Data Communications Company

GIT For Industry

Enhanced Training

September 2016 – London

DCC Public

Who are we?

- Critical Software provides GFI to DCC.

- Presenter:
  Karim Kanso, Engineer @ Critical Software (UK).

- Involved in the development of the GFI framework and with the Test Events organised by DCC.

Agenda

Session 1 (1:30h)

- Introduction
- New Architecture
- GUI Usage

Coffee Break (15 minutes)

Session 2 (1:45h)

- Command Line Usage
- RPC Interface

INTRODUCTION

Introduction: what is it?

- Software test harness provided by Smart DCC for SMETS2
- Helps check interpretation of the Great Britain Companion Specification
- Software tool for risk reduction
  - Allowed tests to be performed ahead of SIT
- Release 1.3 supports all Use Cases for GBCS v0.8.2 (over HAN)
- Enhanced version also supports GPF Use Cases and TOM processing

Introduction: what it means?

GFI = GIT For Industry
GIT = GBCS Integration Testing
GBCS = Great Britain Companion Specification

So unfolding,
GFI = "GBCS Integration Testing For Industry"

"A tool for Industry to test their interpretation of GBCS, namely the smart meter manufacturers."

Introduction: this training?

- Assumption: You have familiarity with previous version of GFI.
- Workshop is a 'how to' for the enhanced GFI.
- Primary focus will be on how to setup the HAN and create new tests using the GUI.
- Advanced topics include command line usage and programmatic control of daemon.

GFI OVERVIEW

Overview: testbed (I)

GFI creates a ZigBee HAN and allows Devices to join.

GFI emulates: Remote Parties, ACB, GPF and CHF.

Communicates directly with Devices in order to execute Tests.

Supported devices: ESME, GSME, PPMID, IHD.

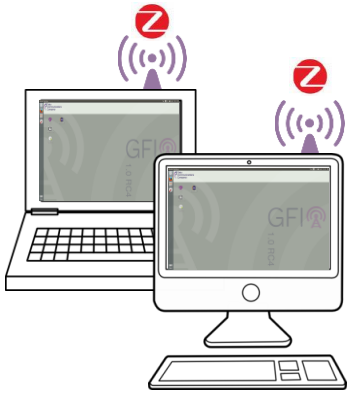
Overview: testbed (II)

GFI consists of three parts:

1. Graphical User Interface – Front End (Session 1)
   - Good for basic to moderate usage.

2. Command Line Interface – Back End (Session 2)
   - Good for advanced usage and custom scenarios.

3. Communications Hub daemon (Both Sessions)
   - Required when not using emulated scenarios.



TOOL CHANGES

Tool Changes: old GFI (I)

- The origins of the GFI testing tool are from a project where the goal was to perform electronic protocol testing of GBCS.
  - Non-functional testing of the protocol.
  - Considered messages between remote parties and metering devices.

- In that project, the focus was on generating commands, responses, and any associated alerts.
  - Completely virtualised environment.
  - Focused on GBCS messages.
  - No consideration for ZigBee layer communications.
  - Not possible to emulate full GBCS as some parts dependent on ZigBee.

Tool Changes: old GFI (II)

- When Uplifted to GFI, so that communication with physical meters over ZigBee was possible, there were a number of limitations.
  - GPF and TOM that depend on ZigBee layer not included as not part of original project.
  - Inherited Architecture:
    1. test setup (Create HAN)
    2. run tests (Execute Tests)
    3. test teardown (Destroy HAN)
  - As the HAN was transient, support for other device types was not included, e.g. IHD.

- The goal of the Enhanced version of GFI was to overcome these limitations, and develop a tool that is useful for industry.

Tool Changes: new GFI (I)

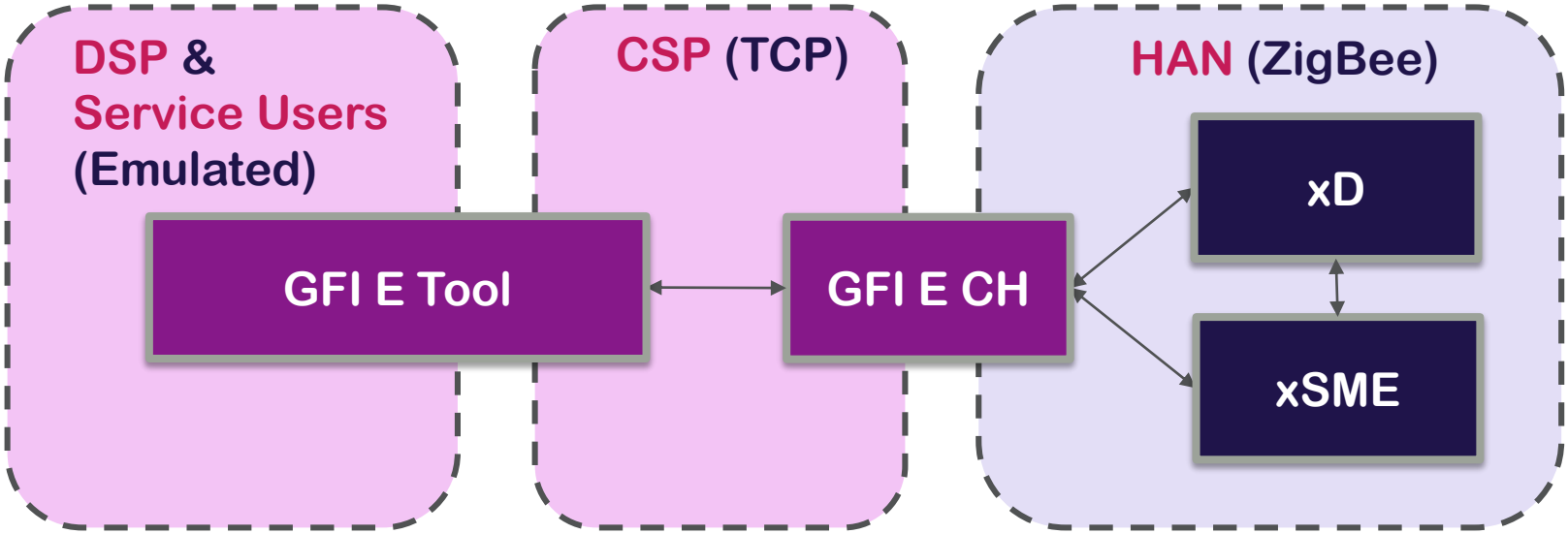
- Foundational change in GFI: de-couple testing tool from the CHF.
  - The HAN is independent of the testing tool.
  - It must be configured before the testing tool is used.

- Support for the GPF and TOM processing was added.
  - Added support for GCS09, GCS59 and GCS61.

- Support for additional device types was added: IHD, PPMID.
  - Added a PPMID emulator (supports PCS01 and PCS02)

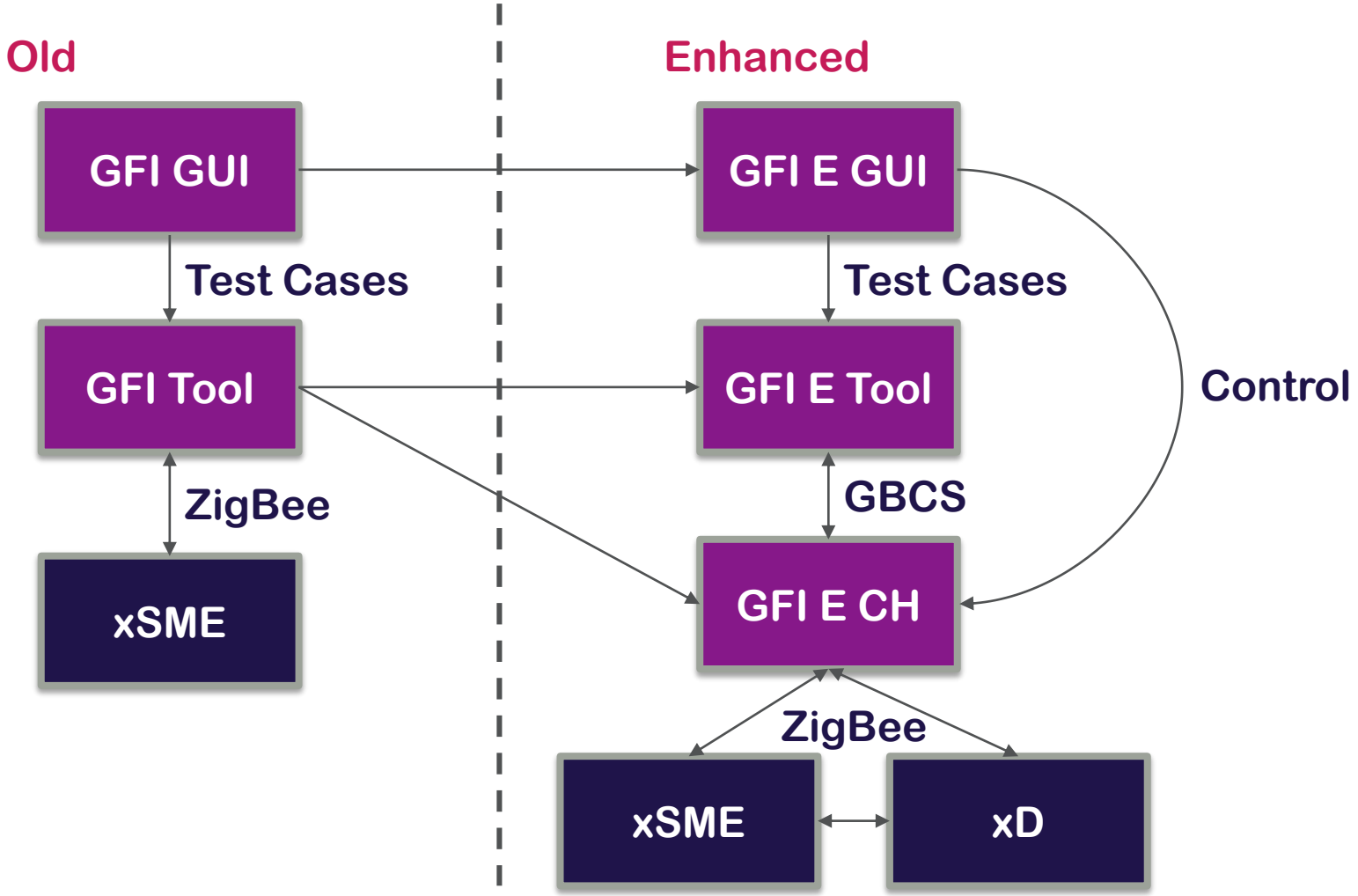
- Support for dummy OTA Update server added.

Tool Changes: new GFI (II)



Architecture of GFI in relation to Smart Metering System.

Tool Changes: new GFI (III)



GRAPHICAL USER INTERFACE

GUI: desktop

Upon a fresh installation this is what you get…

Graphical User Interface

IDE

Command Line

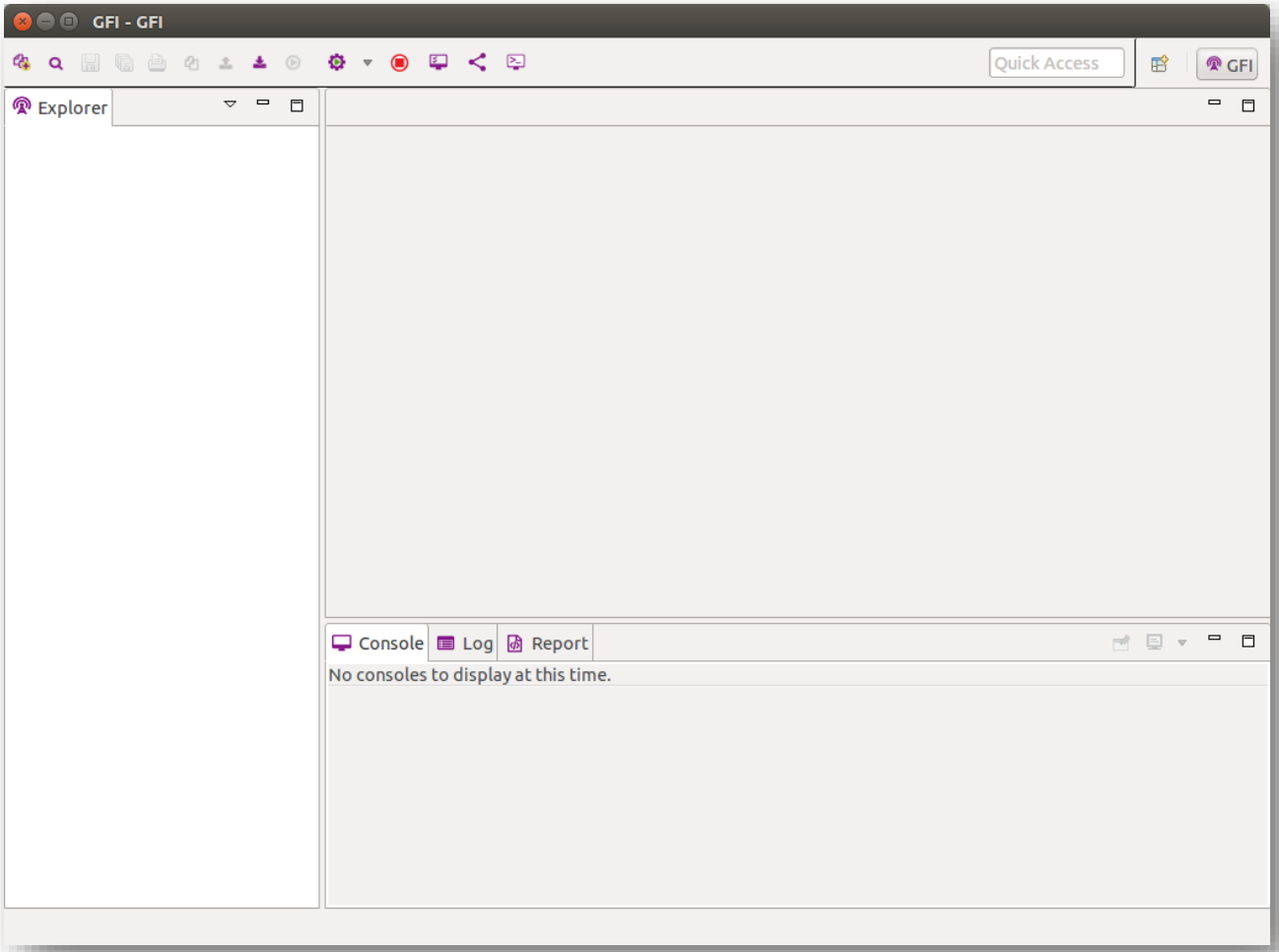
Updater

GUI: desktop

The GUI is launched using this shortcut on the desktop

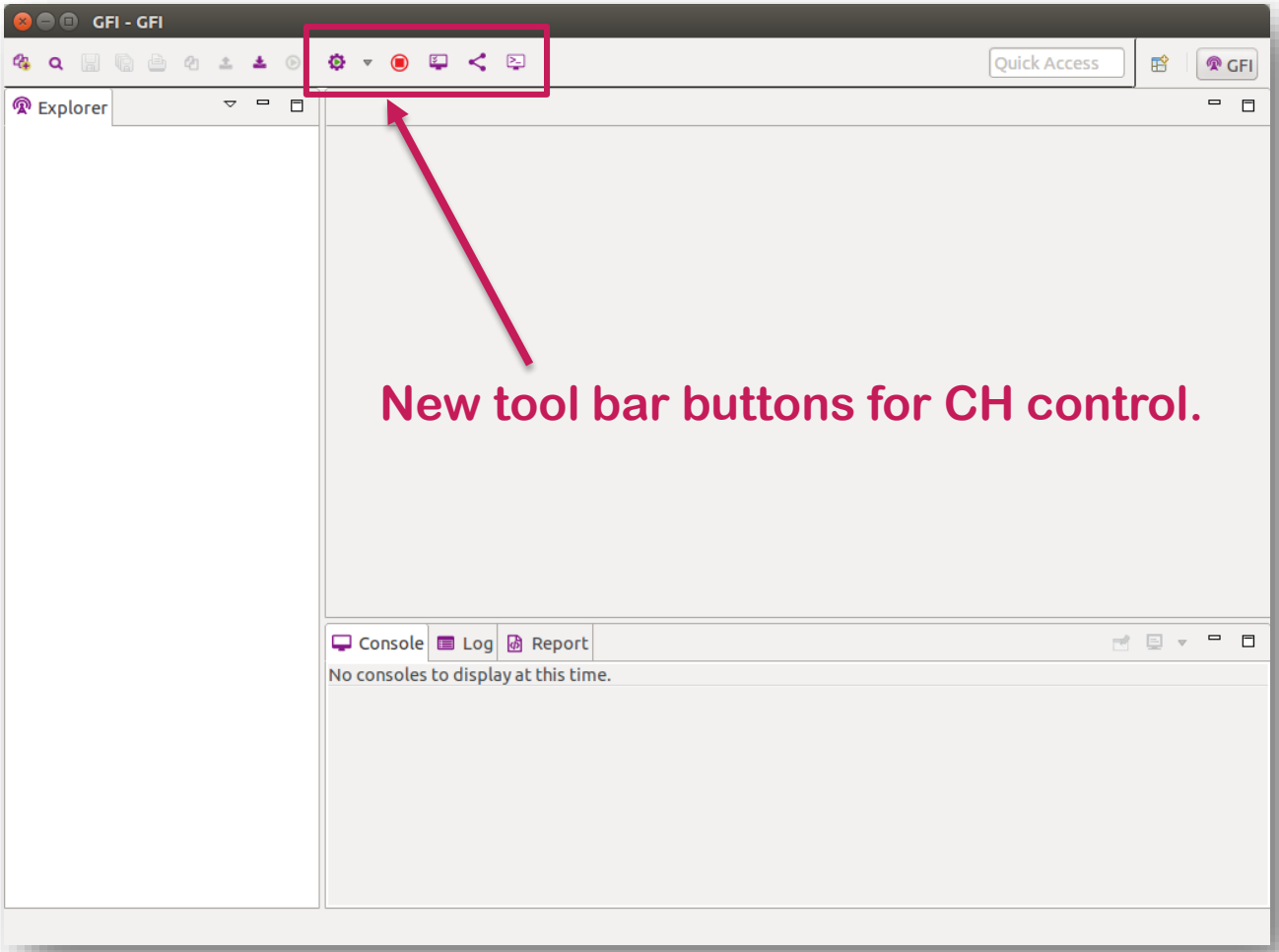
Is a standalone application built to run in Ubuntu 14.04

