

Argus Log

Generated
20190428 14:26:15 UTC-04:00

Test Statistics

Total Statistics	Total	Pass	Fail	Elapsed	Pass / Fail
Critical Tests	6	6	0	00:00:06	<div></div>
All Tests	6	6	0	00:00:06	<div></div>

Statistics by Tag	Total	Pass	Fail	Elapsed	Pass / Fail
No Tags					<div></div>

Statistics by Suite	Total	Pass	Fail	Elapsed	Pass / Fail
Argus	6	6	0	00:00:06	<div></div>

Test Execution Log

SUITE Argus	00:00:05.957
Full Name:	Argus
Documentation:	Tests for the location calculation of argus
Source:	/home/dev/Desktop/testing/argus-dev/tests/Argus.robot
Start / End / Elapsed:	20190428 14:26:09.089 / 20190428 14:26:15.046 / 00:00:05.957
Status:	6 critical test, 6 passed, 0 failed 6 test total, 6 passed, 0 failed

TEST File Check Test	00:00:00.004
Full Name:	Argus.File Check Test
Documentation:	Checks that all argus files have been installed correctly
Start / End / Elapsed:	20190428 14:26:09.123 / 20190428 14:26:09.127 / 00:00:00.004
Status:	PASS (critical)

KEYWORD OperatingSystem.Should Exist argus.py	00:00:00.000
Documentation:	Fails unless the given path (file or directory) exists.
Start / End / Elapsed:	20190428 14:26:09.124 / 20190428 14:26:09.124 / 00:00:00.000
14:26:09.124	INFO Path '/home/dev/Desktop/testing/argus-dev/argus.py' exists.

KEYWORD OperatingSystem.Should Exist config.yml	00:00:00.001
Documentation:	Fails unless the given path (file or directory) exists.
Start / End / Elapsed:	20190428 14:26:09.124 / 20190428 14:26:09.125 / 00:00:00.001
14:26:09.125	INFO Path '/home/dev/Desktop/testing/argus-dev/config.yml' exists.

KEYWORD OperatingSystem.Should Exist install.sh	00:00:00.000
Documentation:	Fails unless the given path (file or directory) exists.
Start / End / Elapsed:	20190428 14:26:09.125 / 20190428 14:26:09.125 / 00:00:00.000
14:26:09.125	INFO Path '/home/dev/Desktop/testing/argus-dev/install.sh' exists.

KEYWORD OperatingSystem.Should Exist location.py	00:00:00.000
Documentation:	Fails unless the given path (file or directory) exists.
Start / End / Elapsed:	20190428 14:26:09.125 / 20190428 14:26:09.125 / 00:00:00.000
14:26:09.125	INFO Path '/home/dev/Desktop/testing/argus-dev/location.py' exists.

KEYWORD OperatingSystem.Should Exist packet_processor.py	00:00:00.000
Documentation:	Fails unless the given path (file or directory) exists.
Start / End / Elapsed:	20190428 14:26:09.126 / 20190428 14:26:09.126 / 00:00:00.000
14:26:09.126	INFO Path '/home/dev/Desktop/testing/argus-dev/packet_processor.py' exists.

KEYWORD OperatingSystem.Should Exist README.md	00:00:00.000
Documentation:	Fails unless the given path (file or directory) exists.
Start / End / Elapsed:	20190428 14:26:09.126 / 20190428 14:26:09.126 / 00:00:00.000
14:26:09.126	INFO Path '/home/dev/Desktop/testing/argus-dev/README.md' exists.

KEYWORD OperatingSystem.Should Exist requirements.txt	00:00:00.001
Documentation:	Fails unless the given path (file or directory) exists.
Start / End / Elapsed:	20190428 14:26:09.126 / 20190428 14:26:09.127 / 00:00:00.001
14:26:09.127	INFO Path '/home/dev/Desktop/testing/argus-dev/requirements.txt' exists.

