also much slower.
293
294     **Example:**
295
296     .. literalinclude:: fstools/_examples_/uid_filelist.py
297
298     .. literalinclude:: fstools/_examples_/uid_filelist.log
299     """
300     SHAhash = hashlib.md5()
301     #
```

```
302     fl = filelist(path, expression, rekursive)
303     fl.sort()
304     for f in fl:
305         with open(f, mode="rb") as fh:
306             d = hashlib.md5()
307             for buf in iter(partial(fh.read, 128), b""):
308                 d.update(buf)
309         SHAhash.update(d.hexdigest().encode())
310     #
311     return SHAhash.hexdigest()
```