LoadCore SCAS Test Library

User Guide

KEYSIGHT

202407081159+03:00

Notices

Copyright Notice

Warranty

The material contained in this document is provided “as is,” and is subject to being changed, without notice, in future editions. Further, to the maximum extent permitted by applicable law, Keysight disclaims all warranties, either express or implied, with regard to this manual and any information contained herein, including but not limited to the implied warranties of merchantability and fitness for a particular purpose. Keysight shall not be liable for errors or for incidental or consequential damages in connection with the furnishing, use, or performance of this document or of any information contained herein. Should Keysight and the user have a separate written agreement with warranty terms covering the material in this document that conflict with these terms, the warranty terms in the separate agreement shall control.

Technology Licenses

The hardware and/or software described in this document are furnished under a license and may be used or copied only in accordance with the terms of such license.

U.S. Government Rights

The Software is "commercial computer software," as defined by Federal Acquisition Regulation ("FAR") 2.101. Pursuant to FAR 12.212 and 27.405-3 and Department of Defense FAR Supplement ("DFARS") 227.7202, the U.S. government acquires commercial computer software under the same terms by which the software is customarily provided to the public. Accordingly, Keysight provides the Software to U.S. government customers under its standard commercial license, which is embodied in its End User License Agreement (EULA), a copy of which can be found at http://www.keysight.com/find/sweula. The license set forth in the EULA represents the exclusive authority by which the U.S. government may use, modify, distribute, or disclose the Software. The EULA and the license set forth therein, does not require or permit, among other things, that Keysight: (1) Furnish technical information related to commercial computer software or commercial computer software documentation that is not customarily provided to the public; or (2) Relinquish to, or otherwise provide, the government rights in excess of these rights customarily provided to the public to use, modify, reproduce, release, perform, display, or disclose commercial computer software or commercial computer software documentation. No additional government requirements beyond those set forth in the EULA shall apply, except to the extent that those terms, rights, or licenses are explicitly required from all providers of commercial computer software pursuant to the FAR and the DFARS and are set forth specifically in writing elsewhere in the EULA. Keysight shall be under no obligation to update, revise or otherwise modify the Software. With respect to any technical data as defined by FAR 2.101, pursuant to FAR 12.211 and 27.404.2 and DFARS 227.7102, the U.S. government acquires no greater than Limited Rights as defined in FAR 27.401 or DFAR 227.7103-5 (c), as applicable in any technical data. 52.227-14 (June 1987) or DFAR 252.227-7015 (b)(2) (November 1995), as applicable in any technical data.

Contacting Us

Keysight headquarters

1400 Fountaingrove Parkway
Santa Rosa, CA 95403-1738
www.ixiacom.com/contact/info

Support

Global Support	+1 818 595 2599	support@ixiacom.com
Regional and local support contacts:		
APAC Support	+91 80 4939 6410	support@ixiacom.com
Australia	+61-742434942	support@ixiacom.com
EMEA Support	+40 21 301 5699	support-emea@ixiacom.com
Greater China Region	+400 898 0598	support-china@ixiacom.com
Hong Kong	+852-30084465	support@ixiacom.com
India Office	+91 80 4939 6410	support-india@ixiacom.com
Japan Head Office	+81 3 5326 1980	support-japan@ixiacom.com
Korea Office	+82 2 3461 0095	support-korea@ixiacom.com
Singapore Office	+65-6215-7700	support@ixiacom.com
Taiwan (local toll-free number)	00801856991	support@ixiacom.com

Table of Contents

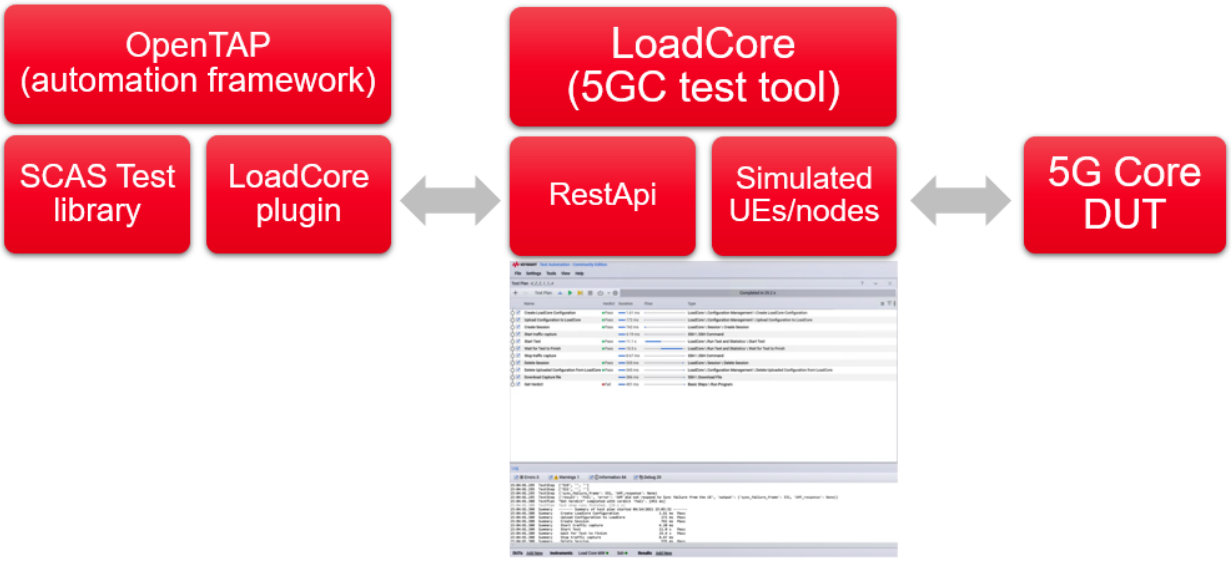
CHAPTER 1

Overview

OpenTAP is an Open Source project for fast and easy development and execution of automated tests.

This guide covers the steps needed to install and configure OpenTap in order to run LoadCore SCAS test suite.

The following components are involved when running the SCAS test library:



Currently, LoadCore SCAS test library offers support for the following validation tests:

Test Case AMF (as described in TS 33.512)	**Description**
4.2.2.1	Authentication and key agreement procedure
• 4.2.2.1.1 (A, B)	Synchronization failure handling
• 4.2.2.1.2 (A, B, C, D)	RES* verification failure handling
4.2.2.3	Security mode command procedure
• 4.2.2.3.1	Replay protection of NAS signaling messages
• 4.2.2.3.2	NAS NULL integrity protection
• 4.2.2.3.3	NAS integrity algorithm selection and use
4.2.2.4	Security in intra-RAT mobility
• 4.2.2.4.1	Bidding down prevention in Xn-handover security in intra-RAT mobility

Test Case AMF (as described in TS 33.512)	Description
• 4.2.2.4.2	NAS protection algorithm selection in AMF change
4.2.2.5	5G-GUTI allocation
• 4.2.2.5.1 (A, B, C)	5G-GUTI allocation
4.2.2.6	Security in registration procedure
• 4.2.2.6.1	Invalid or unacceptable UE security capabilities handling

Test case UPF (as described in TS 33.513)	Description
4.2.2	Security functional requirements on the UPF
• 4.2.2.1, 4.2.2.2, 4.2.2.3	Confidentiality, Integrity and Replay protections over N3 interface
• 4.2.2.6	TEID uniqueness

Test case UDM (as described in TS 33.514)	Description
4.2.1	User Privacy Procedure
• 4.2.1.1	De-concealment of SUPI based on the protection scheme used to generate the SUCI
4.2.2	Authentication and key agreement procedure
• 4.2.2.1	Synchronization failure handling
• 4.2.2.2	Storing of authentication status of UE by UDM

Test case SMF (as described in TS 33.515)	Description
4.2.2	Security functional requirements on the SMF
• 4.2.2.1.1	Priority of UserPlane Security Policy
• 4.2.2.1.3	Security functional requirements on the SMF checking UserPlane security policy
• 4.2.2.1.4	Charging ID Uniqueness

Test case NRF (as described in TS 33.518)	Description
4.2.2.2	NF discovery procedure

Test case NRF (as described in TS 33.518)	Description
• 4.2.2.2.1	NF discovery authorization for specific

The LoadCore SCAS test library can be downloaded from the official Keysight LoadCore support page.

CHAPTER 2
Prerequisites

For a complete and correct functioning setup, the following are required:

- LoadCore system installed and active. This includes at least one Agent, one Middleware and one License server. The required number of licenses need to be available also (Control Plane and Impairment).
- License for the related SCAS test library:

Part Number	Description
P89047A	WRLS 5G LoadCore, SCAS AMF 33512 test library
P89048A	WRLS 5G LoadCore, SCAS AMF, SMF, UPF, AUSF, UDM, NRF test libraries
P89049A	WRLS 5G LoadCore, SCAS UDM 33514 test library
P89052A	WRLS 5G LoadCore, SCAS NRF 33518 test library

- Wireshark - latest version.

This page intentionally left blank.

CHAPTER 3
OpenTap Installation

This section describe the steps needed in order to install OpenTab on different supported platforms.

Windows Installation

1. Download OpenTAP from the homepage here.
2. Start the installer.

It is recommend that you download the Software Development Kit, or simply the Developer's System Community Edition provided by Keysight Technologies. The Developer System is a bundle that contains the SDK as well as a graphical user interface and result viewing capabilities.

It can be installed by typing the following:

```
cd %TAP_PATH%
tap package install "Developer's System CE" -y
```

Linux Installation

1. Install dependencies

On Linux, OpenTAP has a few dependencies that must be manually installed, namely libc6, libunwind, unzip, git, and curl. On Debian derivatives, these can be installed by running the following command:

```
apt-get install libc6-dev libunwind8 unzip git curl
```

Note that the packages may have different names on other distributions. OpenTAP should still work if you install the equivalent packages for your distribution.

In addition to these packages, OpenTAP depends on dotnet core runtime version 2.1. Version 3.0 and greater are not supported. The installation procedure depends on your distribution. Please see the official documentation from Microsoft for further instructions.

2. Install OpenTAP

Download the OpenTAP distribution (`.tar`) from the homepage here.

Install the downloaded distribution:

- `.tar` do the following:
  1. Untar the package in you home directory: `tar -xf OpenTAP*.tar`
  2. Change the permission of the INSTALL.sh file to be executable: `chmod u+x INSTALL.sh`
  3. Run the `INSTALL.sh` script: `./INSTALL.sh`

Docker Installation

We also provide docker images for running OpenTAP. You can find them at hub.docker.com/r/opentapio/opentap.

We maintain two images:

1. a development image which includes all necessary tools to build OpenTAP projects (~2.5GB)
2. a production image which includes only dependencies required to run OpenTAP (~330MB)

The development image is widely used for building and packaging plugins in highly reproducible environments, and we use it internally for continuous deployment. Have a look at the Demonstration plugin's gitlab CI file where we build, test, version, and publish the plugin directly in a continuous integration pipeline.