TEST Argus Missing Config Test	00:00:00.833
---------------------------------------	--------------

Full Name:	Argus.Argus Missing Config Test	
Documentation:	Verifies argus output when config file is missing	
Start / End / Elapsed:	20190428 14:26:09.127 / 20190428 14:26:09.960 / 00:00:00.833	
Status:	PASS (critical)	
<input type="checkbox"/> KEYWORD	<code>\${output} = OperatingSystem.Run python3 argus.py -c test.yml</code>	00:00:00.830
Documentation:	Runs the given command in the system and returns the output.	
Start / End / Elapsed:	20190428 14:26:09.128 / 20190428 14:26:09.958 / 00:00:00.830	
14:26:09.129	INFO Running command 'python3 argus.py -c test.yml 2>&1'.	
14:26:09.958	INFO <code>\${output} = usage: argus.py [-h] [-c CONFIG]</code> argus.py: error: Must provide a valid config file	
<input type="checkbox"/> KEYWORD	BuiltIn.Should Contain <code>\${output}, \${MISSING_CONFIG_STRING1}</code>	00:00:00.001
Documentation:	Fails if <code>container</code> does not contain <code>item</code> one or more times.	
Start / End / Elapsed:	20190428 14:26:09.958 / 20190428 14:26:09.959 / 00:00:00.001	
<input type="checkbox"/> KEYWORD	BuiltIn.Should Contain <code>\${output}, \${MISSING_CONFIG_STRING2}</code>	00:00:00.000
Documentation:	Fails if <code>container</code> does not contain <code>item</code> one or more times.	
Start / End / Elapsed:	20190428 14:26:09.959 / 20190428 14:26:09.959 / 00:00:00.000	
<input type="checkbox"/> KEYWORD	OperatingSystem.Remove File test.yml	00:00:00.001
Documentation:	Removes a file with the given path.	
Start / End / Elapsed:	20190428 14:26:09.959 / 20190428 14:26:09.960 / 00:00:00.001	
14:26:09.960	INFO File '/home/dev/Desktop/testing/argus-dev/test.yml' does not exist.	
<hr/>		
<input type="checkbox"/> TEST	Argus Help Test	00:00:00.834
Full Name:	Argus.Argus Help Test	
Documentation:	Verifies the usage output from argus	
Start / End / Elapsed:	20190428 14:26:09.960 / 20190428 14:26:10.794 / 00:00:00.834	
Status:	PASS (critical)	
<input type="checkbox"/> KEYWORD	<code>\${help_output} = OperatingSystem.Get File tests/test_files/usage.out</code>	00:00:00.000
Documentation:	Returns the contents of a specified file.	
Start / End / Elapsed:	20190428 14:26:09.961 / 20190428 14:26:09.961 / 00:00:00.000	
14:26:09.961	INFO Getting file '/home/dev/Desktop/testing/argus-dev/tests/test_files/usage.out'.	
14:26:09.961	INFO <code>\${help_output} = usage: argus.py [-h] [-c CONFIG]</code> argus optional arguments: -h, --help show this help message and exit -c CONFIG, --config CONFIG Config file. Defaults to confi...	
<input type="checkbox"/> KEYWORD	<code>\${argus_output} = OperatingSystem.Run python3 argus.py --help</code>	00:00:00.832
Documentation:	Runs the given command in the system and returns the output.	
Start / End / Elapsed:	20190428 14:26:09.961 / 20190428 14:26:10.793 / 00:00:00.832	
14:26:09.962	INFO Running command 'python3 argus.py --help 2>&1'.	
14:26:10.793	INFO <code>\${argus_output} = usage: argus.py [-h] [-c CONFIG]</code> argus optional arguments: -h, --help show this help message and exit -c CONFIG, --config CONFIG Config file. Defaults to confi...	
<input type="checkbox"/> KEYWORD	BuiltIn.Should Contain <code>\${help_output}, \${argus_output}</code>	00:00:00.000
Documentation:	Fails if <code>container</code> does not contain <code>item</code> one or more times.	
Start / End / Elapsed:	20190428 14:26:10.794 / 20190428 14:26:10.794 / 00:00:00.000	
<hr/>		
<input type="checkbox"/> TEST	Empty Pcap Test	00:00:00.879
Full Name:	Argus.Empty Pcap Test	
Documentation:	Checks the output from argus with empty pcaps as input	
Start / End / Elapsed:	20190428 14:26:10.795 / 20190428 14:26:11.674 / 00:00:00.879	
Status:	PASS (critical)	
<input type="checkbox"/> KEYWORD	Change AP Config File <code>\$(EMPTY_PCAP01), \$(EMPTY_PCAP02), \$(EMPTY_PCAP03)</code>	00:00:00.018
Documentation:	Changes the configuration files that the APs use for argus	

Start / End / Elapsed: 20190428 14:26:10.795 / 20190428 14:26:10.813 / 00:00:00.018

