How to Build And Run OIC Conformance Test Tool

Platform:

Operating System: Ubuntu 12.04 or higher

Architecture: 32 bit

Prerequisites

1. Java 1.8 or higher(To install jdk 1.8, run the following:)

   ```
   $ sudo add-apt-repository ppa:webupd8team/java
   $ sudo apt-get update
   $ sudo apt-get install oracle-java8-installer
   ```

2. Gradle 2.3 or higher(To install Gradle on linux, run the following:)

   ```
   $ sudo add-apt-repository ppa:cwchien/gradle
   $ sudo apt-get update
   $ sudo apt-get install gradle
   ```

3. Jython 2.7 or higher(To install Jython, follow these steps:)

   - Go to the following Link and Download Jython 2.7.0 installer

   - http://www.jython.org/downloads.html

   - After download is complete, run the jar file using java-8

   - Follow the installation procedure.

4. Robot Framework (To install Robot Framework, follow these steps:)

   - Go to the following link and download zip

   - https://github.com/robotframework/robotframework

   - Extract the zip file

   - Go to the root folder of robot framework, open terminal and run:

     ```
     $ python setup.py install
     ```

5. Iotivity project must be built for linux(Using in DUT Simulator)

   a. Go to the top(root) directory of 'iotivity' project.

b. Follow the prerequisite steps described for IoTivity project.

c. Run the below command (for non-secured resource):

```
$ scons TARGET_OS=linux
```

(Note: C sdk requires tinycbor. Please follow the instruction in the build message to install tinycbor).

d. To build IoTivity using secured resource option, run the below command:

```
$ scons TARGET_OS=linux SECURED=1
```

6. TShark(To install tshark, follow below steps:)

```
$ sudo apt-get install tshark
```

7. Eclipse Luna

- Download Eclipse Luna RCP from:

  http://www.eclipse.org/downloads/download.php?
  file=/technology/epp/downloads/release/luna/SR2/eclipse-rcp-luna-SR2-linux-gtk.tar.gz

- Extract it

- Start Eclipse

- Click on Help->Install New Software..

- Under "work with:", select 'Luna -
  http://download.eclipse.org/releases/luna'

- Wait for the packages to load, then select everything under 'Linux Tools'

- Click Next and proceed with Installation.

- Again click on Help->Install New Software..

- Under "work with:", click on Add

- Provide Name as 'RobotFramework', and Location as
  'http://sourceforge.net/projects/robotide/files/stable/'

- Click Ok

- Now from package list, select 'Robot Framework Eclipse IDE'

- Click on Next and proceed with installation.

- Restart Eclipse

Build Instructions

1. Open Eclipse Luna RCP

- Import Project from git 'conformance-test-tool/res/ConformanceTestTool/'

- Build the project

- Now, Right click on the project, then click Export.

- Under 'Plug-in Development', select Eclipse product, click on Next.

- As Root Directory, type 'ConformanceTestTool'

- As Destination Directory, show 'conformance-test-tool/res/ConformanceTestTool/bin/'
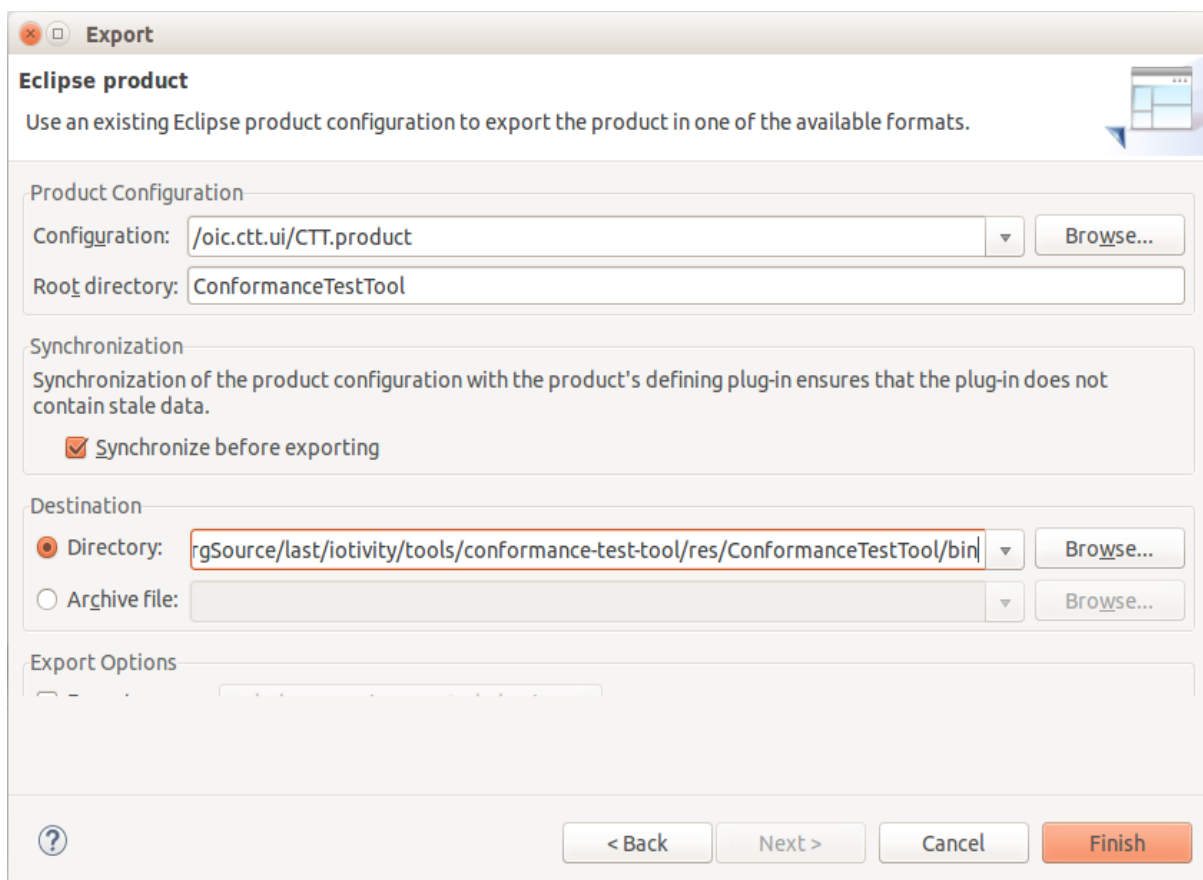