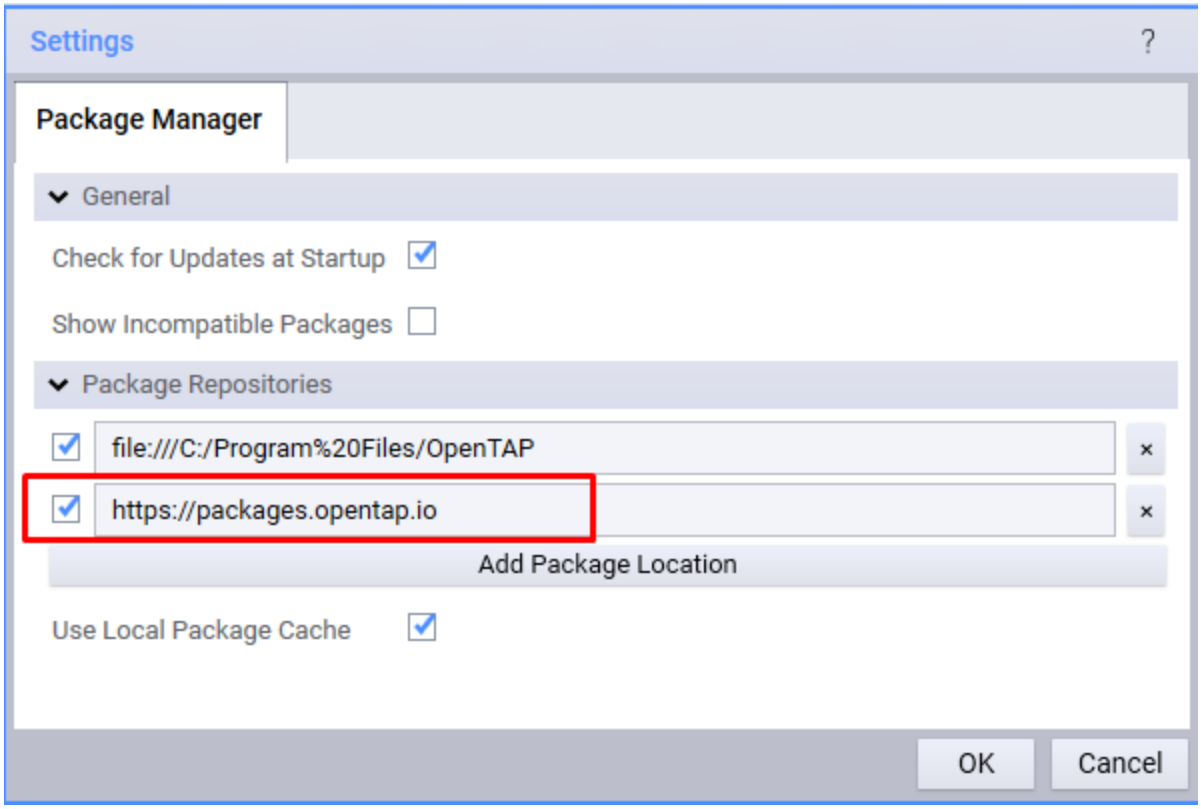
CHAPTER 4
OpenTap Configuration

This section describe the steps needed in order to configure OpenTab to run LoadCore tests.

Install the required packages for OpenTap

The first step is to confirm/configure the URL in order to get the latest packages for OpenTab:

1. Open OpenTap Editor and select **Tools** > **Package Manager**.
   The Package Manager application is displayed.
2. From Package Manager, select **Settings**.
   The Settings window is displayed.
3. From the Package Repositories section, confirm if the following URL exists: https://packages.opentap.io. If not, select the **Add URL** button and add the link to the list:

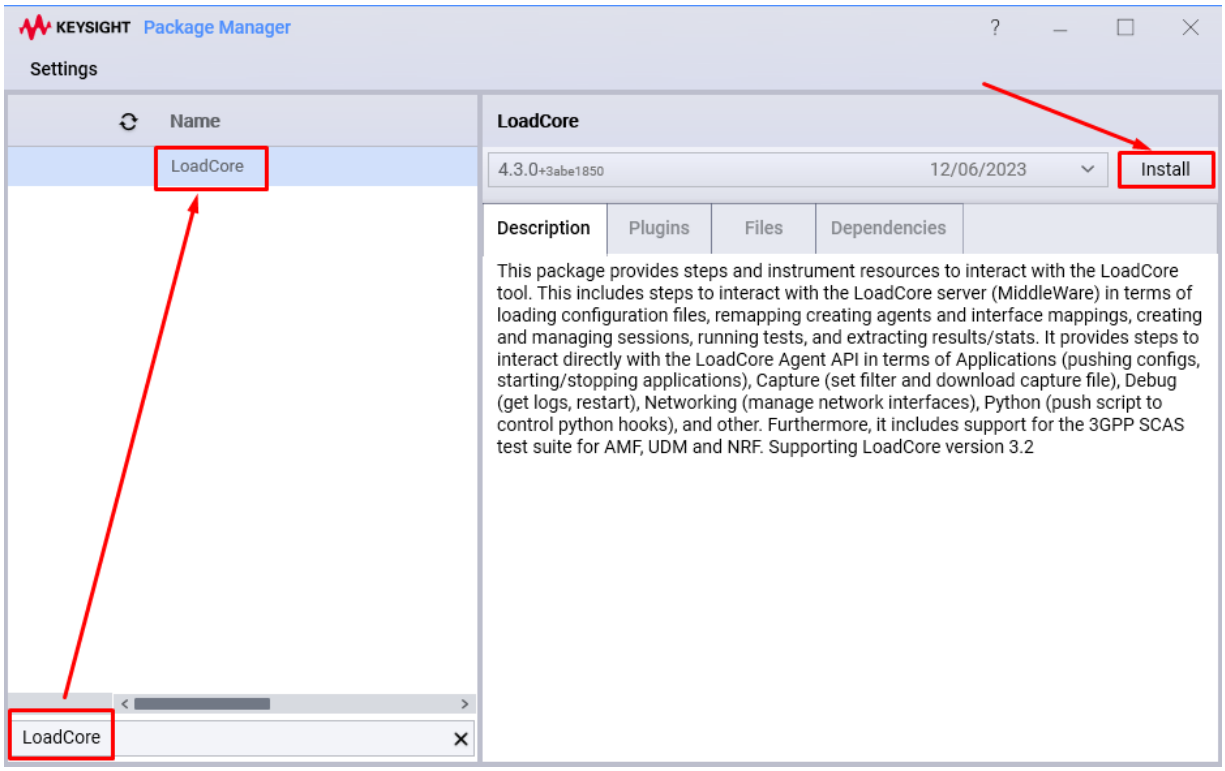


4. Select **OK** to apply the changes.

The second step is to install the required packages for OpenTab. These package are made available through the URL configured above.

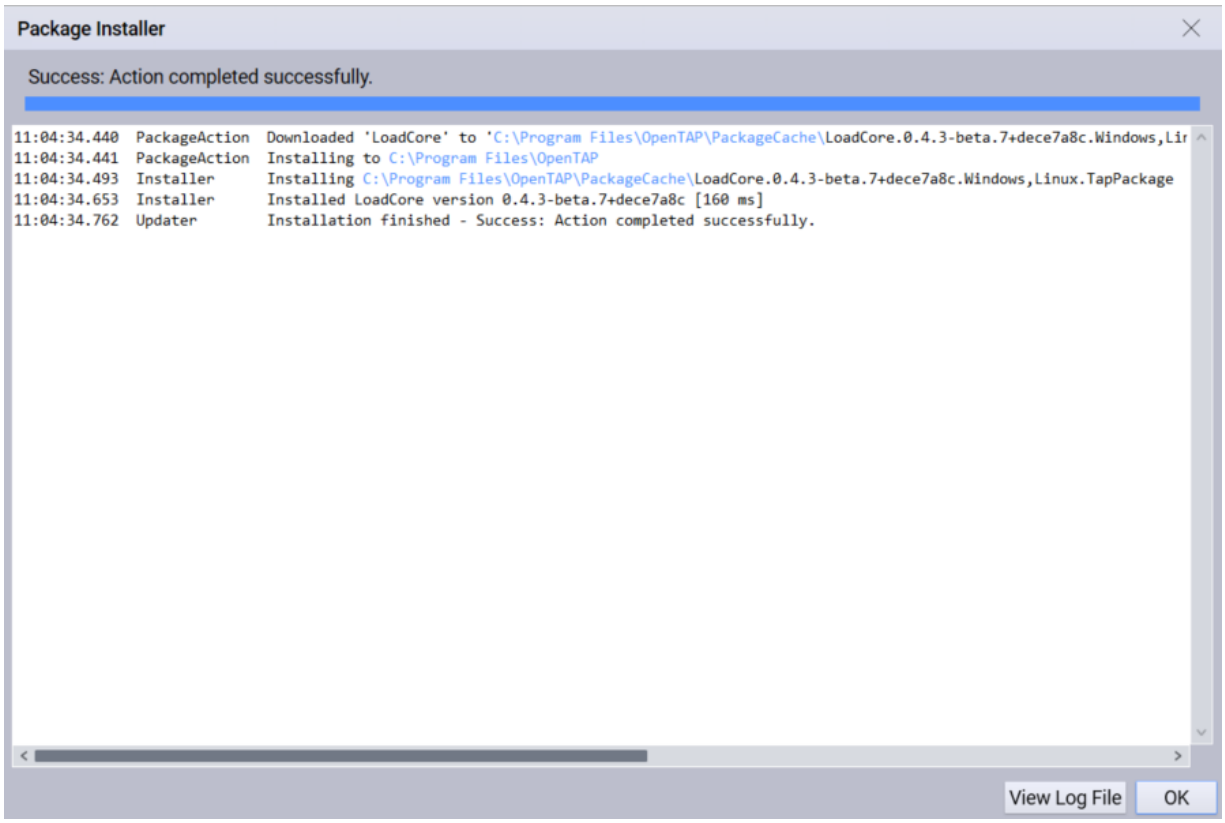
The **LoadCore package** provides steps and instrument resources to interact with the LoadCore tool. It can be installed as follows:

1. In the Search bar, type **LoadCore** (or use the scroll option to manually find the package).



IMPORTANT Make sure that the latest version of the LoadCore package is installed. For this, use the drill-down option for the LoadCore package to check if a newer version is available. Also, when updating to a newer version of the LoadCore package, it is recommended to close OpenTab.

2. Select the Loadcore package and then select **Install**.
   The Package Installer window is displayed.



3. Select **OK**.

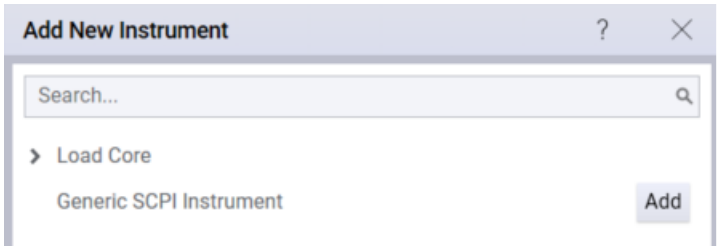
TIP

Alternatively, you can manually download the LoadCore package from https://packages.opentap.io, and use the `tap package install <package_name>` CLI command from your installation directory.

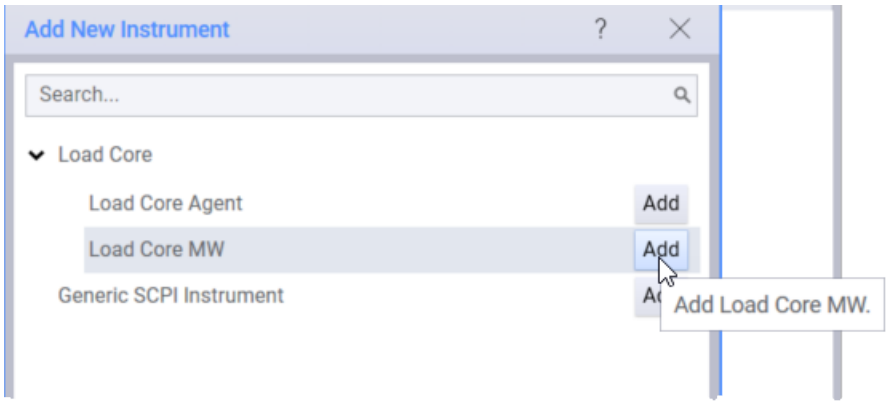
Create and configure OpenTap instruments

Create the required instruments, as follows:

1. From the OpenTap Editor, select the **Add New** button.
   The Bench Settings window is displayed.
2. Select the **Instruments** tab and then select the **Add an Item** button.
   The Add New Instrument window opens and the available instruments are displayed.