GUI: test environment overview (I)



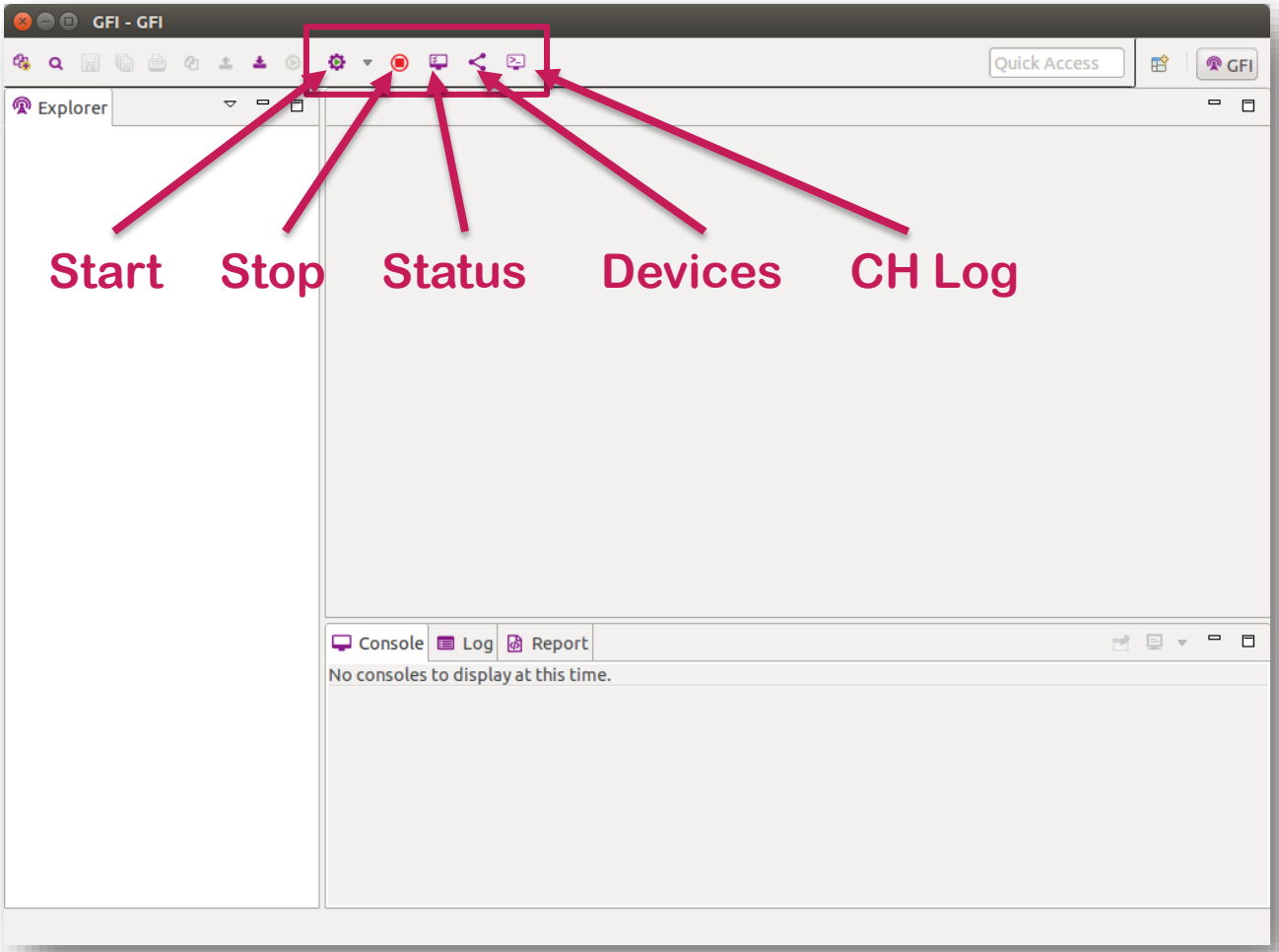
Default test environment setup: Windows and panes can be moved freely.

GUI: test environment overview (II)



Default test environment setup: Windows and panes can be moved freely.

GUI: test environment overview (III)

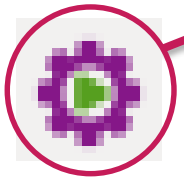


Default test environment setup: Windows and panes can be moved freely.

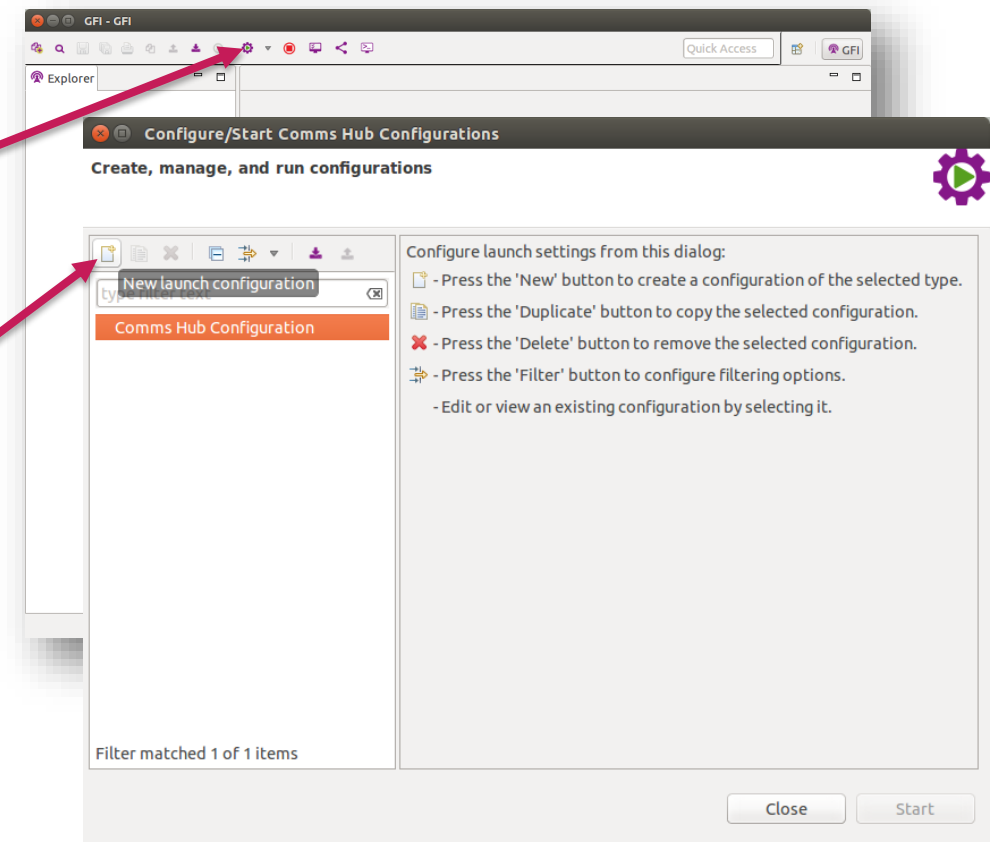
GUI: CH configuration (I)

Before running a test, its required to configure and start the CH.

1) Select the button:



2) Select **New configuration**:



It is possible to store **multiple CH configurations** in the GUI.

Only one of these configuration can be executed at a time.

GUI: CH configuration (II)

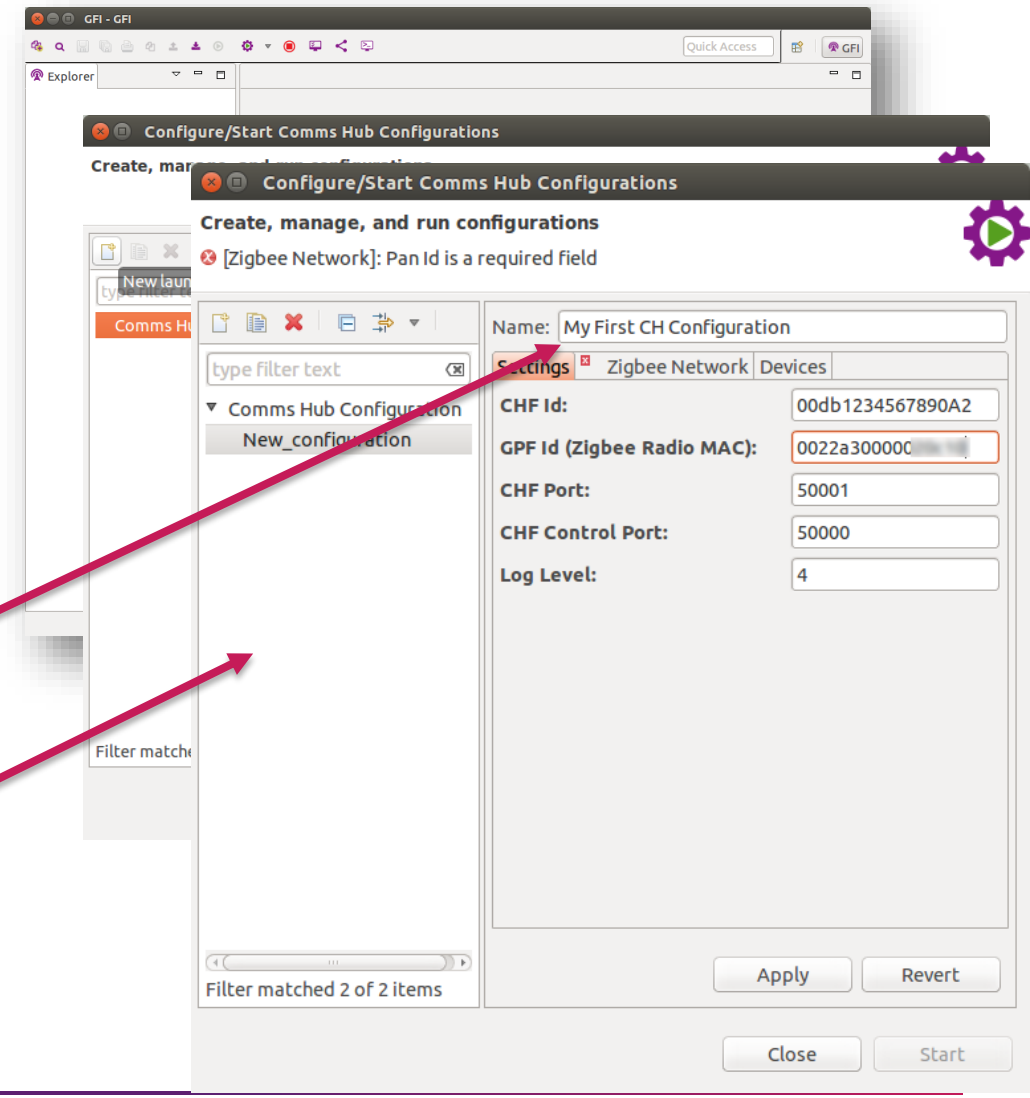
Once a new CH has been created, it needs to be configured.

Configuration is split over 3 tabs:

- General Settings
- ZigBee Network
- Devices (HAN Whitelist)

First, enter a name for the new configuration.

Once saved, this will appear in the list on the left.



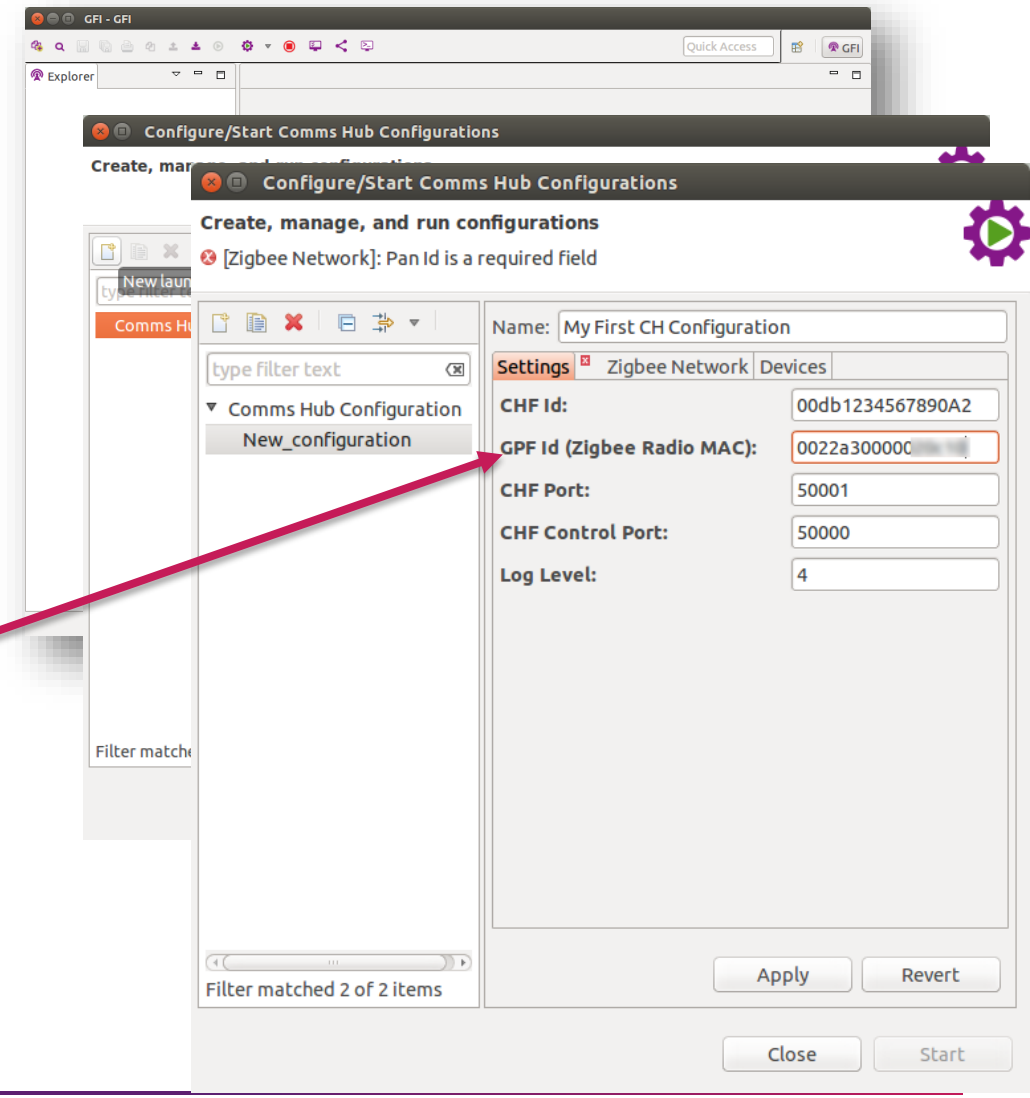
GUI: CH configuration (III)

Settings Tab

Has general settings for the CH.

CHF Id – EUI64 of the CHF
GPF Id – EUI64 of the GPF
CHF Port – TCP Port for GFI
CHF Cont. Port – TCP for GUI
Log Level – Verbosity of Log

The GPF Id must be the same as the GFI Kit EUI64, i.e. same as the ZigBee radio address.



GUI: CH configuration (IV)

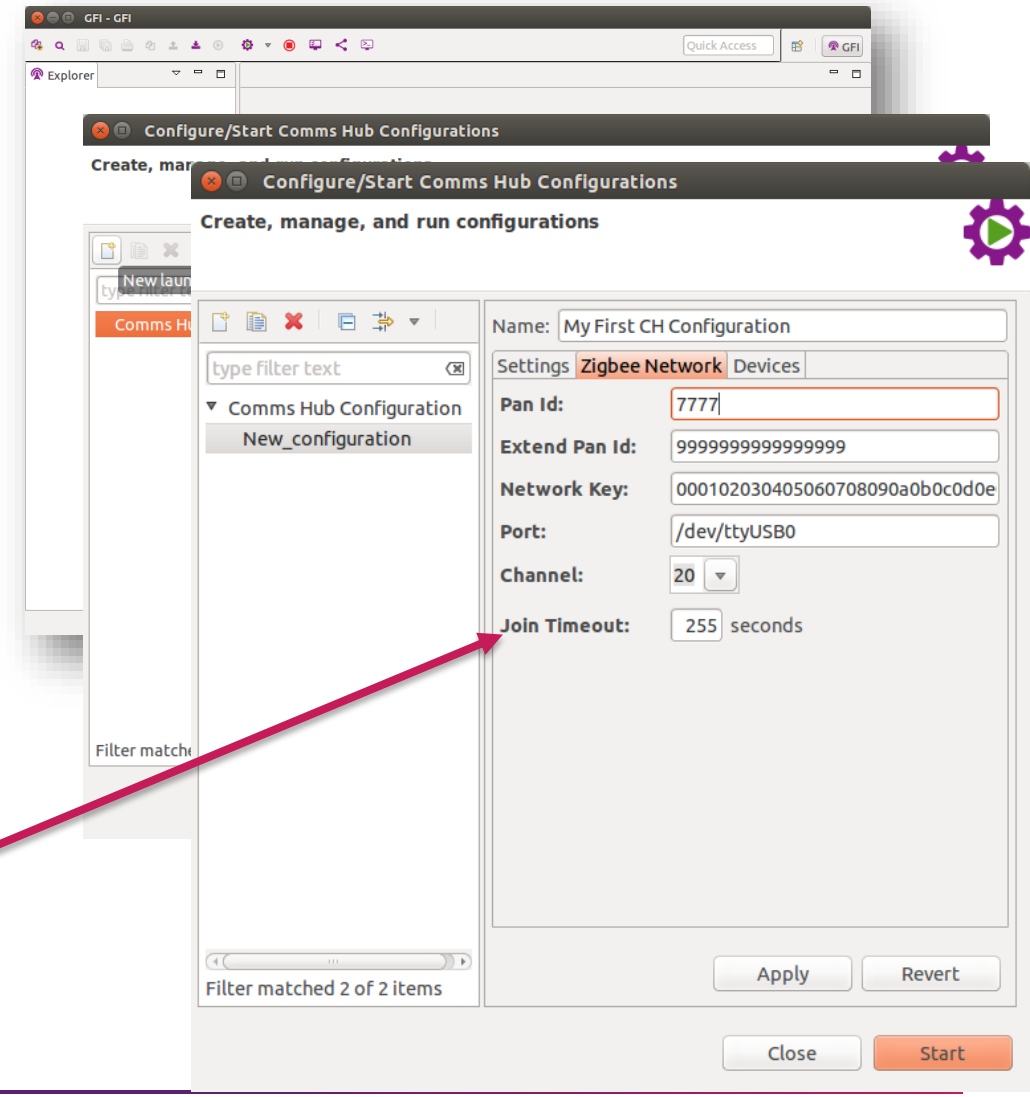
ZigBee Network Tab

Has the ZigBee HAN settings.