Figure: Export Settings Window (Root & Destination Directory must be same as described)

- Click on Finish.

2. Go to the root folder (`iotivity/tools/conformance-test-tool`)

3. From the terminal, run:

```
$ sudo chmod 777 build.sh
$ ./build.sh
```

How to run Conformance TC

1. Start the DUT device to test (or just ConformanceSimulator)

2. Create a description of device/resource representation data structure in json format. An example is provided in the same directory of this documentation. Name the file as 'DUTDescriptor.json'

3. Copy the file to `<iotivity_root>/tools/conformance-test-tool/bin/linux/ConformanceTestTool/libs/` folder. (When using the ConformanceSimulator as DUT, no need to make any DUTDescriptor.json)

4. Go to the root folder of Conformance Test Tool

5. Then Run:

```
$ cd bin/linux/ConformanceTestTool
```

6. To run the GUI App, execute the following:

```
$ ./CTT
```

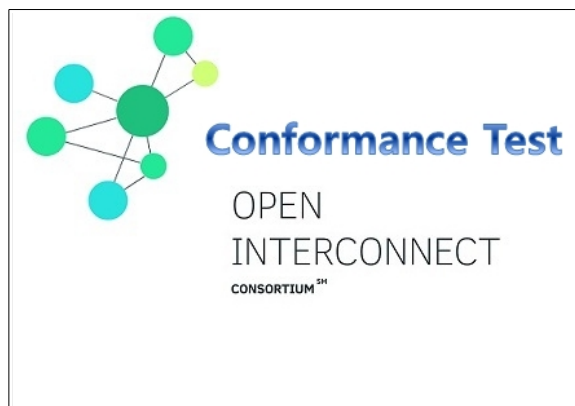7. To run a TC file in command line, execute ./tc_run.sh testsuite/<TC_File_Name>, e.g.

```
$ ./run_tc.sh
testsuiteCT1_1_1_CoAP_Based_Multicast_Endpoint_Discovery-
No_Query.txt
```

8. The result report will be saved at `< path_to_conformance_tool>/bin/linux/ConformanceTestTool/testreport/<timestamp>/`