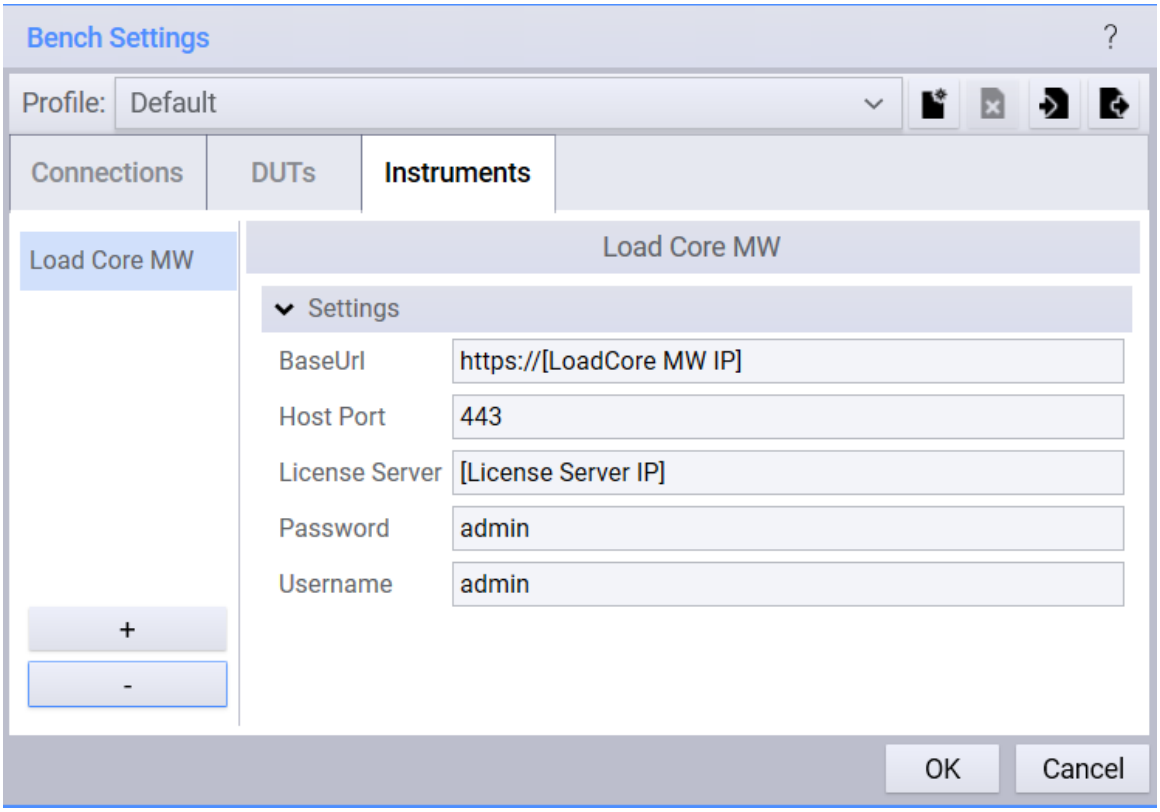


3. Double-click on **LoadCore** and select the **Add** button for Load Core MW option.



Select **Close**.

a. On the Settings section, set the base URL for the MW (the host post is, by default, set to 443).



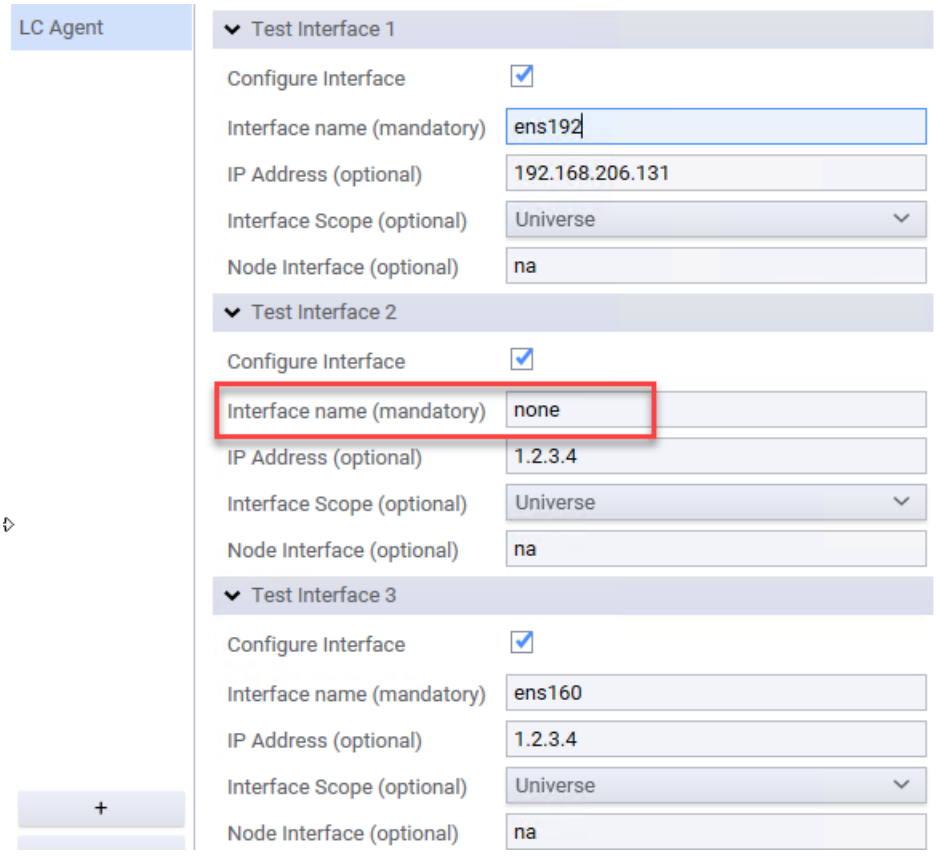
b. Make sure to add the License Server IP. The License Server IP is the same as the one used in LoadCore (gear menu > **Application Settings**).

4. Select the **Add an Item** button in order to start the configuration of the LC Agent instrument. Select the Load Core Agent instrument and then select **Close**.

a. On the Settings section, set the management endpoint path and the management interface for the agent.

b. Configure the SSH connection. Set the host details and provide the authentication credentials. The IP address that is used here is the IP address of the test agent.

c. Configure the test interfaces for the agent (these must match the agent's interfaces from LoadCore). For each interface, select the **Configure Interface** check box in order to enable the interface. The interface name is mandatory.



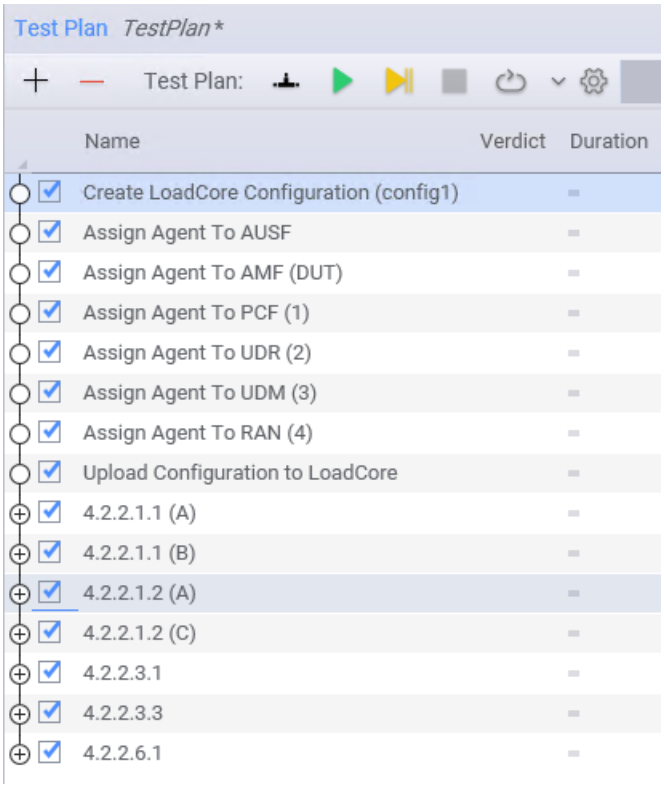
IMPORTANT A **none** interface must be defined (as presented in the image above). The purpose of this interface is to be used later on as a passthrough interface (more details here).

5. Select **OK**.

Test plan import and test steps configuration

The following example describes the steps needed to import a test plan and to configure the test steps.

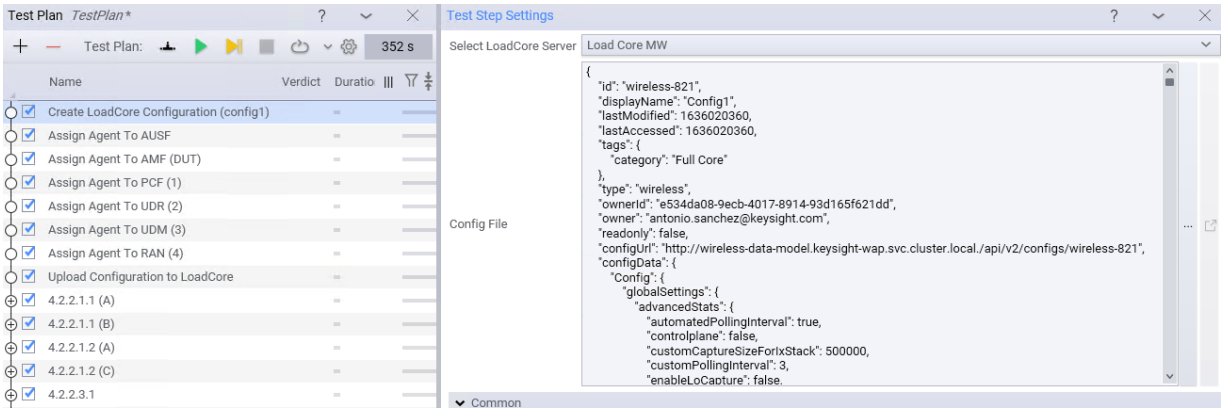
From the OpenTap Editor, we will import the following test plan:



NOTE Additional steps can be added using the **+** button.

After importing the test plan, we need to configure the test steps, as follows:

1. Select the **Create LoadCore Configuration** step and import the `.json` configuration file:
   - Select the **Do you want to import and existing configuration** check box and,
   - Select **Click here to import a .json file** to import the configuration file.



This operation is loading the `.json` test configuration into OpenTap.

