CSIT TOI

CSIT Libraries
Tibor Frank
Feb 6, 2018
CSIT Libraries

• L1 and L2 libraries in CSIT hierarchy
• L1 Python Libraries
• L2 Robot Libraries
• Test Lifecycle Abstraction
• L2 RF Keywords Functional Classification
• CSIT Design Guidelines
L1 and L2 Libraries in CSIT Hierarchy

CSIT System Design Hierarchy
L1 Python Libraries

• Lowest level CSIT libraries abstracting underlying test environment, SUT, DUT and TG specifics;
• Used commonly across multiple L2 keywords.

• Performance and functional tests:
  • L1 keywords are implemented as Python libraries.

Source code: https://git.fd.io/csit/tree/resources/libraries/python
For more information see: https://wiki.fd.io/view/CSIT/Design_Optimizations#CSIT_System_Design_Hierarchy
Performance TG L1 Keywords

- Support for TRex and wrk traffic generators today;
- CSIT IXIA drivers in progress.

Source code: [https://git.fd.io/csit/tree/resources/tools/trex](https://git.fd.io/csit/tree/resources/tools/trex)
Source code: [https://git.fd.io/csit/tree/resources/tools/wrk](https://git.fd.io/csit/tree/resources/tools/wrk)
Traffic profiles and scripts

• Performance data plane traffic profiles
  • TG-specific stream profiles provide full control of:
    • Packet definition;
    • Stream definitions;
    • Stream profiles are independent of CSIT framework;
    • Easily extensible;
    • Same stream profile can be (and is) used for different tests.

• Functional data plane traffic scripts
  • Scapy specific traffic scripts

Source code: https://git.fd.io/csit/tree/resources/traffic_profiles
Source code: https://git.fd.io/csit/tree/resources/traffic_scripts
L2 Robot Libraries

• Higher level CSIT libraries abstracting required functions for executing tests;
• L2 KWs are classified into the following functional categories:
  • Configuration;
  • Test;
  • Verification;
  • StateReport;
  • SuiteSetup;
  • TestSetup;
  • SuiteTeardown;
  • TestTeardown.

Source code: https://git.fd.io/csit/tree/resources/libraries/robot
For more information see: https://wiki.fd.io/view/CSIT/Design_Optimizations#RF_Keywords_Functional_Classification
Test Lifecycle Abstraction

• Anatomy of Good Tests for CSIT:
  • Suite Setup
  • Test Setup
  • Test Case – uses L2 KWs with RF Gherkin style:
    • prefixed with {Given} – Verification of Test setup, reading state: uses
      Configuration KWs, Verification KWs, StateReport KWs;
    • prefixed with {When} – Test execution: Configuration KWs, Test KWs;
    • prefixed with {Then} – Verification of Test execution, reading state: uses
      Verification KWs, StateReport KWs;
  • Test Teardown
  • Suite Teardown

For more information see: https://wiki.fd.io/view/CSIT/Design_Optimizations#Test_Lifecycle_Abstraction
CSIT Design Guidelines

- Strictly follow the [CSIT Design Hierarchy](https://wiki.fd.io/view/CSIT/Design_Optimizations#Applying_CSIT_Design_Guidelines);
- Tests MUST use only L2 Keywords;
- All L2 KWs are composed of L1 KWs and/or other L2 KWs;
- Test and KW naming should be following CSIT naming guidelines;