Mindtree

Welcome to possible

EtherMind

User Guide – Mesh Command Line Interface

Version 0.16 | 10 January 2019

Copyright Information

REVISION HISTORY

Date	Version	Description	Author
04 April 2018	0.1	Initial Version	EtherMind Team
10 April 2018	0.2	Updated Light Client Models section with CTL, HSL and LC	EtherMind Team
16 April 2018	0.3	Updated Provisioning related section	EtherMind Team
03 May 2018	0.4	Updated Foundation Client, Model Client and Provisioning related sections	EtherMind Team
09 May 2018	0.5	Updated Foundation Server related sections	EtherMind Team
17 May 2018	0.6	Updated Foundation Client CLI commands input tables, Foundation Server Test Procedures	EtherMind Team
22 May 2018	0.7	Updated Foundation Server Test Procedures, Corrected Foundation Client Test Procedures	EtherMind Team
28 June 2018	0.8	Added Bearer related section. Move existing scan response data to bearer section	EtherMind Team
11 July 2018	0.9	Added Sensor, Time and Scheduler Model Server/Client related sections	EtherMind Team
27 July 2018	0.10	Added CLI command sections for Mesh Core Layer testing such as Transport and Network layers. Corrected Generic Property Client related	EtherMind Team

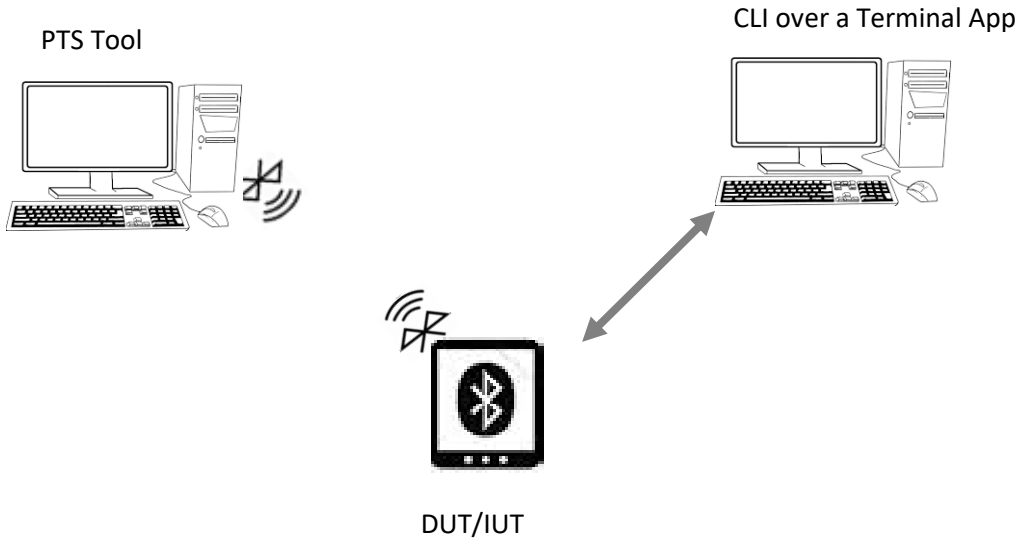
		CLI commands input parameters	
05 August 2018	0.11	Added CLI command to input health status, needed for Friendship Testcases. Added NOTE in section “Setup as Un-Provisioned Device over ADV bearer” and “Setup as Un-Provisioned Device over GATT bearer”	EtherMind Team
17 August 2018	0.12	Updated the Provisioning Module CLI command section	EtherMind Team
31 August 2018	0.13	Corrected hyperlinks in Config Client related Commands section Updated Config Client Model Set Publish Address command section	EtherMind Team
14 October 2018	0.14	Updated additional parameter description for Bind and Unbind config client commands	EtherMind Team
8 January 2019	0.15	Updated Transport related CLI command section with new command to “clear replay cache”.	EtherMind Team
10 January 2019	0.16	Updated Transport related CLI command - “Send Transport Control PDU” command section.	EtherMind Team

Table of Contents

1. Introduction

EtherMind Mesh stack testing for both core layers and models are performed using PTS test suite. EtherMind CLI interface is used to control the “Device Under Test” (DUT) or “Implementation Under Test” (IUT). PTS test reports are used as the evidence.

This document is to capture how various modules (core and models) to be setup and verified to capture the BQB submission evidences.



2. Requirements

Following are list of items required collecting prequalification logs for Mesh stack using PTS test tool application from SIG.

1. PTS Test Application V7.2.0 or Higher
2. PTS BLE Dongle [with Mesh Support, Dongle with following versions or above]

Details	Value	Info
BT HCI Version	0x08	4.2
BT HCI Revision	0x30E8	N/A
BT LMP Version	0x08	4.2
BT Manufacturer Name	0x000A	CSR
BT LMP Subversion	0x30E8	N/A

3. Device with EtherMind Mesh Stack and Models with Command Line Interface
4. TeraTerm or Hyperterminal (Terminal Application)
5. Mesh Model Test Specification
6. Mesh Profile Specification
7. TPG File
8. PTS Workspace (Generated from TPG)
9. Test Plan Spread Sheet (Generated from TPG)

3. Generic CLI Commands for Mesh Setup and Initializations

This section explains few of the CLI commands in detail which are used for generic functionalities of Mesh/Model Applications like Setup and Initialization, Device Reset etc. These CLI commands are used across all Mesh Core Stack/Model layer qualification.

3.1. Setup

This CLI command initializes the Mesh Stack and the Underlying BLE. After each Power Cycle or Re-launch of the IUT, this should be the first command to be sent.

Command Syntax	root->core->setup

Sample Command	root->core->setup
Dependent Parameter	NA
Note	In few ports of EtherMind Mesh, this command shall not be performed as this is done as part of the device startup routine. Please refer to specific port related instructions regarding this.

3.2. Setup as Un-Provisioned Device over ADV bearer

The following sequence of CLI commands are used to setup the IUT as an Un-Provisioned device over ADV bearer and also bind the local application to the core provisioning module

3.2.1. Setup as Device over ADV

Command Syntax	root->core->provision->setup
Sample Command	root->core->provision->setup 1 1
Dependent Parameter	<ROLE : [1 – Device, 2 – Provisioner]> <BEARER: [1 – ADV, 2 – GATT]>
Note	Before issuing this command, it is Mandatory to register “Foundation Models”. Please refer to this section for relevant CLI commands.

3.2.2. Bind the device to Provisioning Module

Command Syntax	root->core->provision->bind
Sample Command	root->core->provision->bind 1 0
Dependent Parameter	<BEARER: [1 – ADV, 2 – GATT]> <INDEX>
Note	NOTE: <INDEX> is “Dummy”/”Don’t Care” in case IUT is setup as Un-Provisioned device on ADV bearer.

3.3. Setup as Un-Provisioned Device over GATT bearer

The following sequence of CLI commands are used to setup the IUT as an Un-Provisioned device over GATT bearer.

3.3.1. Setup as Device over GATT

Command Syntax	root->core->provision->setup
Sample Command	root->core->provision->setup 1 2
Dependent Parameter	<ROLE : [1 – Device, 2 – Provisioner]> <BEARER: [1 – ADV, 2 – GATT]>
Note	Before issuing this command, it is Mandatory to register "Foundation Models". Please refer to this section for relevant CLI commands. For an Un-Provisioned device over GATT, only Setup needs to be issued. "Bind" of the local application can be automatically invoked at the application layer when the `BLEBRR_GATT_IFACE_ENABLE` platform event is received.

3.4. Setup as Provisioner over ADV bearer

The following sequence of CLI commands are used to setup the IUT as a Provisioner over ADV bearer and also bind the desired remote device to the core provisioning module

3.4.1. Setup as Provisioner over ADV

Command Syntax	root->core->provision->setup
Sample Command	root->core->provision->setup 2 1
Dependent Parameter	<ROLE : [1 – Device, 2 – Provisioner]> <BEARER: [1 – ADV, 2 – GATT]>
Note	

3.4.2. Bind the desired remote device to Provisioning Module

Command Syntax	root->core->provision->bind
Sample Command	root->core->provision->bind 1 1
Dependent Parameter	<BEARER: [1 – ADV, 2 – GATT]> <INDEX>
Note	NOTE: <INDEX> is the reference to the desired Remote Un-Provisioned Device. This needs to be invoked once the application is notified of any available Un-Provisioned Device waiting to enter the network through the Provisioning Callback Event "PROV_EVT_UNPROVISIONED_BEACON".

3.5. Setup as Provisioner over GATT bearer

The following sequence of CLI commands are used to setup the IUT as an Un-Provisioned device over GATT bearer.

