

---

# **robotframework-openafslibrary**

**Sine Nomine Associates**

**Nov 29, 2022**



**CONTENTS:**

<b>1</b>	<b>Overview</b>	<b>3</b>
<b>2</b>	<b>Installation</b>	<b>5</b>
<b>3</b>	<b>Variables</b>	<b>7</b>
<b>4</b>	<b>Keywords</b>	<b>9</b>
<b>5</b>	<b>Examples</b>	<b>21</b>
<b>6</b>	<b>License</b>	<b>23</b>
<b>7</b>	<b>Indices and tables</b>	<b>25</b>



The **OpenAFS Library** provides keywords for testing the [OpenAFS](#) distributed filesystem with [Robot Framework](#). See the [openafs-robotest](#) project for a set of test cases using this library.



## OVERVIEW

The **OpenAFS Library** is a **Robot Framework** keyword library for creating test cases for the [OpenAFS distributed filesystem](#).

The OpenAFS clients and servers should be already be installed and configured before running tests. The **OpenAFS Library** and **Robot Framework** should be installed on at least one host in the test cell.





## INSTALLATION

### 2.1 Package

Install the **OpenAFS Library** and **Robot Framework** packages with pip:

```
$ pip install robotframework-openafslibrary
```

### 2.2 Source code

Alternatively, the library may be installed from source code. This may be helpful when developing new keywords.

```
$ git clone https://github.com/openafs-contrib/robotframework-openafslibrary
$ cd robotframework-openafslibrary
$ python configure.py
$ make install-user # or sudo make install
```

You will need to manually install robotframework when pip is not installed.



## VARIABLES

The following variables should be configured in your test suite or in variable files to match your test system setup.

Table 1: Test cell variables

Name	Description
AFS_CELL	Test cell name
KRB_REALM	Authentication realm (akimpersonate)
KRB_AFS_KEYTAB	Authentication keytab (akimpersonate)
AKLOG	aklog command path
BOS	bos command path
FS	fs command path
PAGSH	pagsh command path
PTS	pts command path
RXDEBUG	rxdebug command path
TOKENS	tokens command path
UDEBUG	udebug command path
UNLOG	unlog command path
VOS	vos command path



**KEYWORDS**

Version: 0.7.2

## 4.1 Access Control List Contains

### Arguments

Name	Default value	Notes
path		required
name		required
rights		required

### Documentation

Fails if an ACL does not contain the given rights.

## 4.2 Access Control List Matches

### Arguments

Name	Default value	Notes
path		required
acls		

### Documentation

Fails if an ACL does not match the given ACL.

## 4.3 Access Control Should Exist

### Arguments

Name	Default value	Notes
path		required
name		required

**Documentation**

Fails if the access control does not exist for the the given user or group name.

## 4.4 Access Control Should Not Exist

**Arguments**

Name	Default value	Notes
path		required
name		required

**Documentation**

Fails if the access control exists for the the given user or group name.

## 4.5 Add Access Rights

**Arguments**

Name	Default value	Notes
path		required
name		required
rights		required

**Documentation**

Add access rights to a path.

## 4.6 Command Should Fail

**Arguments**

Name	Default value	Notes
cmd		required

**Documentation**

Fails if command exits with a zero status code.

## 4.7 Command Should Succeed

### Arguments

Name	Default value	Notes
cmd		required
msg	None	

### Documentation

Fails if command does not exit with a zero status code.

## 4.8 Create Dump

### Arguments

Name	Default value	Notes
filename		required
size	small	
contains		

### Documentation

Generate a volume dump file.

## 4.9 Create Files

### Arguments

Name	Default value	Notes
path		required
count	1	
size	0	
depth	0	
width	0	
fill	zero	

### Documentation

Create a directory tree of test files.

#### path

destination path

#### count

number of files to create in each directory

#### size

size of each file

#### depth

sub-directory depth

**width**

number of sub-directories in each directory

**fill**

test files data pattern

Valid fill values:

- zero - fill with zero bits
- one - fill with one bits
- random - fill with pseudo random bits
- fixed - fill with repetitions of fixed bits

## 4.10 Create Volume

**Arguments**

Name	Default value	Notes
name		required
server	None	
part	a	
path	None	
quota	0	
ro	False	
acl	None	
orphan	False	

**Documentation**

Create and mount a volume.

Create a volume and optionally mount the volume. Also optionally create a read-only clone of the volume and release the new new volume. Release the parent volume if it is replicated.

## 4.11 Directory Entry Should Exist

**Arguments**

Name	Default value	Notes
path		required

**Documentation**

Fails if directory entry does not exist in the given path.



## 4.12 File Should Be Executable

### Arguments

Name	Default value	Notes
path		required

### Documentation

Fails if path is not an executable file for the current user.

## 4.13 Get Cache Size

### Documentation

Get the cache size.

Outputs AFS cache size as the number of 1K blocks.

## 4.14 Get Inode

### Arguments

Name	Default value	Notes
path		required

### Documentation

Returns the inode number of a path.

## 4.15 Get Version

### Arguments

Name	Default value	Notes
host		required
port		required

### Documentation

Request the software version number.