KEYWORD YamlLibrary. **Change Config Capture File** argus1, \${config_file_1} 00:00:00.005

Start / End / Elapsed: 20190428 14:26:10.796 / 20190428 14:26:10.801 / 00:00:00.005

14:26:10.801 **INFO** /home/dev/Desktop/testing/argus-dev/tests/libraries/YamlLibrary.py:14:
YAMLLoadWarning: calling yaml.load() without Loader=... is deprecated, as the
default Loader is unsafe. Please read <https://msg.pyyaml.org/load> for full details.
argus_config = yaml.load(f)

KEYWORD OperatingSystem. **Grep File** \${YAML_PATH}, \${config_file_1} 00:00:00.001

Documentation: Returns the lines of the specified file that match the pattern.

Start / End / Elapsed: 20190428 14:26:10.801 / 20190428 14:26:10.802 / 00:00:00.001

14:26:10.801 **INFO** Reading file '/home/dev/Desktop/testing/argus-dev/config.yml'.

14:26:10.802 **INFO** 1 out of 14 lines matched

KEYWORD YamlLibrary. **Change Config Capture File** argus2, \${config_file_2} 00:00:00.005

Start / End / Elapsed: 20190428 14:26:10.802 / 20190428 14:26:10.807 / 00:00:00.005

KEYWORD OperatingSystem. **Grep File** \${YAML_PATH}, \${config_file_2} 00:00:00.001

Documentation: Returns the lines of the specified file that match the pattern.

Start / End / Elapsed: 20190428 14:26:10.807 / 20190428 14:26:10.808 / 00:00:00.001

14:26:10.807 **INFO** Reading file '/home/dev/Desktop/testing/argus-dev/config.yml'.

14:26:10.808 **INFO** 1 out of 14 lines matched

KEYWORD YamlLibrary. **Change Config Capture File** argus3, \${config_file_3} 00:00:00.004

Start / End / Elapsed: 20190428 14:26:10.808 / 20190428 14:26:10.812 / 00:00:00.004

KEYWORD OperatingSystem. **Grep File** \${YAML_PATH}, \${config_file_3} 00:00:00.000

Documentation: Returns the lines of the specified file that match the pattern.

Start / End / Elapsed: 20190428 14:26:10.812 / 20190428 14:26:10.812 / 00:00:00.000

14:26:10.812 **INFO** Reading file '/home/dev/Desktop/testing/argus-dev/config.yml'.

14:26:10.812 **INFO** 1 out of 14 lines matched

KEYWORD OperatingSystem. **Run** python3 argus.py 00:00:00.859

Documentation: Runs the given command in the system and returns the output.

Start / End / Elapsed: 20190428 14:26:10.813 / 20190428 14:26:11.672 / 00:00:00.859

14:26:10.814 **INFO** Running command 'python3 argus.py 2>&1'.

KEYWORD OperatingSystem. **File Should Exist** output.json 00:00:00.001

Documentation: Fails unless the given path points to an existing file.

Start / End / Elapsed: 20190428 14:26:11.672 / 20190428 14:26:11.673 / 00:00:00.001

14:26:11.673 **INFO** File '/home/dev/Desktop/testing/argus-dev/output.json' exists.

KEYWORD OperatingSystem. **File Should Not Be Empty** output.json 00:00:00.000

Documentation: Fails if the specified directory is empty.

Start / End / Elapsed: 20190428 14:26:11.673 / 20190428 14:26:11.673 / 00:00:00.000

14:26:11.673 **INFO** File '/home/dev/Desktop/testing/argus-dev/output.json' contains 19 bytes.

KEYWORD OperatingSystem. **Grep File** output.json, {"loc_updates": []} 00:00:00.001

Documentation: Returns the lines of the specified file that match the pattern.

Start / End / Elapsed: 20190428 14:26:11.673 / 20190428 14:26:11.674 / 00:00:00.001

14:26:11.673 **INFO** Reading file '/home/dev/Desktop/testing/argus-dev/output.json'.

14:26:11.674 **INFO** 1 out of 1 lines matched

TEST Small Pcap Test 00:00:01.908

Full Name: Argus.Small Pcap Test

Documentation: Checks the location calculation algorithms accuracy on a small pcap file extracted from test capture on March 14th.
Checks the values computed against known test output.