3.5.1. Setup as Provisioner over GATT

Command Syntax	root->core->provision->setup
Sample Command	root->core->provision->setup 2 2
Dependent Parameter	<ROLE : [1 – Device, 2 – Provisioner]> <BEARER: [1 – ADV, 2 – GATT]>
Note	For Provisioner over GATT, only Setup needs to be issued. Also, for GATT Provisioner, additional Bearer related CLI commands needs to be issued to create a BLE connection with remote device, Discover the Mesh Provisioning Service and Enable the remote device for Mesh Provisioning Data Out Notifications etc., which are currently not supported by CLI commands. “Bind” of the desired remote device is taken care at the Bearer application layer.

3.6. Storage reset

This CLI command deletes all the data present in the persistent storage location designated for the Mesh Stack, Models and the application. This command is to be used when IUT needs to be removed from the existing networks, similar to a factory reset

Command Syntax	root->reset
Sample Command	root->reset
Dependent Parameter	NA
Note	1. Power Cycle the DUT/IUT after issuing this command.

3.7. Foundation Model Registration

This CLI command shows how to register the IUT as either Foundation Model Server or Foundation Model Client.

3.7.1. Foundation Model Server Registration

Command Syntax	root->model->models->foundation
Sample Command	root->model->models->foundation
Dependent Parameter	NA
Note	NA

3.7.2. Foundation Model Client Registration

1. Configuration Model Client setup

Command Syntax	root->model->modelc->config->setup
Sample Command	root->model->modelc->config->setup
Dependent Parameter	NA
Note	NA

2. Health Model Client setup

Command Syntax	root->model->modelc->health->setup
Sample Command	root->model->modelc->health->setup
Dependent Parameter	NA
Note	NA

3.8. Bearer Commands

The following sequence of CLI commands are to perform BLE Bearer related procedures, specifically for the BLE-GATT Bearer.

3.8.1. Connect

Command Syntax	root->brr->connect

Sample Command	root->brr->connect 0 A0A1A2A3A4A5A6
Dependent Parameter	<BLE ADDR TYPE : [0 – PUBLIC, 1 – RANDOM]> <BLE ADDR : [6 Octets of BLE Device Address MSB-LSB]>
Note	This CLI command is used to initiate connection from the Local device as BLE GAP Central to the desired remote device identified by the <BLE ADDR TYPE> and <BLE ADDR>. The <BLE ADDR TYPE> is the Bluetooth device address type which either PUBLIC or RANDOM i.e. 0 or 1 The <BLE ADDR> is the Bluetooth device address of the remote, which is in MSB-LSB format. i.e. for a Bluetooth Address 0xA0A1A2A3A4A5, input the <BLE ADDR> as A0A1A2A3A4A5

3.8.2. Disconnect

Command Syntax	root->brr->disconnect
Sample Command	root->brr->disconnect
Dependent Parameter	NA
Note	This CLI command initiates Disconnection of the current active BLE Link with the remote device.

3.8.3. Scan Response Set

Command Syntax	root->brr->srsp_set
Sample Command	root->brr->srsp_set
Dependent Parameter	NA
Note	This CLI command sets the desired Scan Response Data, which the application wants to use along with the Connectable Advertisements. For example to include “Complete Name” along with Mesh Connectable Un-Provisioned Device beacons or with Mesh Proxy Connectable Advertisements. The default value of the Scan Response Data is “MT-MESH-DEMO”. To have any other desired data to be sent in scan response, the array “cli_brr_scanrsp_data” present in cli_brr.c file [function cli_scan_rsp_data_set(…)] needs to be updated.

3.8.4. Discover Services

Command Syntax	root->brr->discover
Sample Command	root->brr->discover 0
Dependent Parameter	<SERVICE MODE : [0 – Provisioning Service, 1 – Proxy Service]>
Note	This CLI command discovers the desired services from the connected remote device [server]. Currently, the CLI option is enabled to discover only Mesh Provisioning or Proxy Services.

3.8.5. Enable/Disable Notifications

Command Syntax	root->brr->config_ntf
Sample Command	root->brr->config_ntf 1 0
Dependent Parameter	<Ntf state : [0 – Disable, 1 – Enable]> <SERVICE MODE : [0 – Provisioning Service, 1 – Proxy Service]>
Note	This CLI command is used to enable or disable the Mesh Provisioning or Mesh Proxy Data OUT characteristics for Notifications in the connected remote device [server].

3.9. Proxy Commands

The following sequence of CLI commands are used to realize Proxy Server related procedures such as Triggering and Terminating Proxy Server connectable advertisements. Also Proxy Client related procedures to send relevant Proxy Configuration messages such as setting list filters and adding/removing addresses from filter lists etc.

3.9.1. Start Proxy ADV using Network ID

Command Syntax	root->core->proxy->proxys->start
Sample Command	root->core->proxy->proxys->start 1 0

Dependent Parameter	<Type:[1 - Network ID, 2 - Node Identity]> <Subnet Handle: [2 Bytes of Subnet Handle]> [in HEX]
Note	This CLI command is used to start advertising proxy service using network ID

3.9.2. Start Proxy ADV using Node Identity

Command Syntax	root->core->proxy->proxys->start
Sample Command	root->core->proxy->proxys->start 2 0
Dependent Parameter	<Type:[1 - Network ID, 2 - Node Identity]> <Subnet Handle: [2 Bytes of Subnet Handle]> [in HEX]
Note	This CLI command is used to start advertising proxy service using Node Identity

3.9.3. Stop Proxy ADV

Command Syntax	root->core->proxy->proxys->stop
Sample Command	root->core->proxy->proxys->stop
Dependent Parameter	NA
Note	This CLI command is used to sop advertising proxy service

3.9.4. Set Whitelist Filter

Command Syntax	root->core->proxy->proxyc->setwl
Sample Command	root->core->proxy->proxyc->setwl
Dependent Parameter	NA
Note	This CLI command is used to set the Proxy filter Type as Whitelist filter

3.9.5. Set BlackList Filter

Command Syntax	root->core->proxy->proxyc->setbl
Sample Command	root->core->proxy->proxyc->setbl
Dependent Parameter	NA
Note	This CLI command is used to set the Proxy filter Type as blacklist filter

3.9.6. Add Address to FIlter

Command Syntax	root->core->proxy->proxyc->add_addr
Sample Command	root->core->proxy->proxyc->add_addr 0 1001 1002 ... 1005]>
Dependent Parameter	<Subnet Handle: [2 Bytes of Subnet Handle]> [in HEX] <List of Addresses[Max of Address count define by CLI_PROXY_MAX_PROXY_ADDR_CNT]> [in HEX]
Note	This CLI command is used to add destination addresses to the proxy filter list.

3.9.7. Remove Address to FIlter

Command Syntax	root->core->proxy->proxyc->rm_addr
Sample Command	root->core->proxy->proxyc->rm_addr 0 1001 1002 ... 1005]>
Dependent Parameter	<Subnet Handle: [2 Bytes of Subnet Handle]> [in HEX] <List of Addresses[Max of Address count define by CLI_PROXY_MAX_PROXY_ADDR_CNT]> [in HEX]
Note	This CLI command is used to remove destination addresses to the proxy filter list.

3.10. Transport Commands

The following sequence of CLI commands are used to realize transport layer specific procedures.

3.10.1 Send

Command Syntax	root->core->transport->send
Sample Command	root-> core->transport->send 0 0 0 0 0002 0001 root-> core->transport->send 0 0 0 0 0002 0001
Dependent Parameter	<Flag [0 - Unsegmented 1- segmented]> <Subnet Handle: [2 Bytes of Subnet Handle]> [in HEX] <Appkey Handle: [2 Bytes of Appkey Handle]> [in HEX] <TTL :[1 Byte of TTL]> [in HEX] <SRC ADDR: [2 Bytes of Source Address]> [in HEX] <DST ADDR: Non virtual - 2 octets Virtual - 16 octets>[in HEX]
Note	This CLI command is used to transmit segmented or unsegmented access pdu to either non virtual address or virtual address. For sending the access pdu to non virtual address, value for DST ADDR should be 2 byte value. For sending the access pdu to virtual address, value for DST ADDR should be 16 byte value

3.10.2 Friend Request

Command Syntax	root->core->transport->frndreq
Sample Command	root->core->transport->frndreq 0 4b 100 50 10000
Dependent Parameter	<Subnet Handle: [2 Bytes of Subnet Handle]> [in HEX] <criteria : [1 Byte of Criteria]> [in HEX] <rx_delay : [1 Byte of RX Delay (in milliseconds)]> [in Decimal] <poll_timeout : [4 Bytes of POLL timeout (in 100s of milliseconds)]> [in Decimal] [for 5 seconds, parameter to input is 50] <setup_timeout: [4 Byte of Setup Timeout (in milliseconds)]> [in Decimal] [for 10 seconds, parameter to input in 10000]
Note	This CLI command is used to send the friend Request to Friend Node Before using this command need to enable mesh friend feature(refer 3.12.1 section)

3.10.3 Send Transport Control PDU

Command Syntax	root->core->transport->ctrlmsg
Sample Command	root->core->transport->ctrlmsg 0
Dependent Parameter	Mandatory Parameters - <Control Message Type : [1 Bytes of Control Message Type, Valid types are – 0 - Friend Subscription List Add 1 – Friend Subscription List Remove 2 – Friend Clear > Optional Parameters – <Group Address : [2 Bytes of Group Address]> [in HEX]
Note	This CLI command is used to send either Friend Subscription List Add, remove and clear messages. NOTE: The Optional parameter “Group Address” is mandatory when the “Control Message Type” is 0 or 1 i.e. “Friend Subscription List Add” or “Friend Subscription List Remove”.