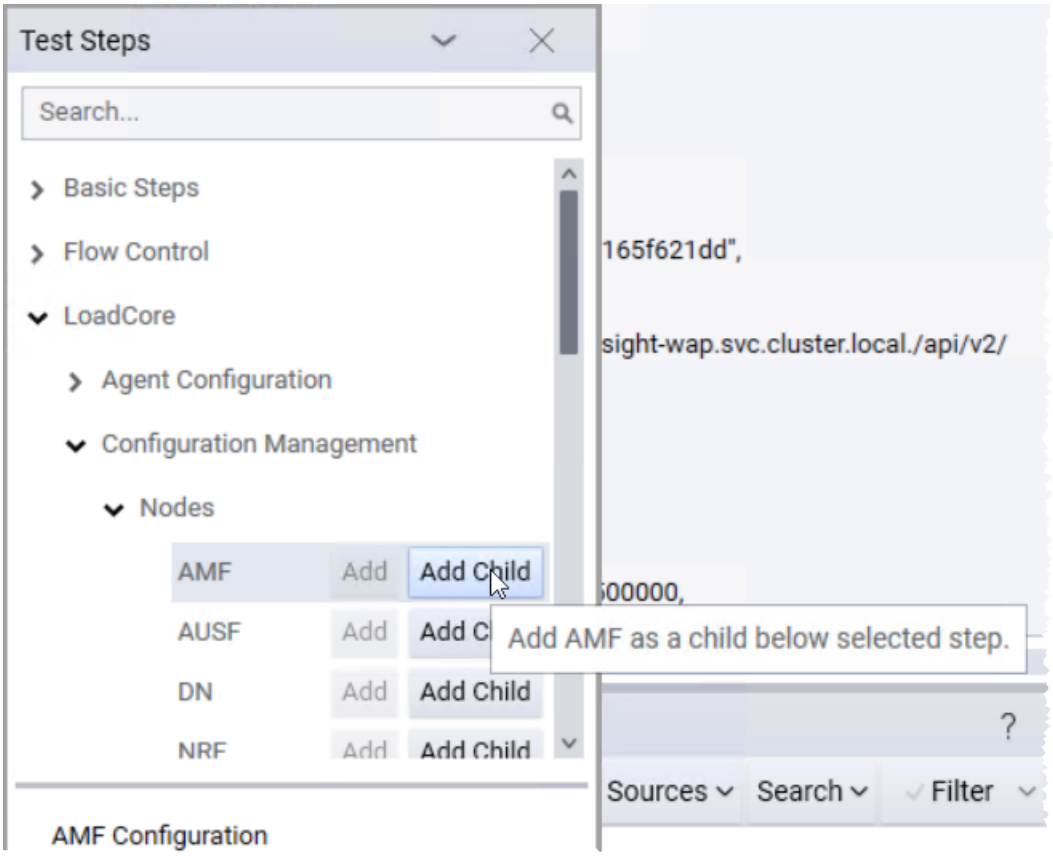
You need to load the `.json` configuration file corresponding to this test plan.

This configuration should be adjusted to match the current setup you have. The easiest way is to import the configuration in LoadCore, configure the system (assign the agents and define the DUT, select the impairment profile – SCAS), export the configuration and use it in OpenTap.

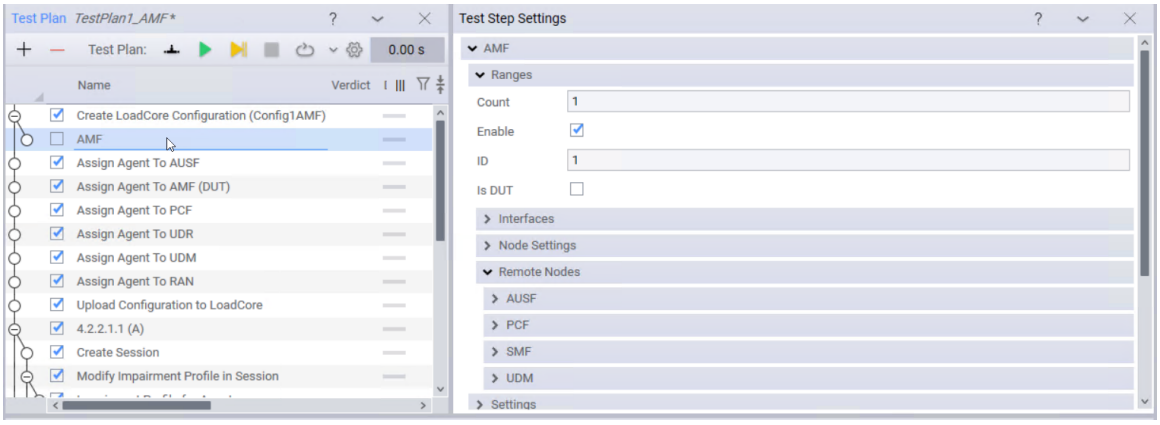
Also, if required, the parameters can be changed directly from the test plan:

a. Select the **+** button from the upper-left corner.

b. From the Test Steps window, select **Loadcore** > **Configuration Management** > **Nodes** and then, select the child node that you want to configure (in the example below, the AMF child is selected)



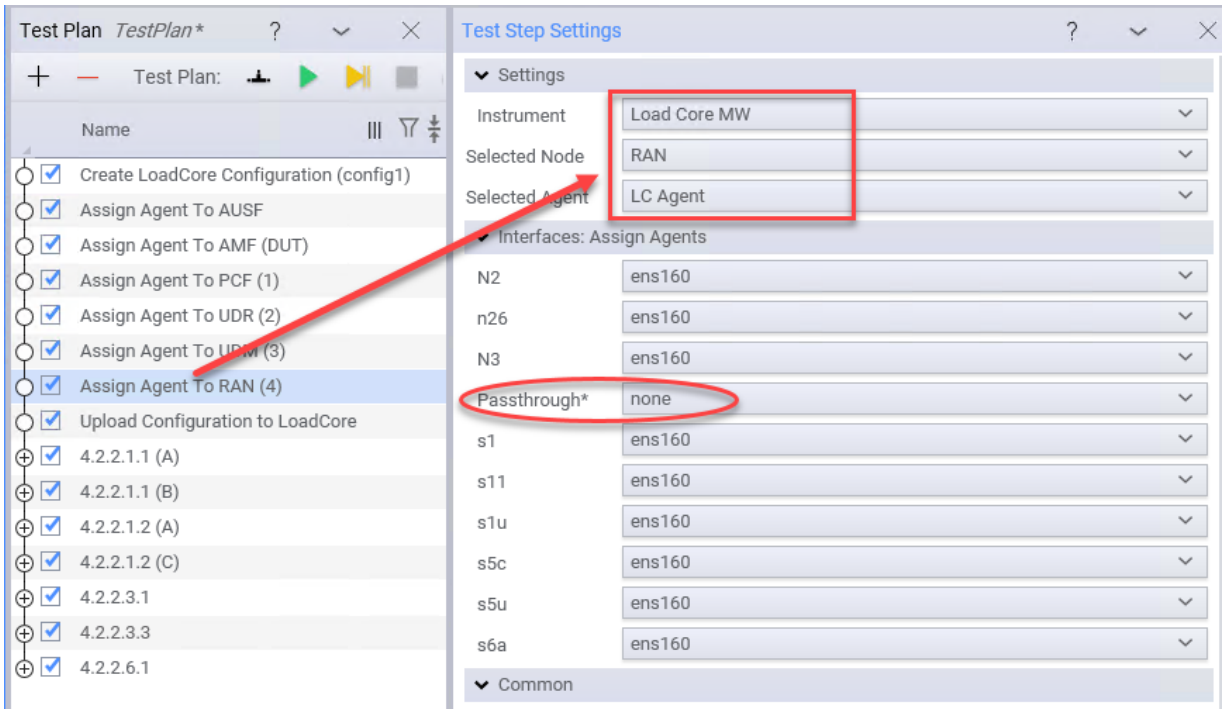
c. The newly created child inherits the configuration from the test plan. This configuration can be modified from the Test Step Settings window, Ranges section.



Make sure that the **Enabled** check box from the Common section is selected in order to enable this test step to be used when the test plan is executed (this recommendation

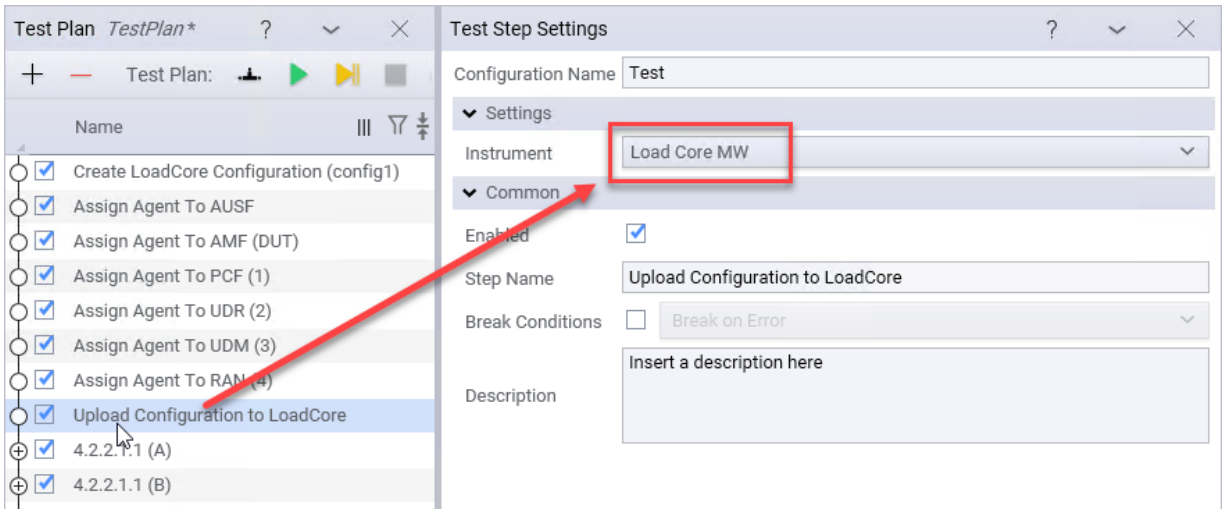
applies to all test steps). Also, if required, you can enable the **Break Conditions** check box an specify the break condition (this will stop the test when the condition is met).

2. Assign the agents to the LoadCore MW nodes. Select the MW instrument, the node and the agent instrument.
   Make sure that the interfaces are correctly assigned.

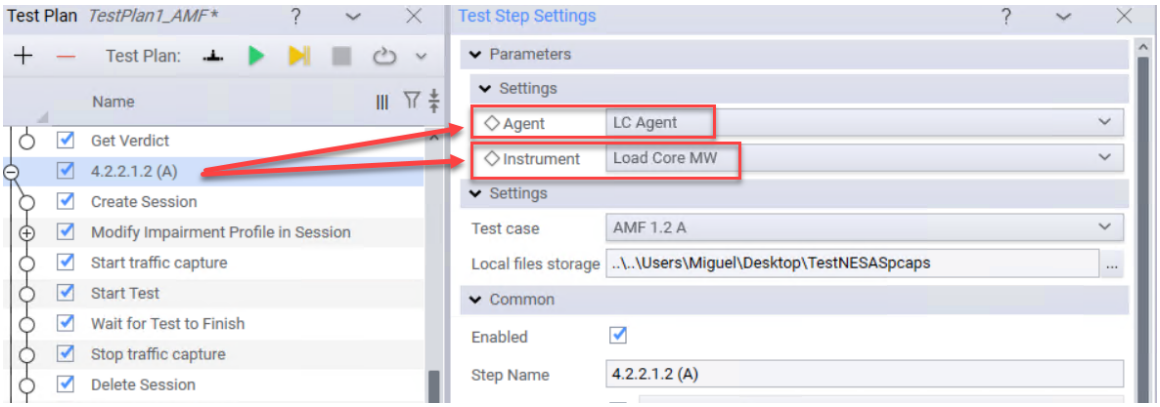


IMPORTANT When assigning the agent to the RAN node, make sure that the **Passthrough** interface is set to **none**.
To set this interface to **none**, it has to be defined in the agent instrument, as described here.