Pan Id – ZB Pan
Extended Pan Id – ZB Ex. Pan
Network Key – ZB NWK Key
Port – Dongle USB Port
Channel – ZB Channel
Join Timeout – Window of permit join.

Normally its not required to change the UBB Port.

Setting Join Timeout to a value of 255 means its unlimited.



GUI: CH configuration (V)

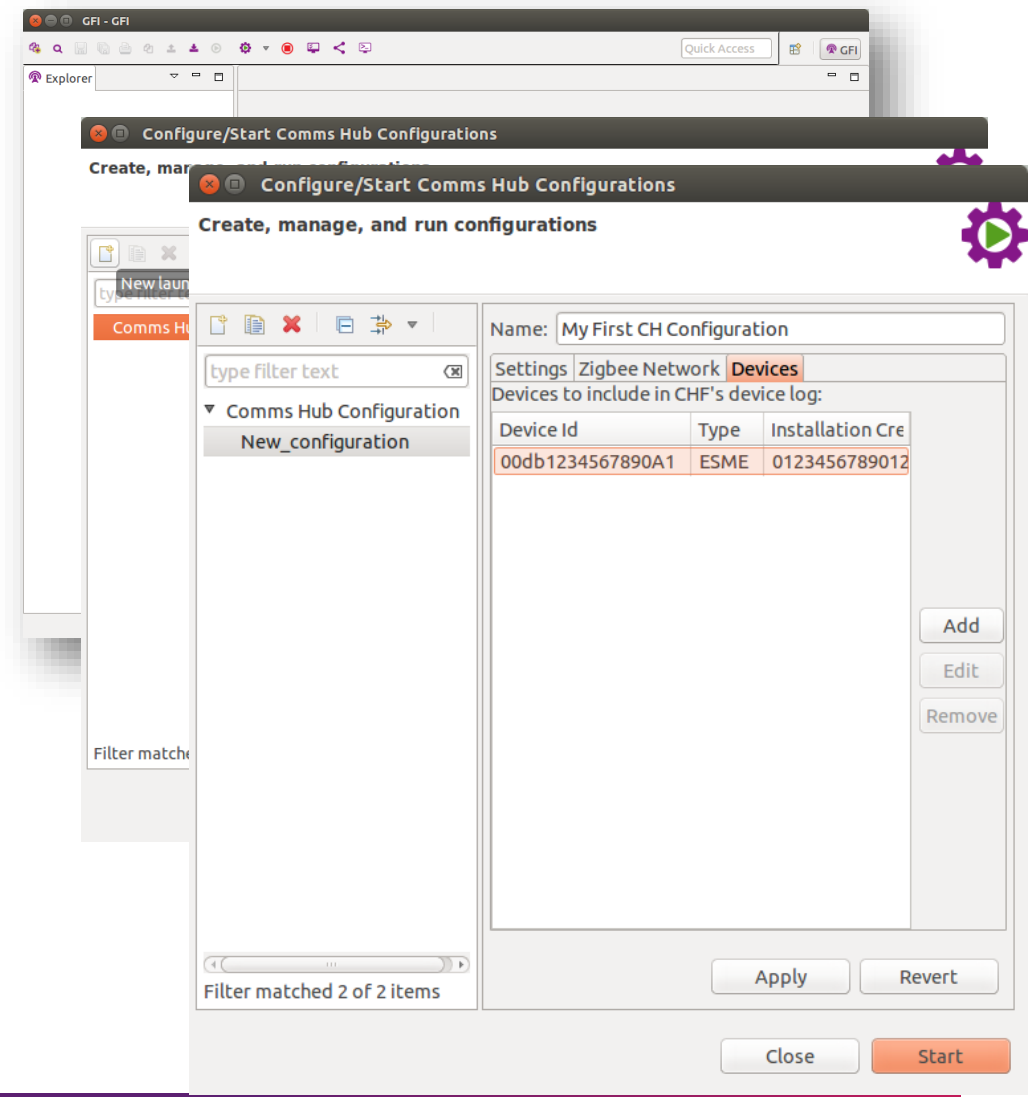
Devices Tab

Has the ZigBee HAN whitelist.

The whitelist is made out of a list of entries, consisting of:

- Device EUI64
- Device Type
- Installation Credentials

It is not essential to configure the whitelist here as it can be configured using CCS0x use cases.



GUI: CH configuration (V)

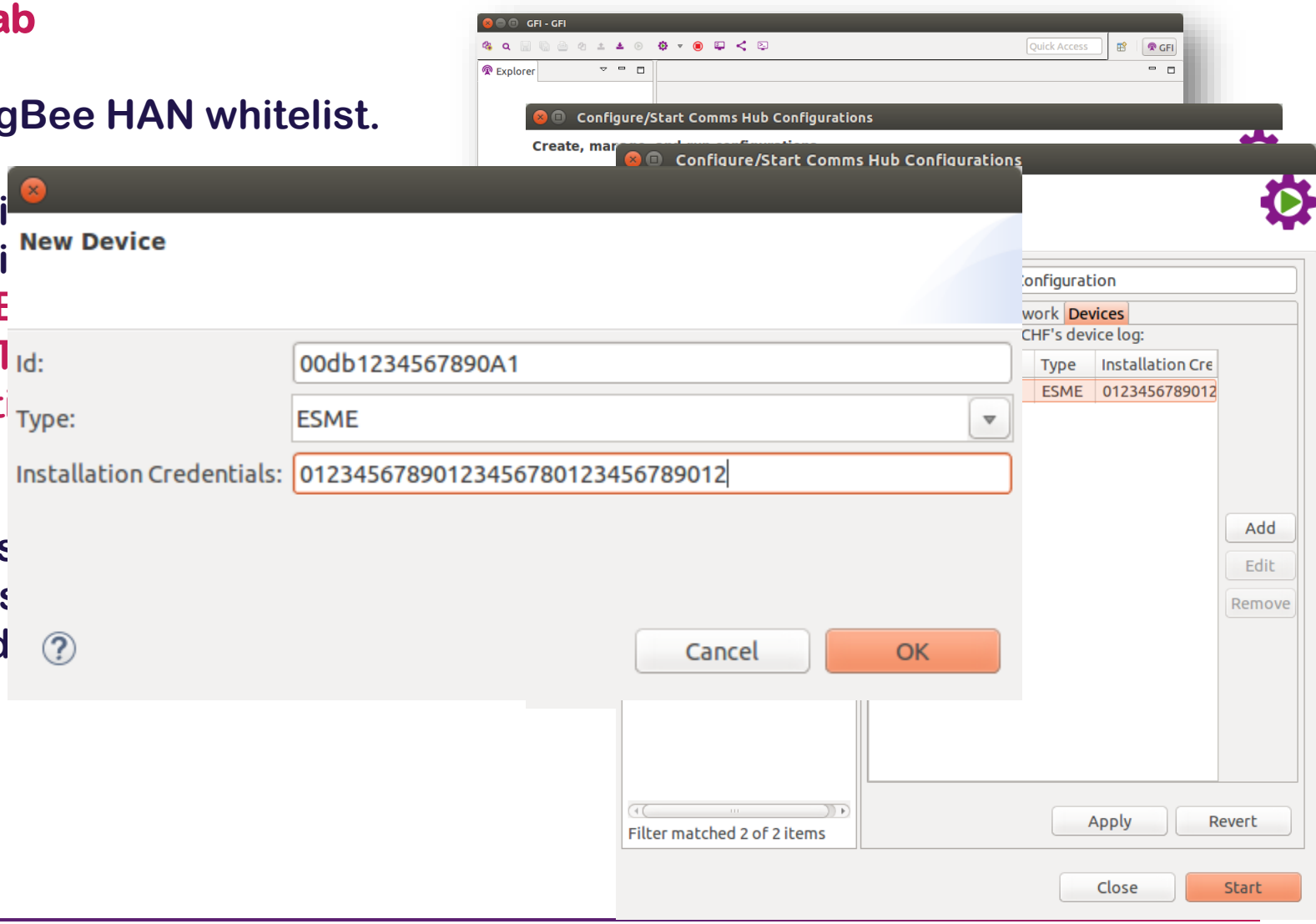
Devices Tab

Has the ZigBee HAN whitelist.

The whiteli
list of entri

- Device E
- Device T
- Installat

It is not ess
the whitelis
configured
cases.

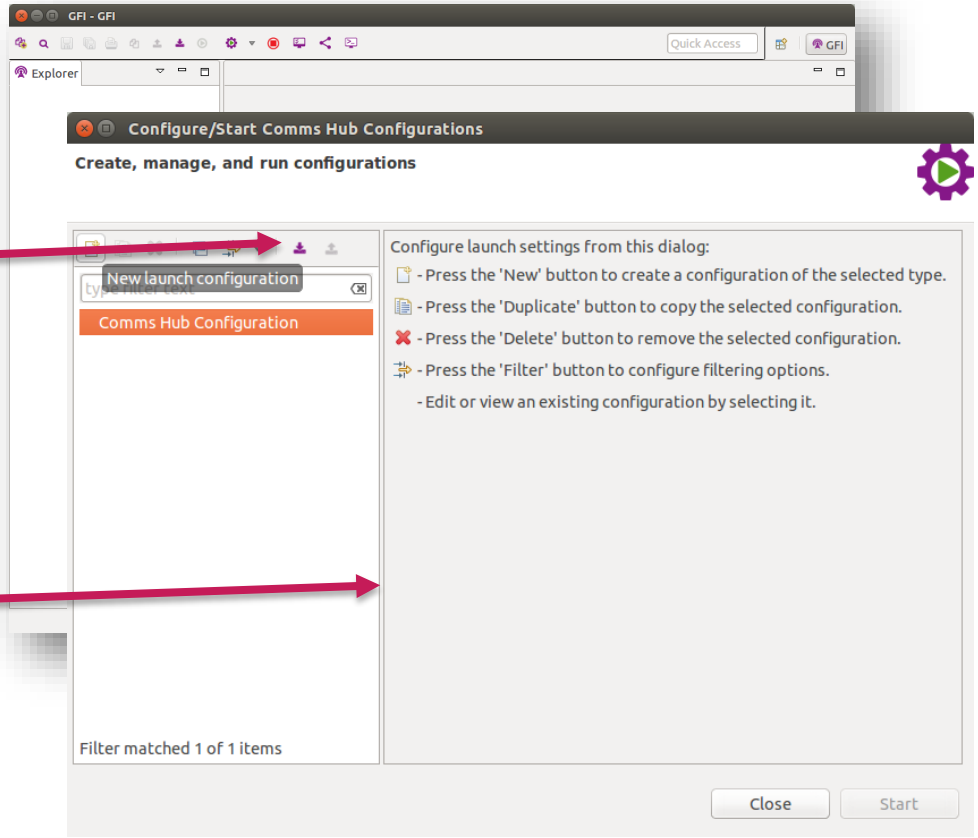


GUI: CH configuration (VI)

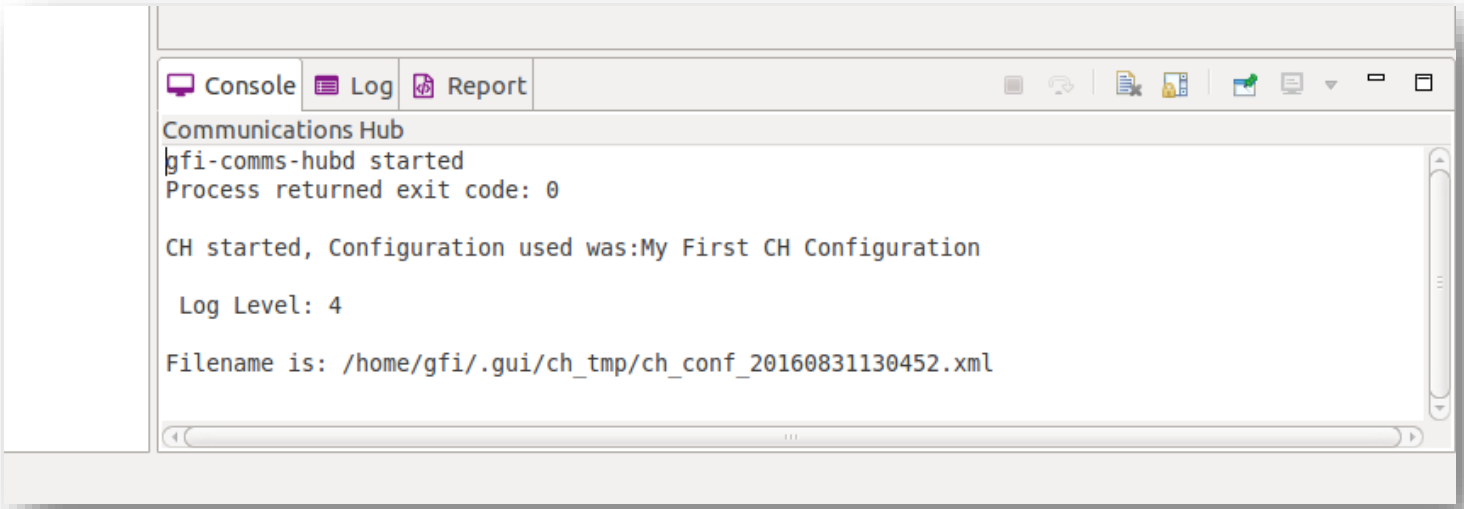
It is also possible to import and export CH configurations. These are stored as XML files.

This is managed by the two purple buttons on the toolbar.

These buttons are not normally visible and require the divisor to be moved right to be seen.



GUI: CH started



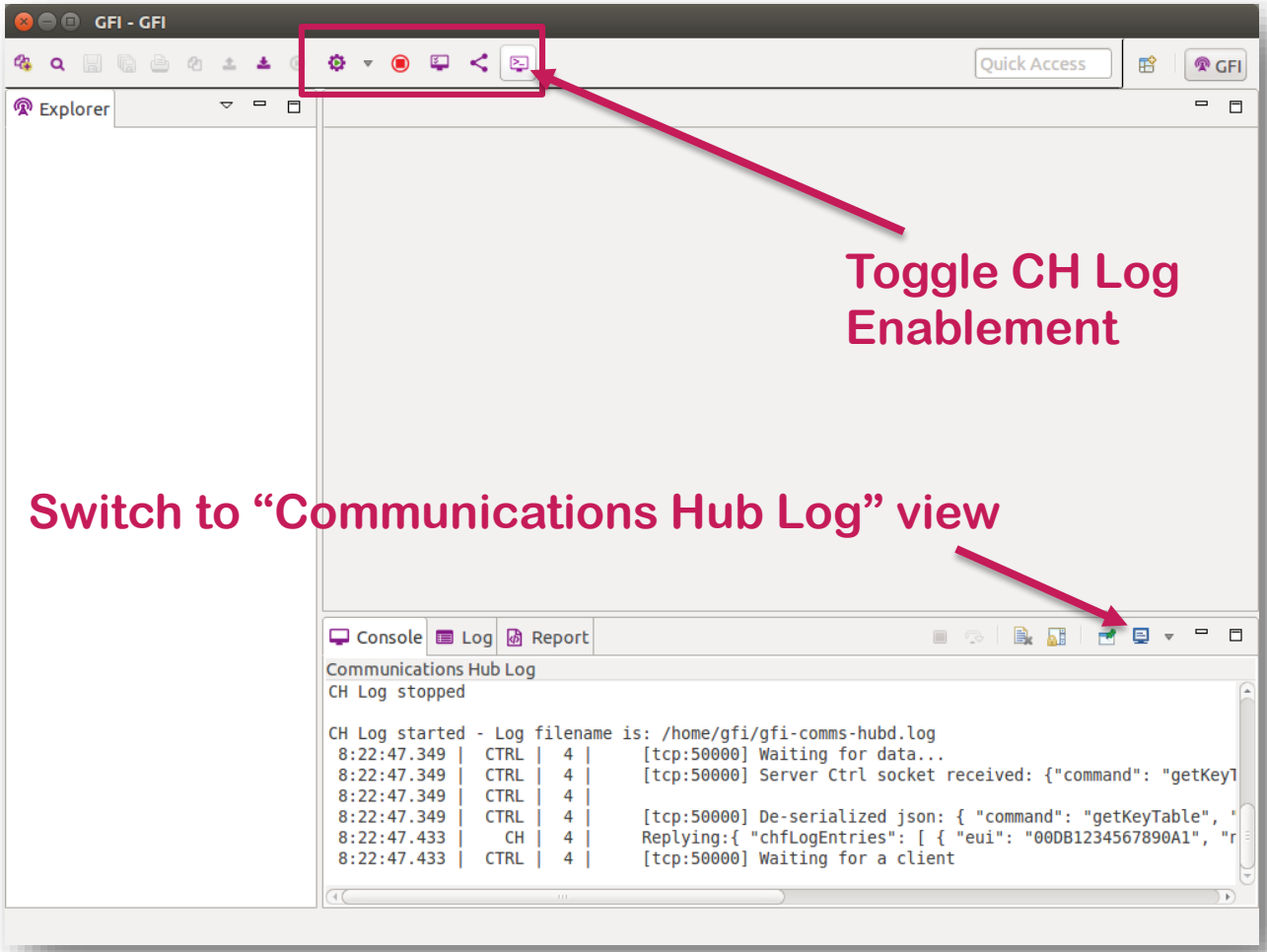
If successful, the exit code will be zero. If non-zero, check the GPF Id aligns with the EUI64 of the ZigBee radio (this is a common issue).