3.10.4 Clear Replay Cache

Command Syntax	root->core->transport->clrreplaycache
Sample Command	root->core->transport-> clrreplaycache
Dependent Parameter	None
Note	This CLI command is used to clear the replay cache.

3.11. Network Commands

The following sequence of CLI commands are used to transmit Network Layer related commands.

3.11.1 Set the network packet Header

Command Syntax	root->core->network->set_hdr
Sample	root->core->network->set_hdr 0 0 0002 0001

Command	
Dependent Parameter	<TTL : 1 Byte of TTL value> [in HEX] <CTL: 1 Byte of CTL value > [in HEX] <SRC Addr : 2 Bytes of Source Address> [in HEX] <DST Addr: 2 Bytes of Destination Address> [in HEX]
Note	This CLI command is used to set the headers of the network packet before sending the network packet

3.11.2 Send

Command Syntax	root->core->network->send
Sample Command	root->core->network->send 0001 0 root->core->network->send 0001 1
Dependent Parameter	<DST Addr: 2 Bytes of Destination Address> [in HEX] <Flag[0: Unsegmented, 1: Segmented]>
Note	This CLI command is used to send the unsegmented or segmented network packet. Before using this command need to set the network packet header(refer section 3.11.1)

3.11.3 Get Current IV Index

Command Syntax	root->core->network->get_ivindex
Sample Command	root->core->network->get_ivindex
Dependent Parameter	NA
Note	This CLI command is used to get the current IV Index

3.11.4 Secure Network Beacon [SNB]

These CLI commands are used to

1. Transmit Secure network beacon with both Key Refresh and IV Update flag bits set to 0[see 3.11.4.1]
2. Transmit Secure network beacons with Key Refresh update [see 3.11.4.2]
3. Transmit Secure network beacons with IV update [see 3.11.4.3]
4. Transmit Secure network beacons when peer device initiated Key refresh procedure [see 3.11.4.1]
5. Transmit Secure network beacons when local device needs to initiate Key refresh procedure [see 3.11.4.2]

6. Transmit Secure network bacons with both Key Refresh and IV update flag bits set to 1 [see 3.11.4.4]

3.11.4.1 Transmit SNB

Command Syntax	root->core->network->snb
Sample Command	root->core->network->snb 0
Dependent Parameter	<Subnet Handle: [2 Bytes of Subnet Handle]> [in HEX]
Note	Command to send secure network beacon with both Key Refresh and IV Update bits set to 0.

3.11.1.2 Transmit SNB for Key Refresh Procedure

Command Syntax	root->core->network->snb
Sample Command	root->core->network->snb 0 2 root->core->network-> snb 0 3
Dependent Parameter	<Subnet Handle: [2 Bytes of Subnet Handle]> [in HEX] {Optional Parameters:} <Key Refresh state 2- Phase2, 3- Phase3>
Note	1. Command to keep the device in Phase 2 and send the secure network beacons with Key Refresh flag set to 1, secured using new Netkey 2. Command to keep the device in Phase 3 and send the secure network beacons with Key Refresh flag set to 0, secured using new Netkey

3.11.1.3 Transmit SNB for IV Update procedure

Command Syntax	root->core->network->snb
Sample Command	root->core->network->snb 0 0 a root->core->network->snb 0 1 b
Dependent Parameter	<Subnet Handle: [2 Bytes of Subnet Handle]> [in HEX] {Optional Parameters :}

	<IV update state 0- Normal 1- IV Update in Progress> <IV index : 4 Bytes of IV Index> [in HEX]
Note	1. Command to keep the device in IV Update Normal State and send the secure network beacons with IV Update flag set to 0 and IV index = n 2. Command to keep the device in IV Update State and send the secure network beacons with IV Update flag set to 1 and IV index = m where (m = n+1)

3.11.1.4 Transmit SNB with both Key Refresh and IV Update Procedure

Command Syntax	root->core->network->snb
Sample Command	root->core->network->snb 0 1 1 a root->core->network->snb 0 1 1 b
Dependent Parameter	<Subnet Handle: [2 Bytes of Subnet Handle]> [in HEX] <Key Refresh state : [1 Byte of Key Refresh State]> [in HEX] <IV update state : [1 Byte of IV Update State]> [in HEX] <IV index : [4 Bytes of IV Index]> [in HEX]
Note	Command to send secure network beacon with both Key Refresh and IV Update flag bits set to 1 and with current IV index value

3.12. Select Mesh Core Features

This CLI command shows how to Enable or Disable the Mesh Core Features such as Relay, Proxy, Friend and Low Power Node.

3.12.1. Enable Mesh Core Features Support

Command Syntax	root->core->enable
Sample Command	root->core->enable relay
Dependent Parameter	<Selected Feature : ["relay", "proxy", "friend", "lpn"]>
Note	The parameters to this command are strings and it is case sensitive.

	The valid string values are as mentioned above.

3.12.2. Disable Mesh Core Features Support

Command Syntax	root->core->disable
Sample Command	root->core->disable relay
Dependent Parameter	<Selected Feature : ["relay", "proxy", "friend", "lpn"]>
Note	The parameters to this command are strings and it is case sensitive. The valid string values are as mentioned above.

3.13. Health

This CLI command publishes the current health status with the given health status code.

Command Syntax	root->health
Sample Command	root->health 0
Dependent Parameter	<Status Code: [1 Byte of Status Code]> [in HEX]
Note	This Command needs to be issued for the LPN testcase MESH/NODE/FRND/LPN/BV-02-C

3.14. Vendor Model Registration

This CLI command shows how to register the IUT as either Foundation Model Server or Foundation Model Client.

3.14.1. Vendor Model Server Registration

Command Syntax	root->model->models->vendor
Sample Command	root->model->models->vendor
Dependent Parameter	NA

Note	NA

4. Mesh Core Testing

This section captures the CLI command sequences needed to setup IUT for testing Mesh Core procedures against PTS.

4.1. Provisioning Procedures

This section covers the CLI commands needed to setup IUT as either Un-Provisioned Device waiting to be provisioned into a network or the Provisioner which provisions other devices into the Mesh Network and test the corresponding Mesh Core Provisioning testcases with the PTS test tool.

4.1.1. Provisioning CLI Procedures

1. Setup

This CLI command is used to setup the IUT as either Provisioner or Un-Provisioned Device over either ADV or GATT bearer.

Command Syntax	root->core->provision->setup
Sample Command	root->core->provision->setup 1 1
Dependent Parameter	<ROLE : [1 – Device, 2 – Provisioner]> <BEARER: [1 – ADV, 2 – GATT]>
Note	

2. Bind

This CLI command is used to bind either the Local Device or the Remote Device pointed by the Index to the Provisioning Module of the Mesh Stack.

Command Syntax	root->core->provision->bind
Sample Command	root->core->provision->bind 1 0
Dependent Parameter	<BEARER: [1 – ADV, 2 – GATT]> <INDEX>
Note	NOTE: <INDEX> is “Dummy”/”Don’t Care” in case IUT is setup as Un-Provisioned device on ADV bearer.

3. Set Authentication Action/Mode

This CLI command is used to set the desired authentication mode to be used for current provisioning, both when IUT plays the Device role or the Provisioner role.

