

Apache NimBLE Test Plan

The following test plan should be executed against an Apache Mynewt release candidate before a vote is called. This is a very minimal test plan and is a work in progress

- [Units under test](#)

[bleprph](#)

- [Steps](#)
- [Testing Notes](#)

[bttester](#)

[btshell](#)

[blehci](#)

[Bluetooth 5 features](#)

[Over-the-Air Image Upgrade Over BLE](#)

- [Steps](#)

[Bluetooth Mesh](#)

- [Steps](#)
- [Test with Android](#)

[Bluetooth Qualification](#)

- [Ensure that host is passing all relevant PTS \(latest available version\) tests.](#)
- [Ensure that controller is passing all relevant Harmony tests.](#)

Units under test

- nordic_pca10028 (optional if application doesn't fit)
- nordic_pca10040
- nordic_pca10056
- nordic_pca10095
- dialog_da1469x-dk-pro

TODO: provide targets for testing in repo

bleprph

Steps

- *[Peer devices: iPhone/LightBlue, OSX/LightBlue, Galaxy/NRFConnect]*
- Central connects and disconnects several times; ensure:
 - bleprph does not crash
 - bleprph immediately resumes advertising after disconnect
- Central initiates pairing
- Central restores encrypted connection via bonding

Targets

Here are the targets that were used. See the testing notes below for BLE_SM_SC. To test legacy pairing this value should be 0.

```
targets/nordic_pca10028_bleprph
  app=@apache-mynewt-nimble/apps/bleprph
  bsp=@apache-mynewt-core/hw/bsp/nordic_pca10028
  build_profile=optimized
  syscfg=BLE_SM_BONDING=1:BLE_SM_OUR_KEY_DIST=7:BLE_SM_THEIR_KEY_DIST=7:LOG_LEVEL=255
targets/nordic_pca10040_bleprph
  app=@apache-mynewt-nimble/apps/bleprph
  bsp=@apache-mynewt-core/hw/bsp/nordic_pca10040
  build_profile=optimized
  syscfg=BLE_SM_BONDING=1:BLE_SM_OUR_KEY_DIST=7:BLE_SM_SC=1:BLE_SM_THEIR_KEY_DIST=7:LOG_LEVEL=0:
OS_MAIN_STACK_SIZE=1024:BLE_MAX_CONNECTIONS=2
targets/nordic_pca10056_bleprph
  app=@apache-mynewt-nimble/apps/bleprph
  bsp=@apache-mynewt-core/hw/bsp/nordic_pca10056
  build_profile=optimized
  syscfg=BLE_SM_BONDING=1:BLE_SM_OUR_KEY_DIST=7:BLE_SM_SC=1:BLE_SM_THEIR_KEY_DIST=7:LOG_LEVEL=0:
OS_MAIN_STACK_SIZE=1024:BLE_MAX_CONNECTIONS=2
```

Testing Notes

1. To test the legacy pairing, you need to read the characteristic with the encryption required permission. The bleprph application has two services. One of the services has one characteristic, the other has two of them. The UUID with two characteristics is the one you want.
2. With OSX/Light Blue, the application does not ask you to pair. It just does it automatically. When using LightBlue on the iPhone, it asks you to pair.
3. To test the third bullet (central restores encrypted connection), after you have paired all you need to do is disconnect and reconnect. You should then be able to read the characteristic again without having to pair.
4. To test the fourth bullet item, you need to build with BLE_SM_SC: 1. Normally, you build with BLE_SM_SC: 0 for the previous tests.

bttester

Run automated test suite against supported devices with [BTPTesterCore](#) automation tool.

Currently supported devices:

- Mynewt with bttester app
- Android running [BTPTesterAndroid](#)

Targets

```
targets/nordic_pca10028_bttester
  app=@apache-mynewt-nimble/apps/bttester
  bsp=@apache-mynewt-core/hw/bsp/nordic_pca10028
  build_profile=optimized
  syscfg=BLE_MESH=0:CONSOLE_RTT=0:MSYS_1_BLOCK_COUNT=40:MSYS_1_BLOCK_SIZE=120:OS_MAIN_STACK_SIZE=320:
  SHELL_TASK=0
targets/nordic_pca10040_bttester
  app=@apache-mynewt-nimble/apps/bleprph
  bsp=@apache-mynewt-core/hw/bsp/nordic_pca10040
  build_profile=optimized
targets/nordic_pca10056_bttester
  app=@apache-mynewt-nimble/apps/bleprph
  bsp=@apache-mynewt-core/hw/bsp/nordic_pca10056
  build_profile=optimized
```

btshell

Steps

- btshell as peripheral
 - *[Peer devices: iPhone/LightBlue or nRFConnect, Linux/Bluez]*
 - Peripheral performs undirected advertising
 - Central pairs with peripheral
 - Central restores encrypted connection via bonding
 - Central reads characteristic
 - Central writes characteristic
 - Central enables indications
 - Peripheral sends indication when characteristic changes
 - Two centrals connect to peripheral
- btshell as central
 - *[Peer devices: iPhone/LightBlue or nRFConnect, Linux/Bluez]*
 - General connection establishment procedure
 - Service discovery
 - Central pairs with peripheral
 - Central restores encrypted connection via bonding
 - Central reads characteristic
 - Central writes characteristic
 - Central enables indications
 - Central connects to two peripheral

blehci

Steps

- Using Bluez as a host
 - Connect to a third-party device (either master or slave).
 - Send some data (e.g., service discovery).
 - Ensure connection stays up.

Bluetooth 5 features

Steps:

- follow btshell procedures with BT5 enabled (using legacy advertising and connections creation)
- repeat above using different PHYs combinations (1M/1M, 1M/coded, 1M/2M, coded/coded)
- advertise with large (>31 bytes) data and verify if all data is received on scanner
- repeat above for all PHYs combinations
- using blehci sample and Linux host test advertising using 2 instances (TODO add hcitool instructions)

Over-the-Air Image Upgrade Over BLE

Steps

- Build and load boot loader
- Build, create image with version 1.0.0, and load bleprph using serial connection.
- Create newtmgr conn profile for ble. Example:

```
newtmgr conn add mybleprph type=ble connstring="peer_name=nimble-bleprph"
```

- Run newtmgr image list command to verify the connection and the image is loaded. Example:

```
newtmgr image list -c mybleprph
```

- Run newt create-image command to create bleprph image with version 2.0.0
- Run: newtmgr image upload command to ensure the new image is uploaded. Example:

```
newtmgr image upload -c mybleprph ~/dev/myproj/bin/targets/myperiph/app/apps/bleprph/bleprph.img
```

- Run newtmgr image list command to verify the image is uploaded.

Bluetooth Mesh

Steps

- configure and flash blemesh app
- connect to provisioner and provision the device
- test on/off model

Test with Android

- download and install Bluetooth Mesh by Silicon Labs app from Play Store on a phone
- configure and flash blemesh app
- in app scan for the device
- connect to the device, enter authentication code
- set model for the device to on/off model
- test on/off model

Bluetooth Qualification

Ensure that host is passing all relevant PTS (latest available version) tests.

Ensure that controller is passing all relevant Harmony tests.