Start / End / Elapsed: 20190428 14:26:11.675 / 20190428 14:26:13.583 / 00:00:01.908

Status: **PASS** (critical)

KEYWORD OperatingSystem. **Run** tshark -r \${VALID_PCAP01} -Y '(frame.time >= \${CAP_TIME_VALID_START}) && (frame.time <= \${CAP_TIME_VALID_END})' -w tests/test_caps/small_cap_test-01.cap 00:00:00.196

Documentation: Runs the given command in the system and returns the output.

Start / End / Elapsed: 20190428 14:26:11.675 / 20190428 14:26:11.871 / 00:00:00.196

14:26:11.676 **INFO** Running command 'tshark -r tests/test_caps/capture-small-01.cap -Y '(frame.time >= "Mar 14, 2019 17:09:33.000000000") && (frame.time <= "Mar 14, 2019 17:09:33.603000000")' -w tests/test_caps/small_cap_test-01.cap'.

KEYWORD OperatingSystem. **Run** tshark -r \${VALID_PCAP02} -Y '(frame.time >= \${CAP_TIME_VALID_START}) && (frame.time <= \${CAP_TIME_VALID_END})' -w tests/test_caps/small_cap_test-02.cap 00:00:00.198

Documentation: Runs the given command in the system and returns the output.
Start / End / Elapsed: 20190428 14:26:11.872 / 20190428 14:26:12.070 / 00:00:00.198
 14:26:11.873 **INFO** Running command 'tshark -r tests/test_caps/capture-small-adj-02.cap -Y '(frame.time >= "Mar 14, 2019 17:09:33.000000000") && (frame.time <= "Mar 14, 2019 17:09:33.603000000")' -w tests/test_caps/small_cap_test-02.cap'.
 - **KEYWORD** OperatingSystem. **Run** tshark -r \${VALID_PCAP03} -Y '(frame.time >= \${CAP_TIME_VALID_START}) && (frame.time <= \${CAP_TIME_VALID_END})' -w tests/test_caps/small_cap_test-03.cap 00:00:00.192

Documentation: Runs the given command in the system and returns the output.
Start / End / Elapsed: 20190428 14:26:12.071 / 20190428 14:26:12.263 / 00:00:00.192
 14:26:12.072 **INFO** Running command 'tshark -r tests/test_caps/capture-small-adj-03.cap -Y '(frame.time >= "Mar 14, 2019 17:09:33.000000000") && (frame.time <= "Mar 14, 2019 17:09:33.603000000")' -w tests/test_caps/small_cap_test-03.cap'.
 - **KEYWORD** OperatingSystem. **Should Exist** tests/test_caps/small_cap_test-01.cap 00:00:00.001

Documentation: Fails unless the given path (file or directory) exists.
Start / End / Elapsed: 20190428 14:26:12.263 / 20190428 14:26:12.264 / 00:00:00.001
 14:26:12.264 **INFO** Path '/home/dev/Desktop/testing/argus-dev/tests/test_caps/small_cap_test-01.cap' exists.
 - **KEYWORD** OperatingSystem. **Should Exist** tests/test_caps/small_cap_test-02.cap 00:00:00.000

Documentation: Fails unless the given path (file or directory) exists.
Start / End / Elapsed: 20190428 14:26:12.264 / 20190428 14:26:12.264 / 00:00:00.000
 14:26:12.264 **INFO** Path '/home/dev/Desktop/testing/argus-dev/tests/test_caps/small_cap_test-02.cap' exists.
 - **KEYWORD** OperatingSystem. **Should Exist** tests/test_caps/small_cap_test-03.cap 00:00:00.000

Documentation: Fails unless the given path (file or directory) exists.
Start / End / Elapsed: 20190428 14:26:12.265 / 20190428 14:26:12.265 / 00:00:00.000
 14:26:12.265 **INFO** Path '/home/dev/Desktop/testing/argus-dev/tests/test_caps/small_cap_test-03.cap' exists.
 - **KEYWORD** **Change AP Config File** tests/test_caps/small_cap_test-01.cap, tests/test_caps/small_cap_test-02.cap, tests/test_caps/small_cap_test-03.cap 00:00:00.016

Documentation: Changes the configuration files that the APs use for argus
Start / End / Elapsed: 20190428 14:26:12.265 / 20190428 14:26:12.281 / 00:00:00.016
 - **KEYWORD** YamlLibrary. **Change Config Capture File** argus1, \${config_file_1} 00:00:00.005