If unsuccessful a log file is saved at the location:

/home/gfi/gfi-comms-hubd.log

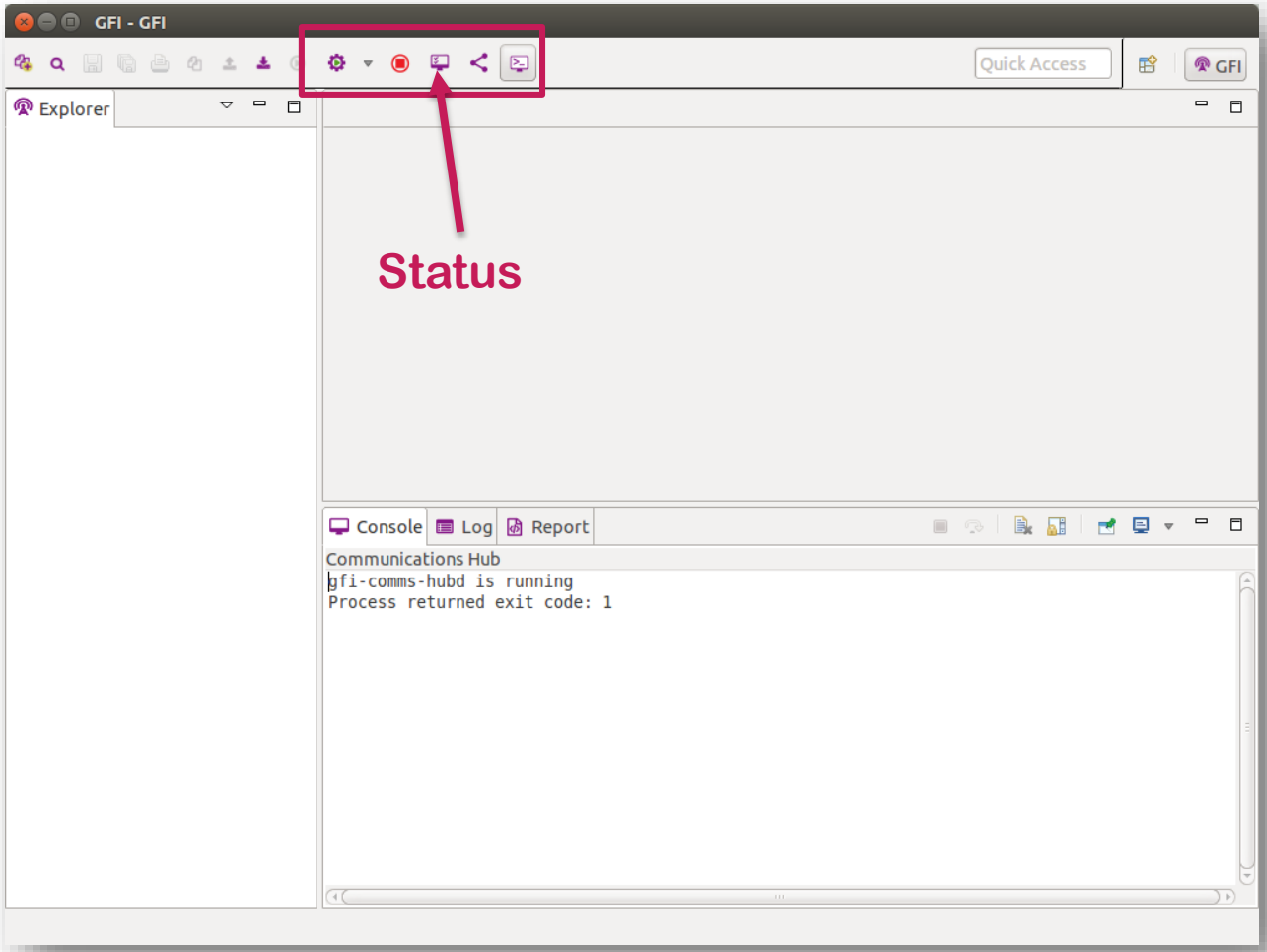
This details the error in more detail. If requesting support from GFI, please attach the log file to the support request.

GUI: CH log area



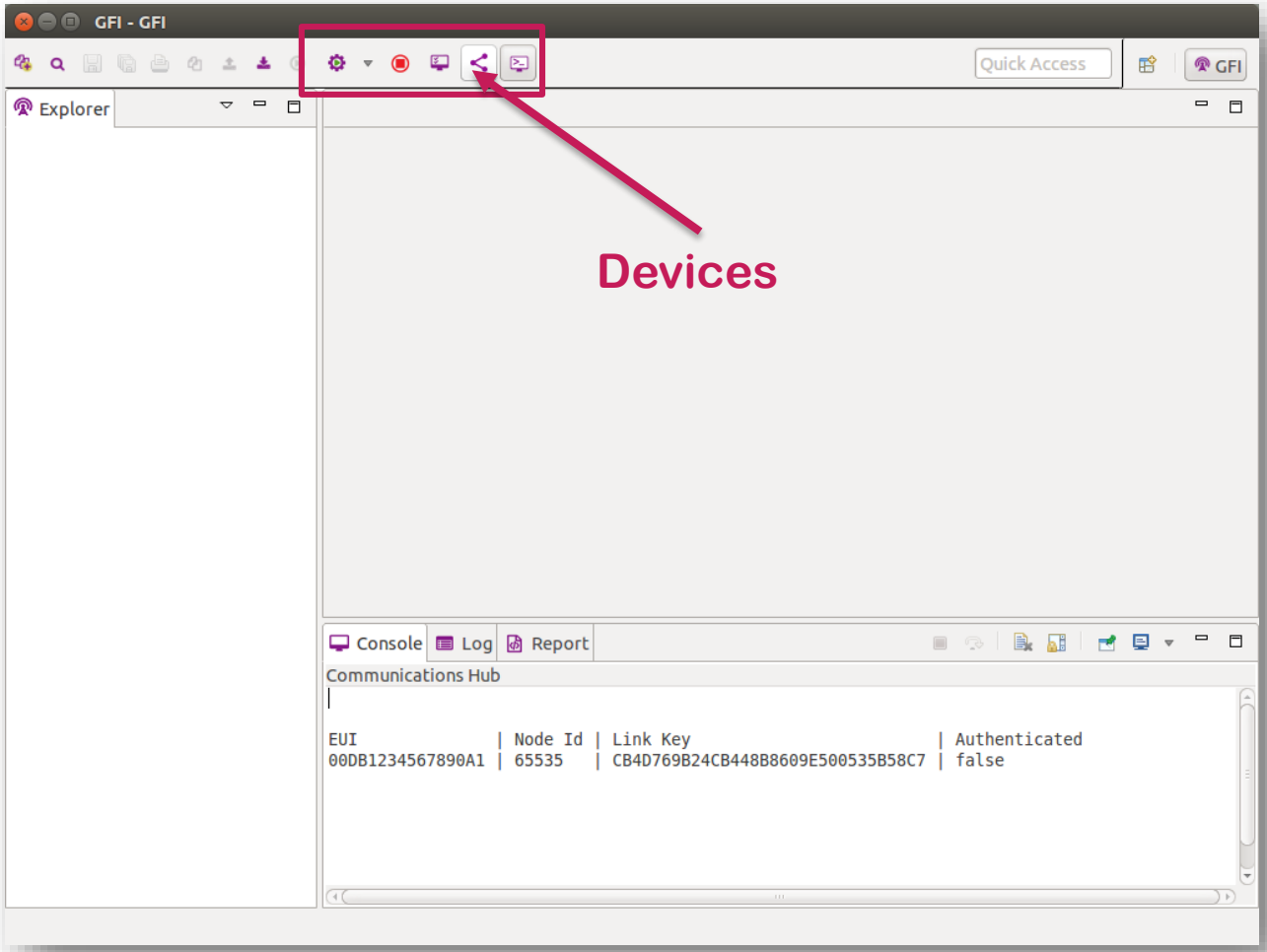
To view the CH log, it is required to first enable it via the tool bar, then switch to it via the console log selector.

GUI: CH status (I)



Identifies whether the CH daemon is running, or not.

GUI: CH status (II)



Displays the current state of the HAN. Which devices are in the whitelist and whether they are associated with the network.

GUI: step-by-step examples

Step-by-step examples will be presented to illustrate the testing tool operation.

GFI supports two distinct operation scenarios:

1) Emulated Scenario – Where the ESME/GSME is emulated

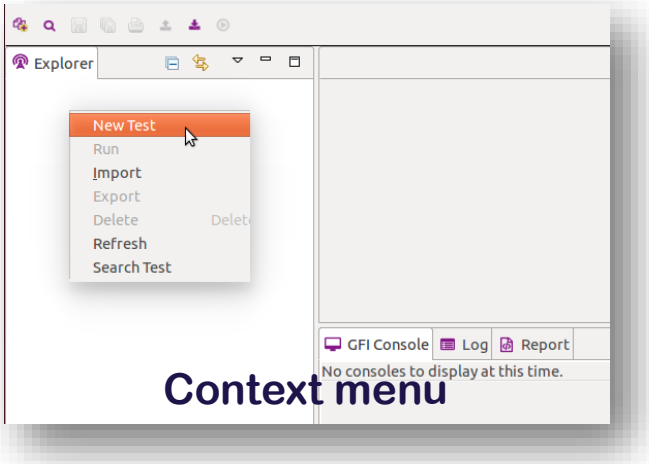
2) Physical Scenario – Nominal operation with real ESME/GSME

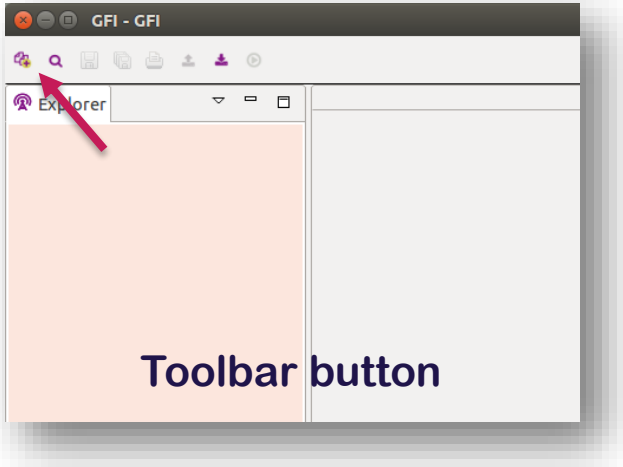
GUI: step-by-step example 1

Emulated Scenario

Step 1: Create New Test

This may be done by invoking the New Test wizard from the context menu on the Explorer, from the File menu or from the respective button on the toolbar



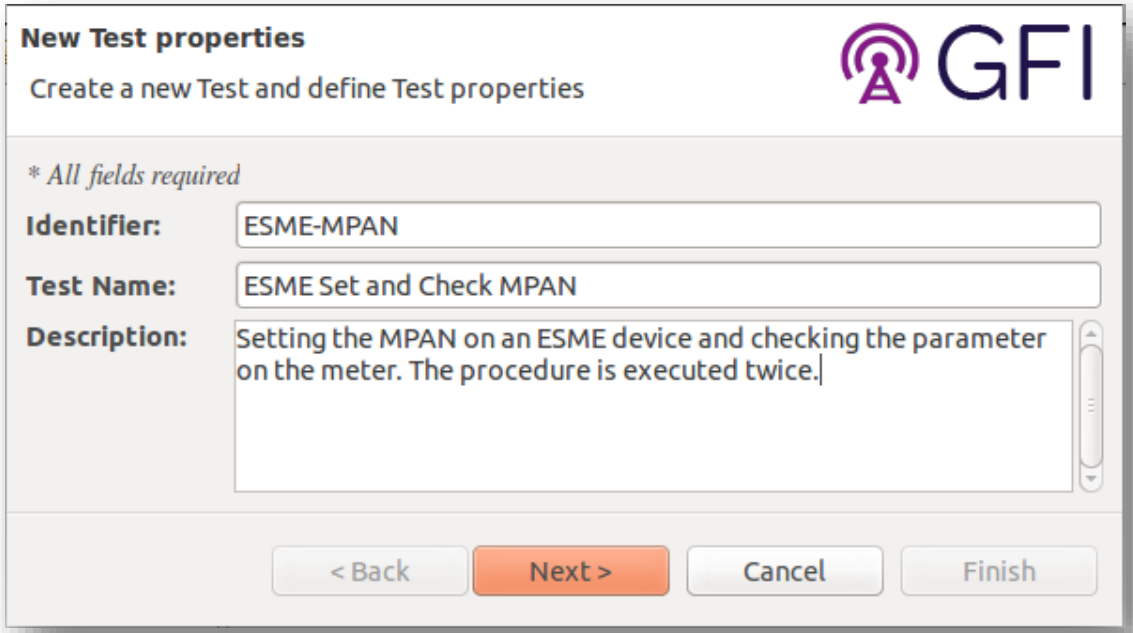


GUI: step-by-step example 1

Emulated Scenario

Step 2: Setup Properties

Gathers the initial information about the Test



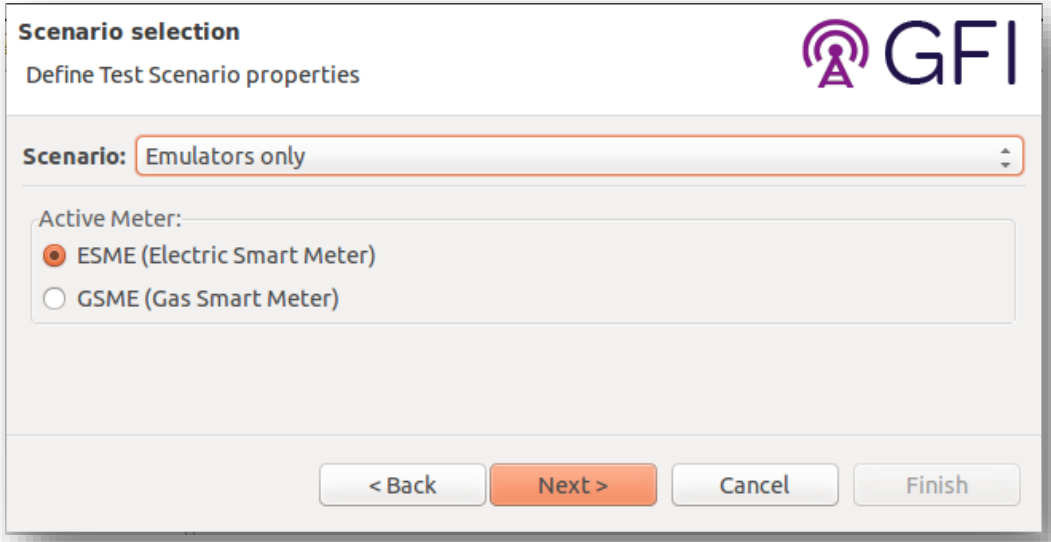
GUI: step-by-step example 1

Emulated Scenario

Step 3: Setup Scenario

Gathers the information required by the Test Scenario such as: network settings, MACs, Device certificates, etc.

In this example, we will select "Emulators only"

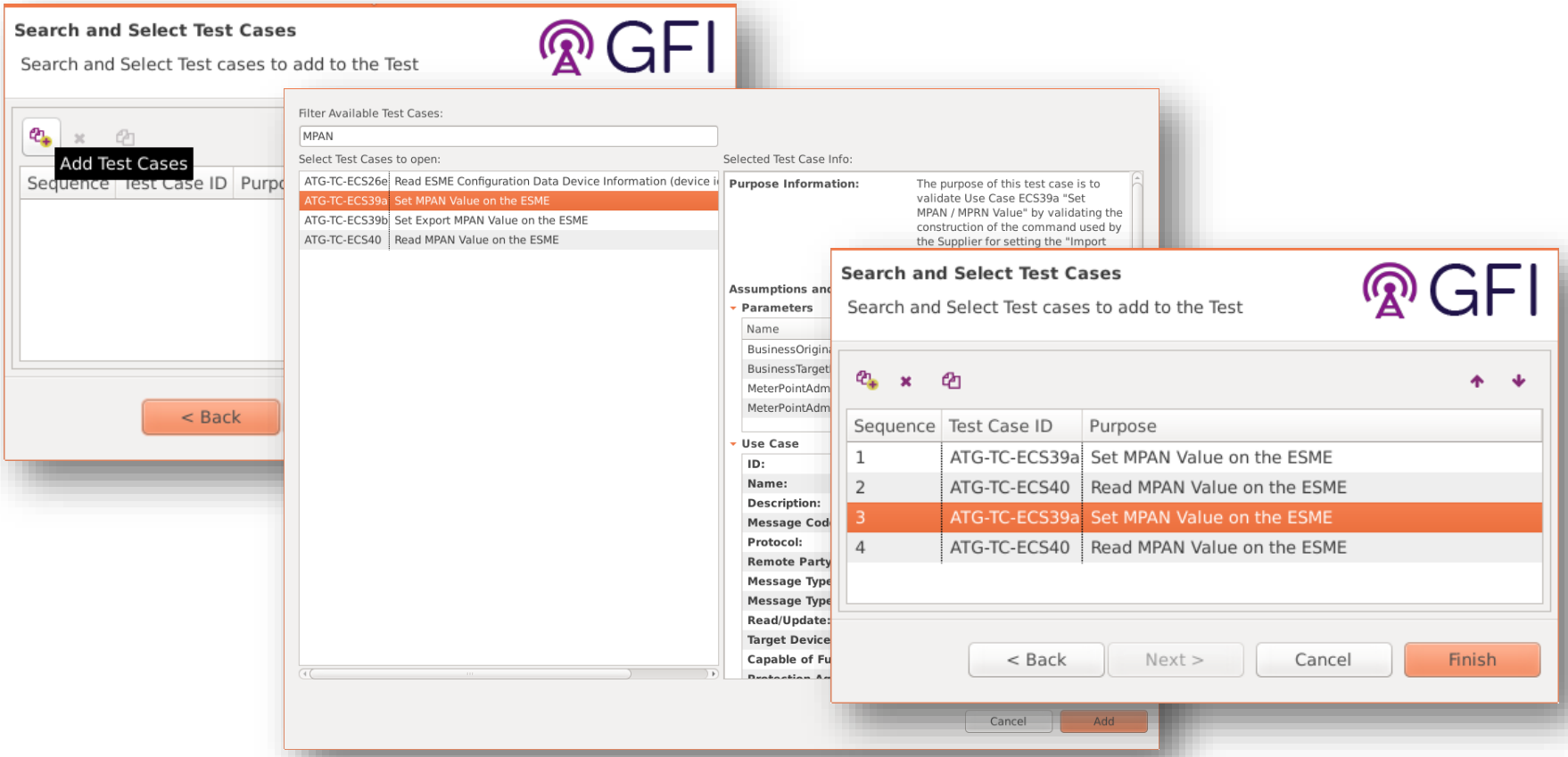


GUI: step-by-step example 1

Emulated Scenario

Step 4: Add Use Cases

Select the Use Cases to build the sequence for the new Test.