Command Syntax	root->core->provision->auth_act
Sample Command	root->core->provision->auth_act 2 3 4
Dependent Parameter	<MODE: [0 – None, 1 – Static OOB, 2 – Output OOB, 3 – Input OOB]> If Mode is Output OOB ->> <OOB Action: [0-Blink,1-Beep,2-Vibrate,3-Numeric, 4-Alphanumeric]> Else, If Mode is Input OOB->> <OOB Action: [0-Push,1-Twist,2-Numeric,3-Alphanumeric]> <OOB Size: [1 – 8 Bytes]> Optional Parameters – <Static OOB Value: [16 Bytes]>
Note	NOTE: If MODE is “Static OOB” then currently the “Static OOB Value” is “Don’t Care”. The current Provisioning application uses the following hard coded “Static OOB value”: 0x11111111111111111111111111111111[16 Octets 0f 0x11]. While testing against PTS tool, Please input the “Static OOB Value” mentioned above in the IXIT “TSPX_OOB_code”.

4. Set the Authentication Value

This CLI command is used to set the authentication value to the Provisioning Module in the Mesh Stack.

Command Syntax	root->core->provision->auth_val
Sample Command	root->core->provision->auth_val 0 7
Dependent Parameter	<MODE: [0 – Numeric, 1 – Alphanumeric]> <Value>
Note	

5. Get Device Public Key

This CLI command is used to fetch the current ECDH P256 Public key provided by the crypto module to the Mesh Stack, used in the Un-Provisioned device.

Command Syntax	root->core->provision->get_pkey
Sample Command	root->core->provision->get_pkey
Dependent Parameter	None
Note	NOTE_1: While testing against PTS tool, Set the local ECDH P256 Public-key in PTS IXIT item “TSPX_device_public_key”. NOTE_2: Once TC is completed, Reset the PTS IXIT item “TSPX_device_public_key” with the default value

6. Set Device Public Key

This CLI command is used by the Provisioner to set the ECDH P256 Public key of the Remote Un-Provisioned device to the Provisioning Module in the Mesh Stack.

Command Syntax	root->core->provision->dev_pkey
Sample Command	root->core->provision->dev_pkey
Dependent Parameter	None
Note	NOTE: Currently, the “dev_pkey” that gets set from the provisioning application is a hardcoded 64 Octet value defined as the default ECDH P256 Public Key used by PTS tool in the IXIT item “TSPX_device_public_key”.

7. Get list of Provisioned Devices

This CLI command is used to get the list of Provisioned nodes by the Provisioner from the Provisioning Module in the Mesh Stack.

Command Syntax	root->core->provision->dev_list
Sample Command	root->core->provision->dev_list

Dependent Parameter	None
Note	NOTE: Currently, the "dev_list" command will return the Unicast Address assigned and the total number of elements present in each of the Provisioned nodes.

8. Remove all Device Keys

This CLI command is used to delete all the device keys maintained in the Mesh Stack.

Command Syntax	root->core->provision->rm_devkeys
Sample Command	root->core->provision->rm_devkeys
Dependent Parameter	None
Note	NOTE: After this command is issued, the Device can **"Neither initiate nor respond to any Configuration Message".**

4.1.2. Provisioning PTS Test Procedures

1. PTS Test of Un-Provisioned Device

To test IUT as an Un-Provisioned Device, follow the below given procedures:

1. Reset and power cycle/ Re-launch the IUT.
2. Setup and Initialize the IUT [if needed by the platform]
3. Register the Foundation Model Servers.
4. Register the specific Model Server which is of interest. For example, to register the Generic ONOFF model server follow the instructions from here.
5. Setup the IUT [Board/executable] as a "New Device" which is to be provisioned to a Mesh Network.
6. Based on the testcase, set the Authentication Mode and Update the Public Key of Device. Please refer to the below table for sample authentication modes and commands to be issued in CLI depending on each test category of Provisioning when IUT plays the Device/Node role.

TABLE 1: Authentication and Public Key Combinations for Un-Provisioned Device

OOB Authentication Mode

Device's Public Key OOB Mode		INPUT OOB Device Inputs/ Provisioner Outputs	OUTPUT OOB Device Outputs/ Provisioner Inputs	STATIC OOB Device & Provisioner Inputs	NONE
	NO	auth_act -> 3 0 2 auth_val -> <M><V>	auth_act-> 2 3 4	auth_act-> 1 0 0	
	YES	get_pkey -> auth_act -> 3 0 2 auth_val -> <M><V>	get_pkey -> auth_act -> 2 3 4	get_pkey -> auth_act -> 1 0 0	get_pkey ->

2. PTS Test of Provisioner

To test IUT as a Provisioner, follow the below given procedures:

1. Reset and power cycle/ Re-launch the IUT.
2. Setup and Initialize the IUT [if needed by the platform]
3. Register the Foundation Model Servers and Client.
4. Register the specific Model Client which is of interest. For example, to register the Generic ONOFF model Client follow the instructions from here.
5. Setup the IUT [Board/executable] as a "Provisioner" which has to provision other devices to a Mesh Network.
6. Based on the testcase, set the Authentication Mode and Update the Public Key of Device. Please refer to the below table for sample authentication modes and commands to be issued in CLI depending on each test category of Provisioning when IUT plays the Provisioner role.

TABLE 2: Authentication and Public Key Combinations for Provisioner

		OOB Authentication Mode			
Device's Public		INPUT OOB Device Outputs/ Provisioner Inputs	OUTPUT OOB Device Inputs/ Provisioner Outputs	STATIC OOB Device & Provisioner Inputs	NONE
	NO	auth_act -> 2 3 1	auth_act-> 3 0 1	auth_act-> 1 0 0	

Key OOB Mode		auth_val -> <M><V>			
	YES	dev_pkey -> auth_act -> 2 3 1 auth_val -> <M><V>	dev_pkey -> auth_act -> 3 3 1	dev_pkey -> auth_act -> 1 0 0	dev_pkey ->

4.2.Configuration Server Procedures

This section covers the CLI commands needed to setup IUT as a Configuration Server and test the corresponding Configuration Server testcases with the PTS test tool.

4.2.1. Configuration Server CLI Procedures

Currently, there are no specific CLI sub procedures for Configuration Server.

4.2.2. Configuration Server PTS Test Procedures

To test IUT as Configuration Server, follow the below given procedures:

1. Reset and power cycle/ Re-launch the IUT.
2. Setup and Initialize the IUT [if needed by the platform]
3. Register the Foundation Model Servers.
4. Register any Model Server, which is of interest. For example, to register the Generic ONOFF model server follow the instructions from here.
5. Setup the IUT [Board/executable] as a "New Device" which is to be provisioned to a Mesh Network.
6. Based on the testcase from the corresponding category, "Mesh->Configuration Model" in PTS, enable or disable the features that is of interest. Please refer to the above section for CLI commands to enable and disable the desired mesh features such as Relay/Proxy/Friend/LPN etc.
7. Few test cases are dependent on features that are supported by IUT and also the PICS and PIXIT that are configured at PTS during the testing based on the features supported by IUT. For example, in testcase MESH/NODE/CFG/CFGR/BV-01-C, PTS PIXIT configuration "TSPX_iut_supports_relay" needs to be set as TRUE if MS_RELAY_SUPPORT is defined in MS_features.h
8. For few other testcases, the test procedure mandates the IUT to support a reduced resource sets. The corresponding tunable parameter should be updated in MS_limits.h to successfully execute these testcases. For example, in testcase MESH/NODE/CFG/NKL/BI-03-C, clear the persistent storage and configure MS_MAX_SUBNETS to 1 and rebuild the binary and execute the testcase.

9. For testcases related testing Configuration Server along with a Vendor Specific Model, for example, in testcases such as MESH/NODE/CFG/SL/BV-03-C and MESH/NODE/CFG/MAKL/BI-04-C follow the instructions from here to register the Vendor Specific Model along with “Step 4”. Also, for these testcases, update the PTS PIXIT configuration 'TSPX_vendor_model_id' as 'A001A001' and run the testcase.

4.3. Configuration Client Procedures

This section covers the CLI commands needed to setup IUT as a Configuration Client and test the corresponding Configuration Client testcases with the PTS test tool.

4.3.1. Configuration Client CLI Procedures

1. Configuration Client Model Set Publish Address

Follow the below CLI command to set the desired Publish address to the Configuration Client from the application.