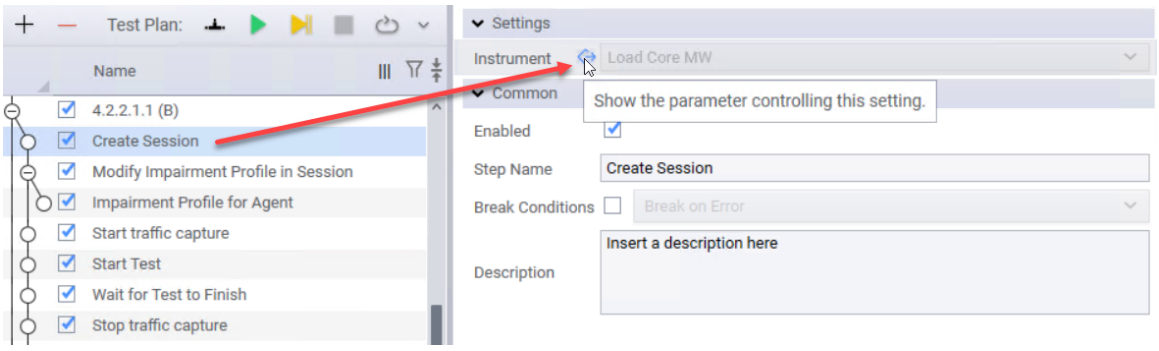
3. Select the **Upload configuration to LoadCore** step and select the Middleware instrument.



4. For each test case:
   a. Make sure that on the Test Step Settings window > **Parameters** > **Settings**, the Agent and the MW instrument are correctly assigned.

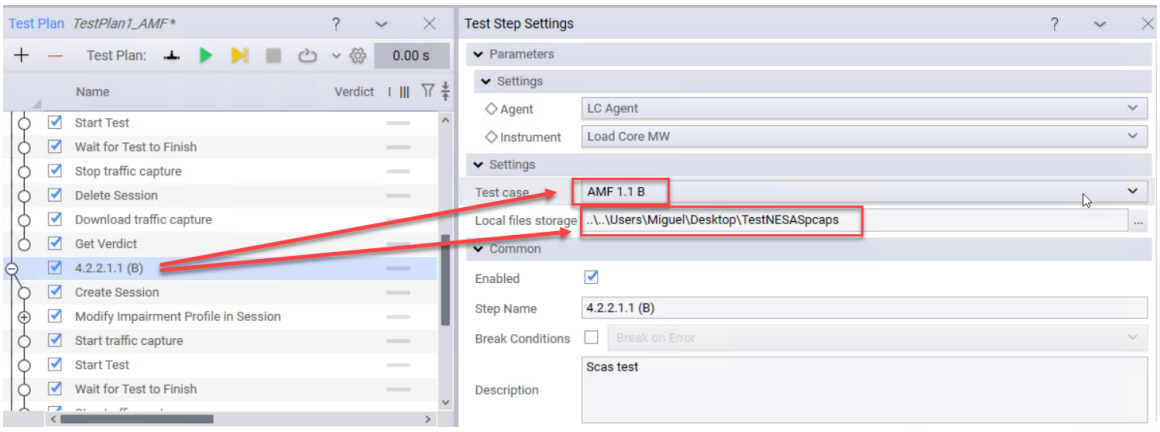


   For all test steps under a test case, the instruments used are assigned by default (Agent, MW or both). All test steps are using the same instruments defined above as parameters.



   If needed, in order to change this, right-click on Instrument and select **Unparameterize**. This will make this field editable and will allow you to select another value from the drop-down list.

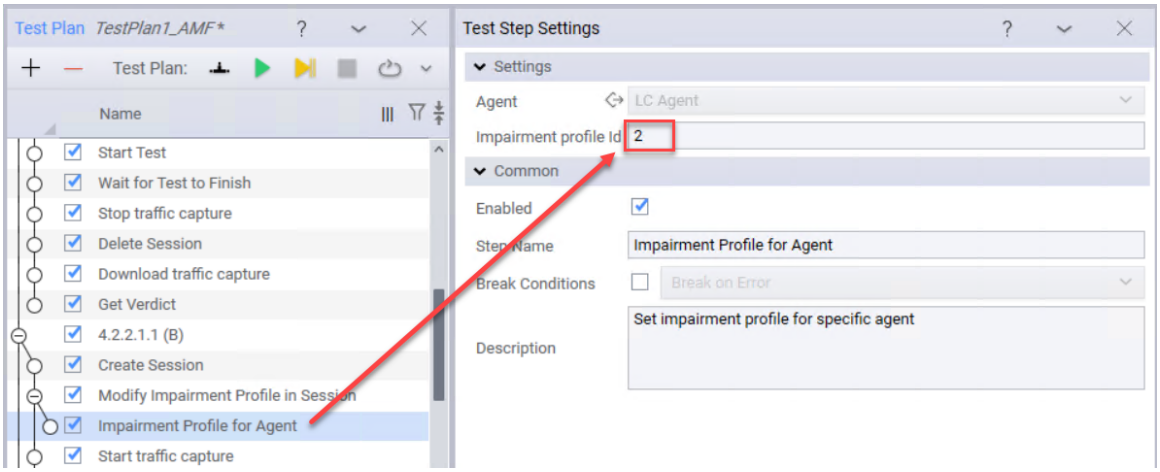
   b. Make sure that the test case is correctly assigned and set the capture file location.



IMPORTANT Selecting a different test case from the drop-down list will cause the test to fail (refer to the table presented below for more details).

NOTE It is recommended to delete the capture files folders when they are not needed anymore in order to save disk space.

c. Expand **Modify Impairment Profile in Session** and select **Impairment Profile for Agent**. Make sure that the correct Impairment profile Id is correctly assigned (refer to the table presented below for more details).



Test Plan	Test Step	Test Case	Impairment Profile Id
TestPlan_AMF	4.2.2.1.1 (A)	AMF 1.1 A	1
	4.2.2.1.1 (B)	AMF 1.1 B	2
	4.2.2.1.2 (A)	AMF 1.2 A	3
	4.2.2.1.2 (B)	AMF 1.2 B	1
	4.2.2.1.2 (C)	AMF 1.2 C	4
	4.2.2.1.2 (D)	AMF 1.2 D	2

Test Plan	Test Step	Test Case	Impairment Profile Id
	4.2.2.3.1	AMF 3.1	5
	4.2.2.3.2 (A)	AMF 3.2 A	
	4.2.2.3.3	AMF 3.3	
	4.2.2.4.1	AMF 4.1	1
	4.2.2.4.2	AMF 4.2	
	4.2.2.5.1 (A)	AMF 5.1 A	
	4.2.2.5.1 (B)	AMF 5.1 B	
	4.2.2.5.1 (C)	AMF 5.1 C	
	4.2.2.6.1	AMF 6.1	6
TestPlan_UDM	4.2.1.1	UDM 1.1	
	4.2.2.1	UDM 2.1	1
	4.2.2.2	UDM 2.2	
TestPlan_UPF	4.2.2.1	UPF 2.1	
	4.2.2.2	UPF 2.2	
	4.2.2.3	UPF 2.3	
	4.2.2.6	UPF 2.6	
TestPlan_SMF	4.2.2.1.1	SMF 1.1	
	4.2.2.1.3	SMF 1.3	1
	4.2.2.1.4	SMF 1.4	
TestPlan_NRF	4.2.2.2.1	NRF 2.1	1

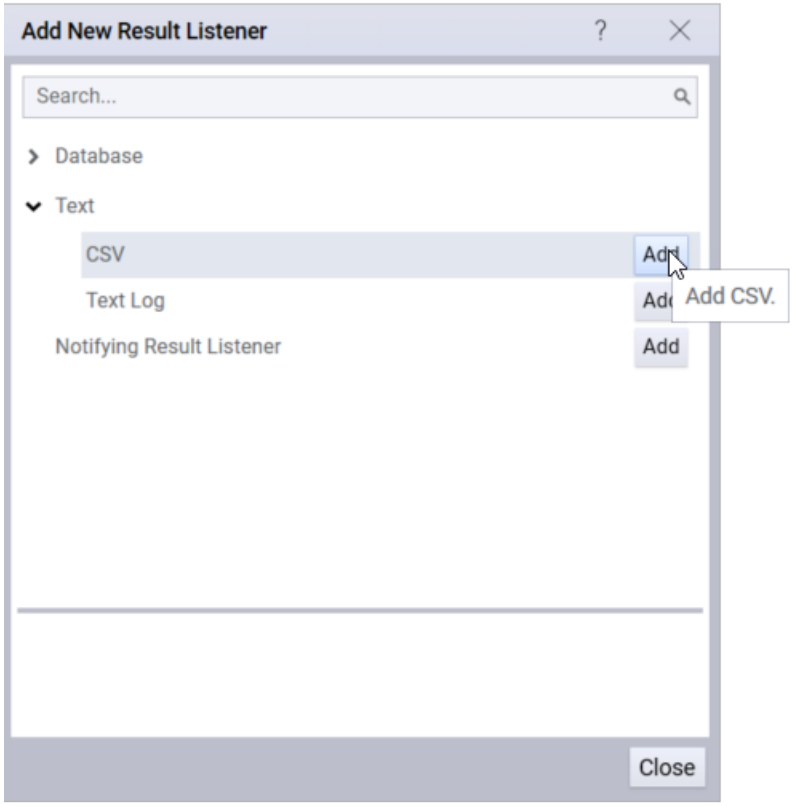
This page intentionally left blank.

CHAPTER 5
Results

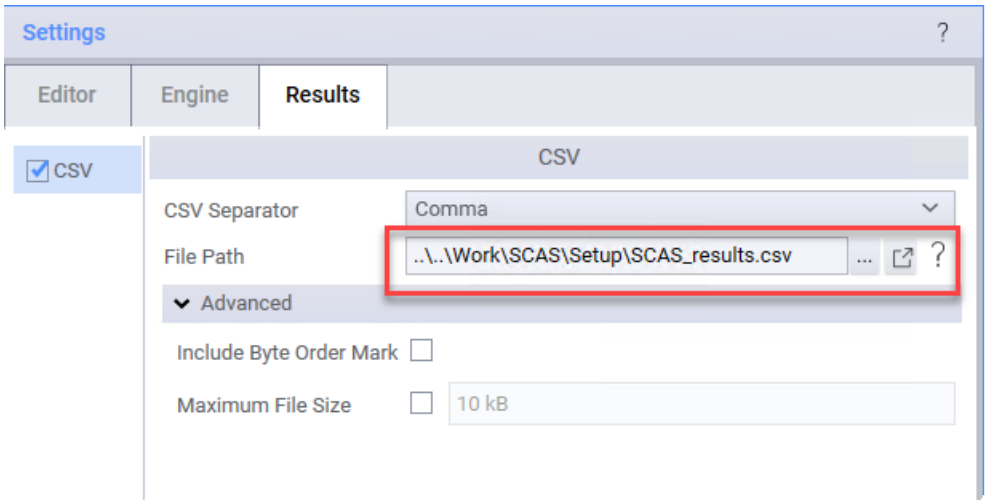
OpenTap Results

The test results from OpenTap can be stored into a `.csv` file. To do this, you must enable CSV generation and also to provide the file location, as follows:

1. From the OpenTap Editor, select **Settings** > **Results**. The Settings window is displayed.
2. On the Results Tab, select the **Add an Item** button. The Add New Results Listener window is displayed.
3. Double-click on **Text** and select the **Add** button from the CSV line.

