

### show\_techsupport.html

Report generated on 28-May-2024 at 01:14:32 by [@test.html](#) v4.1.1

#### Environment

Python	3.8.10
Platform	Linux-5.4.0-162-generic-x86_64-with-glibc2.29
Packages	<ul style="list-style-type: none"> <li>• pytest 7.4.0</li> <li>• pluggy 1.4.0</li> </ul>
Plugins	<ul style="list-style-type: none"> <li>• forked:1.6.0</li> <li>• xdist:1.29.0</li> <li>• ansible:4.0.0</li> <li>• html:4.1.1</li> <li>• allure-pytest:2.8.22</li> <li>• metadata:3.1.1</li> <li>• repeat:0.9.3</li> </ul>

#### Summary

4 tests took 00:38:08.

(Un)check the boxes to filter the results.

2 Failed,  2 Passed,  5 Skipped,  0 Expected failures,  0 Unexpected passes,  3 Errors,  0 Reruns

[Show all details](#) / [Hide all details](#)

Result	Test
Failed	<pre>show_techsupport[test_techsupport.py:test_techsupport[mirroring-ixre-egl-board15]]  request = &lt;FixtureRequest for &lt;Function test_techsupport[mirroring-ixre-egl-board15]&gt;&gt;, config = None duthosts = [MultiAsicSonicHost ixre-egl-board15], &lt;MultiAsicSonicHost ixre-cpm-chassis11&gt;], enum_rand_one_per_hwsku_frontend_hostname = 'ixre-egl-board15'  def test_techsupport(request, config, duthosts, enum_rand_one_per_hwsku_frontend_hostname):     """     test the "show techsupport" command in a loop     :param config: fixture to configure additional setups_list on dut.     :param duthost: DUT host     """     duthost = duthosts[enum_rand_one_per_hwsku_frontend_hostname]     loop_range = request.config.getoption("--loop_num") or DEFAULT_LOOP_RANGE     loop_delay = request.config.getoption("--loop_delay") or DEFAULT_LOOP_DELAY     since = request.config.getoption("--logs_since") or str(randint(1, 5)) + " minute ago"      logger.debug("Loop_range is {} and loop_delay is {}".format(loop_range, loop_delay))</pre>
Failed	<pre>show_techsupport[test_techsupport_no_secret.py:test_secret_removed_from_show_techsupport[ixre-cpm-chassis11]]  DEBUG tests.common.helpers.parallel:parallel.py:153 Started process 1761557 running target "analyzer_logrotate--&lt;MultiAsicSonicHost ixre-egl-board15&gt;" DEBUG tests.common.helpers.parallel:parallel.py:153 Started process 1761561 running target "analyzer_logrotate--&lt;MultiAsicSonicHost ixre-egl-board7&gt;" DEBUG tests.common.helpers.parallel:parallel.py:153 Started process 1761565 running target "analyzer_logrotate--&lt;MultiAsicSonicHost ixre-cpm-chassis11&gt;" INFO tests.common.helpers.parallel:parallel.py:85 process analyzer_logrotate--&lt;MultiAsicSonicHost ixre-egl-board15&gt; terminated with exit code None INFO tests.common.helpers.parallel:parallel.py:85 process analyzer_logrotate--&lt;MultiAsicSonicHost ixre-egl-board7&gt; terminated with exit code None INFO tests.common.helpers.parallel:parallel.py:85 process analyzer_logrotate--&lt;MultiAsicSonicHost ixre-cpm-chassis11&gt; terminated with exit code None DEBUG tests.common.helpers.parallel:parallel.py:161 task completed 3, running 0 INFO tests.common.helpers.parallel:parallel.py:221 Completed running processes for target "analyzer_logrotate" in 0:00:15.513764 seconds DEBUG root:loganalyzer.py:219 Loaded common config. DEBUG root:loganalyzer.py:219 Loaded common config. DEBUG tests.common.helpers.parallel:parallel.py:153 Started process 1761780 running target "analyzer_add_marker--&lt;MultiAsicSonicHost ixre-egl-board15&gt;" DEBUG tests.common.helpers.parallel:parallel.py:153 Started process 1761783 running target "analyzer_add_marker--&lt;MultiAsicSonicHost ixre-egl-board7&gt;" DEBUG tests.common.helpers.parallel:parallel.py:153 Started process 1761787 running target "analyzer_add_marker--&lt;MultiAsicSonicHost ixre-cpm-chassis11&gt;" INFO tests.common.helpers.parallel:parallel.py:85 process analyzer_add_marker--&lt;MultiAsicSonicHost ixre-cpm-chassis11&gt; terminated with exit code None</pre>
Error	<pre>show_techsupport[test_techsupport.py:test_techsupport[ac1-ixre-egl-board15]:setup]  duthosts = [MultiAsicSonicHost ixre-egl-board15], &lt;MultiAsicSonicHost ixre-egl-board7&gt;, &lt;MultiAsicSonicHost ixre-cpm-chassis11&gt;], enum_rand_one_per_hwsku_frontend_hostname = 'ixre-egl-board15' acl_setup = ('dut_tmp_dir': 'acl_tmp/sciphwqs9')  @pytest.fixture(scope='function') def acl(duthosts, enum_rand_one_per_hwsku_frontend_hostname, acl_setup):     """     setup/teardown ACL rules based on test class requirements     :param duthost: DUT host object     :param acl_setup: setup information     :return:     """     duthost = duthosts[enum_rand_one_per_hwsku_frontend_hostname]     acl_facts = duthost.acl_facts()['ansible_facts']['ansible_acl_facts']     pytest_require(ACL_TABLE_NAME in acl_facts, ("acl table not exists"))</pre>
Error	<pre>show_techsupport[test_techsupport.py:test_techsupport[mirroring-ixre-egl-board15]:teardown]  duthosts = [MultiAsicSonicHost ixre-egl-board15], &lt;MultiAsicSonicHost ixre-egl-board7&gt;, &lt;MultiAsicSonicHost ixre-cpm-chassis11&gt;] request = &lt;SubRequest 'loganalyzer' for &lt;Function test_techsupport[mirroring-ixre-egl-board15]&gt;&gt;  @pytest.fixture(autouse=True) def loganalyzer(duthosts, request):     if request.config.getoption("--disable_loganalyzer") or "disable_loganalyzer" in request.keywords:         logging.info("Log analyzer is disabled")         yield         return      # Analyze all the duts     fail_test = not (request.config.getoption("--ignore_la_failure"))     store_la_logs = request.config.getoption("--store_la_logs")     analyzers = {}     should_rotate_log = request.config.getoption("--loganalyzer_rotate_logs")</pre>
Error	<pre>show_techsupport[test_techsupport_no_secret.py:test_secret_removed_from_show_techsupport[ixre-egl-board15]:teardown]  duthosts = [MultiAsicSonicHost ixre-egl-board15], &lt;MultiAsicSonicHost ixre-egl-board7&gt;, &lt;MultiAsicSonicHost ixre-cpm-chassis11&gt;] request = &lt;SubRequest 'loganalyzer' for &lt;Function test_secret_removed_from_show_techsupport[ixre-egl-board15]&gt;&gt;  @pytest.fixture(autouse=True) def loganalyzer(duthosts, request):     if request.config.getoption("--disable_loganalyzer") or "disable_loganalyzer" in request.keywords:         logging.info("Log analyzer is disabled")         yield         return      # Analyze all the duts     fail_test = not (request.config.getoption("--ignore_la_failure"))     store_la_logs = request.config.getoption("--store_la_logs")     analyzers = {}     should_rotate_log = request.config.getoption("--loganalyzer_rotate_logs")</pre>