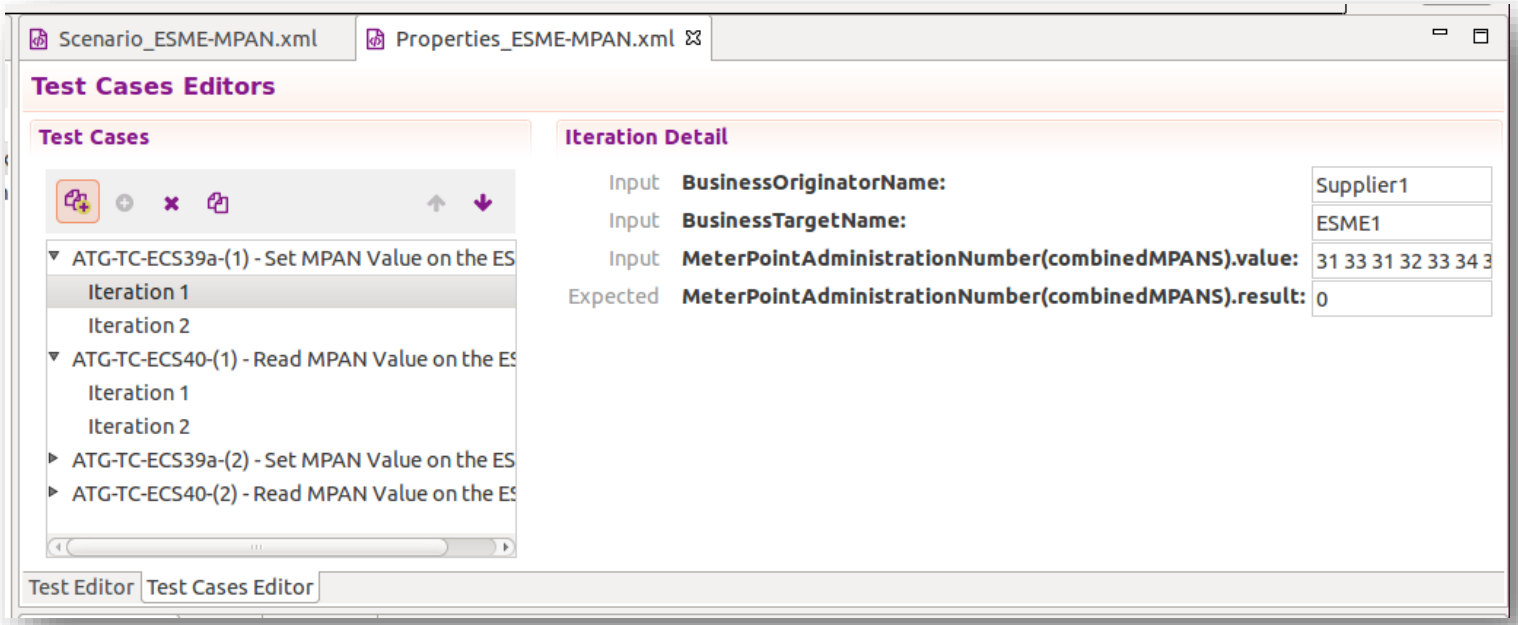


GUI: step-by-step example 1

Emulated Scenario

Step 5: Customize Use Cases Iterations

In this step, the User may adjust the Use Case inputs/expected outputs for each iteration. Iterations/Use Cases can be freely added/removed.



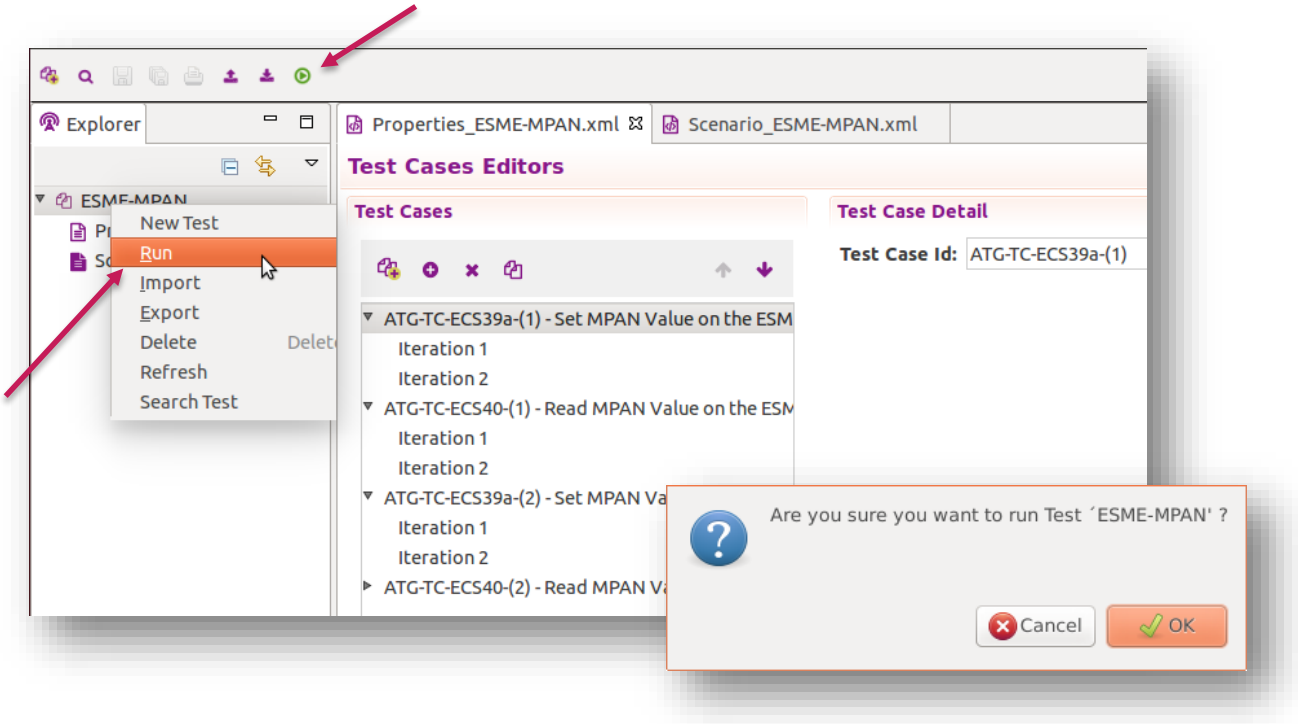
The number of iterations and the values for inputs and expected outputs are created from default values present in a Use Case library.

GUI: step-by-step example 1

Emulated Scenario

Step 6: Execute Test

The Test can be executed through the “Run” button in the toolbar or the option “Run” in the Test’s context menu.

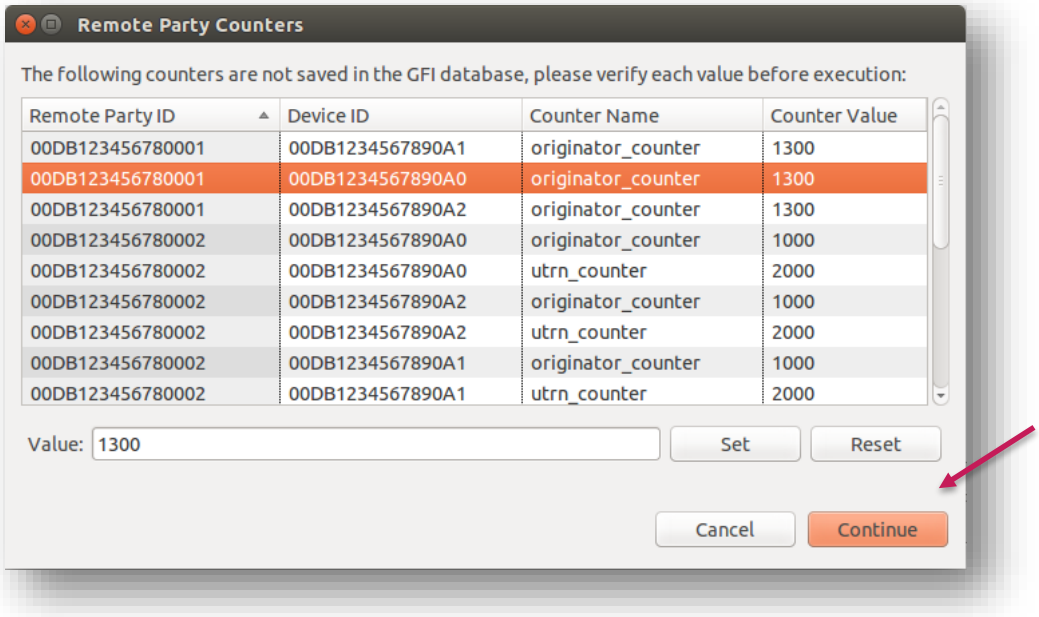


GUI: step-by-step example 1

Emulated Scenario

Step 6: Execute Test

A database of counters per Remote Party/Device is maintained by the application for the situations where protection against replay is used.



Click for the Test to be executed

The first time a Remote Party/Device association is used, their counter needs to be initialised. In this case, a dialog is presented to define the starting values of the counters.

GUI: step-by-step example 1

Emulated Scenario

Step 6: Execute Test

The console activity may then be observed in the Console Pane.
It is a real time output of the GFI execution.

When the execution is finished, the execution log and the report are presented in the Execution Log pane.

GFI Console | Log | Report

```
Found 1º argument. Searching for file [/home/gfiuser/.gui/tmp/tmpScenario8539409256231373 88gfi.xml].
Found 2º argument. Searching for file [/home/gfiuser/workspace-scrsht/ESME-MPAN/Properties_ESME-MPAN.xml].


-------------------------------------------------------------------------------------------------------
Initializing the Software Database (SDB)...
Starting GFI Execution...
-------------------------------------------------------------------------------------------------------
 STARTING EXECUTION TIME: 2015-09-25 19:24:28 (WEST)
-------------------------------------------------------------------------------------------------------
    TIME      | AREA  | LV | MESSAGE
-------------------------------------------------------------------------------------------------------
00:00:00.219 |  MNGE |  0 | Preparing Test Execution...
00:00:00.220 | RPORT |  4 |     Report location: /home/gfiuser/.gui/tmp/GFIServRun580155825909432952 1/test-report.html
00:00:00.235 |   ACB |  4 |     ACBEmulator: Start processing thread for messages received from xSME
00:00:00.235 |   ACB |  4 |     ACBEmulator: Waiting for new messages from xSME
00:00:00.236 | EQUIP |  4 |     GSMEEmulator: Start processing thread
00:00:00.236 | EQUIP |  4 |     GSMEEmulator: Waiting for new messages
00:00:00.254 |  MNGE |  0 | Starting Test Execution...
00:00:00.268 |   ACB |  4 |     ACBEmulator: Start processing thread for messages received from the KRP
00:00:00.268 |   ACB |  4 |     ACBEmulator: Waiting for new messages from KRP
00:00:00.269 | EQUIP |  4 |     ESMEEmulator: Start processing thread
00:00:00.269 | EQUIP |  4 |     ESMEEmulator: Waiting for new messages
00:00:00.269 | EQUIP |  4 |     CHFEmulator: Start processing thread
00:00:00.269 | EQUIP |  4 |     CHFEmulator: Waiting for new messages
00:00:00.270 |  MNGE |  0 | Starting Run 1 of 1...
00:00:00.270 |  MNGE |  0 | Executing Run 1 of 1...
00:00:00.270 |  MNGE |  2 |   Testcase: ATG-TC-ECS39a-(1), Starting...
00:00:00.278 |  MNGE |  2 |   Testcase: ATG-TC-ECS39a-(1), Starting Iteration: 1 of 2
00:00:00.278 |  MNGE |  2 |   Testcase: ATG-TC-ECS39a-(1), Iteration: 1 of 2, Starting Test Step: 1 of 1
00:00:00.279 |   KRP |  4 |     RPEmulator: Start processing thread
00:00:00.279 |   KRP |  4 |     RPEmulator: Waiting for new messages
00:00:00.284 |  TEST |  4 |     CGBCSUseCase: Preparing to send GENERAL_SIGNING_MESSAGE
00:00:00.285 |  TEST |  4 |     CGBCSUseCase: Adding signature to GENERAL_SIGNING_MESSAGE
00:00:00.286 | EQUIP |  4 |     KRP: GENERAL_SIGNING_MESSAGE detailed byte array (68 bytes):
```

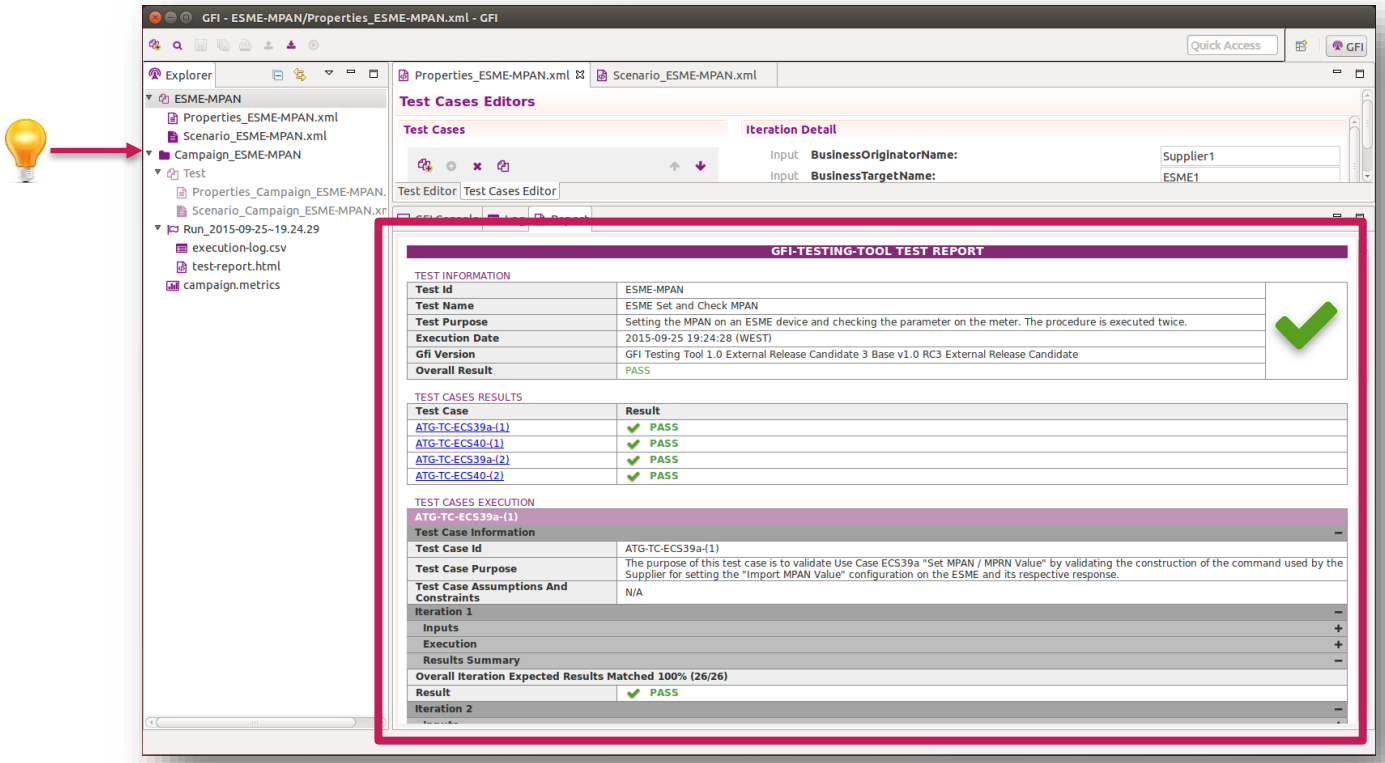
Only the report corresponding to the last execution of a Test is displayed in this pane. All previously saved reports are displayed in the Edit/View pane.

GUI: step-by-step example 1

Emulated Scenario

Step 6: Execute Test

Upon running the Test a new Campaign is added to the Explorer pane. The Test report is automatically presented.