How to use the GUI

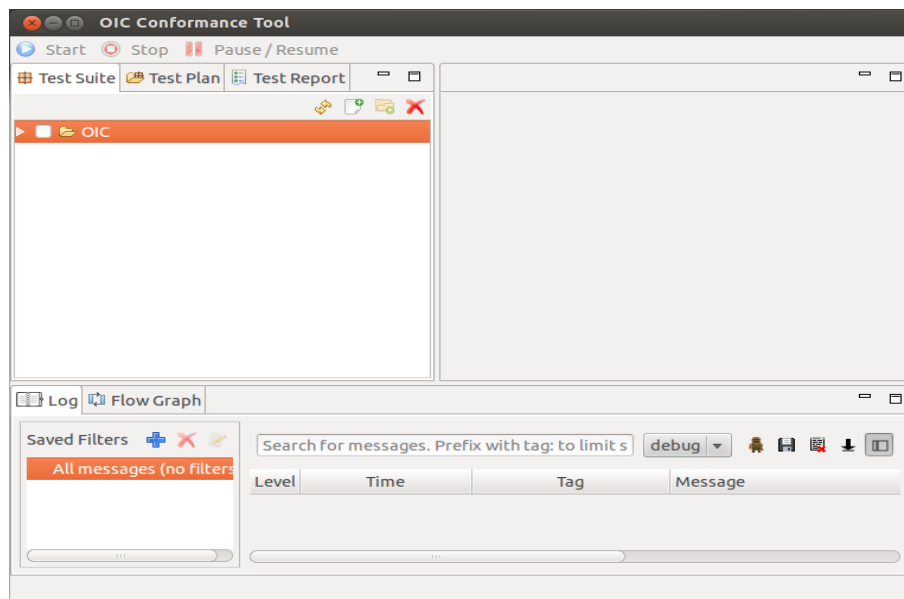The GUI will start like this:
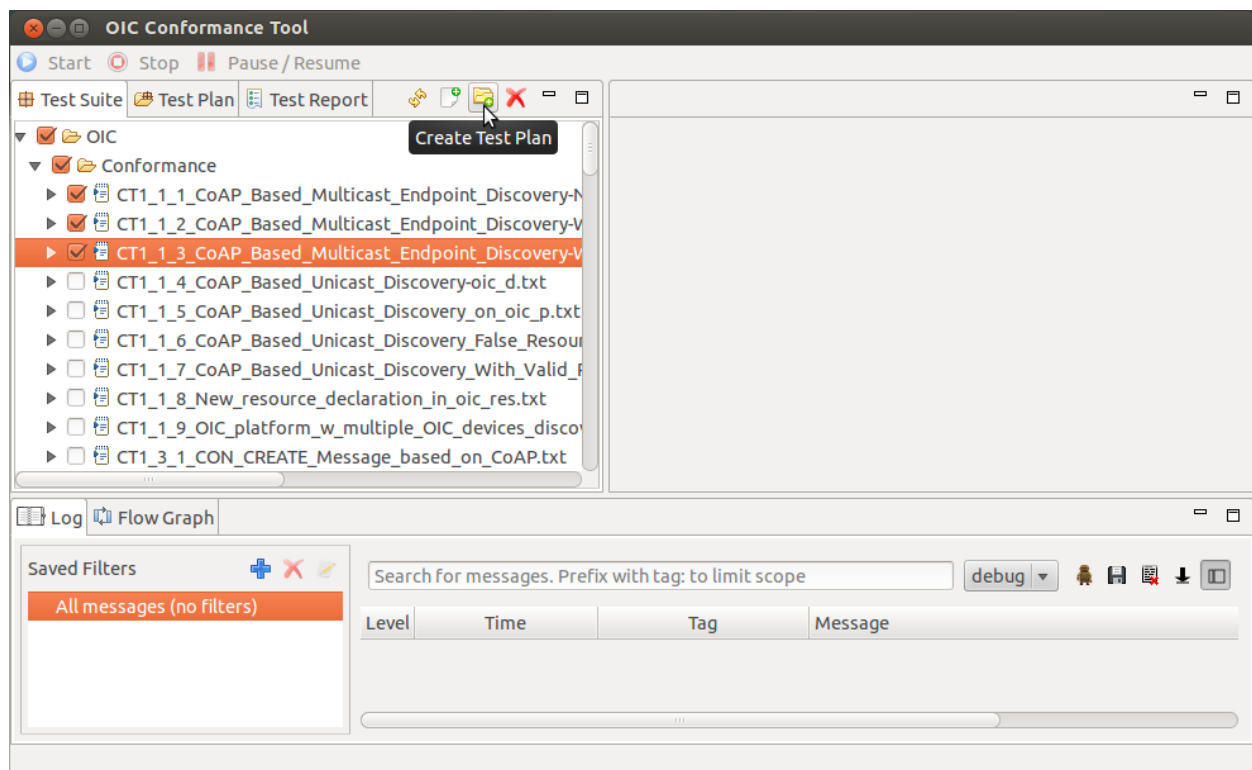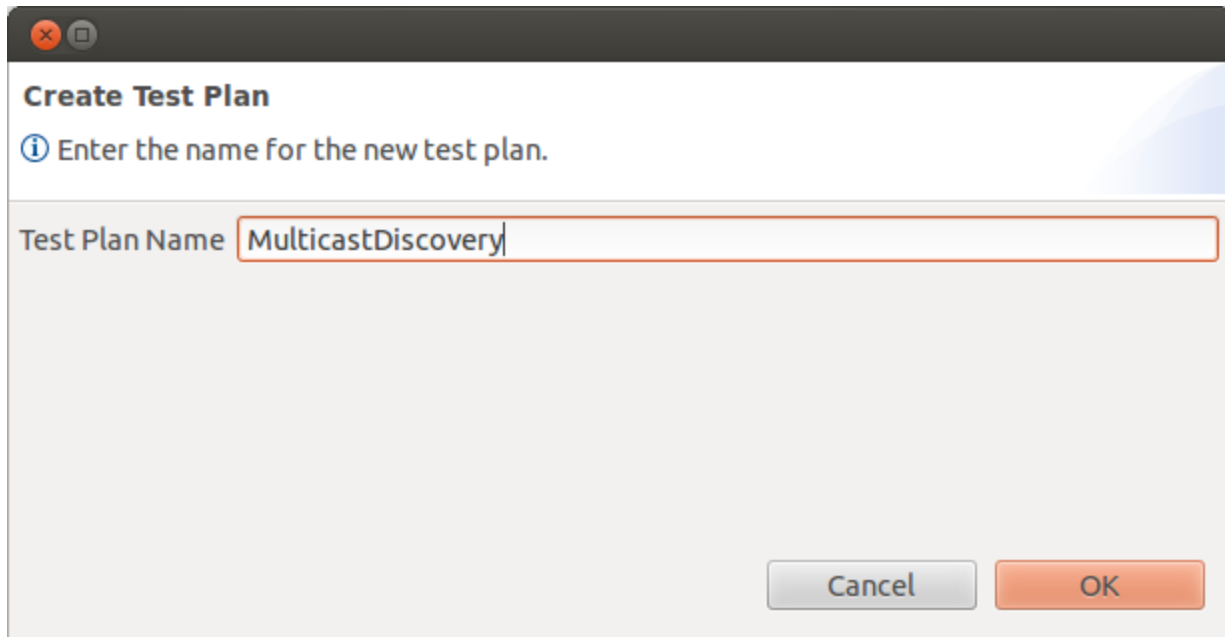
Figure: Splash Screen