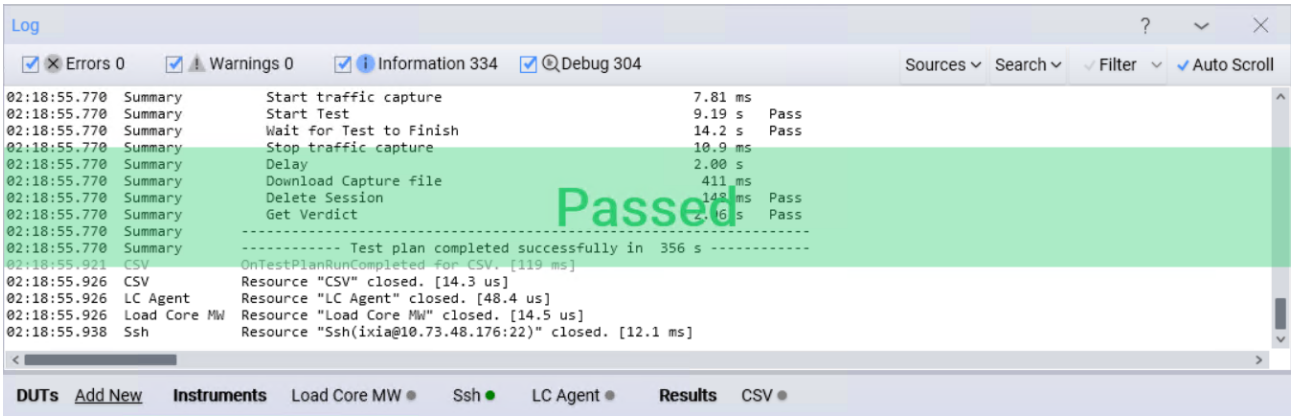


   Select **Close**.
4. Select the CSV delimiter. Supported delimiters are semicolon, comma and tab.
5. Set the path for the `.csv` file.

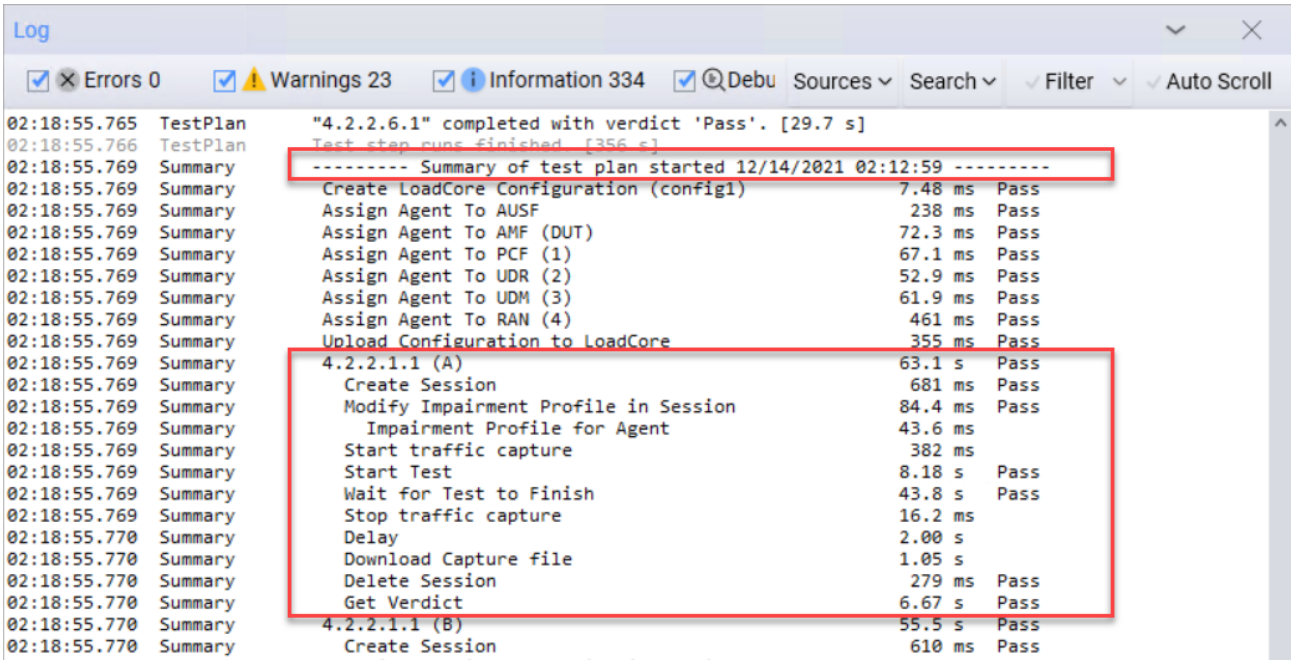


6. Select **OK**.

The test result can be also checked from the Log panel. After successfully running the LoadCore SCAS test library, the following message is displayed in the Log panel:



The results for each test are displayed under the Summary section in the Log panel, as presented in the example below:



A sample of the `SCAS_results.csv` file is attached to this document.

LoadCore Results

From LoadCore, the test results can be retrieved from Test Results window.

To access the Test Results window, select **Browse Results**.

From here, select the test and then, select **Download** > **CSV** in order to download the results as a `.csv` file.

This page intentionally left blank.

CHAPTER 6
Tests Description

This section offers more details about the tests from the SCAS test library.

AMF Tests Description

The following tests require the AMF to be set as DUT.

4.2.2.1.1 (A, B) Synchronization failure handling

The tests verify that synchronization failure is correctly handled by the AMF.

4.2.2.1.1 (A) AUSF responds to Authentication message form DUT (AMF)

Execution Steps:

1. The UE sends an authentication failure message to the AMF with *synchronisation failure* (AUTS).
2. The AMF sends a `Nausf_UEAuthentication_Authenticate Request` message with a *synchronisation failure indication* to the AUSF.
3. The AUSF sends a `Nausf_UEAuthentication_Authenticate Response` message to the AMF immediately after receiving the request from the AMF, to make sure the AMF will receive the response before timeout.

The test is marked as **PASSED** if the AMF drops the ongoing authentication process and may initiate a new authentication towards the UE. Otherwise, is marked as **FAILED**.

4.2.2.1.1 (B) AUSF doesn’t respond to Authentication message form DUT (AMF)

Execution Steps:

1. The UE sends an authentication failure message to the AMF with *synchronisation failure* (AUTS).
2. The AMF sends a `Nausf_UEAuthentication_Authenticate Request` message with a *synchronisation failure indication* to the AUSF.
3. The AUSF does not send a `Nausf_UEAuthentication_Authenticate Response` message to the AMF before timeout.

The test is marked as **PASSED** if the AMF does not send any new authentication request to the UE.

4.2.2.1.2 (A, B, C, D) RES verification failure handling*

The tests verify:

- that the AMF correctly handles RES* verification failure detected in the AMF or/and in the AUSF, when the SUCI is included in the initial NAS message.
- that the AMF correctly handles RES* verification failure detected in the AMF or/and in the AUSF, when the 5G-GUTI is included in the initial NAS message.

4.2.2.1.2 (A) The UE sends RR with SUCI to the DUT (AMF) and incorrect RES*

Execution Steps:

1. The UE sends RR with SUCI to the AMF under test, to trigger the AMF under test to initiate the authentication.
2. The AUSF, after receiving the request from the AMF under test, responds with a `Nausf_UEAuthentication_Authenticate Response` message with an authentication vector to the AMF under test.
3. The UE, after receiving the Authentication Request message from the AMF under test, returns an incorrect RES* to the AMF under test in the NAS Authentication Response message, which will trigger the AMF to compute HRES*, compare HRES* with HXRES* and send an authentication request to the AUSF. The tester captures the value of RES* in the request.
4. The AUSF returns to the AMF under test the indication of RES* verification failure.

The test is marked as **PASSED** if the AMF rejects the authentication by sending an Authentication Reject to the UE. Otherwise, is marked as **FAILED**.

4.2.2.1.2 (B) The UE sends RR with 5G-GUTI to the DUT (AMF) and incorrect RES*

Execution Steps:

1. The UE sends RR with 5G-GUTI to the AMF under test, to trigger the AMF under test to initiate the authentication.
2. The AUSF, after receiving the request from the AMF under test, responds with a `Nausf_UEAuthentication_Authenticate Response` message with an authentication vector to the AMF under test.
3. The UE, after receiving the Authentication Request message from the AMF under test, returns an incorrect RES* to the AMF under test in the NAS Authentication Response message, which will trigger the AMF to compute HRES*, compare HRES* with HXRES* and send an authentication request to the AUSF. The tester captures the value of RES* in the request.
4. The AUSF returns to the AMF under test the indication of RES* verification failure.

The test is marked as **PASSED** if the AMF initiates an Identification procedure with the UE to retrieve the SUCI.

4.2.2.1.2 (C) The UE sends RR with SUCI to the DUT (AMF) and RES*

Execution Steps:

1. The UE sends RR with SUCI to the AMF under test, to trigger the AMF under test to initiate the authentication.
2. The AUSF, after receiving the request from the AMF under test, responds with a `Nausf_UEAuthentication_Authenticate Response` message with an authentication vector to the AMF under test.
3. The UE, after receiving the Authentication Request message from the AMF under test, returns an incorrect RES* to the AMF under test in the NAS Authentication Response message, which will trigger the AMF to compute HRES*, compare HRES* with HXRES* and send an

authentication request to the AUSF. The tester captures the value of RES* in the request.

4. The AUSF returns to the AMF under test the indication of RES* verification failure.

The test is marked as **PASSED** if the AMF rejects the authentication by sending an Authentication Reject to the UE.

4.2.2.1.2 (D) The UE sends RR with 5G-GUTI to the DUT (AMF) and RES*

Execution Steps:

1. The UE sends RR with 5G-GUTI to the AMF under test, to trigger the AMF under test to initiate the authentication.
2. The AUSF, after receiving the request from the AMF under test, responds with a `Nausf_UEAuthentication_Authenticate Response` message with an authentication vector to the AMF under test.
3. The UE, after receiving the Authentication Request message from the AMF under test, returns an incorrect RES* to the AMF under test in the NAS Authentication Response message, which will trigger the AMF to compute HRES*, compare HRES* with HXRES* and send an authentication request to the AUSF. The tester captures the value of RES* in the request.
4. The AUSF returns to the AMF under test the indication of RES* verification failure.

The test is marked as **PASSED** if the AMF initiates an Identification procedure with the UE to retrieve the SUCI.

4.2.2.3.1 Replay protection of NAS signalling messages

The test verifies that NAS signaling messages are replay protected by AMF over N1 interface between UE and AMF.

Execution Steps:

1. The tester will capture the NAS SMC procedure taking place between UE and AMF over N1 interface using any network analyzer.
2. The tester will filter the NAS Security Mode Complete message by using a filter.
3. The tester will check for the NAS SQN of filtered NAS Security Mode Complete message and using any packet crafting tool the tester shall create a NAS Security Mode Complete message containing same NAS SQN of the filtered NAS Security Mode Complete message or the tester will replay the captured NAS signaling packets.
4. Tester will check whether the replayed NAS signaling packets were processed by the AMF by capturing over N1 interface to see if any corresponding response message is received from the AMF.
5. Tester will confirm that AMF provides replay protection by dropping/ignoring the replayed packet if no corresponding response is sent by the AMF to the replayed packet.
6. Tester will verify from the result that if the crafted NAS Security Mode Complete message or replayed NAS signaling messages are not processed by the AMF when the N1 interface is replay protected.