Start / End / Elapsed: 20190428 14:26:12.265 / 20190428 14:26:12.270 / 00:00:00.005
 - **KEYWORD** OperatingSystem. **Grep File** \${YAML_PATH}, \${config_file_1} 00:00:00.001

Documentation: Returns the lines of the specified file that match the pattern.
Start / End / Elapsed: 20190428 14:26:12.270 / 20190428 14:26:12.271 / 00:00:00.001
 14:26:12.270 **INFO** Reading file '/home/dev/Desktop/testing/argus-dev/config.yml'.
 14:26:12.271 **INFO** 1 out of 14 lines matched
 - **KEYWORD** YamlLibrary. **Change Config Capture File** argus2, \${config_file_2} 00:00:00.004

Start / End / Elapsed: 20190428 14:26:12.271 / 20190428 14:26:12.275 / 00:00:00.004
 - **KEYWORD** OperatingSystem. **Grep File** \${YAML_PATH}, \${config_file_2} 00:00:00.001

Documentation: Returns the lines of the specified file that match the pattern.
Start / End / Elapsed: 20190428 14:26:12.275 / 20190428 14:26:12.276 / 00:00:00.001
 14:26:12.276 **INFO** Reading file '/home/dev/Desktop/testing/argus-dev/config.yml'.
 14:26:12.276 **INFO** 1 out of 14 lines matched
 - **KEYWORD** YamlLibrary. **Change Config Capture File** argus3, \${config_file_3} 00:00:00.004

Start / End / Elapsed: 20190428 14:26:12.276 / 20190428 14:26:12.280 / 00:00:00.004
 - **KEYWORD** OperatingSystem. **Grep File** \${YAML_PATH}, \${config_file_3} 00:00:00.001

Documentation: Returns the lines of the specified file that match the pattern.
Start / End / Elapsed: 20190428 14:26:12.280 / 20190428 14:26:12.281 / 00:00:00.001
 14:26:12.281 **INFO** Reading file '/home/dev/Desktop/testing/argus-dev/config.yml'.
 14:26:12.281 **INFO** 1 out of 14 lines matched
 - **KEYWORD** OperatingSystem. **Run** python3 argus.py 00:00:01.295

Documentation: Runs the given command in the system and returns the output.
Start / End / Elapsed: 20190428 14:26:12.281 / 20190428 14:26:13.576 / 00:00:01.295
 14:26:12.283 **INFO** Running command 'python3 argus.py 2>&1'.
 - **KEYWORD** OperatingSystem. **File Should Exist** output.json 00:00:00.000

Documentation: Fails unless the given path points to an existing file.
Start / End / Elapsed: 20190428 14:26:13.577 / 20190428 14:26:13.577 / 00:00:00.000
 14:26:13.577 **INFO** File '/home/dev/Desktop/testing/argus-dev/output.json' exists.