Command Syntax	root->model->modelc->config->publishaddr
Sample Command	root->model->modelc->config->publishaddr 1
Dependent Parameter	<Publish Address : [2 Bytes of Address]> <App Key ID : [2 Byte of App Key Index to used]>
Note	In the PTS testing of Config Client, the Publish Address is usually the Unicast Address of the PTS device(which is the Provisioner and the Configuration Client) and its address is typically 0x0001(Unicast Address of Provisioner/PTS). NOTE: The App Key ID parameter need not be provided as the CLI application tries to fetch the corresponding App Key ID/ Device Key ID for the given Address. If there is no corresponding mapping for the provided address, then the first Device Key ID i.e. MS_MAX_APPS is used.

2. Configuration Client Beacon GET

Follow the below CLI command to send the Configuration Client Beacon GET message to the Mesh Network.

Command Syntax	root->model->modelc->config->beaconget
Sample	root->model->modelc->config->beaconget

Command	
Dependent Parameter	NA
Note	The "Publish Address" needs to be set before issuing this command while testing with PTS.

3. Configuration Client Beacon SET

Follow the below CLI command to send the Configuration Client Beacon SET-ACKNOWLEDGED message to the Mesh Network.

Command Syntax	root->model->modelc->config->beaconset
Sample Command	root->model->modelc->config->beaconset 1
Dependent Parameter	<Beacon Flag : [1 Byte of Beacon Flag Value]> [in HEX]
Note	The "Publish Address" needs to be set before issuing this command while testing with PTS. All values to be populated in HEX as prompted by the PTS test tool during the particular test execution.

4. Configuration Client Composition Data GET

Follow the below CLI command to send the Configuration Client Composition Data GET message to the Mesh Network.

Command Syntax	root->model->modelc->config->compositiondataget
Sample Command	root->model->modelc->config->compositiondataget 1
Dependent Parameter	<Page Num : [1 Byte of Page Value]> [in HEX]
Note	The "Publish Address" needs to be set before issuing this command while testing with PTS. All values to be populated in HEX as prompted by the PTS test tool during the particular test execution.

5. Configuration Client Default TTL GET

Follow the below CLI command to send the Configuration Client Default TTL GET message to the Mesh Network.

Command Syntax	root->model->modelc->config->defaultttlget
Sample Command	root->model->modelc->config->defaultttlget
Dependent Parameter	NA
Note	The "Publish Address" needs to be set before issuing this command while testing with PTS.

6. Configuration Client Default TTL SET

Follow the below CLI command to send the Configuration Client Default TTL SET-ACKNOWLEDGED message to the Mesh Network.

Command Syntax	root->model->modelc->config->defaultttlset
Sample Command	root->model->modelc->config->defaultttlset 7F
Dependent Parameter	<Default TTL Value : [1 Byte of Default TTL Value]> [in HEX]
Note	The "Publish Address" needs to be set before issuing this command while testing with PTS. All values to be populated in HEX as prompted by the PTS test tool during the particular test execution.

7. Configuration Client Proxy GET

Follow the below CLI command to send the Configuration Client Proxy GET message to the Mesh Network.

Command Syntax	root->model->modelc->config->proxyget
Sample Command	root->model->modelc->config->proxyget
Dependent Parameter	NA
Note	The "Publish Address" needs to be set before issuing this command while testing with PTS.

8. Configuration Client Proxy SET

Follow the below CLI command to send the Configuration Client Proxy SET-ACKNOWLEDGED message to the Mesh Network.

Command Syntax	root->model->modelc->config->proxyset
Sample Command	root->model->modelc->config->proxyset 1
Dependent Parameter	<Proxy Flag : [1 Byte of Proxy Flag Value]> [in HEX]
Note	The "Publish Address" needs to be set before issuing this command while testing with PTS. All values to be populated in HEX as prompted by the PTS test tool during the particular test execution.

9. Configuration Client Friend GET

Follow the below CLI command to send the Configuration Client Friend GET message to the Mesh Network.

Command Syntax	root->model->modelc->config->friendget
Sample Command	root->model->modelc->config->friendget
Dependent Parameter	NA
Note	The "Publish Address" needs to be set before issuing this command while testing with PTS.

10. Configuration Client Friend SET

Follow the below CLI command to send the Configuration Client Friend SET-ACKNOWLEDGED message to the Mesh Network.

Command Syntax	root->model->modelc->config->friendset
Sample Command	root->model->modelc->config->friendset 1
Dependent Parameter	<Friend Flag : [1 Byte of Friend Flag Value]> [in HEX]
Note	The "Publish Address" needs to be set before issuing this command while testing with PTS. All values to be populated in HEX as prompted by the PTS test tool during the particular test execution.

11. Configuration Client Relay GET

Follow the below CLI command to send the Configuration Client Relay GET message to the Mesh Network.

Command Syntax	root->model->modelc->config->relayget
Sample Command	root->model->modelc->config->relayget
Dependent Parameter	NA
Note	The "Publish Address" needs to be set before issuing this command while testing with PTS.

12. Configuration Client Relay SET

Follow the below CLI command to send the Configuration Client Relay SET-ACKNOWLEDGED message to the Mesh Network.

Command Syntax	root->model->modelc->config->relayset
Sample Command	root->model->modelc->config->relayset 1 0 0
Dependent Parameter	<Relay Flag : [1 Byte of Relay Flag Value]> [in HEX] <Relay RTX Count : [1 Byte of Relay RTX Count Value{Only 3 Usable Bits}]> [in HEX] <Relay Interval Steps: [1 Byte of Relay Retransmit Interval Steps{Only 5 Usable Bits}]> [in HEX]
Note	The "Publish Address" needs to be set before issuing this command while testing with PTS. All values to be populated in HEX as prompted by the PTS test tool during the particular test execution.

13. Configuration Client Model Publication GET

Follow the below CLI command to send the Configuration Client Model Publication GET message to the Mesh Network.

Command Syntax	root->model->modelc->config->modelpublicationget
Sample Command	root->model->modelc->config->modelpublicationget

Dependent Parameter	<Element Address : [2 Bytes of Element Address]> [in HEX] <Model ID : [4 Bytes of Model ID]> [in HEX]
Note	The "Publish Address" needs to be set before issuing this command while testing with PTS. All values to be populated in HEX as prompted by the PTS test tool during the particular test execution. Model ID Type will be derived from the length of Model ID (16 bits for SIG Model ID and 32 bits for Vendor Model ID)

14. Configuration Client Model Publication SET

Follow the below CLI command to send the Configuration Client Model Publication SET-ACKNOWLEDGED message to the Mesh Network.

Command Syntax	root->model->modelc->config->modelpublicationset
Sample Command	root->model->modelc->config->modelpublicationset
Dependent Parameter	<Element Address : [2 Bytes of Element Address]> [in HEX] <Publish Address : [2 Bytes of Publish Address]> [in HEX] <AppKey Index : [2 Bytes of AppKey Index]> [in HEX] <Credential Flag : [1 Byte of Credential Flag]> [in HEX] <Publish TTL : [1 Byte of Publish TTL]> [in HEX] <Publish Period : [1 Byte of Publish Period]> [in HEX] <Publish RTX Count : [1 Byte of Publish Retransmit Count]> [in HEX] <Publish RTX Steps : [1 Byte of Publish Retransmit Interval Steps]> [in HEX] <Model ID : [4 Bytes of Model ID]> [in HEX]
Note	The "Publish Address" needs to be set before issuing this command while testing with PTS. All values to be populated in HEX as prompted by the PTS test tool during the particular test execution. Model ID Type will be derived from the length of Model ID (16 bits for SIG Model ID and 32 bits for Vendor Model ID)

15. Configuration Client Model Subscription Add

Follow the below CLI command to send the Configuration Client Model Subscription Add message to the Mesh Network.

Command Syntax	root->model->modelc->config->modelsubscriptionadd
Sample Command	root->model->modelc->config->modelsubscriptionadd
Dependent Parameter	<Element Address : [2 Bytes of Element Address]> [in HEX] <Address : [2 Bytes of Address]> [in HEX] <Model ID : [4 Bytes of Model ID]> [in HEX]
Note	The "Publish Address" needs to be set before issuing this command while testing with PTS. All values to be populated in HEX as prompted by the PTS test tool during the particular test execution. Model ID Type will be derived from the length of Model ID (16 bits for SIG Model ID and 32 bits for Vendor Model ID)

16. Configuration Client Model Subscription Overwrite

Follow the below CLI command to send the Configuration Client Model Subscription Overwrite message to the Mesh Network.