Figure: Conformance Tool Graphical User Interface (GUI)
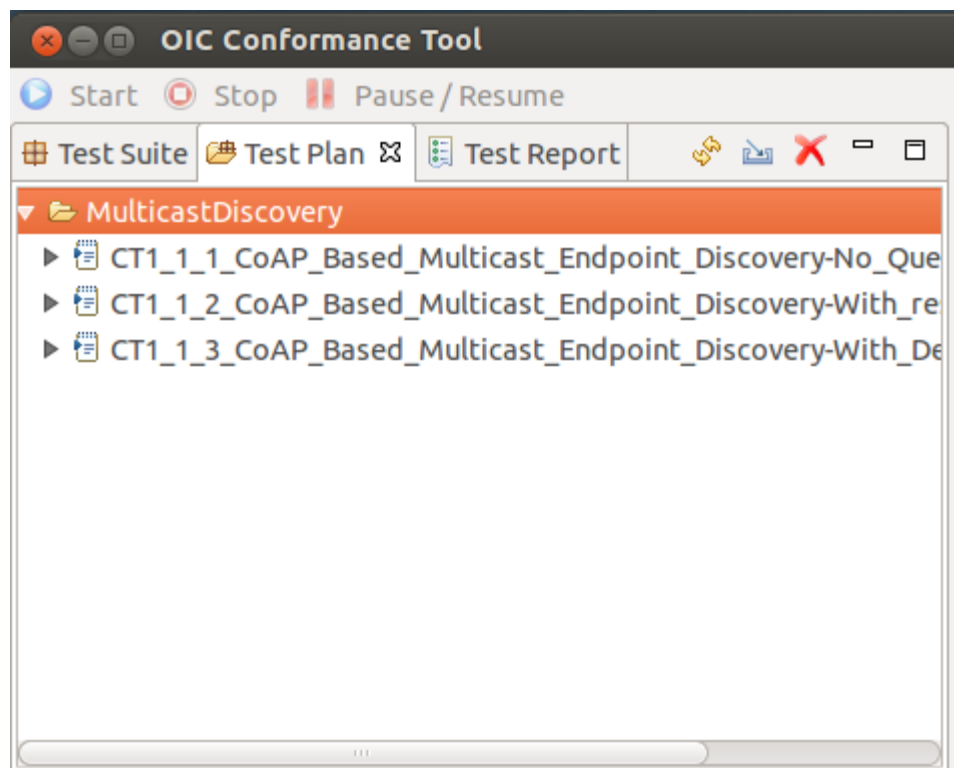
How to make a Test plan

- Select your desired test cases in the **Test Suite** tab. Click Create Test Plan

- A **Create Test Plan** dialogue box will appear. Enter a name for it and click OK.

- A test plan will be created. The created test plan will be visible in the **Test Plan** tab.