The test is marked as **PASSED** if the NAS signaling messages sent between UE and AMF over N1 interface are replay protected.

4.2.2.3.2 NAS NULL integrity protection

The test verify that NAS NULL integrity protection algorithm is used correctly.

Execution Steps:

1. The UE initiates an emergency registration.
2. The AMF derives the KAMF and NAS signaling keys after successful authentication of the UE.
3. The AMF sends the NAS Security Mode Command message to the UE containing the selected NAS algorithms.

The test is marked as **PASSED** if the UE was successfully authenticated and the integrity algorithm selected by the AMF is NAS SMC message is different from NIA0.

4.2.2.3.3 NAS integrity algorithm selection and use

The test verifies that the AMF selects the NAS integrity algorithm which has the highest priority according to the ordered list of supported integrity algorithms and is contained in the 5G security capabilities supported by the UE, and selected NAS security algorithm is being used.

Execution Steps:

1. The UE sends a Registration Request with Initial Registration type to the AMF under test.
2. The tester filters the Security Mode Command and Security Mode Complete messages.
3. The tester examines the selected integrity algorithm in the SMC against the list of ordered NAS integrity algorithm and the 5G security capabilities supported by the UE. The tester examines the MAC verification of the Security Mode Complete at the AMF under test.

The test is marked as **PASSED** if the selected integrity algorithm has the highest priority according to the list of ordered NAS integrity algorithm and is contained in the UE 5G security capabilities and the MAC verification of the Security Mode Complete message is successful.

4.2.2.4.1 Bidding down prevention in Xn-handover

The test verifies that bidding down is prevented by the AMF under test in Xn handovers.

Execution Steps:

1. The tester sends 5G security capabilities for the UE, different from the ones stored in the AMF, to the AMF under test using a Path-Switch message.

The test is marked as **PASSED** if there is a capability mismatch in the Path-Switch Acknowledge message sent by AMF under test to the target gNB (which includes the locally stored 5G security capabilities in the AMF under test for that UE).

4.2.2.4.2 NAS protection algorithm selection in AMF change

The test verifies that NAS protection algorithms are selected correctly during N2 Handover.

Execution Steps:

1. The AMF under test receives the UE security capabilities and the NAS algorithms used by the source AMF from the source AMF.
2. The AMF under test selects the NAS algorithms which have the highest priority according to the ordered lists. The lists are configured such that the algorithms selected by the AMF under test are different from the ones received from the source AMF.

The test is marked as **PASSED** if the NASC of the NGAP HANDOVER REQUEST message sent by the AMF under test to the gNB includes the chosen algorithm and the AMF under test initiates a NAS security mode command procedure, including the chosen algorithms.

4.2.2.5.1 (A, B, C) 5G-GUTI allocation

The tests verify that a new 5G-GUTI is allocated by the AMF under test in these scenarios accordingly.

4.2.2.5.1 (A) 5G-GUTI allocation

Execution Steps:

1. The UE sends a Registration Request message of type *initial registration* to the AMF.
2. The AMF sends a new 5G-GUTI to the UE during the registration procedure.

The test is marked as **PASSED** if in the NAS signalling packets a new 5G-GUTI is received by UE during registration procedure. The DUT (AMF) should use NAS security context to encapsulate the messages.

The new 5G-GUTI is different from the old 5G-GUTI.

4.2.2.5.1 (B) 5G-GUTI allocation

Execution Steps:

1. The UE sends a Registration Request message of type *mobility registration update* to the AMF.
2. The AMF sends a new 5G-GUTI to the UE during the registration procedure.

The test is marked as **PASSED** if in the NAS signalling packets a new 5G-GUTI is received by UE during registration procedure. The DUT (AMF) should use NAS security context to encapsulate the messages.

The new 5G-GUTI is different from the old 5G-GUTI.

4.2.2.5.1 (C) 5G-GUTI allocation

Execution Steps:

1. The UE sends a Service Request message in response to a Paging message to the AMF.
2. The AMF sends a new 5G-GUTI to the UE during the registration procedure.

The test is marked as **PASSED** if in the NAS signalling packets a new 5G-GUTI is received by UE during registration procedure. The DUT (AMF) should use NAS security context to encapsulate the messages.

The new 5G-GUTI is different from the old 5G-GUTI.

4.2.2.6.1 Invalid or unacceptable UE security capabilities handling

The test verifies that UE security capabilities invalid or unacceptable are not accepted by the AMF under test in registration procedure.

Execution Steps:

1. The UE sends UE security capabilities to the AMF under test using registration request message.

The test is marked as **PASSED** if the UE receives Registration reject message sent by the AMF.

UPF Tests Description

The following tests require the UPF to be set as DUT.

4.2.2.1, 4.2.2.2, 4.2.2.3 Confidentiality, Integrity and Replay protections over N3 interface

The test verifies that the transported user data between gNB and UPF is confidentiality, integrity and replay protected.

Execution Steps:

1. The tester intercepts the traffic between the UPF under test and the gNB.
2. The tester establishes a N3 secure communication (IPSec as agreed in specifications).

The test is marked as **PASSED** if the UPF creates a secure IPSec communication and send the user data over it, protecting them.

4.2.2.6 TEID uniqueness

The test verifies that the TEID generated by UPF under test for each new GTP tunnel is unique.

Execution Steps:

1. The tester intercepts the traffic between the UPF under test and the SMF.
2. The tester triggers the maximum number of concurrent N4 session establishment requests.
3. The tester captures the N4 session establishment responses sent from UPF to SMF and verifies that the F-TEID created for each generated response is unique.

The test is marked as **PASSED** if the UPF generates F-TEID in each different N4 session establishment response is unique.

UDM Tests Description

The following tests require the UDM to be set as DUT.

4.2.1.1 De-concealment of SUPI from the SUCI based on the protection scheme used to generate the SUCI

The test verifies that the SIDF De-conceals the SUPI from the SUCI based on the protection scheme used to generate the SUCI.

Execution Steps:

1. Analyse the entire authentication procedure between UE and AMF over N1, N12 and N13 interface looking for `Nudm_Authentication_Get Response` message sent from UDM to AUSF over N13 interface containing the SUPI.
2. Compare the SUPI gotten from UE and the SUPI retrieved from `Nudm_Authentication_Get Response` message.

The test is marked as **PASSED** if the UDM resolves the SUPI from the SUCI based on the protection scheme used to generate the SUCI.

4.2.2.1 Synchronization failure handling

The test verifies that synchronization failure is recovered correctly in the home network.

Execution Steps:

1. The AUSF sends an `Nudm_UEAuthentication_Get Request` message to the UDM with a *synchronization failure indication* and parameters RAND and AUTS.

The test is marked as **PASSED** if the UDM sends an `Nudm_UEAuthentication_Get Response` message with a new authentication vector to the AUSF.

4.2.2.2 Storing of authentication status of UE by UDM

The test verifies that the UDM under test stores the authentication status of UE, which is identical to the UE authentication information sent to/from the AUSF and the AMF.

Execution Steps:

1. Filter the `Nudm_UEAuthentication_Get Request` message sent over the N13 interface.
2. Filter the `Nausf_UEAuthentication_Authenticate Response` message sent over N12 interface to retrieve the Authentication result (EAP success/failure for EAP-AKA' or Result for 5G AKA).
3. Filter the `Nudm_UEAuthentication_ResultConfirmation Request` message to retrieve the authentication result and time of authentication procedure sent from the AUSF to the UDM over N13 interface.
4. Compare the serving network name stored in the UDM against the serving network name retrieved from the `Nudm_Authentication_Get Request` message and the serving network name retrieved from the `Nudm_UEAuthentication_ResultConfirmation Request` message.
5. Compare the authentication status stored in the UDM against the authentication result retrieved from N12 interface.
6. Compare the SUPI stored in the UDM against the SUPI retrieved from the `Nudm_Authentication_Get Response` message and the SUPI retrieved from the `Nudm_UEAuthentication_ResultConfirmation Request` message.
7. Compare the timestamp stored in the UDM against the time of authentication procedure retrieved from the `Nudm_UEAuthentication_ResultConfirmation Request` message.

The test is marked as **PASSED** if the UDM stores all parameters related to the authentication status (SUPI, authentication result, timestamp, and the serving network name) of UE.

SMF Tests Description

The following tests require the SMF to be set as DUT.

4.2.2.1.1 Priority of UserPlane Security Policy

The test verifies that the user plane security policy from the UDM takes precedence at the SMF under test over locally configured user plane security policy.

Execution Steps:

1. The system initiates a PDU session establishment procedure by sending `Nsmf_PDUSession_CreateSMContext Request` message to the SMF.
2. The SMF under test retrieves the Session Management Subscription data using `Nudm_SDM_Get` service from UDM, where the Session Management Subscription data includes the user plane security policy stored in UDM.

The test is marked as **PASSED** if the Security Indication IE in the N2 SM information contained in the `Namf_Communication_N1N2MessageTransfer` message is the same as the UP security policy configured in the UDM.

4.2.2.1.3 Security functional requirements on the SMF checking UserPlane security policy

The test verifies that the SMF checks the UserPlane security policy that is sent by the ng-eNB/gNB during handover.

Execution Steps:

1. The system sends the `Nsmf_PDUSession_UpdateSMContext Request` message to the SMF under test. A UE UP security policy different than the one preconfigured at the SMF under test is included in the Request message.
2. `Nsmf_PDUSession_UpdateSMContext Response` message sent from the SMF under test is captured.

The test is marked as **PASSED** if the preconfigured UE security policy is contained in the *n2SmInf* IE in the captured Response message.

4.2.2.1.4 Charging ID Uniqueness

The test verifies that the charging ID generated by the SMF for each PDU session is unique.

Execution Steps:

1. The system triggers the establishment of the maximum number of concurrent PDU sessions that the SMF under test can handle.
2. The Charging Data Request [initial] sent from the SMF under test to the CHF are captured.

The test is marked as **PASSED** if the charging ID contained in the *PDU Session Charging Information* IE in each Charging Data Request [initial] is unique.

NRF Tests Description

The following tests require the NRF to be set as DUT.

4.2.2.2.1 NF discovery authorization for specific slice

The test verifies that the NRF under test does not authorize slice specific discovery request for the NF instance, which is not part of the requested slice, according to the slice specific discovery configuration of the requested NF instance.

Execution Steps:

1. The NF2 (AUSF) registers at the NRF under test with a list of S-NSSAI.
2. The NF1 (AMF) sends an `Nnrf_NFDiscovery_Request` to the NRF under test with the expected service name of NF2, NF type of the expected NF2.
3. The NRF under test determines that NF2 instance only allows discovery from NFs belonging to slice A, according to the *allowedNssais* list stored in NF2 Profile.

The test is marked as **PASSED** if the NRF returns a response with *403 Forbidden* status code, otherwise it **FAILED**.

This page intentionally left blank.

CHAPTER 7

Troubleshooting

In the event you encounter a problem when using OpenTap to run LoadCore SCAS test suite, use the information in this section to try and identify the problem in order to correct it.

Retrieving OpenTap Logs

Log messages provide useful insight to the process of debugging. Log messages are displayed in the Editor Log panel and saved in the log file.