Command Syntax	root->model->modelc->config->modelsubscriptionoverwrite
Sample Command	root->model->modelc->config->modelsubscriptionoverwrite
Dependent Parameter	<Element Address : [2 Bytes of Element Address]> [in HEX] <Address : [2 Bytes of Address]> [in HEX] <Model ID : [4 Bytes of Model ID]> [in HEX]
Note	The "Publish Address" needs to be set before issuing this command while testing with PTS. All values to be populated in HEX as prompted by the PTS test tool during the particular test execution. Model ID Type will be derived from the length of Model ID (16 bits for SIG Model ID and 32 bits for Vendor Model ID)

17. Configuration Client Model Subscription Delete

Follow the below CLI command to send the Configuration Client Model Subscription Delete message to the Mesh Network.

Command Syntax	root->model->modelc->config->modelsubscriptiondelete

Sample Command	root->model->modelc->config->modelsubscriptiondelete
Dependent Parameter	<Element Address : [2 Bytes of Element Address]> [in HEX] <Address : [2 Bytes of Address]> [in HEX] <Model ID : [4 Bytes of Model ID]> [in HEX]
Note	The "Publish Address" needs to be set before issuing this command while testing with PTS. All values to be populated in HEX as prompted by the PTS test tool during the particular test execution. Model ID Type will be derived from the length of Model ID (16 bits for SIG Model ID and 32 bits for Vendor Model ID)

18. Configuration Client Model Subscription Delete All

Follow the below CLI command to send the Configuration Client Model Subscription Delete All message to the Mesh Network.

Command Syntax	root->model->modelc->config->modelsubscriptiondeleteall
Sample Command	root->model->modelc->config->modelsubscriptiondeleteall
Dependent Parameter	<Element Address : [2 Bytes of Element Address]> [in HEX] <Model ID : [4 Bytes of Model ID]> [in HEX]
Note	The "Publish Address" needs to be set before issuing this command while testing with PTS. All values to be populated in HEX as prompted by the PTS test tool during the particular test execution. Model ID Type will be derived from the length of Model ID (16 bits for SIG Model ID and 32 bits for Vendor Model ID)

19. Configuration Client Model Publication Virtual Address SET

Follow the below CLI command to send the Configuration Client Model Publication Virtual Address SET-ACKNOWLEDGED message to the Mesh Network.

Command Syntax	root->model->modelc->config->modelpublicationvirtualaddressset
Sample Command	root->model->modelc->config->modelpublicationvirtualaddressset
Dependent Parameter	<Element Address : [2 Bytes of Element Address]> [in HEX]

	<Publish Address : [16 Bytes of Publish Address Label UUID]> [in HEX] <AppKey Index : [2 Bytes of AppKey Index]> [in HEX] <Credential Flag : [1 Byte of Credential Flag]> [in HEX] <Publish TTL : [1 Byte of Publish TTL]> [in HEX] <Publish Period : [1 Byte of Publish Period]> [in HEX] <Publish RTX Count : [1 Byte of Publish Retransmit Count]> [in HEX] <Publish RTX Steps : [1 Byte of Publish Retransmit Interval Steps]> [in HEX] <Model ID : [4 Bytes of Model ID]> [in HEX]
Note	The "Publish Address" needs to be set before issuing this command while testing with PTS. All values to be populated in HEX as prompted by the PTS test tool during the particular test execution. Model ID Type will be derived from the length of Model ID (16 bits for SIG Model ID and 32 bits for Vendor Model ID)

20. Configuration Client Model Subscription Virtual Address Add

Follow the below CLI command to send the Configuration Client Model Subscription Virtual Address Add message to the Mesh Network.

Command Syntax	root->model->modelc->config->modelsubscriptionvirtualaddressadd
Sample Command	root->model->modelc->config->modelsubscriptionvirtualaddressadd
Dependent Parameter	<Element Address : [2 Bytes of Element Address]> [in HEX] <Address : [16 Bytes of Address Label UUID]> [in HEX] <Model ID : [4 Bytes of Model ID]> [in HEX]
Note	The "Publish Address" needs to be set before issuing this command while testing with PTS. All values to be populated in HEX as prompted by the PTS test tool during the particular test execution. Model ID Type will be derived from the length of Model ID (16 bits for SIG Model ID and 32 bits for Vendor Model ID)

21. Configuration Client Model Subscription Virtual Address Overwrite

Follow the below CLI command to send the Configuration Client Model Subscription Virtual Address Overwrite message to the Mesh Network.

Command Syntax	root->model->modelc->config-> modelsubscriptionvirtualaddressoverwrite
Sample Command	root->model->modelc->config-> modelsubscriptionvirtualaddressoverwrite
Dependent Parameter	<Element Address : [2 Bytes of Element Address]> [in HEX] <Address : [16 Bytes of Address Label UUID]> [in HEX] <Model ID : [4 Bytes of Model ID]> [in HEX]
Note	The "Publish Address" needs to be set before issuing this command while testing with PTS. All values to be populated in HEX as prompted by the PTS test tool during the particular test execution. Model ID Type will be derived from the length of Model ID (16 bits for SIG Model ID and 32 bits for Vendor Model ID)

22. Configuration Client Model Subscription Virtual Address Delete

Follow the below CLI command to send the Configuration Client Model Subscription Virtual Address Delete message to the Mesh Network.

Command Syntax	root->model->modelc->config-> modelsubscriptionvirtualaddressdelete
Sample Command	root->model->modelc->config-> modelsubscriptionvirtualaddressdelete
Dependent Parameter	<Element Address : [2 Bytes of Element Address]> [in HEX] <Address : [16 Bytes of Address Label UUID]> [in HEX] <Model ID : [4 Bytes of Model ID]> [in HEX]
Note	The "Publish Address" needs to be set before issuing this command while testing with PTS. All values to be populated in HEX as prompted by the PTS test tool during the particular test execution. Model ID Type will be derived from the length of Model ID (16 bits for SIG Model ID and 32 bits for Vendor Model ID)

23. Configuration Client SIG Model Subscription GET

Follow the below CLI command to send the Configuration Client SIG Model Subscription GET message to the Mesh Network.

Command Syntax	root->model->modelc->config->sigmodelsubscriptionget
Sample Command	root->model->modelc->config->sigmodelsubscriptionget
Dependent Parameter	<Element Address : [2 Bytes of Element Address]> [in HEX] <Model ID : [4 Bytes of Model ID]> [in HEX]
Note	The "Publish Address" needs to be set before issuing this command while testing with PTS. All values to be populated in HEX as prompted by the PTS test tool during the particular test execution.

24. Configuration Client Vendor Model Subscription GET

Follow the below CLI command to send the Configuration Client Vendor Model Subscription GET message to the Mesh Network.

Command Syntax	root->model->modelc->config->vendormodelsubscriptionget
Sample Command	root->model->modelc->config->vendormodelsubscriptionget
Dependent Parameter	<Element Address : [2 Bytes of Element Address]> [in HEX] <Model ID : [4 Bytes of Model ID]> [in HEX]
Note	The "Publish Address" needs to be set before issuing this command while testing with PTS. All values to be populated in HEX as prompted by the PTS test tool during the particular test execution.

25. Configuration Client NetKey GET

Follow the below CLI command to send the Configuration Client NetKey GET message to the Mesh Network.

Command Syntax	root->model->modelc->config->netkeyget
Sample Command	root->model->modelc->config->netkeyget
Dependent Parameter	NA

Note	The "Publish Address" needs to be set before issuing this command while testing with PTS.

26. Configuration Client NetKey Add

Follow the below CLI command to send the Configuration Client NetKey Add message to the Mesh Network.

Command Syntax	root->model->modelc->config->netkeyadd
Sample Command	root->model->modelc->config->netkeyadd 1 1
Dependent Parameter	<NetKey Index : [2 Bytes of NetKey Index]> [in HEX] <NetKey Value : [16 Bytes of NetKey]> [in HEX]
Note	The "Publish Address" needs to be set before issuing this command while testing with PTS. All values to be populated in HEX as prompted by the PTS test tool during the particular test execution.

27. Configuration Client NetKey Update

Follow the below CLI command to send the Configuration Client NetKey Update message to the Mesh Network.

Command Syntax	root->model->modelc->config->netkeyupdate
Sample Command	root->model->modelc->config->netkeyupdate1 1
Dependent Parameter	<NetKey Index : [2 Bytes of NetKey Index]> [in HEX] <NetKey Value : [16 Bytes of NetKey]> [in HEX]
Note	The "Publish Address" needs to be set before issuing this command while testing with PTS. All values to be populated in HEX as prompted by the PTS test tool during the particular test execution.

28. Configuration Client NetKey Delete

Follow the below CLI command to send the Configuration Client NetKey Delete message to the Mesh Network.