- Double click on the test plan to select it.
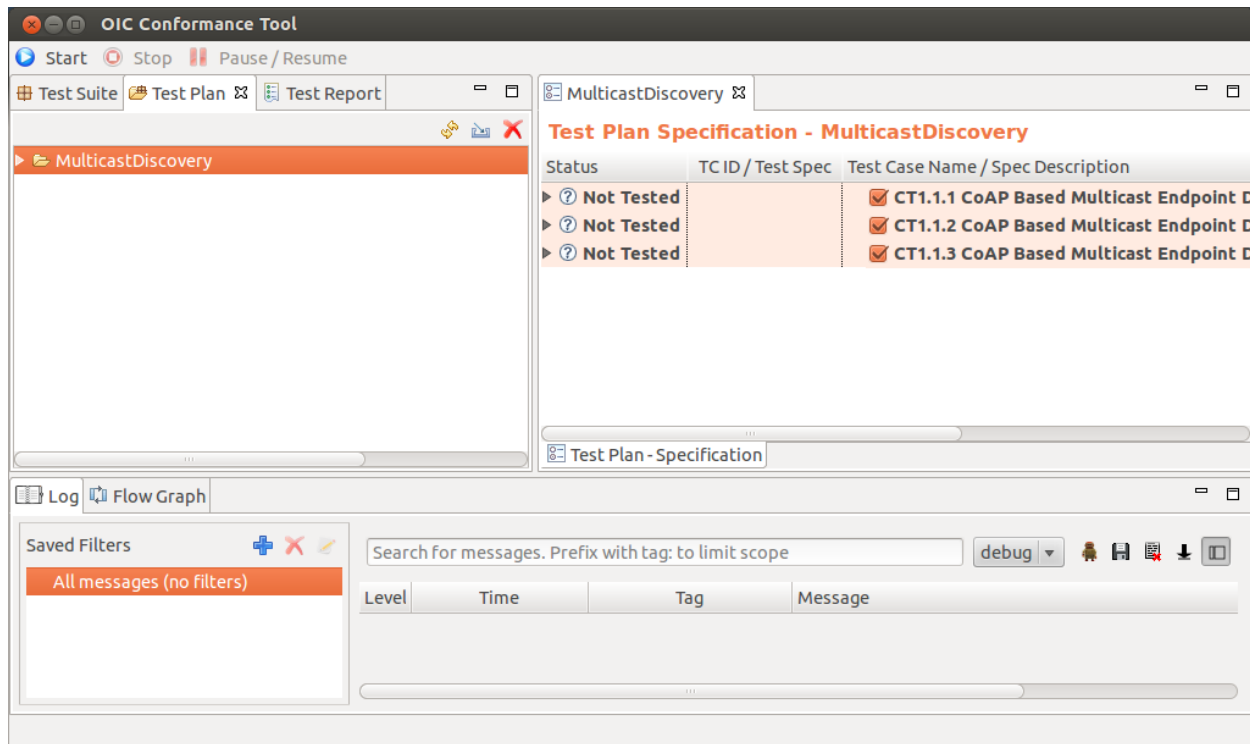
How to run the Test plan

- Once the test plan is selected, click the **Start** button to start the test.
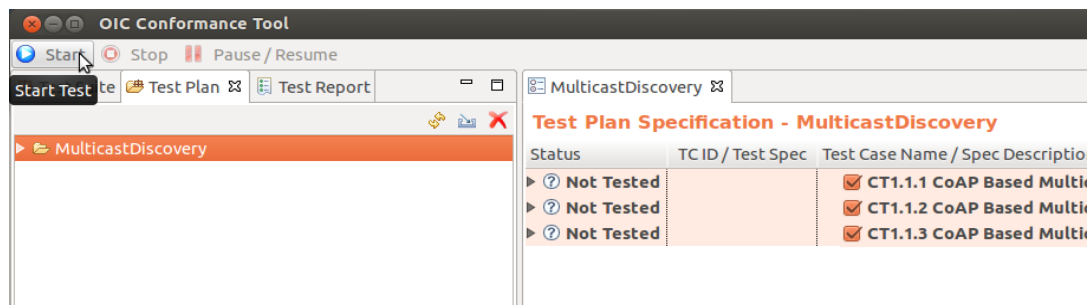
- When the test is running, you can **Stop** or **Pause**/**Resume** the execution. The Flow Graph, Progress & Log windows will show real-time test execution information.
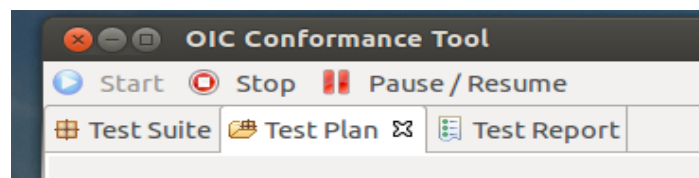
How to view & interpret Reports