	KEYWORD OperatingSystem. File Should Not Be Empty output.json	00:00:00.000
Documentation: Fails if the specified directory is empty.		
Start / End / Elapsed: 20190428 14:26:13.578 / 20190428 14:26:13.578 / 00:00:00.000		
14:26:13.578	INFO File '/home/dev/Desktop/testing/argus-dev/output.json' contains 9953 bytes.	
	KEYWORD \${output} = JSONLibrary. Get Dict From Json File output.json	00:00:00.001
Start / End / Elapsed: 20190428 14:26:13.578 / 20190428 14:26:13.579 / 00:00:00.001		
14:26:13.579	INFO \${output} = {'loc_updates': [{'timestamp': 1552597773.602372, 'location': [2.51615513578063, -39.039146902938], 'device': '40:4e:36:22:b8:01'}, {'timestamp': 1552597773.602372, 'location': [2.51615513578063, -39....	
	KEYWORD BuiltIn. Should Be Equal \${output['loc_updates'][0]['device']}, 40:4e:36:22:b8:01	00:00:00.000
Documentation: Fails if the given objects are unequal.		
Start / End / Elapsed: 20190428 14:26:13.579 / 20190428 14:26:13.579 / 00:00:00.000		
	KEYWORD BuiltIn. Should Be Equal \${output['loc_updates'][0]['location'][0]}, \${2.51615513578063}	00:00:00.001
Documentation: Fails if the given objects are unequal.		
Start / End / Elapsed: 20190428 14:26:13.579 / 20190428 14:26:13.580 / 00:00:00.001		
	KEYWORD BuiltIn. Should Be Equal \${output['loc_updates'][0]['location'][1]}, \${-39.039146902938}	00:00:00.000
Documentation: Fails if the given objects are unequal.		
Start / End / Elapsed: 20190428 14:26:13.580 / 20190428 14:26:13.580 / 00:00:00.000		
	KEYWORD \${locations} = BuiltIn. Get Length \${output['loc_updates']}	00:00:00.001
Documentation: Returns and logs the length of the given item as an integer.		
Start / End / Elapsed: 20190428 14:26:13.580 / 20190428 14:26:13.581 / 00:00:00.001		
14:26:13.581	INFO Length is 84	
14:26:13.581	INFO \${locations} = 84	
	KEYWORD BuiltIn. Should Be Equal \${84}, \${locations}	00:00:00.000
Documentation: Fails if the given objects are unequal.		
Start / End / Elapsed: 20190428 14:26:13.581 / 20190428 14:26:13.581 / 00:00:00.000		
	TEARDOWN Clean Up Files	00:00:00.002
Documentation: Removes files created in tests		
Start / End / Elapsed: 20190428 14:26:13.581 / 20190428 14:26:13.583 / 00:00:00.002		
	KEYWORD OperatingSystem. Remove File tests/test_cap_test-01.cap	00:00:00.000
Documentation: Removes a file with the given path.		
Start / End / Elapsed: 20190428 14:26:13.582 / 20190428 14:26:13.582 / 00:00:00.000		
14:26:13.582	INFO File '/home/dev/Desktop/testing/argus-dev/tests/test_cap_test-01.cap' does not exist.	
	KEYWORD OperatingSystem. Remove File tests/test_cap_test-02.cap	00:00:00.000
Documentation: Removes a file with the given path.		
Start / End / Elapsed: 20190428 14:26:13.582 / 20190428 14:26:13.582 / 00:00:00.000		
14:26:13.582	INFO File '/home/dev/Desktop/testing/argus-dev/tests/test_cap_test-02.cap' does not exist.	
	KEYWORD OperatingSystem. Remove File tests/test_cap_test-03.cap	00:00:00.000
Documentation: Removes a file with the given path.		
Start / End / Elapsed: 20190428 14:26:13.583 / 20190428 14:26:13.583 / 00:00:00.000		
14:26:13.583	INFO File '/home/dev/Desktop/testing/argus-dev/tests/test_cap_test-03.cap' does not exist.	
	KEYWORD OperatingSystem. Remove File output.json	00:00:00.000
Documentation: Removes a file with the given path.		
Start / End / Elapsed: 20190428 14:26:13.583 / 20190428 14:26:13.583 / 00:00:00.000		
14:26:13.583	INFO Removed file '/home/dev/Desktop/testing/argus-dev/output.json'.	