Command Syntax	root->model->modelc->config->netkeydelete
Sample	root->model->modelc->config->netkeydelete 1

Command	
Dependent Parameter	<NetKey Index : [2 Bytes of NetKey Index]> [in HEX]
Note	The "Publish Address" needs to be set before issuing this command while testing with PTS. All values to be populated in HEX as prompted by the PTS test tool during the particular test execution.

29. Configuration Client AppKey GET

Follow the below CLI command to send the Configuration Client AppKey GET message to the Mesh Network.

Command Syntax	root->model->modelc->config->appkeyget
Sample Command	root->model->modelc->config->appkeyget 0
Dependent Parameter	<NetKey Index : [2 Bytes of NetKey Index]> [in HEX]
Note	The "Publish Address" needs to be set before issuing this command while testing with PTS. All values to be populated in HEX as prompted by the PTS test tool during the particular test execution.

30. Configuration Client AppKey Add

Follow the below CLI command to send the Configuration Client AppKey Add message to the Mesh Network.

Command Syntax	root->model->modelc->config->appkeyadd
Sample Command	root->model->modelc->config->appkeyadd 0 1 1
Dependent Parameter	<NetKey Index : [2 Bytes of NetKey Index]> [in HEX] <AppKey Index : [2 Bytes of AppKey Index]> [in HEX] <AppKey Value : [16 Bytes of AppKey]> [in HEX]
Note	The "Publish Address" needs to be set before issuing this command while testing with PTS. All values to be populated in HEX as prompted by the PTS test tool during the particular test execution.

31. Configuration Client AppKey Update

Follow the below CLI command to send the Configuration Client AppKey Update message to the Mesh Network.

Command Syntax	root->model->modelc->config->appkeyupdate
Sample Command	root->model->modelc->config->appkeyupdate 0 1 1
Dependent Parameter	<NetKey Index : [2 Bytes of NetKey Index]> [in HEX] <AppKey Index : [2 Bytes of AppKey Index]> [in HEX] <AppKey Value : [16 Bytes of AppKey]> [in HEX]
Note	The "Publish Address" needs to be set before issuing this command while testing with PTS. All values to be populated in HEX as prompted by the PTS test tool during the particular test execution.

32. Configuration Client AppKey Delete

Follow the below CLI command to send the Configuration Client AppKey Delete message to the Mesh Network.

Command Syntax	root->model->modelc->config->appkeydelete
Sample Command	root->model->modelc->config->appkeydelete 0 1
Dependent Parameter	<NetKey Index : [2 Bytes of NetKey Index]> [in HEX] <AppKey Index : [2 Bytes of AppKey Index]> [in HEX]
Note	The "Publish Address" needs to be set before issuing this command while testing with PTS. All values to be populated in HEX as prompted by the PTS test tool during the particular test execution.

33. Configuration Client Bind

Follow the below CLI command to send the Configuration Client Bind message to the Mesh Network.

Command Syntax	root->model->modelc->config->bind
Sample Command	root->model->modelc->config->bind

Dependent Parameter	<Element Address : [2 Bytes of Element Address]> [in HEX] <AppKey Index : [2 Bytes of AppKey Index]> [in HEX] <Model ID : [4 Bytes of Model ID]> [in HEX] <Complementary Model ID : [4 Bytes of Model ID]> [in HEX]
Note	The "Publish Address" needs to be set before issuing this command while testing with PTS. All values to be populated in HEX as prompted by the PTS test tool during the particular test execution. The complementary Model ID parameter will be used if the corresponding complementary model is initialized in the local device. For example, if the local node is a configuration client with a Generic ONOFF client and the bind is done for a remote node's Generic ONOFF server model, the Generic ONOFF client in the local node also will be bound to the Appkey at the given Appkey index.

34. Configuration Client Unbind

Follow the below CLI command to send the Configuration Client Unbind message to the Mesh Network.

Command Syntax	root->model->modelc->config->unbind
Sample Command	root->model->modelc->config->unbind
Dependent Parameter	<Element Address : [2 Bytes of Element Address]> [in HEX] <AppKey Index : [2 Bytes of AppKey Index]> [in HEX] <Model ID : [4 Bytes of Model ID]> [in HEX] <Complementary Model ID : [4 Bytes of Model ID]> [in HEX]
Note	The "Publish Address" needs to be set before issuing this command while testing with PTS. All values to be populated in HEX as prompted by the PTS test tool during the particular test execution. The complementary Model ID parameter will be used if the corresponding complementary model is initialized in the local device. For example, if the local node is a configuration client with a Generic ONOFF client and the unbind is done for a remote node's Generic ONOFF server model, the binding of Generic ONOFF client in the local node also will be done with the Appkey at the given Appkey index.

35. Configuration Client SIG Model App GET

Follow the below CLI command to send the Configuration Client SIG Model App GET message to the Mesh Network.

Command Syntax	root->model->modelc->config->sigmodelappget
Sample Command	root->model->modelc->config->sigmodelappget
Dependent Parameter	<Element Address : [2 Bytes of Element Address]> [in HEX] <Model ID : [4 Bytes of Model ID]> [in HEX]
Note	The "Publish Address" needs to be set before issuing this command while testing with PTS. All values to be populated in HEX as prompted by the PTS test tool during the particular test execution.

36. Configuration Client Vendor Model App GET

Follow the below CLI command to send the Configuration Client Vendor Model App GET message to the Mesh Network.

Command Syntax	root->model->modelc->config->vendormodelappget
Sample Command	root->model->modelc->config->vendormodelappget
Dependent Parameter	<Element Address : [2 Bytes of Element Address]> [in HEX] <Model ID : [4 Bytes of Model ID]> [in HEX]
Note	The "Publish Address" needs to be set before issuing this command while testing with PTS. All values to be populated in HEX as prompted by the PTS test tool during the particular test execution.

37. Configuration Client Node Identity GET

Follow the below CLI command to send the Configuration Client Node Identity GET message to the Mesh Network.

Command Syntax	root->model->modelc->config->nodeidentityget
Sample Command	root->model->modelc->config->nodeidentityget
Dependent Parameter	NA
Note	The "Publish Address" needs to be set before issuing this command while testing with PTS.

38. Configuration Client Node Identity SET

Follow the below CLI command to send the Configuration Client Node Identity SET message to the Mesh Network.

Command Syntax	root->model->modelc->config->nodeidentityset
Sample Command	root->model->modelc->config->nodeidentityset
Dependent Parameter	<NetKey Index : [2 Bytes of NetKey Index]> [in HEX] <Identity : [1 Byte of Identity]> [in HEX]
Note	The "Publish Address" needs to be set before issuing this command while testing with PTS. All values to be populated in HEX as prompted by the PTS test tool during the particular test execution.

39. Configuration Client Node Reset

Follow the below CLI command to send the Configuration Client Node Reset message to the Mesh Network.

Command Syntax	root->model->modelc->config->nodereset
Sample Command	root->model->modelc->config->nodereset
Dependent Parameter	NA
Note	The "Publish Address" needs to be set before issuing this command while testing with PTS.

40. Configuration Client Heartbeat Publication GET

Follow the below CLI command to send the Configuration Client Heartbeat Publication GET message to the Mesh Network.

Command Syntax	root->model->modelc->config->heartbeatpublicationget
Sample Command	root->model->modelc->config->heartbeatpublicationget
Dependent Parameter	NA
Note	The "Publish Address" needs to be set before issuing this command while testing with PTS.

41. Configuration Client Heartbeat Publication SET

Follow the below CLI command to send the Configuration Client Heartbeat Publication SET message to the Mesh Network.

Command Syntax	root->model->modelc->config->heartbeatpublicationset
Sample Command	root->model->modelc->config->heartbeatpublicationset
Dependent Parameter	<Destination : [2 Bytes of Destination Address]> [in HEX] <Count Log : [1 Byte of Count Log]> [in HEX] <Period Log : [1 Byte of Period Log]> [in HEX] <TTL : [1 Byte of TTL]> [in HEX] <Features : [2 Bytes of Features]> [in HEX] <NetKey Index : [2 Bytes of NetKey Index]> [in HEX]
Note	The "Publish Address" needs to be set before issuing this command while testing with PTS. All values to be populated in HEX as prompted by the PTS test tool during the particular test execution.

42. Configuration Client Heartbeat Subscription GET

Follow the below CLI command to send the Configuration Client Heartbeat Subscription GET message to the Mesh Network.

Command Syntax	root->model->modelc->config->heartbeatsubscriptionget
Sample Command	root->model->modelc->config->heartbeatsubscriptionget
Dependent Parameter	NA
Note	The "Publish Address" needs to be set before issuing this command while testing with PTS.