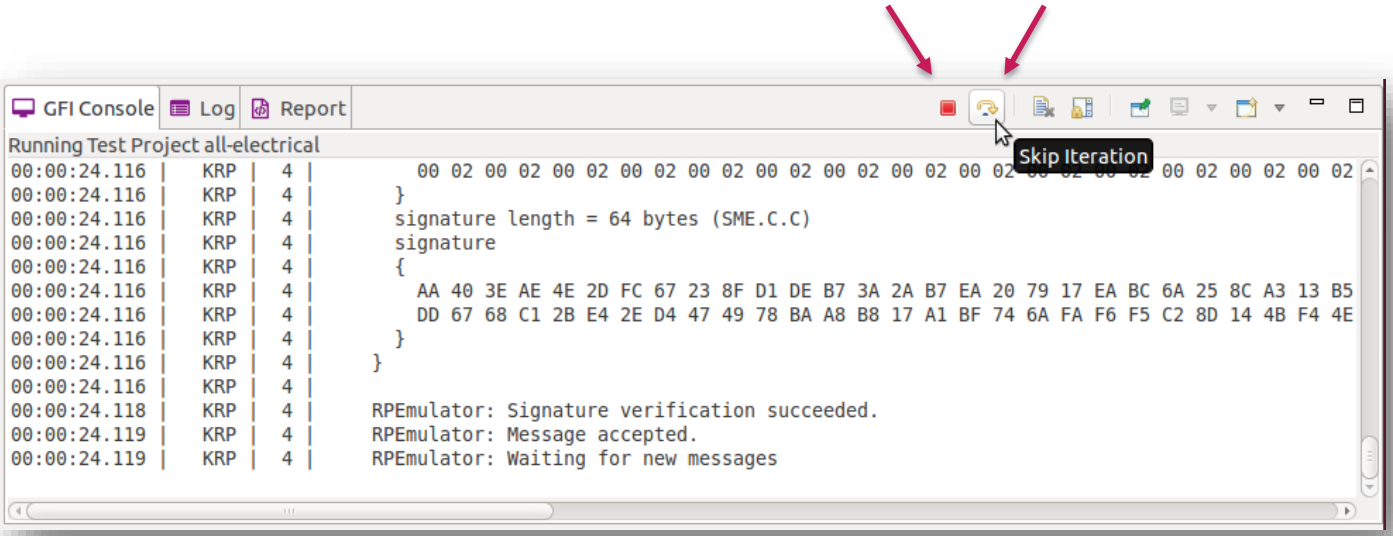


GUI: step-by-step example 1

Emulated Scenario

Step 6: Execute Test

During the execution, the User can skip the current iteration by clicking the skip button in the console pane. In much the same way a Test execution may be stopped using the stop button.

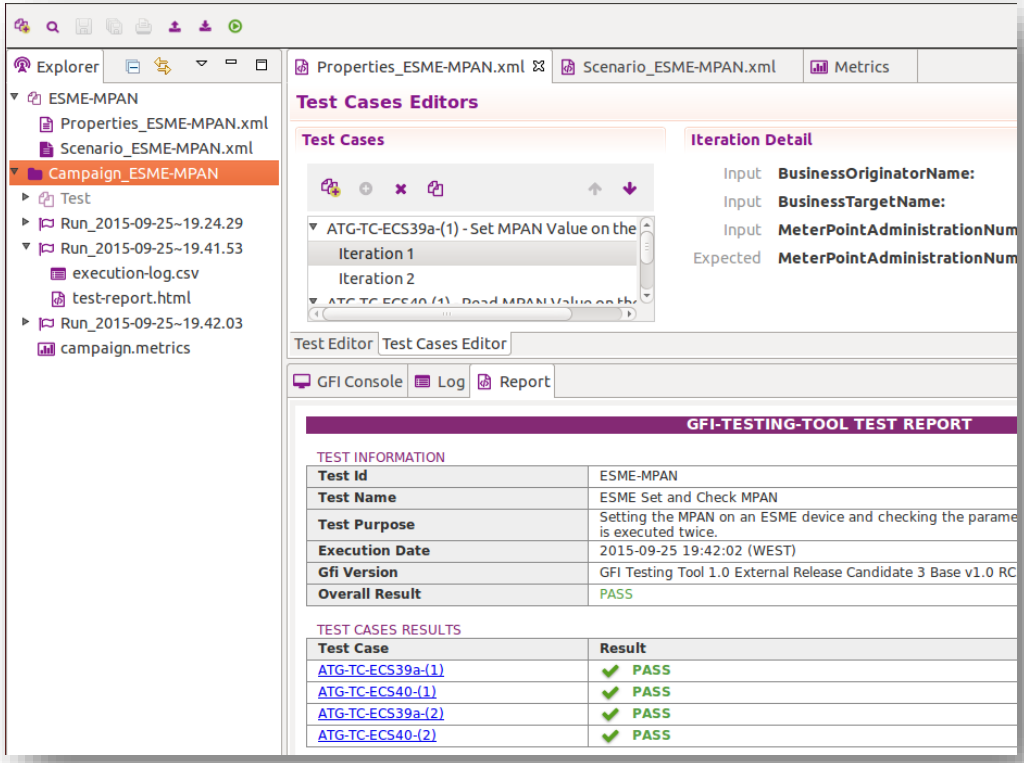


GUI: step-by-step example 1

Emulated Scenario

Step 6: Execute Test

Whenever a Test is executed, a new Campaign is generated.
On the other hand, if a Campaign is executed, a new Run is generated.



Campaigns are ALWAYS executed based on the same Test conditions (Properties and Scenario).

GUI: step-by-step example 1

Emulated Scenario

Step 7: Check Results

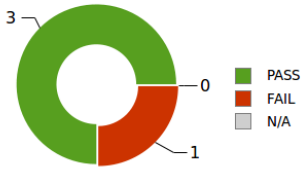
Campaign Metrics Information

Test ID: ESME-MPAN

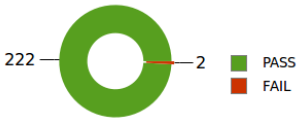
Purpose: Setting the MPAN on an ESME device and checking the parameter on the meter. The procedure is executed twice.

Campaign Run Dates	Total	Fail	Pas
2015/09/26 00:15:22	4	1	3

TEST CASES



CHECKS



GFI Console | Log | Report

00:00:00.505	CHECK	ECS40_RESPONSE.PAYLOAD.LONG_INVOKE_ID_PRIO				
00:00:00.505	CHECK	ECS40_RESPONSE.PAYLOAD.LONG_INVOKE_ID_PRIO				
00:00:00.505	CHECK	ECS40_RESPONSE.PAYLOAD.LONG_INVOKE_ID_PRIO				
00:00:00.505	CHECK	ECS40_RESPONSE.PAYLOAD.DATE_TIME.LENGTH				
00:00:00.505	CHECK	ECS40_RESPONSE.PAYLOAD.TAG				
00:00:00.505	CHECK	GENERAL_SIGNING_MESSAGE.GROUPING_HEADER.C				
00:00:00.505	CHECK	ECS40_RESPONSE.PAYLOAD.ACCESS_RESPONSE_BO N.NUMBER_OF_REQUESTS	0	equal to	0	PASS
00:00:00.505	CHECK	ECS40_RESPONSE.PAYLOAD.ACCESS_RESPONSE_BODY.ACCESS_RESPONSE_LIST_OF_DAT A.NUMBER_OF_REQUEST_RESPONSES	2	equal to	2	PASS
00:00:00.505	CHECK	ECS40_RESPONSE.PAYLOAD.ACCESS_RESPONSE_BODY.ACCESS_RESPONSE_SPECIFICATI ON.NUMBER_OF_REQUEST_RESULTS	2	equal to	2	PASS
00:00:00.505	CHECK	ECS40_RESPONSE.PAYLOAD.ACCESS_RESPONSE_BODY.ACCESS_RESPONSE_SPECIFICATI ON.REQUEST_RESULTS.METER_ADMINISTRATION_POINT_NUMBER_COMBINED_MPANS.AC CESS_RESPONSE_TYPE	1	equal to	1	PASS
00:00:00.505	CHECK	ECS40_RESPONSE.PAYLOAD.ACCESS_RESPONSE_BODY.ACCESS_RESPONSE_SPECIFICATI ON.REQUEST_RESULTS.METER_ADMINISTRATION_POINT_NUMBER_EXPORT_MPAN.ACCES S_RESPONSE_TYPE	1	equal to	1	PASS
00:00:00.505	CHECK	ECS40_RESPONSE.PAYLOAD.ACCESS_RESPONSE_BODY.ACCESS_RESPONSE_LIST_OF_DAT A.REQUEST_RESPONSES.METER_ADMINISTRATION_POINT_NUMBER_COMBINED_MPANS	12 34 56	equal to	31 33 31 32 33 34 35 36 37 38 31 31 31 31 33 31 32 33 34 35 36 37 38 32 32 32	FAIL
00:00:00.505	CHECK	ECS40_RESPONSE.PAYLOAD.ACCESS_RESPONSE_BODY.ACCESS_RESPONSE_LIST_OF_DAT A.REQUEST_RESPONSES.METER_ADMINISTRATION_POINT_NUMBER_EXPORT_MPAN	31 33 31 32 33 34 35 36 37 38 33 33 33	equal to	31 33 31 32 33 34 35 36 37 38 33 33 33	PASS
00:00:00.505	CHECK	ECS40_RESPONSE.PAYLOAD.ACCESS_RESPONSE_BODY.ACCESS_RESPONSE_SPECIFICATI ON.REQUEST_RESULTS.METER_ADMINISTRATION_POINT_NUMBER_COMBINED_MPANS.RES ULT	0	equal to	0	PASS
00:00:00.505	CHECK	ECS40_RESPONSE.PAYLOAD.ACCESS_RESPONSE_BODY.ACCESS_RESPONSE_SPECIFICATI ON.REQUEST_RESULTS.METER_ADMINISTRATION_POINT_NUMBER_EXPORT_MPAN.RESULT	0	equal to	0	PASS

Results Summary

Overall Iteration Expected Results Matched 96.7% (29/30)

Result FAIL

Iteration 2

GUI: step-by-step example 1

Emulated Scenario

Step 7: Check Results

GFI - Campaign_ESME-MPAN_003/Run_2015-09-26~00.15.22/test-report.html - GFI

Explorer

- ESME-MPAN
  - Properties_ESME-MPAN.xml
  - Scenario_ESME-MPAN.xml
- Campaign_ESME-MPAN
  - Test
  - Run_2015-09-25~19.24.29
  - Run_2015-09-25~19.41.53
  - Run_2015-09-25~19.42.03
  - campaign.metrics
- Campaign_ESME-MPAN_001
- Campaign_ESME-MPAN_002
- Campaign_ESME-MPAN_003
  - Test
  - Run_2015-09-26~00.15.22
    - execution-log.csv
    - test-report.html
  - campaign.metrics
- Campaign_ESME-MPAN_004
  - Test
  - Run_2015-09-26~00.38.55
  - campaign.metrics

Properties_ESME-M... | Scenario_ESME-MPA... | Metrics | Campaign_ESME-MPA...

GFI-TESTING-TOOL TEST REPORT

TEST INFORMATION

Test Id	ESME-MPAN
Test Name	ESME Set and Check MPAN
Test Purpose	Setting the MPAN on an ESME device and checking the parameter on the meter. The procedure is executed twice.
Execution Date	2015-09-26 00:15:21 (WEST)
Gfi Version	GFI Testing Tool 1.0 External Release Candidate 3 Base v1.0 RC3 External Release Candidate
Overall Result	FAIL

TEST CASES RESULTS

Test Case	Result
ATG-TC-ECS39a-(1)	PASS
ATG-TC-ECS40-(1)	PASS
ATG-TC-ECS39a-(2)	PASS
ATG-TC-ECS40-(2)	FAIL

TEST CASES EXECUTION

ATG-TC-ECS39a-(1)

Test Case Information

Test Case Id	ATG-TC-ECS39a-(1)
Test Case Purpose	The purpose of this test case is to validate Use Case ECS39a "Set MPAN / MPRN Value" by validating the construction of the command used by the Supplier for setting the "Import MPAN Value" configuration on the ESME and its respective response.
Test Case Assumptions And Constraints	N/A

Iteration 1

Report

GFI-TESTING-TOOL TEST REPORT

TEST INFORMATION

Test Id	ESME-MPAN
Test Name	ESME Set and Check MPAN
Test Purpose	Setting the MPAN on an ESME device and checking the parameter on the meter. The procedure is executed twice.
Execution Date	2015-09-26 00:38:55 (WEST)
Gfi Version	GFI Testing Tool 1.0 External Release Candidate 3 Base v1.0 RC3 External Release Candidate
Overall Result	PASS

TEST CASES RESULTS

Test Case	Result
ATG-TC-ECS39a-(1)	PASS
ATG-TC-ECS40-(1)	PASS
ATG-TC-ECS39a-(2)	PASS
ATG-TC-ECS40-(2)	PASS

TEST CASES EXECUTION

ATG-TC-ECS39a-(1)

Test Case Information

Test Case Id	ATG-TC-ECS39a-(1)
Test Case Purpose	The purpose of this test case is to validate Use Case ECS39a "Set MPAN / MPRN Value" by validating the construction of the command used by the Supplier for setting the "Import MPAN Value" configuration on the ESME and its respective response.
Test Case Assumptions And Constraints	N/A

Iteration 1

GFI Console | Log

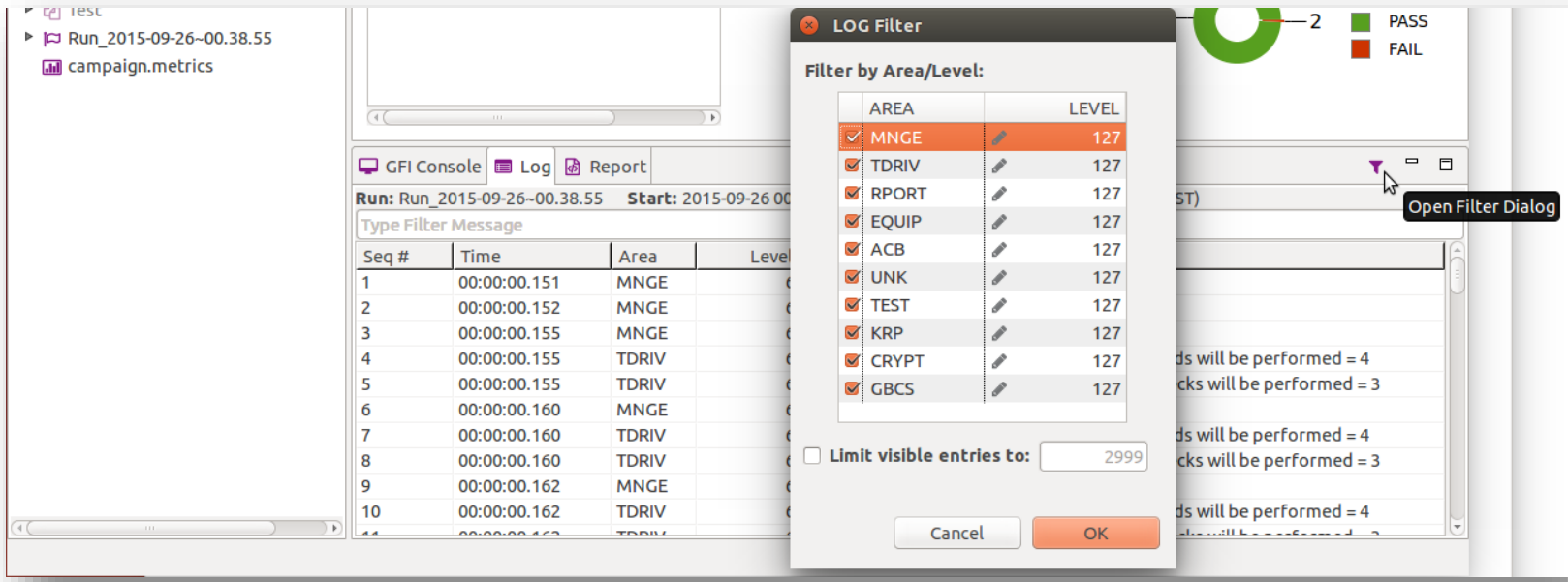
```
Running Test Project ESME-MPAN
00:00:00.562 | RPORT |  0 | ----------------------------------------------------------------
00:00:00.562 | RPORT |  0 | OVERALL TEST RESULT: PASS
00:00:00.562 | RPORT |  0 | ----------------------------------------------------------------
--------------------------------------------------------------------------------------------
Complete GFI Execution.
--------------------------------------------------------------------------------------------
All clean up, exiting.
```

1 item selected

GUI: step-by-step example 1

Emulated Scenario

Step 7: Check Results



GUI: step-by-step example 1

Emulated Scenario

Step 8: Import and Export Tests

To import:

- Go to File -> Import
- Enter a Test name
- Select the scenario and properties files
- Press OK

To export:

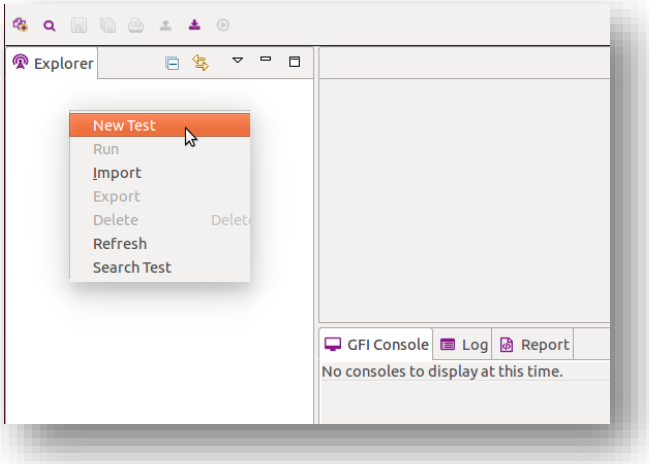
- Right click on a Test
- Select Export
- Select directory where to save the scenario and properties files

