

RUN IOTIVITY

(For 16.04 LTS Ubuntu Version x64
architecture)

Contents

DOWNLOAD IoTIVITY AND SET UP SYSTEM CONFIGURATIONS	3
MAKE CHANGES TO FILES FOR BUILDING IOTIVITY	4
BUILD AND RUN IOTIVITY.....	5
RUN EXAMPLES	6

DOWNLOAD IoTIVITY AND SET UP SYSTEM CONFIGURATIONS

- Install Build essentials

```
sudo apt-get install build-essential git scons libtool autoconf valgrind doxygen wget  
unzip
```

- Install External Libraries

```
sudo apt-get install libboost-dev libboost-program-options-dev libboost-thread-dev uuid-  
dev libexpat1-dev libglib2.0-dev libsqlite3-dev libcurl4-gnutls-dev
```

- Download 1.3-rel branch from the following github location

<https://github.com/iotivity/iotivity>

MAKE CHANGES TO FILES FOR BUILDING IOTIVITY

- Only for Ubuntu x64 architecture,
 - Navigate to `~/iotivity/resource/csdk/stack/test`
 - Open `stacktest.cpp`
 - Change the `SHORT_TEST_TIMEOUT` from 5 second to 55 second

BUILD AND RUN IOTIVITY

- Navigate to iotivity home directory. Run 'scons' command on the terminal. (It will take around 5-10 mins to finish building).

NOTE: There are various options with which scons can be used. See following for various options: https://wiki.iotivity.org/build_iotivity_with_ubuntu_build_machine

- To run test cases, run following command
scons TEST = 1

NOTE: If you run into any problem such as this one,

```
[-----] 9 tests from PrimitiveResourceTest
RUN      OK PrimitiveResourceTest.RequestGetInvokesOCResourceGet (0 ms)
RUN      OK PrimitiveResourceTest.RequestGetThrowsOCResourceGetReturnsNotOK (0 ms)
RUN      OK PrimitiveResourceTest.RequestGetThrowsOCResourceGetReturnsNotOK (0 ms)
RUN      OK PrimitiveResourceTest.RequestSetInvokesOCResourcePost (0 ms)
RUN      OK PrimitiveResourceTest.RequestSetInvokesOCResourcePost (0 ms)
RUN      OK PrimitiveResourceTest.RequestSetThrowsOCResourcePostReturnsNotOK (0 ms)
RUN      OK PrimitiveResourceTest.RequestSetThrowsOCResourcePostReturnsNotOK (0 ms)
RUN      OK PrimitiveResourceTest.RequestSetPassResourceAttributesToOCResourcePost (0 ms)
RUN      OK PrimitiveResourceTest.RequestSetPassResourceAttributesToOCResourcePost (0 ms)
RUN      OK PrimitiveResourceTest.RequestObserveInvokesOCResourceObserve (0 ms)
RUN      OK PrimitiveResourceTest.RequestObserveInvokesOCResourceObserve (0 ms)
RUN      OK PrimitiveResourceTest.RequestObserveThrowsOCResourceObserveReturnsNotOK (0 ms)
RUN      OK PrimitiveResourceTest.RequestObserveThrowsOCResourceObserveReturnsNotOK (0 ms)
RUN      OK PrimitiveResourceTest.DelegateGettersToOCResource (0 ms)
RUN      OK PrimitiveResourceTest.DelegateGettersToOCResource (0 ms)
RUN      OK PrimitiveResourceTest.ResponseStatementHasSameValuesWithOCRepresentationReceived
RUN      OK PrimitiveResourceTest.ResponseStatementHasSameValuesWithOCRepresentationReceived (0 ms)
[-----] 9 tests from PrimitiveResourceTest (0 ms total)

[-----] 2 tests from DiscoverResourceTest
RUN      DiscoverResourceTest.CallbackIsInvokedWhenResourceIsDiscovered
Illegal instruction (core dumped)
scons: *** [out/linux/x86_64/release/service/resource-encapsulation/src/common/utservice/resource-encapsulation/src/common/rcs_common_test] Error 132
scons: building terminated because of errors.
root@mbacl9470:/home/mbachant/iotivity-1.3-rel#
```