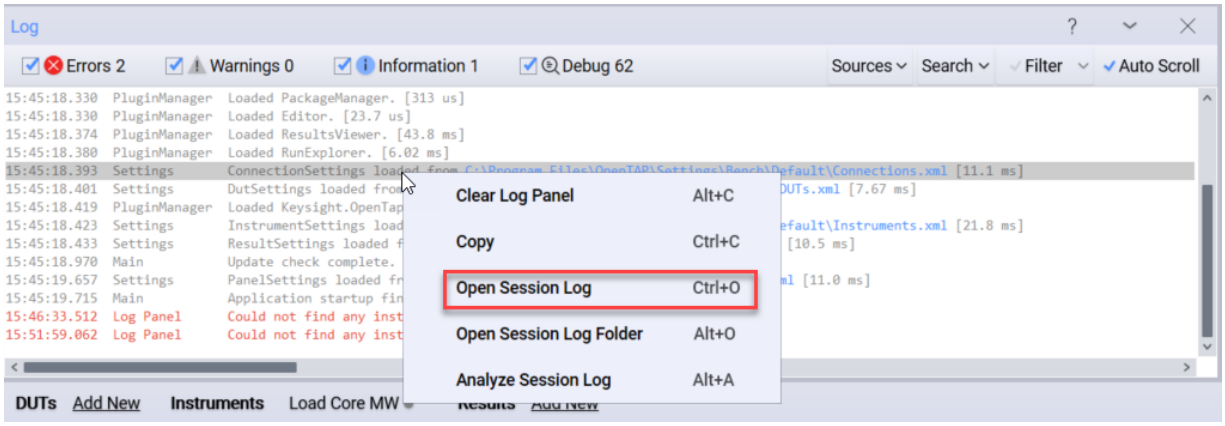
Four levels of log messages — **Error**, **Warning**, **Information**, and **Debug** — allow messages to be grouped in order of importance and relevance.

IMPORTANT The debug messages are enabled by selecting the **Debug** check box from the Log panel (by default this option is not enabled).

Log messages are shown in the Log panel and are stored in the session's log file, named `SessionLogs\SessionLog [DateTime].txt`.

The log file can be also accessed and inspected from the Log panel:

- right-click on the Log panel and select **Open Session Log** to open the `.txt` file.



- right-click on the Log panel and select **Open Session Log Folder** and select a specific `.txt` file.

Retrieving LoadCore Logs

Another useful source of info for troubleshooting is represented by the `.csv` results file and the logs file from Loadcore.

These can be retrieved from the Test Results window (Browse Results section).

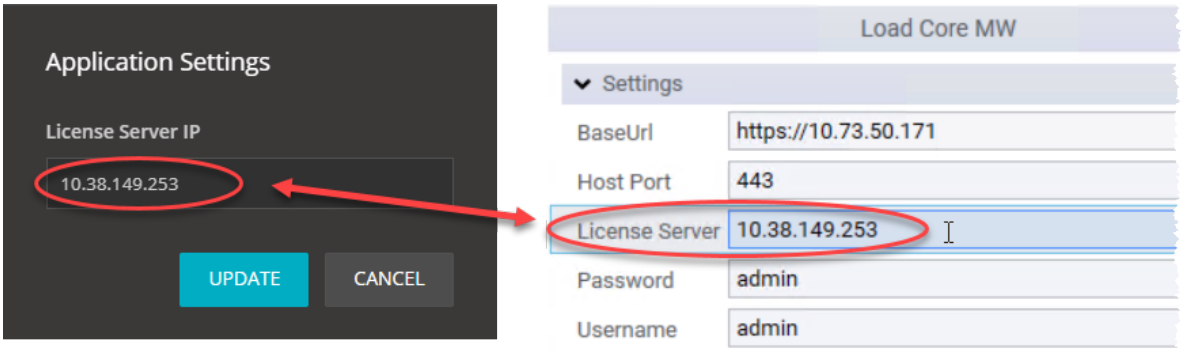
TIP You can follow both OpenTap and LoadCore in parallel when running the SCAS test suite. Basically, this will create, run and then, delete the tests from LoadCore. This way, it could be easier to identify potential errors.

Common issues

One common problem you could encounter is related to licensing error messages. This can be avoided by following these steps:

- Make sure that the License Server IP address used in OpenTab is the same as the one used in LoadCore.
  - For LoadCore, the License Server IP address can be retrieved/updated from gear menu > **Application Settings**.
  - For OpenTap, the License Server IP address must be specified when configuring the Loadcore MW instrument (details here).

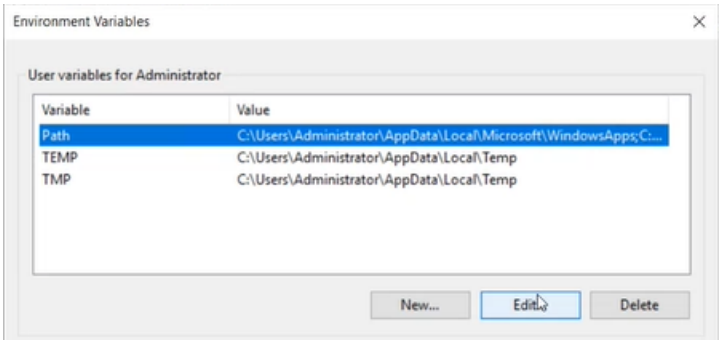
  For Example



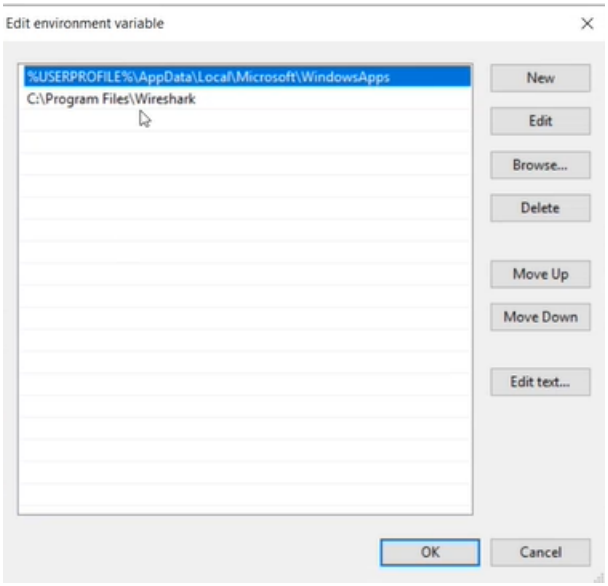
- Make sure that you meet the license requirements specified here.

Another common source of error messages is related to Wireshark. To avoid this, do the following:

1. Make sure that the Wireshark path is added to the environment variables:
   a. Open Command Prompt (as Administrator) and run the `tshark` command. If the following message is displayed, you need to edit the environment variables and add the path to the Wireshark installation directory:
   ```
   'tshark' is not recognized as an internal or external command, operable program or batch file.
   ```
   b. Open the Start Search, type in **env** and select **Edit the system environment variables**.
   c. Select **Environment Variables**.
   d. Select **Path** and then, select **Edit**.



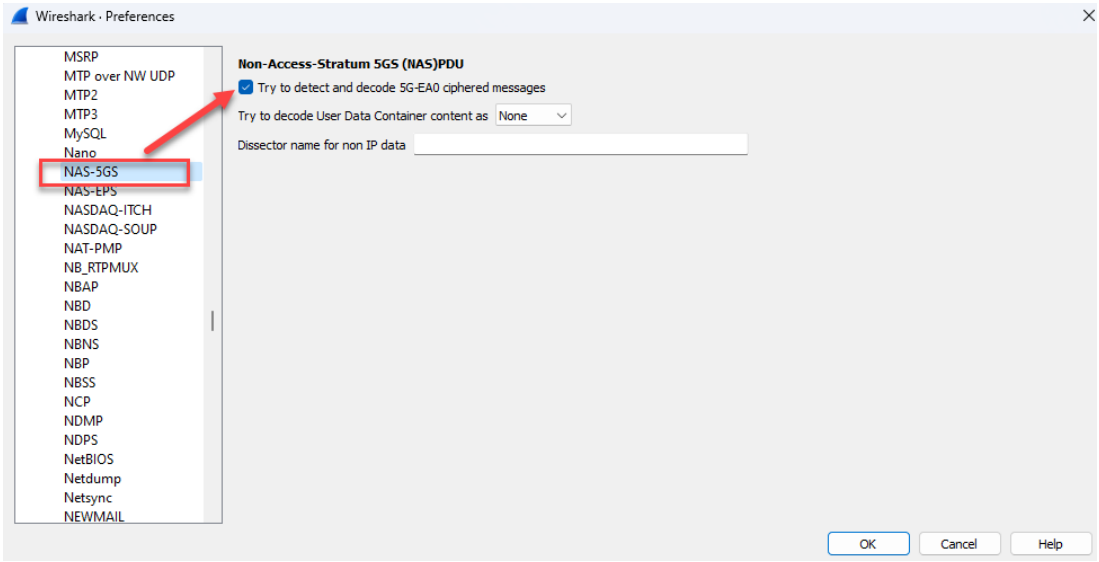
   e. Select the Wireshark path and then, select **OK**.



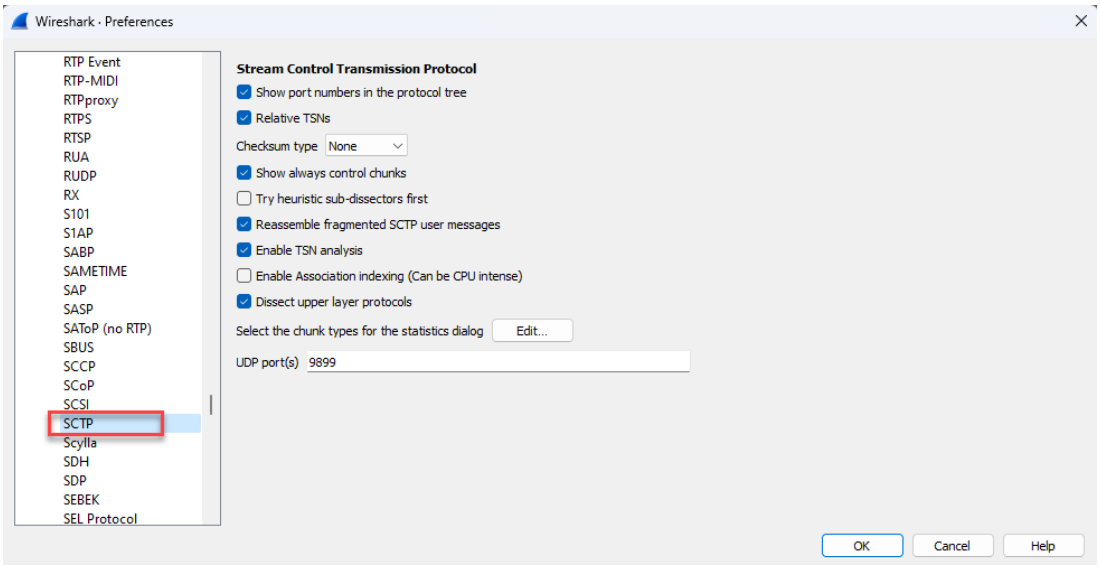
f. Close the Command Prompt and open it again. Run the `tshark` command to make sure that the message above is no longer displayed:

```
C:\Users\Administrator>tshark
Capturing on 'Local Area Connection* 9'
0 packets captured
```

2. Make sure that the NAS-5GS protocol is enabled in WireShark:
   a. From Wireshare, select **Edit** > **Preferences**.
   b. Expand the Protocols drop-down line and make sure that the following WireShark settings are configured in order to show frames decoded properly:
      - ***NAS-5GS*** - make sure that the **Try to detect and decode 5G-EA0 ciphered messages** check box is selected.



- ***SCTP*** settings:



c. Select **OK**.

Index

www.keysight.com