	TEST Location Time Interval Test	00:00:01.461
Full Name: Argus.Location Time Interval Test		
Documentation: Verifies that the location ignores packets that have too large of a time delta.		
Start / End / Elapsed: 20190428 14:26:13.584 / 20190428 14:26:15.045 / 00:00:01.461		
Status: PASS (critical)		
	KEYWORD OperatingSystem. Run tshark -r \${VALID_PCAP01} -Y '(frame.time >= \${CAP_TIME_INVALID_START01}) && (frame.time <= \${CAP_TIME_INVALID_END01})' -w tests/test_caps/small_cap_test-01.cap	00:00:00.216
Documentation: Runs the given command in the system and returns the output.		
Start / End / Elapsed: 20190428 14:26:13.584 / 20190428 14:26:13.800 / 00:00:00.216		
14:26:13.586	INFO Running command 'tshark -r tests/test_caps/capture-small-01.cap -Y '(frame.time >= "Mar 14, 2019 17:09:33.000000000") && (frame.time <= "Mar 14, 2019 17:09:36.000000000")' -w tests/test_caps/small_cap_test-01.cap'.	


```

- KEYWORD OperatingSystem.Run tshark -r ${VALID_PCAP02} -Y '(frame.time >= ${CAP_TIME_INVALID_START02})
  && (frame.time <= ${CAP_TIME_INVALID_END02})' -w tests/test_caps/small_cap_test-02.cap 00:00:00.203
Documentation:      Runs the given command in the system and returns the output.
Start / End / Elapsed: 20190428 14:26:13.800 / 20190428 14:26:14.003 / 00:00:00.203
14:26:13.802 INFO Running command 'tshark -r tests/test_caps/capture-small-adj-02.cap -Y '(frame.time
  >= "Mar 14, 2019 17:09:39.000000000") && (frame.time <= "Mar 14, 2019
  17:09:41.000000000")' -w tests/test_caps/small_cap_test-02.cap'.

- KEYWORD OperatingSystem.Run tshark -r ${VALID_PCAP03} -Y '(frame.time >= ${CAP_TIME_INVALID_START03})
  && (frame.time <= ${CAP_TIME_INVALID_END03})' -w tests/test_caps/small_cap_test-03.cap 00:00:00.186
Documentation:      Runs the given command in the system and returns the output.
Start / End / Elapsed: 20190428 14:26:14.003 / 20190428 14:26:14.189 / 00:00:00.186
14:26:14.005 INFO Running command 'tshark -r tests/test_caps/capture-small-adj-03.cap -Y '(frame.time
  >= "Mar 14, 2019 17:09:45.000000000") && (frame.time <= "Mar 14, 2019
  17:09:47.603000000")' -w tests/test_caps/small_cap_test-03.cap'.

- KEYWORD OperatingSystem.Should Exist tests/test_caps/small_cap_test-01.cap 00:00:00.001
Documentation:      Fails unless the given path (file or directory) exists.
Start / End / Elapsed: 20190428 14:26:14.189 / 20190428 14:26:14.190 / 00:00:00.001
14:26:14.190 INFO Path '/home/dev/Desktop/testing/argus-dev/tests/test_caps/small_cap_test-01.cap'
  exists.

- KEYWORD OperatingSystem.Should Exist tests/test_caps/small_cap_test-02.cap 00:00:00.000
Documentation:      Fails unless the given path (file or directory) exists.
Start / End / Elapsed: 20190428 14:26:14.190 / 20190428 14:26:14.190 / 00:00:00.000
14:26:14.190 INFO Path '/home/dev/Desktop/testing/argus-dev/tests/test_caps/small_cap_test-02.cap'
  exists.

- KEYWORD OperatingSystem.Should Exist tests/test_caps/small_cap_test-03.cap 00:00:00.000
Documentation:      Fails unless the given path (file or directory) exists.
Start / End / Elapsed: 20190428 14:26:14.190 / 20190428 14:26:14.190 / 00:00:00.000
14:26:14.190 INFO Path '/home/dev/Desktop/testing/argus-dev/tests/test_caps/small_cap_test-03.cap'
  exists.

- KEYWORD Change AP Config File tests/test_caps/small_cap_test-01.cap, tests/test_caps/small_cap_test-
  02.cap, tests/test_caps/small_cap_test-03.cap 00:00:00.015
Documentation:      Changes the configuration files that the APs use for argus
Start / End / Elapsed: 20190428 14:26:14.191 / 20190428 14:26:14.206 / 00:00:00.015

- KEYWORD YamlLibrary.Change Config Capture File argus1, ${config_file_1} 00:00:00.004
Start / End / Elapsed: 20190428 14:26:14.191 / 20190428 14:26:14.195 / 00:00:00.004

- KEYWORD OperatingSystem.Grep File ${YAML_PATH}, ${config_file_1} 00:00:00.001
Documentation:      Returns the lines of the specified file that match the pattern.
Start / End / Elapsed: 20190428 14:26:14.195 / 20190428 14:26:14.196 / 00:00:00.001
14:26:14.196 INFO Reading file '/home/dev/Desktop/testing/argus-dev/config.yml'.
14:26:14.196 INFO 1 out of 14 lines matched

- KEYWORD YamlLibrary.Change Config Capture File argus2, ${config_file_2} 00:00:00.004
Start / End / Elapsed: 20190428 14:26:14.196 / 20190428 14:26:14.200 / 00:00:00.004

- KEYWORD OperatingSystem.Grep File ${YAML_PATH}, ${config_file_2} 00:00:00.001
Documentation:      Returns the lines of the specified file that match the pattern.
Start / End / Elapsed: 20190428 14:26:14.200 / 20190428 14:26:14.201 / 00:00:00.001
14:26:14.200 INFO Reading file '/home/dev/Desktop/testing/argus-dev/config.yml'.
14:26:14.201 INFO 1 out of 14 lines matched

- KEYWORD YamlLibrary.Change Config Capture File argus3, ${config_file_3} 00:00:00.004
Start / End / Elapsed: 20190428 14:26:14.201 / 20190428 14:26:14.205 / 00:00:00.004

- KEYWORD OperatingSystem.Grep File ${YAML_PATH}, ${config_file_3} 00:00:00.000
Documentation:      Returns the lines of the specified file that match the pattern.
Start / End / Elapsed: 20190428 14:26:14.205 / 20190428 14:26:14.205 / 00:00:00.000
14:26:14.205 INFO Reading file '/home/dev/Desktop/testing/argus-dev/config.yml'.
14:26:14.205 INFO 1 out of 14 lines matched

- KEYWORD OperatingSystem.Run python3 argus.py 00:00:00.835
Documentation:      Runs the given command in the system and returns the output.
Start / End / Elapsed: 20190428 14:26:14.206 / 20190428 14:26:15.041 / 00:00:00.835
14:26:14.207 INFO Running command 'python3 argus.py 2>&1'.

- KEYWORD OperatingSystem.File Should Exist output.json 00:00:00.000
Documentation:      Fails unless the given path points to an existing file.
Start / End / Elapsed: 20190428 14:26:15.041 / 20190428 14:26:15.041 / 00:00:00.000