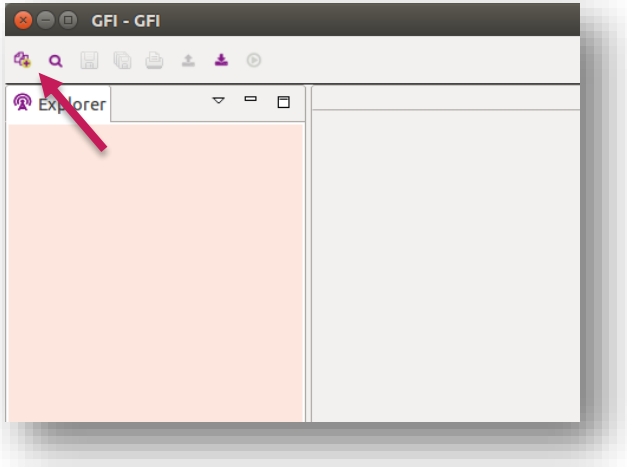
GUI: step-by-step example 2

Physical Scenario

Step 1: Create New Test

This may be done by invoking the New Test wizard from the context menu on the Explorer, from the File menu or from the respective button on the toolbar



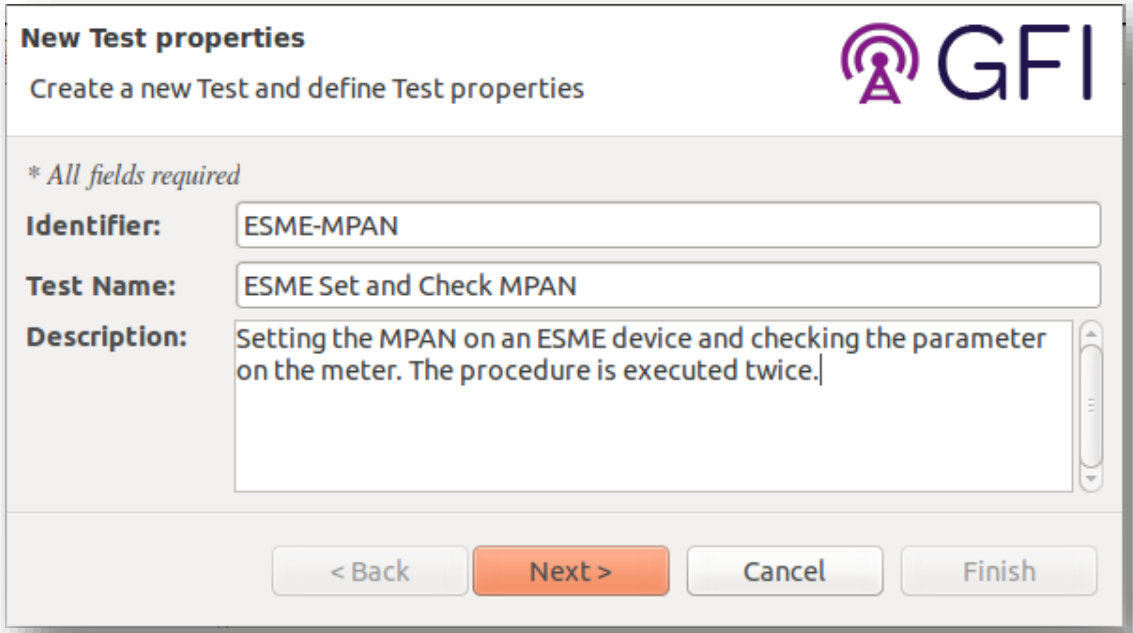


GUI: step-by-step example 2

Physical Scenario

Step 2: Setup Properties

Gathers the initial information about the Test



GUI: step-by-step example 2

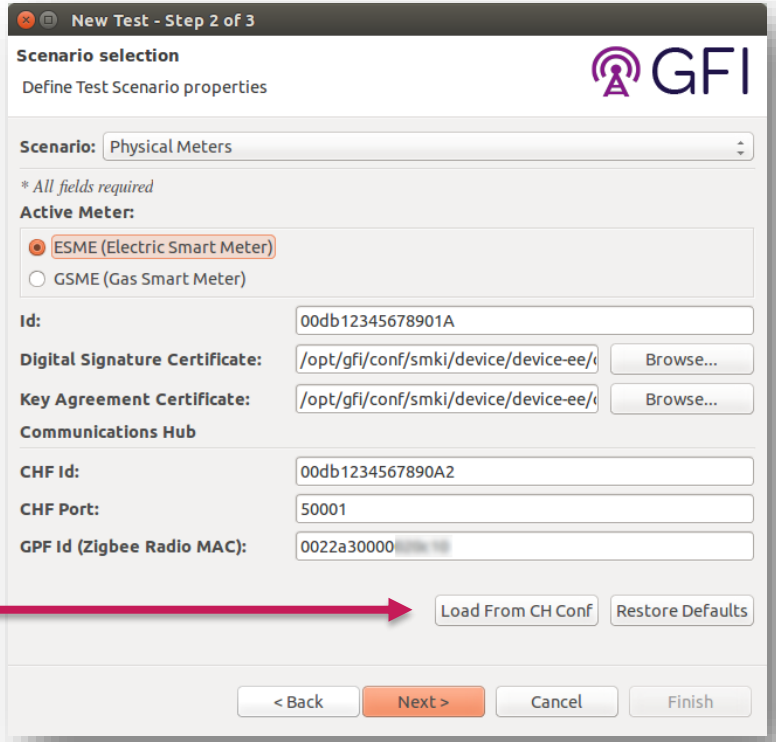
Physical Scenario

Step 3: Setup Scenario

Gathers the information required by the Test Scenario such as: MACs, Device certificates, etc.

Initially, the fields in this dialog are shown blank.
After a Test is created successfully the configuration is saved.

Some of this information is duplicated from the CH configuration, such as the CHF and GPF Id's. This can be directly loaded from the CH configuration.

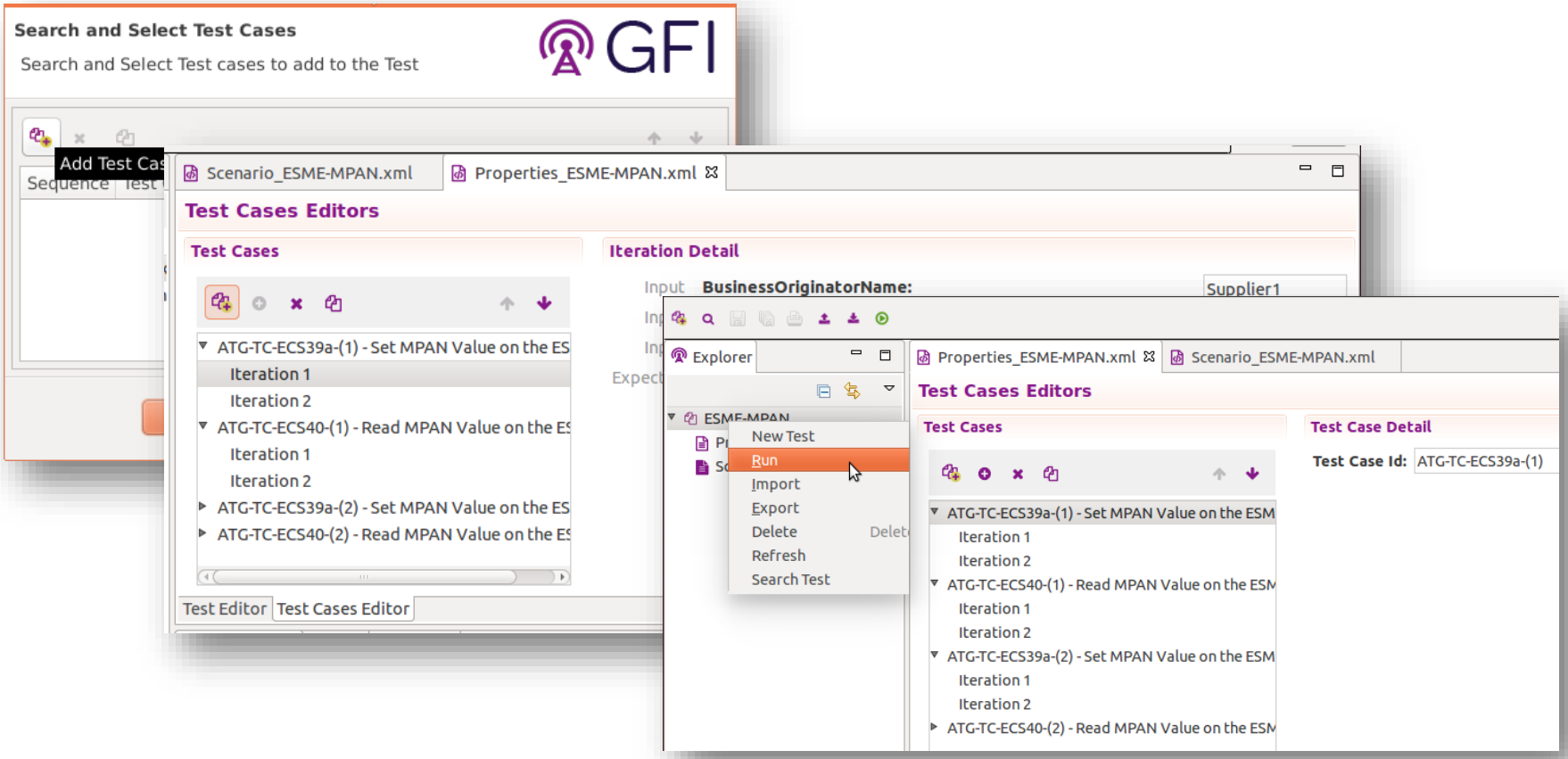


GUI: step-by-step example 2

Physical Scenario

Step 4 to 6:

The same as in example 1



GUI: ADVANCED OPERATIONS

GUI: advanced operations

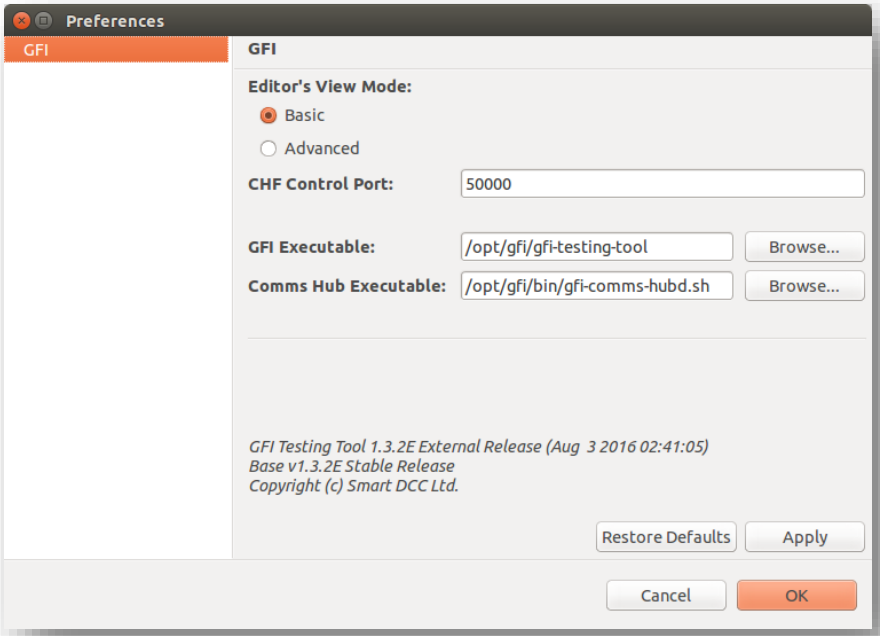
Edit Modes

a) Basic mode

Default mode, with a graphical interface

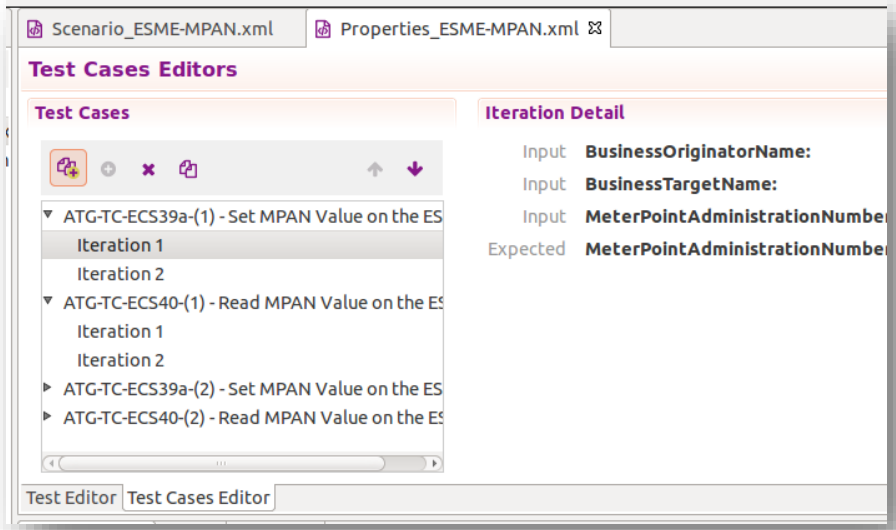
a) Advanced mode

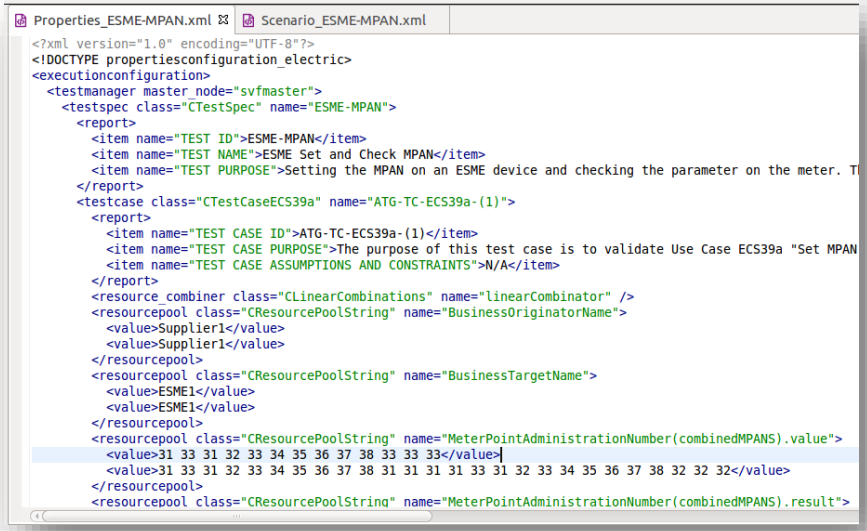
For experienced users, edition in the XML file



GUI: advanced operations

Edit Modes





CH: ADVANCED OPERATIONS

CH: command-line interface (I)

GFI Communications Hub can be invoked in the command-line.

Nominal invocation:

1 2

```
gfi@gfi:/opt/gfi$ ./bin/gfi-comms-hub conf/my-ch-configuration.xml
```

Here, the only parameter given is the configuration file. This file contains the ZigBee HAN settings and CH device log defaults.

CH: command-line interface (II)

Option	Description
configuration-file.xml	The configuration file
--daemon	Starts the communications hub without an interactive menu, i.e. it only displays the output.
--loglevel=n	Sets the console log level to level n, defaults to 4 if not specified.
--logdir=dir	Sets the directory where the logfiles will be saved, defaults to current directory if not specified. If existing log files are located in the same directory, they will be removed when the tool runs.
--version	Display the tool version

CH: command-line interface (III)

Communications Hub interactive menu

```
====================== Debug console ======================
Commands available:
         d        : print connected devices information
         t <node>: Start Tunnel
         s <node>: Stop Tunnel
         T <node>: Send message through Tunnel
         m <node>: read Metering status
         C        : Print Counters
         k        : Print Keys
         a        : Print address table
         i <endp>: Print this node info
         A <eui> : Add device to key tables
         R <eui> : Remove device from key tables
         U <eui> : Unjoin device link key
         ?        : List available options
```

CH: command-line interface (IV)

Console will show in real time the relevant actions performed in the Test

```
-----------------------------------------------------------------------------------------------
 STARTING EXECUTION TIME: Fri Aug  5 10:38:55 2016

-----------------------------------------------------------------------------------------------
    TIME      | AREA   | LV | MESSAGE
-----------------------------------------------------------------------------------------------
10:38:55.760 |     CH |  4 |     Loading CH configuration from conf/ch.my.xml
10:38:55.761 |   CTRL |  4 |     [tcp:50000] Waiting for a client
10:38:55.761 | STORE  |  4 |     Loading Coordinator configuration from conf/ch.my.xml
10:38:55.762 | ZIGBE  |  4 |     Using serial port /dev/ttyUSB0
10:38:55.910 | ZIGBE  |  4 |     ezsp ver 0x4 stack type 0x2 stack ver. [5.3.1 Pre-Releasex build 551]
10:38:55.917 | ZIGBE  |  4 |     Ezsp Config: set 0x7f63cbb26c94 to 0x 7:Success: set
10:38:55.923 | ZIGBE  |  4 |     Ezsp Config: set 0x7f63cbb26cac to 0x 5:Success: set
10:38:55.930 | ZIGBE  |  4 |     Ezsp Config: set 0x7f63cbb26cbb to 0x10:Success: set
10:38:55.937 | ZIGBE  |  4 |     Ezsp Config: set 0x7f63cbb26cce to 0x 2:Success: set
10:38:55.943 | ZIGBE  |  4 |     Ezsp Config: set 0x7f63cbb26cdc to 0x 2:Success: set
10:38:55.949 | ZIGBE  |  4 |     Ezsp Config: set 0x7f63cbb26cea to 0x1e00:Success: set
10:38:55.955 | ZIGBE  |  4 |     Ezsp Config: set 0x7f63cbb26d02 to 0x1e:Success: set
10:38:55.961 | ZIGBE  |  4 |     Ezsp Config: set 0x7f63cbb26d0b to 0x8000:Success: set
10:38:55.968 | ZIGBE  |  4 |     Ezsp Config: set 0x7f63cbb26d19 to 0x 1:Success: set
10:38:55.974 | ZIGBE  |  4 |     Ezsp Policy: set 0x7f63cbb26d5a to "0x7f63cbb26d30":Success: set
```

CH: configuration (I)

```
<CommsHubConfiguration>
    <CHFId>00DB1234567890A2</CHFId>
    <GPFId>00DB1234567890A3</GPFId>
    <ControlPort>50000</ControlPort>
    <CHFPort>50001</CHFPort>
    <FirmwareVersion>GFIv1.3.2e</FirmwareVersion>
    <TimeoutMessages>3600000</TimeoutMessages>
    <ZigbeeNetwork>
        <PanId>1123</PanId>
        <ExtendedPanId>5122553344553344</ExtendedPanId>
        <NetworkKey>000102030405060708090A0b0c0d0e0f</NetworkKey>
        <Port>/dev/ttyUSB0</Port>
        <FlowControl>xonxoff</FlowControl>
        <PermitJoin>1</PermitJoin>
        <JoinTimeout>255</JoinTimeout>
        <Channel>12</Channel>
    </ZigbeeNetwork>
```

Same settings as located in the GUI configuration dialog.