## 4.16 Get Volume Id

### Arguments

Name	Default value	Notes
name		required

### Documentation

Lookup the volume numeric id.

## 4.17 Inode Should Be Equal

### Arguments

Name	Default value	Notes
a		required
b		required

### Documentation

Fails if paths have different inodes.

## 4.18 Link

### Arguments

Name	Default value	Notes
src		required
dst		required
code_should_be	0	

### Documentation

Create a hard link.

## 4.19 Link Count Should Be

### Arguments

Name	Default value	Notes
path		required
count		required

### Documentation

Fails if the path has an unexpected inode link count.

## 4.20 Login

### Arguments

Name	Default value	Notes
user		required
password	None	
keytab	None	

### Documentation

Acquire an AFS token for authenticated access.

## 4.21 Logout

### Documentation

Release the AFS token.

## 4.22 Mount Volume

### Arguments

Name	Default value	Notes
path		required
vol		required
options		

### Documentation

Mount a volume on a path.

## 4.23 Pag From Groups

### Arguments

Name	Default value	Notes
gids	None	

### Documentation

Return the PAG from the given group id list.

## 4.24 Pag Shell

### Arguments

Name	Default value	Notes
script		required

### Documentation

Run a command in the pagsh and returns the output.

## 4.25 Pag Should Be Valid

### Arguments

Name	Default value	Notes
pag		required

### Documentation

Fails if the given PAG number is out of range.

## 4.26 Pag Should Exist

### Documentation

Fails if a PAG is not set.

## 4.27 Pag Should Not Exist

### Documentation

Fails if a PAG is set.

## 4.28 Release Volume

### Arguments

Name	Default value	Notes
name		required

### Documentation

Release the volume.

## 4.29 Remove Volume

### Arguments

Name	Default value	Notes
name_or_id		required
path	None	
flush	False	
server	None	
part	None	
zap	False	

### Documentation

Remove a volume.

Remove the volume and any clones. Optionally remove the given mount point.

## 4.30 Should Be A Dump File

### Arguments

Name	Default value	Notes
filename		required

### Documentation

Fails if filename is not an AFS dump file.

## 4.31 Should Be Dir

### Arguments

Name	Default value	Notes
path		required

### Documentation

Fails if path is not a directory.

## 4.32 Should Be File

### Arguments

Name	Default value	Notes
path		required

### Documentation

Fails if path is not a file.

## 4.33 Should Be Symlink

### Arguments

Name	Default value	Notes
path		required

### Documentation

Fails if path is not a symlink.

## 4.34 Should Not Be Dir

### Arguments

Name	Default value	Notes
path		required

### Documentation

Fails if path is a directory.

## 4.35 Should Not Be Symlink

### Arguments

Name	Default value	Notes
path		required

### Documentation

Fails if path is a symlink.

## 4.36 Symlink

### Arguments

Name	Default value	Notes
src		required
dst		required
code_should_be	0	

### Documentation

Create a symlink.

## 4.37 Unlink

### Arguments

Name	Default value	Notes
path		required
code_should_be	0	

### Documentation

Unlink the directory entry.

## 4.38 Volume Location Matches

### Arguments

Name	Default value	Notes
name_or_id		required
server		required
part		required
vtype	rw	

### Documentation

Fails if volume is not located on the given server and partition.

## 4.39 Volume Should Be Locked

### Arguments

Name	Default value	Notes
name		required

### Documentation

Fails if the volume is not locked.

## 4.40 Volume Should Be Unlocked

### Arguments

Name	Default value	Notes
name		required

### Documentation

Fails if the volume is locked.

## 4.41 Volume Should Exist

### Arguments

Name	Default value	Notes
name_or_id		required

### Documentation

Verify the existence of a read-write volume.

Fails if the volume entry is not found in the VLDB or the volume is not present on the fileserver indicated by the VLDB.

## 4.42 Volume Should Not Exist

### Arguments

Name	Default value	Notes
name_or_id		required

### Documentation

Fails if volume exists.



## EXAMPLES

The following example demonstrates a test of a basic OpenAFS volume release.

*** Settings ***			
Library	OpenAFSLibrary		
*** Test Cases ***			
Volume Release			
	[Documentation]	Example test.	
	Login	admin	keytab=admin.keytab
	Create Volume	testvol	afs01   a
	Command Should Succeed	\${VOS}	addsite afs01 a testvol
	Release Volume	testvol	
	Volume Should Exist	testvol.readonly	
	Volume Location Matches	testvol	server=afs01   part=a
↪vtype=ro			
	Remove Volume	testvol	
	Logout		



**LICENSE**

Copyright (c) 2014-2021, Sine Nomine Associates

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

THE SOFTWARE IS PROVIDED ‘AS IS’ AND THE AUTHOR DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS SOFTWARE INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY SPECIAL, DIRECT, INDIRECT, OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

six.py: Copyright (c) 2010-2018 Benjamin Peterson

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the “Software”), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED “AS IS”, WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.



## INDICES AND TABLES

- `genindex`
- `search`