```

```

14:26:15.041  INFO  File '/home/dev/Desktop/testing/argus-dev/output.json' exists.

- KEYWORD OperatingSystem. File Should Not Be Empty output.json 00:00:00.001
Documentation: Fails if the specified directory is empty.
Start / End / Elapsed: 20190428 14:26:15.041 / 20190428 14:26:15.042 / 00:00:00.001
14:26:15.042  INFO  File '/home/dev/Desktop/testing/argus-dev/output.json' contains 19 bytes.

- KEYWORD ${output} = JSONLibrary. Get Dict From Json File output.json 00:00:00.000
Start / End / Elapsed: 20190428 14:26:15.042 / 20190428 14:26:15.042 / 00:00:00.000
14:26:15.042  INFO  ${output} = {'loc_updates': []}

- KEYWORD ${locations} = Builtin. Get Length ${output['loc_updates']} 00:00:00.001
Documentation: Returns and logs the length of the given item as an integer.
Start / End / Elapsed: 20190428 14:26:15.042 / 20190428 14:26:15.043 / 00:00:00.001
14:26:15.043  INFO  Length is 0
14:26:15.043  INFO  ${locations} = 0

- KEYWORD Builtin. Should Be Equal ${0}, ${locations} 00:00:00.000
Documentation: Fails if the given objects are unequal.
Start / End / Elapsed: 20190428 14:26:15.043 / 20190428 14:26:15.043 / 00:00:00.000

- TEARDOWN Clean Up Files 00:00:00.002
Documentation: Removes files created in tests
Start / End / Elapsed: 20190428 14:26:15.043 / 20190428 14:26:15.045 / 00:00:00.002

- KEYWORD OperatingSystem. Remove File tests/test_cap_test-01.cap 00:00:00.000
Documentation: Removes a file with the given path.
Start / End / Elapsed: 20190428 14:26:15.044 / 20190428 14:26:15.044 / 00:00:00.000
14:26:15.044  INFO  File '/home/dev/Desktop/testing/argus-dev/tests/test_cap_test-01.cap' does not exist.

- KEYWORD OperatingSystem. Remove File tests/test_cap_test-02.cap 00:00:00.000
Documentation: Removes a file with the given path.
Start / End / Elapsed: 20190428 14:26:15.044 / 20190428 14:26:15.044 / 00:00:00.000
14:26:15.044  INFO  File '/home/dev/Desktop/testing/argus-dev/tests/test_cap_test-02.cap' does not exist.

- KEYWORD OperatingSystem. Remove File tests/test_cap_test-03.cap 00:00:00.001
Documentation: Removes a file with the given path.
Start / End / Elapsed: 20190428 14:26:15.044 / 20190428 14:26:15.045 / 00:00:00.001
14:26:15.044  INFO  File '/home/dev/Desktop/testing/argus-dev/tests/test_cap_test-03.cap' does not exist.

- KEYWORD OperatingSystem. Remove File output.json 00:00:00.000
Documentation: Removes a file with the given path.
Start / End / Elapsed: 20190428 14:26:15.045 / 20190428 14:26:15.045 / 00:00:00.000
14:26:15.045  INFO  Removed file '/home/dev/Desktop/testing/argus-dev/output.json'.

```