The **Log** tab will show all the detailed logs generated by the test cases.
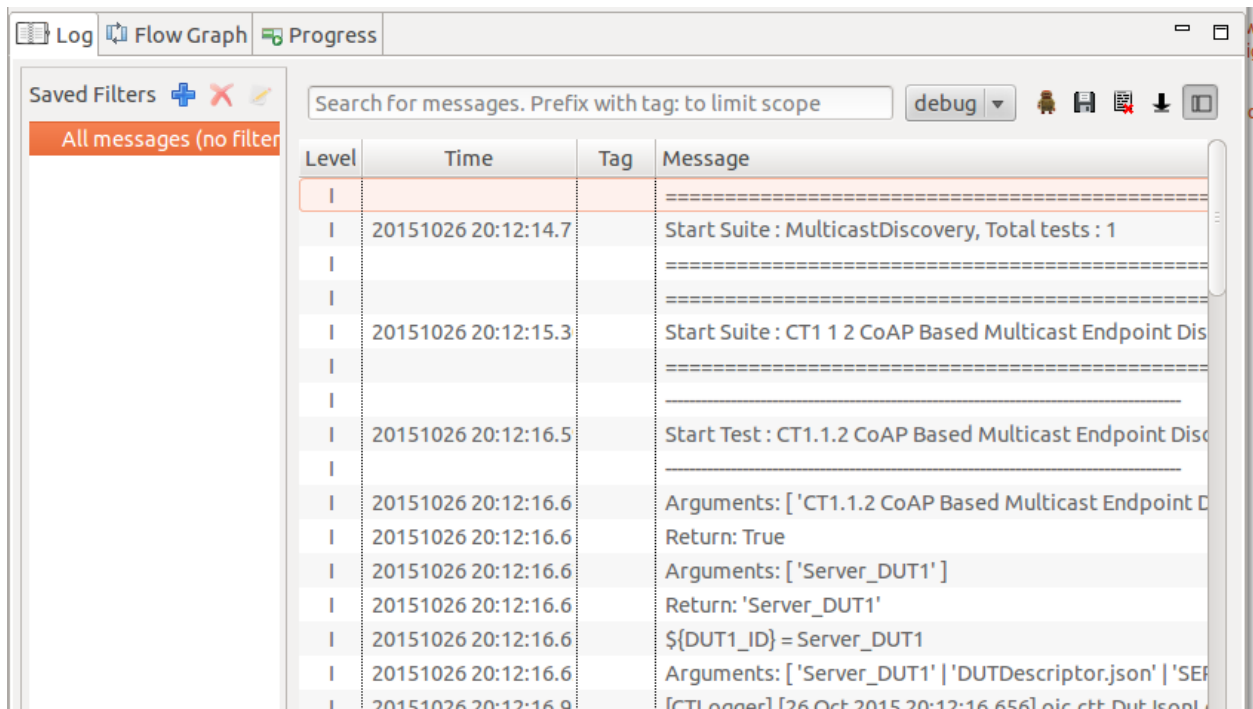
When the test execution is finished, a **Test Report** window will open with a complete report on the test plan.
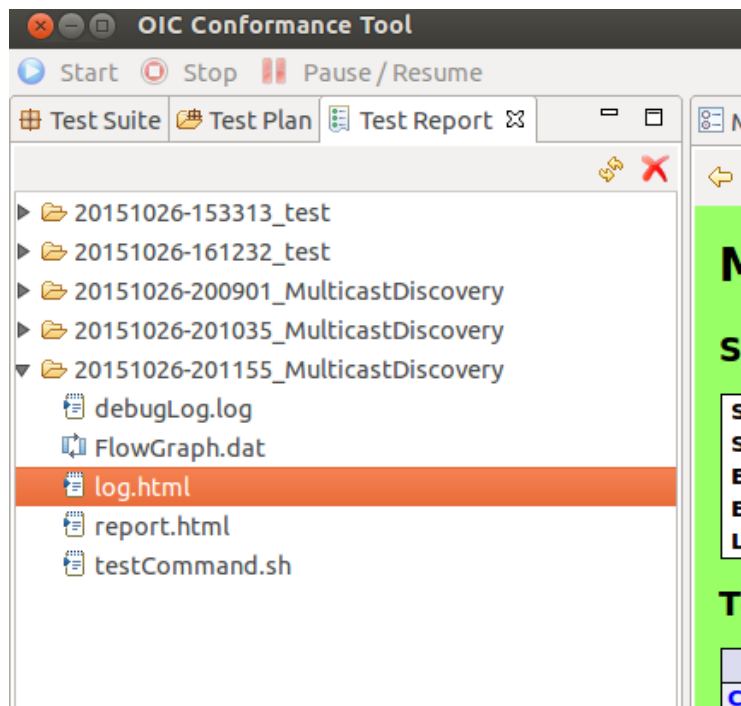
The reports will be saved in the storage drive. You can view all the generated test reports from the **Test Report** tab.

How to run Conformance TC in CLI(command line interface)

1. Go to the root folder of Conformance Tool

2. Then Run:

    ```
    $ cd bin/linux/ConformanceTestTool/ConformanceTool/
    testsuite/OIC/Conformance/
    ```

3. To run a TC file, execute ./tc_run.sh <TC_File_Name>, e.g.

    ```
    $ ./run_tc.sh
    testsuite/CT1_1_1_CoAP_Based_Multicast_Endpoint_Discovery-
    No_Query.txt
    ```
4. The result report will be saved at < path_to_conformance_tool>/bin/linux/ ConformanceTestTool/ConformanceTool/testreport/<timestamp>/

```
●─□   antu@antu-VirtualBox: ~/git/oictest_repo/IotivityOrgSource/ctt/iotivity/tools/conformance-test-tool/bin/lin

===============================================================================
CT1.1.2 CoAP Based Multicast Endpoint Discovery - With resource ty...  .....
TE -------->        Multicast Discover with query        --------> DUT
    Query : rt=core.light

TE <--------            Response for Discover            <-------- DUT
    Response Code : 2.05
    Source Address : 10.0.2.15:46362
    payload[{"di":"1Qi6GcBmRWmpF1nUXizXmg==","links":[{"href":"/device/light-1","rt":"core.light
","if":"oic.if.baseline","p":{"bm":3}}]}]


TE <--------            Response for Discover            <-------- DUT
    Response Code : 2.05
    Source Address : fe80:0:0:0:a00:27ff:fe85:b783%2:46881
    payload[{"di":"1Qi6GcBmRWmpF1nUXizXmg==","links":[{"href":"/device/light-1","rt":"core.light
","if":"oic.if.baseline","p":{"bm":3}}]}]


TE -------->        Multicast Discover with query        --------> DUT
    Query : rt=core.fan

TE <--------            Response for Discover            <-------- DUT
    Response Code : 2.05
    Source Address : fe80:0:0:0:a00:27ff:fe85:b783%2:46881
    payload[{"di":"1Qi6GcBmRWmpF1nUXizXmg==","links":[{"href":"/device/fan-1","rt":"core.fan","i
f":"oic.if.baseline","p":{"bm":1}}]}]


TE <--------            Response for Discover            <-------- DUT
    Response Code : 2.05
    Source Address : 10.0.2.15:46362
    payload[{"di":"1Qi6GcBmRWmpF1nUXizXmg==","links":[{"href":"/device/fan-1","rt":"core.fan","i
f":"oic.if.baseline","p":{"bm":1}}]}]

..
TE -------->        Multicast Discover with query        --------> DUT
    Query : rt=core.dummydummy
log4j:WARN No appenders could be found for logger (CTLogger).
log4j:WARN Please initialize the log4j system properly.
log4j:WARN See http://logging.apache.org/log4j/1.2/faq.html#noconfig for more info.

CT1.1.2 CoAP Based Multicast Endpoint Discovery - With resource ty... | PASS |
-------------------------------------------------------------------------------
CT1 1 2 CoAP Based Multicast Endpoint Discovery-With resource type... | PASS |
1 critical test, 1 passed, 0 failed
1 test total, 1 passed, 0 failed
===============================================================================
```