Its possible to use the CH configuration exported from the GUI instead of manually creating this file.

There is a template of this file provided in the GFI installation.

CH: configuration (II)

```
    <DeviceLogCHFEntries>
        <Entry>
          <EntityIdentifier>00DB1234567890A1</EntityIdentifier>
          <InstallationCredentials>83FED3407A939723A5C639B200000000</InstallationCredentials>
          <DeviceType>0</DeviceType>
        </Entry>
    </DeviceLogCHFEntries>
</CommsHubConfiguration>
```

Multiple "entry" elements can be defined.

The device type aligns with DLMS definitions.

CH: remote interface

It is possible to interface with the CH over the network by directing JSON object to the chosen TCP port.

The following actions are supported:

- Exit
- View Whitelist and Associations
- Add Device to Whitelist
- Remove Device from Whitelist
- Un-Join with Device

These actions are limited to controlling the CH, with the goal to simplify integration of GFI into testing harnesses.

Currently it is not possible to directly inject GBCS messages into the CH.

CH: remote interface: exit

Requests the communications hub shutdown.

```
gfi@gfi:/opt/gfi$ echo {\"command\":\"exit\"} | nc localhost 50000
{"action": "exit"}
gfi@gfi:/opt/gfi$
```

CH: remote interface: view whitelist

Retrieve the whitelist for the HAN (CHF device log)

If successful, the command returns a JSON object representing the whitelist.

This is the same information that is displayed by the GUI.

```
gfi@gfi:/opt/gfi$ echo {\"command\":\"getKeyTable\"} | nc localhost 50000 | json_pp
{
   "chfLogEntries" : [
      {
         "nodeId" : 65535,
         "authenticated" : false,
         "eui" : "00DB1234567890A2",
         "linkKey" : "FC049DA8FAE3D6B2638B70DFA7471396"
      },
      {
         "nodeId" : 65535,
         "authenticated" : false,
         "linkKey" : "FC049DA8FAE3D6B2638B70DFA7471396",
         "eui" : "00DB1234567890A0"
      }
   ]
}

gfi@gfi:/opt/gfi$
```

CH: remote interface: add device

Add device to whitelist for the HAN (CHF device log)

💡 If the permit join window has closed this will not reopen that window. It is required to use CCS01 to do that.

```
gfi@gfi:/opt/gfi$ echo \
> {\"command\":\"addDevice\", \
>  \"parameters\":{\"eui\":\"0022A30000020C16\" , \
>                  \"installationCredentials\":\"83 FE D3 40 7A 93 97 23 A5 C6 39 B2 69 16 D5 10\" , \
>                  \"deviceType\" : \"1\" } } | nc localhost 50000
{"action": "addDevice"}
gfi@gfi:/opt/gfi$
```

CH: remote interface: remove device

Remove device to whitelist for the HAN (CHF device log)

```
gfi@gfi:/opt/gfi$ echo {\"command\":\"removeDevice\", \"parameters\":{\"eui\":\"0022A30000020C16\" } }
| nc localhost 50000
{"action": "removeDevice"}
gfi@gfi:/opt/gfi$
```

Un-Join device from the HAN.

```
gfi@gfi:/opt/gfi$ echo {\"command\":\"unJoinDevice\", \"parameters\":{\"eui\":\"0022A30000020C16\" } }
| nc localhost 50000
{"action": "unJoinDevice"}
gfi@gfi:/opt/gfi$
```

GFI: ADVANCED OPERATIONS

GFI: command-line interface

GFI testing tool can be invoked on the command-line.

Nominal invocation:

1 2 3

```
./gfi-testing-tool conf/gfi-scenario-emulators.xml tests/ATG-TEST-ECS40.xml
t. Searching for file [conf/gfi-scenario-emulators.xml].
t. Searching for file [tests/ATG-TEST-ECS40.xml].
```

GFI: command-line interface

Console will show in real time the relevant actions performed in the Test

```
gfi@gfi:/opt/gfi$ ./gfi-testing-tool conf/gfi-scenario-emulators.xml tests/ATG-TEST-ECS40.xml
Found 1◆◆ argument. Searching for file [conf/gfi-scenario-emulators.xml].
Found 2◆◆ argument. Searching for file [tests/ATG-TEST-ECS40.xml].


-----------------------------------------------------------------------------------------------
Initializing the Software Database (SDB)...
Starting GFI Execution...
-----------------------------------------------------------------------------------------------
 STARTING EXECUTION TIME: 2016-03-11 10:17:20 (GMT)
-----------------------------------------------------------------------------------------------
    TIME      | AREA   | LV | MESSAGE
-----------------------------------------------------------------------------------------------
00:00:00.121 |  MNGE |  0 | Preparing Test Execution...
00:00:00.121 | RPORT |  4 |     Report location: /opt/gfi/tests/test-report.html
00:00:00.123 |   ACB |  4 |     ACBEmulator: Start processing thread for messages received from xSME
00:00:00.123 |   ACB |  4 |     ACBEmulator: Waiting for new messages from xSME
00:00:00.123 |   ACB |  4 |     ACBEmulator: Start processing thread for messages received from the KRP
00:00:00.123 |   ACB |  4 |     ACBEmulator: Waiting for new messages from KRP
00:00:00.124 | EQUIP |  4 |     CHFEmulator: Start processing thread
00:00:00.124 | EQUIP |  4 |     CHFEmulator: Waiting for new messages
00:00:00.125 | EQUIP |  4 |     ESMEEmulator: Start processing thread
00:00:00.125 | EQUIP |  4 |     ESMEEmulator: Waiting for new messages
00:00:00.126 | EQUIP |  4 |     GSMEEmulator: Start processing thread
00:00:00.126 | EQUIP |  4 |     GSMEEmulator: Waiting for new messages
00:00:00.130 |   KRP |  4 |     RPEmulator: Start processing thread
00:00:00.130 |   KRP |  4 |     RPEmulator: Waiting for new messages
00:00:00.130 |  MNGE |  0 | Starting Test Execution...
00:00:00.130 |  MNGE |  0 | Starting Run 1 of 1...
00:00:00.130 |  MNGE |  0 | Executing Run 1 of 1...
00:00:00.130 |  MNGE |  2 |   Testcase: ATG-TC-ECS40, Starting...
00:00:00.132 |  MNGE |  2 |   Testcase: ATG-TC-ECS40, Starting Iteration: 1 of 4
00:00:00.133 |  MNGE |  2 |   Testcase: ATG-TC-ECS40, Iteration: 1 of 4, Starting Test Step: 1 of 1
00:00:00.135 | EQUIP |  4 |     CHFEmulator: New message available
00:00:00.135 | EQUIP |  4 |     CHFEmulator: Control message received
00:00:00.135 | EQUIP |  4 |     CHFEmulator: Processing ControlCommandHandler
00:00:00.135 | EQUIP |  4 |     Received Control Parameter [TwinElement] with value = True
00:00:00.135 | EQUIP |  4 |     CHFEmulator: Processing ControlResponseHandler
00:00:00.136 | EQUIP |  4 |     ESMEEmulator: New message available
```

GFI: command-line interface

In the directory where the Test file is, these files are created:

Last execution log and report:

- test-report.xml
- test-report.html
- execution-log.csv

Log and report backups:

- reports/<properties-filename>-test-report-<date>.xml
- reports/<properties-filename>-test-report-<date>.html
- logs/<properties-filename>-execution-log-<date>.csv

GFI: command-line interface options

Command line options available for the Tester

Help (--help)

Displays a help menu

```
gfi@gfi:/opt/gfi$ ./gfi-testing-tool --help
Usage: gfi-testing-tool <scenario-file.xml> <test-properties-file.xml> [options]
<scenario-file.xml>         The scenario configuration file
<test-properties-file.xml>  The Test properties file, containing the Test Cases to be executed.
Options:
  --help                    Display this information.
  --version                 Display the testing tool version information.
  --dcrypto                 Disable the rejection of messages due to invalid MAC or Signature.
  --checkmsg                Check if the cryptographic protections (MAC and/or Signature) of the given message are correct.
                            When using this option, the <test-properties-file.xml> should be replaced by a file with the message to che
ck.
  --loglevel  <level>       Configure the console output log level to <level>.
  --outputdir <dir>         Set <dir> as the base output directory for logs and reports.

gfi@gfi:/opt/gfi$
```

GFI: command-line interface options

Command line options available for the Tester

Version (--version)

Displays the software version of the GFI testing tool

```
gfi@gfi:/opt/gfi$ ./gfi-testing-tool --version
GFI Testing Tool 1.3 External Release Candidate 3 (Mar  6 2016 02:57:11)
Base v1.3 RC3 External Release Candidate
Copyright (c) Smart DCC Ltd.

gfi@gfi:/opt/gfi$
```

GFI: command-line interface options

Command line options available for the Tester

Cryptography (--dcrypto)

Disable the rejection of messages due to invalid MAC and Signature

```
gfi@gfi:/opt/gfi$ ./gfi-testing-tool conf/gfi-scenario-emulators.xml tests/ATG-TEST-ECS40.xml --dcrypto
Found 1◆◆ argument. Searching for file [conf/gfi-scenario-emulators.xml].
Found 2◆◆ argument. Searching for file [tests/ATG-TEST-ECS40.xml].


-----------------------------------------------------------------------------------------------
Initializing the Software Database (SDB)...
Starting GFI Execution...
-----------------------------------------------------------------------------------------------
 STARTING EXECUTION TIME: 2016-03-11 10:13:13 (GMT)
-----------------------------------------------------------------------------------------------
    TIME      | AREA   | LV | MESSAGE
-----------------------------------------------------------------------------------------------
00:00:00.134 |  MNGE |  0 | Preparing Test Execution...
00:00:00.134 | RPORT |  4 |     Report location: /opt/gfi/tests/test-report.html
00:00:00.136 |   ACB |  4 |     ACBEmulator: Start processing thread for messages received from xSME
00:00:00.136 |   ACB |  4 |     ACBEmulator: Waiting for new messages from xSME
00:00:00.136 |   ACB |  4 |     ACBEmulator: Start processing thread for messages received from the KRP
00:00:00.136 |   ACB |  4 |     ACBEmulator: Waiting for new messages from KRP
00:00:00.137 | EQUIP |  4 |     CHFEmulator: Start processing thread
00:00:00.138 | EQUIP |  4 |     CHFEmulator: Waiting for new messages
00:00:00.139 | EQUIP |  4 |     ESMEEmulator: Start processing thread
00:00:00.139 | EQUIP |  4 |     ESMEEmulator: Waiting for new messages
00:00:00.140 | EQUIP |  4 |     GSMEEmulator: Start processing thread
00:00:00.140 | EQUIP |  4 |     GSMEEmulator: Waiting for new messages
00:00:00.144 |   KRP |  2 |   WARNING: Rejection of messages due to invalid MAC or Signature was disabled by User.
00:00:00.144 |   KRP |  4 |     RPEmulator: Start processing thread
00:00:00.144 |   KRP |  4 |     RPEmulator: Waiting for new messages
00:00:00.144 |  MNGE |  0 | Starting Test Execution...
00:00:00.144 |  MNGE |  0 | Starting Run 1 of 1...
00:00:00.144 |  MNGE |  0 | Executing Run 1 of 1...
00:00:00.144 |  MNGE |  2 |   Testcase: ATG-TC-ECS40, Starting...
```

GFI: command-line interface options

Command line options available for the Tester

Verbose level (--loglevel)

Overrides the verbose level of the Console Output

```
gfi@gfi:/opt/gfi$ ./gfi-testing-tool conf/gfi-scenario-emulators.xml tests/ATG-TEST-ECS40.xml --loglevel 6
Found 1◆◆ argument. Searching for file [conf/gfi-scenario-emulators.xml].
Found 2◆◆ argument. Searching for file [tests/ATG-TEST-ECS40.xml].


-----------------------------------------------------------------------------------------------------
Initializing the Software Database (SDB)...
Starting GFI Execution...
-----------------------------------------------------------------------------------------------------
 STARTING EXECUTION TIME: 2016-03-11 10:13:59 (GMT)
-----------------------------------------------------------------------------------------------------
    TIME      | AREA   | LV | MESSAGE
-----------------------------------------------------------------------------------------------------
00:00:00.096 |  MNGE |  6 |       Registering core codecs
00:00:00.096 |  MNGE |  6 |       Created Codec genericCodec with name genericCodec
00:00:00.096 |  MNGE |  6 |       Registering GFI codecs
```

GFI: command-line interface options

Command line options available for the Tester

Output directory (--outputdir)

Allows to specify an output path

```
gfi@gfi:/opt/gfi$ ./gfi-testing-tool conf/gfi-scenario-emulators.xml tests/ATG-TEST-ECS40.xml --outputdir MY_TEST_LOCATION
Found 1◆◆ argument. Searching for file [conf/gfi-scenario-emulators.xml].
Found 2◆◆ argument. Searching for file [tests/ATG-TEST-ECS40.xml].


-----------------------------------------------------------------------------------------------------
Initializing the Software Database (SDB)...
Starting GFI Execution...
-----------------------------------------------------------------------------------------------------
 STARTING EXECUTION TIME: 2016-03-11 10:15:11 (GMT)
-----------------------------------------------------------------------------------------------------
    TIME      | AREA   | LV | MESSAGE
-----------------------------------------------------------------------------------------------------
00:00:00.120 |   MNGE |   0 | Preparing Test Execution...
00:00:00.120 | RPORT |   4 |      Report location: /opt/gfi/MY_TEST_LOCATION/test-report.html
00:00:00.122 |    ACB |   4 |      ACBEmulator: Start processing thread for messages received from xSME
00:00:00.122 |    ACB |   4 |      ACBEmulator: Waiting for new messages from xSME
```

COMMON ISSUES

Common Issues (I)

1. **Tests created with previous versions of the tool fail.**
   - The test needs to be recreated with current version of tool.

1. **How to add a new remote party?**
   - Define the remote party in the scenario file (using XML).

3. **Pre-defined values in the use-case library are not always correct.**
   - Verify use-case parameters are within valid range.

4. **GFI appears to hang (live lock) during test.**
   - If test involves receiving alerts, and these are future dated.
   - If MAC invalid, then message is discarded (see --dcrypto).
     - Sometimes wrong certificates are the cause.
   - Inspecting the log file will provide more information.

Common Issues (II)

5. **CH does not start, and returns error code 11 in GUI**
   a) Ensure ZigBee dongle is attached to correct port: e.g. /dev/ttyUSB0
      - Port can be check with updater tool.
   b) Ensure ZigBee dongle EUI64 aligns with GPF ID
      - EUI64 can be read with updater tool.
   c) Inspect the CH log file for error cause.

6. **ZigBee device does not associate with GFI**
   - Verify install code and EUI64 specified in the CH Device Log.
   - Investigate with ZigBee sniffer
   - If GSME, ensure joining process is aligned with DCC guidance notes.
     a) Physical Environment == 0x01
     b) Identify Price Cluster where Commodity Type == 0x01

Common Issues (III)

7. **Fatal error or "message limit reached" error encountered**
   - These are transient issues, restart the CH daemon and try again.
   - If the problem is reproducible, please contact GFI Support.
   - Fatal errors are more likely when running on a virtual machine.

8. **What CBKE Crypto Suite GFI supports?**
   - GFI supports CBKE Crypto Suite 2

9. **Is Partner Link Key Exchange Supported?**
   - Yes in GFI 1.3.2E
   - Its enablement is configurable in GFI 1.3.3E

Common Issues (IV)

10. Can I integrate GFI into a wider test framework?

- Yes, but not directly supported
  a) Use of the command line application
  b) Use CH control JSON interface
  c) Using CM tools such as SVN (or shared folder on network)
     - Please don't interfere with gfi directory.

11. Can I install GFI on a Virtual Machine?

- Yes, but this is not officially supported.
  - Issues regarding ZigBee stability reported.
- Its easiest to use the ISO file instead of USB pen.

12. Can I install GFI on a partition in my personal computer?

- No. GFI installer wipes the hard drive/SSD.

Common Issues (V)

13. **Where are the certificates and keys used by GFI?**
    - Certificates and keys are installed to: `/opt/gfi/conf/smki`

14. **How long will GFI wait for a message before failing?**
    - 1 Hour

15. **How can I change the default timeout (useful for automating testing)**
    - In the scenario file there is a tag "`timeout_messages`"
    - Change the value for this for the ACB and Remote Party Emulators
    - Value is in milliseconds

Further Information

Available from (Search for: “dcc gfi support”):
https://www.smartdcc.co.uk/implementation/test-assurance/gbcs-integration-testing-for-industry/

- Installation Guide
- User Manual
- Technical Reference Manual

Further support is available from: GFISupport@smartdcc.co.uk

SUMMARY AND QUESTIONS

Summary

Topics covered during training:

- What is Enhanced GFI
- How to use the Enhanced GFI GUI
  - Emulated
  - Basic Use-Case Execution: ECS39a
- Introduction to the command line interface
  - Communications Hub
  - Testing Tool
- Common Issues

THANK YOU