Solution to it is following:

- Comment the following lines in the specified file
 - **File location:** ~/iotivity/service/resource-encapsulation/src/common/SConscript
 - **Lines to be commented:** 129-136
- scons -c
- scons
- scons TEST = 1

Note: Navigate to 'out/linux/x64_x86/release' directory in iotivity folder. There will be '.a' and '.so' files. These are shared libraries. ('.so' files corresponds to shared object and '.a' files are archived objects).

- export LD_LIBRARY_PATH=<iotivity>/out/linux/x86_64/release
- C++ example are in <iotivity>/out/linux/x86_64/release/resource/examples
- C examples are in
 <iotivity>/out/linux/x86_64/release/resource/csdk/stack/samples/linux/SimpleClientServer.
- Run basic client and server example

```
cd
${project_dir}/out/${TARGET_OS}/${TARGET_ARCH}/${BUILD_MODE}/resource/examples/

killall simpleserver simpleclient # make sure none are running

./simpleserver 2>&1 | tee simpleserver.log.txt &

./simpleclient 2>&1 | tee simpleclient.log.txt

killall simpleserver # since simpleclient is over let's stop server too
```

- Run specific server and client
 - Navigate to <iotivity>/out/resources/examples
 - Following examples will be present

```

devicediscoveryclient.o      oic_svr_db_server.dat
devicediscoveryclient.o      presenceclient
devicediscoveryserver        presenceclient.o
devicediscoveryserver.o      presenceserver
directpairingclient          presenceserver.o
directpairingclient.o        rdclient
Fridgeclient                 rdclient.o
Fridgeclient.o               roomclient
Fridgeserver                 roomclient.o
Fridgeserver.o               roomserver
garageclient                 roomserver.o
garageclient.o               simpleclient
garageserver                 simpleclientHQ
garageserver.o               simpleclientHQ.o
groupclient                  simpleclient.o
groupclient.o                simpleclientserver
groupserver                  simpleclientserver.o
groupserver.o                simpleserver
light_introspection.json     simpleserverHQ
lightserver                  simpleserverHQ.o
lightserver.o                simpleserver.o
oic_svr_db_client.dat        threadingsample
oic_svr_db_client_directpairing.dat threadingsample.o

```

Run any pair of server and client.

`./fridgeclient` and `./fridge server` on separate terminals.

➤ Run security provisioning example:

Provisioning Client test steps:

1. \$ cd <out>/resource/csdk/security/provisioning/sample
2. \$./sampleserver_justworks (optionally, "\$./sampleserver_justworks |& tee ss-jw.log" to capture output for later study)
3. <open a second console in same dir>
4. \$./provisioningclient (using | tee not suggested as it tends to make menu unusable, but once the process is memorized, you can then use | tee to capture client-side logs)
5. at provisioningclient app menu, choose "10" <enter>
6. after discovery completes, one device should appear in "un-owned" list and the menu is re-displayed
7. at provisioningclient app menu, choose "20" <enter>
8. after doing JustWorks OTM completes and the menu is re-displayed
9. at provisioningclient app menu, choose "10" <enter>
10. after discovery, same deviceuuid as step 6. should appear but now in "owned" device list

Also you can run the security unit tests:

Security Unit Tests:

1. \$ cd <out>/resource/csdk/security/unittest
2. \$./unittest

On successful build, following message will be seen:

```
[-----] 27 tests from remoteresourceobjecttest (2903 ms total)
[-----] Global test environment tear-down
[=====] 33 tests from 2 test cases ran. (4926 ms total)
[  PASSED  ] 33 tests.

YOU HAVE 2 DISABLED TESTS

scons: done building targets.
root@osboxes:/home/osboxes/Downloads/iotivity-1.3-rel#
```