Figure : Execution of a Test Case using Command Line Interface of ConformanceTestTool
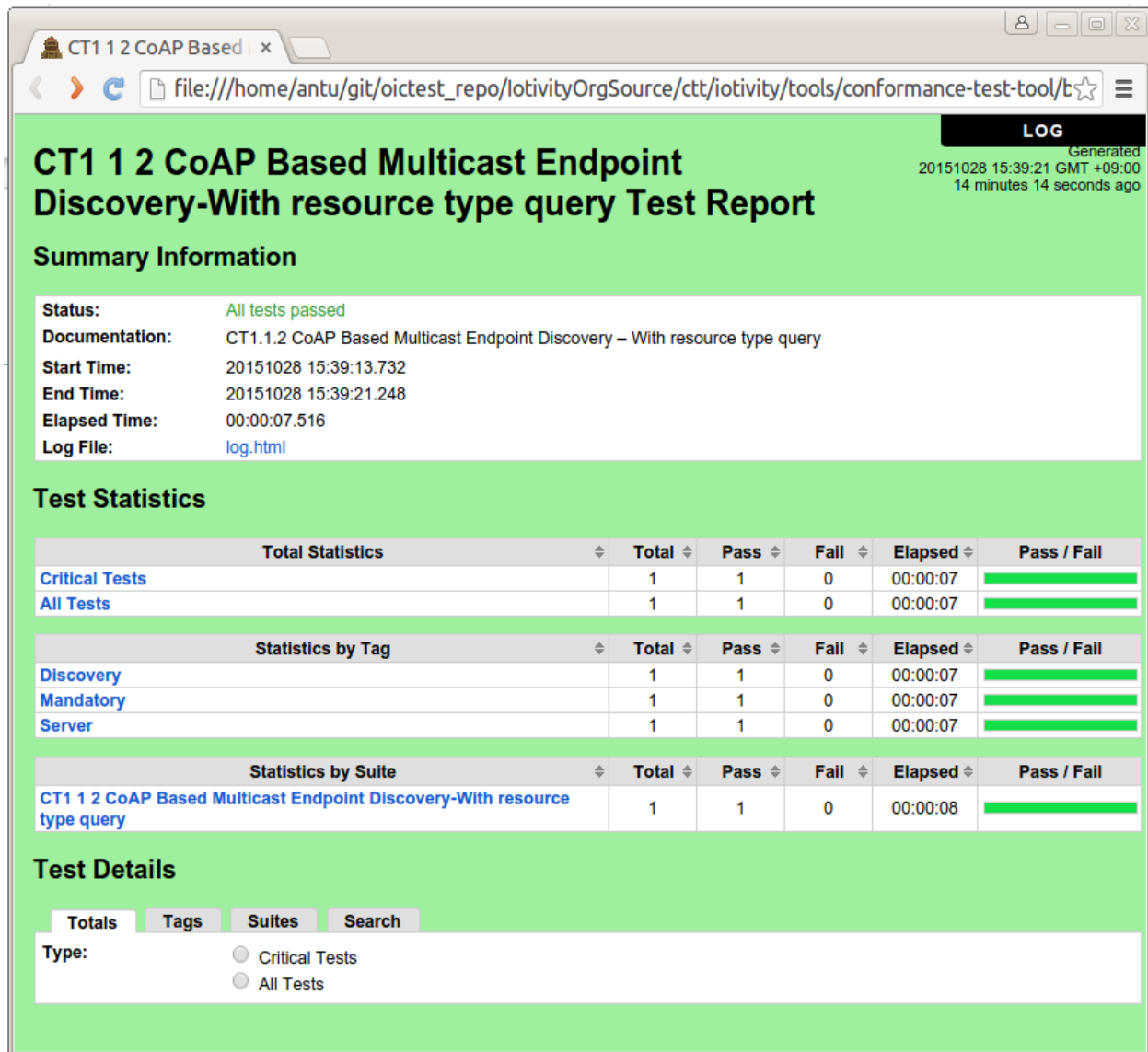
Figure : View of Conformance Test Tool TC Execution Result

How to Build And Run ConformanceSimulator

ConformanceSimulator

ConformanceSimulator is an OIC client-server simulator which supports the basic client and server features mentioned in the OIC Core Spec. It is based upon IoTivity implementation of OIC specification. The server part supports discoverable, non-discoverable, observable, non-observable, secured and non-secured resources. Both the server and client supports all CRUDN operations. Resource, Platform, Device and Collection discovery is supported for the client. The client supports both CON & NON type messaging.

ConformanceSimulator is build automatically with ConfonformanceTestTool project. It is a command line application which performs operation by option menu selection.

Pre-requisite for ConformanceSimulator

1. scons is required to build ConformanceSimulator

2. ConformanceSimulator is based upon Iotivity. So Iotivity project must be built for linux

   a. Go to the top(root) directory of 'iotivity' project.

   b. Follow the pre-requisite steps described ifor IoTivity project.

   c. Run the below command (for non-secured resource):

      ```
      $ scons TARGET_OS=linux
      ```

      (Note: C sdk requires tinycbor. Please follow the instruction in the build message to install tinycbor).

   d. To build Iotivity using Secured resource option, run the below command:

      ```
      $ scons TARGET_OS=linux SECURED=1
      ```

How to build ConformanceSimulator

1. Go to the top(root) directory of 'ConformanceSimulator' project.

2. Run the below command to build Conformance simulator

   ```
   $ scons
   ```

3. Help:

```
$ scons -h
```

4. Clear:

```
$ scons -c
```

N.B. : ConformanceSimulator is automatically build during ConformanceTestTool build, so no need to perform these steps when ConformanceTestTool project is previously built.

How to run ConformanceSimulator

1. From the top(root) directory of ConformanceSimulator, go to linux binary folder:

```
$ cd bin/linux/
```

2. Append the library location of iotivity to system library path:

```
$ export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:<path-to-iotivity-
root>/out/linux/x86/release/
```

3. Alternatively, for 64 bit operating system:

```
$ export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:<path-to-iotivity-
root>/out/linux/x86_64/release/
```

4. To run ConformanceSimulator using default settings(Non-conformable, IPv6, non-secure server):

```
$ ./ConformanceSimulator
```

5. General command to run ConformanceSimulator:

```
$ ./ConformanceSimulator [QoS<0/1>] [IP_Version<4/6>]
[Security<0/1/2>]
```

QoS: 0 = NON, 1 = CON

IP_Version: 4 = IPv4, 6 = IPv6

Security: 0 = non-secure, 1 = Secure Client, 2 = Secure Server

e.g.,to Run IpV4 non-secure NON-type server:

```
$ ./ConformancSiulator 0 4 0
```

6. Alternatively,

```
$ ./ConformancSiulator 0 4
```

7. To run IpV6 secure CON-type client:\

```
$ ./ConformancSiulator 1 6 1
```

8. To run IpV6 secure CON server:

```
$ ./ConformancSiulator 1 6 2
```


터미널

```
              20. Send POST Request - Partial Update
              21. Send POST Request - Create Sub-Ordinate Resource
              22. Send Delete Request
              23. Observe Resource - Retrive Request with Observe
              24. Cancel Observing Resource
              25. Cancel Observing Resource Passively
              26. Discover Device - Unicast
              27. Discover Device - Multicast
              28. Discover Platform - Unicast
              29. Find Group
              30. Join Found Resource To Found Group
              31. Quit Conformance Simulator App
1
createResource called!!
Current resource info:
Server Started
Inside onResourceServerStarted...
Resource created successfully
Current resource info:
Server Started
Inside onResourceServerStarted...
Resource created successfully
Please Select an option from the menu and press Enter
         Server Operations:
              1. Create Normal Resource
              2. Create Invisible Resource
              3. Create Resource With Complete URL
              4. Create Secured Resource
              5. Create 100 Light Resources
              6. Create Group Resource
              7. Delete All Resources
              8. Delete Created Group
         Client Operations:
              10. Find core.light Type Resource
              11. Find Specific Type Of Resource
              12. Find All Resources
              13. Find core.light Type Resource - Unicast
              14. Find Specific Type Of Resource - Unicast
              15. Find All Resources - Unicast
              16. Join Found Resource To The Group
              17. Send GET Request
              18. Send PUT Request - Create Resource
              19. Send PUT Request - Complete Update
              20. Send POST Request - Partial Update
              21. Send POST Request - Create Sub-Ordinate Resource
              22. Send Delete Request
              23. Observe Resource - Retrive Request with Observe
              24. Cancel Observing Resource
              25. Cancel Observing Resource Passively
              26. Discover Device - Unicast
              27. Discover Device - Multicast
              28. Discover Platform - Unicast
              29. Find Group
              30. Join Found Resource To Found Group
              31. Quit Conformance Simulator App
```

Figure: Conformance Simulator Execution