43. Configuration Client Heartbeat Subscription SET

Follow the below CLI command to send the Configuration Client Heartbeat Subscription SET message to the Mesh Network.

Command Syntax	root->model->modelc->config->heartbeatsubscriptionset

Sample Command	root->model->modelc->config->heartbeatsubscriptionset
Dependent Parameter	<Source : [2 Bytes of Source Address]> [in HEX] <Destination : [2 Bytes of Destination Address]> [in HEX] <Period Log : [1 Byte of Period Log]> [in HEX]
Note	The "Publish Address" needs to be set before issuing this command while testing with PTS. All values to be populated in HEX as prompted by the PTS test tool during the particular test execution.

44. Configuration Client Network Transmit GET

Follow the below CLI command to send the Configuration Client Network Transmit GET message to the Mesh Network.

Command Syntax	root->model->modelc->config->networktransmitget
Sample Command	root->model->modelc->config->networktransmitget
Dependent Parameter	NA
Note	The "Publish Address" needs to be set before issuing this command while testing with PTS.

45. Configuration Client Network Transmit SET

Follow the below CLI command to send the Configuration Client Network Transmit SET message to the Mesh Network.

Command Syntax	root->model->modelc->config->heartbeatsubscriptionset
Sample Command	root->model->modelc->config->heartbeatsubscriptionset
Dependent Parameter	<TX Count : [2 Bytes of Network TX Count]> [in HEX] <TX interval Steps : [2 Bytes of Network TX Interval Steps]> [in HEX]
Note	The "Publish Address" needs to be set before issuing this command while testing with PTS. All values to be populated in HEX as prompted by the PTS test tool during the particular test execution.

46. Configuration Client LPN Poll Timeout GET

Follow the below CLI command to send the Configuration Client LPN Poll Timeout GET message to the Mesh Network.

Command Syntax	root->model->modelc->config->lpnpolltimeoutget
Sample Command	root->model->modelc->config->lpnpolltimeoutget
Dependent Parameter	<LPN Address : [2 Bytes of LPN Address]> [in HEX]
Note	The "Publish Address" needs to be set before issuing this command while testing with PTS. All values to be populated in HEX as prompted by the PTS test tool during the particular test execution.

47. Configuration Client Key Refresh Phase GET

Follow the below CLI command to send the Configuration Client Key Refresh Phase GET message to the Mesh Network.

Command Syntax	root->model->modelc->config->keyrefreshphaseget
Sample Command	root->model->modelc->config->keyrefreshphaseget
Dependent Parameter	<NetKey Index : [2 Bytes of NetKey Index]> [in HEX]
Note	The "Publish Address" needs to be set before issuing this command while testing with PTS. All values to be populated in HEX as prompted by the PTS test tool during the particular test execution.

48. Configuration Client Key Refresh Phase SET

Follow the below CLI command to send the Configuration Client Key Refresh Phase SET message to the Mesh Network.

Command Syntax	root->model->modelc->config->keyrefreshphaseset
Sample Command	root->model->modelc->config->keyrefreshphaseset
Dependent	<NetKey Index : [2 Bytes of NetKey Index]> [in HEX]

Parameter	<Transition : [1 Byte of Transition]> [in HEX]
Note	The "Publish Address" needs to be set before issuing this command while testing with PTS. All values to be populated in HEX as prompted by the PTS test tool during the particular test execution.

4.3.2. Configuration Client PTS Test Procedures

To test the above-mentioned Foundation Model Configuration Client, follow the below given procedures:

1. Reset and power cycle/ Re-launch the IUT.
2. Setup and Initialize the IUT [if needed by the platform]
3. Register the Foundation Model Servers.
4. Register the specific Foundation Model Client, which is of interest. For example, to register the Configuration model client follow the instructions from here.
5. Setup the IUT [Board/executable] as a "New Device" which is to be provisioned to a Mesh Network.
6. Run the corresponding "Configuration Model" test section from PTS software. For example, in this case run the "MESH->Configuration Model->MESH/CFGCL/CFG/<xyz>" tests from PTS.
7. Depending upon the individual test case procedures, one might have to use the specific Configuration Client related procedures as prompted by PTS. For example, in Configuration model client to issue a Beacon GET message follow the procedures as mentioned here.

4.4. Health Client Procedures

This section covers the CLI commands needed to setup IUT as a Health Client and test the corresponding Health Client testcases with the PTS test tool.

4.4.1. Health Client CLI Procedures

1. Health Client Model Set Publish Address

Follow the below CLI command to set the desired Publish address to the Health Client from the application.

Command Syntax	root->model->modelc->health->publishaddr
Sample Command	root->model->modelc->health->publishaddr 1 0
Dependent Parameter	<Publish Address : [2 Bytes of Address]> <App Key ID : [2 Byte of App Key Index to used]>
Note	In the PTS testing of Health Client, the Publish Address is usually the Unicast Address of

	the PTS device(which is the Provisioner and the Configuration Client) and its address is typically 0x0001(Unicast Address of Provisioner/PTS).

2. Health Client Period GET

Follow the below CLI command to send the Health Client Period GET message to the Mesh Network.

Command Syntax	root->model->modelc->health->periodget
Sample Command	root->model->modelc->health->periodget
Dependent Parameter	NA
Note	The "Publish Address" needs to be set before issuing this command while testing with PTS.

3. Health Client Period SET

Follow the below CLI command to send the Health Client SET-ACKNOWLEDGED message to the Mesh Network.

Command Syntax	root->model->modelc->health->periodset
Sample Command	root->model->modelc->health->periodset 1
Dependent Parameter	<Fast Period Divisor : [1 Byte of Fast Period Divisor]> [in HEX]
Note	The "Publish Address" needs to be set before issuing this command while testing with PTS. All values to be populated in HEX as prompted by the PTS test tool during the particular test execution.

4. Health Client Period SET Unacknowledged

Follow the below CLI command to send the Health Client SET-UNACKNOWLEDGED message to the Mesh Network.

Command Syntax	root->model->modelc-> health->periodsetun
Sample Command	root->model->modelc-> health->periodsetun 1

Dependent Parameter	<Fast Period Divisor : [1 Byte of Fast Period Divisor]> [in HEX]
Note	The "Publish Address" needs to be set before issuing this command while testing with PTS. All values to be populated in HEX as prompted by the PTS test tool during the particular test execution.

5. Health Client Fault GET

Follow the below CLI command to send the Health Client Fault GET message to the Mesh Network.

Command Syntax	root->model->modelc->health->faultget
Sample Command	root->model->modelc->health->faultget 1
Dependent Parameter	<Company ID : [2 Bytes of Company ID]> [in HEX]
Note	The "Publish Address" needs to be set before issuing this command while testing with PTS.

6. Health Client Fault Test

Follow the below CLI command to send the Health Client Fault Test ACKNOWLEDGED message to the Mesh Network.

Command Syntax	root->model->modelc->health->faulttest
Sample Command	root->model->modelc->health->faulttest 1 1
Dependent Parameter	<Test ID : [1 Byte of Test ID]> [in HEX] <Company ID : [2 Bytes of Company ID]> [in HEX]
Note	The "Publish Address" needs to be set before issuing this command while testing with PTS. All values to be populated in HEX as prompted by the PTS test tool during the particular test execution.

7. Health Client Fault Test Unacknowledged

Follow the below CLI command to send the Health Client Fault Test UNACKNOWLEDGED message to the Mesh Network.

Command Syntax	root->model->modelc->health->faulttestun
Sample Command	root->model->modelc->health->faulttestun 1 1
Dependent Parameter	<Test ID : [1 Byte of Test ID]> [in HEX] <Company ID : [2 Bytes of Company ID]> [in HEX]
Note	The "Publish Address" needs to be set before issuing this command while testing with PTS. All values to be populated in HEX as prompted by the PTS test tool during the particular test execution.

8. Health Client Fault Clear

Follow the below CLI command to send the Health Client Fault Clear ACKNOWLEDGED message to the Mesh Network.

Command Syntax	root->model->modelc->health->faultclear
Sample Command	root->model->modelc->health->faultclear 1
Dependent Parameter	<Company ID : [2 Bytes of Company ID]> [in HEX]
Note	The "Publish Address" needs to be set before issuing this command while testing with PTS. All values to be populated in HEX as prompted by the PTS test tool during the particular test execution.

9. Health Client Fault Clear Unacknowledged

Follow the below CLI command to send the Health Client Fault Clear UNACKNOWLEDGED message to the Mesh Network.

Command Syntax	root->model->modelc->health->faultclearun
Sample Command	root->model->modelc->health->faultclearun 1
Dependent Parameter	<Company ID : [2 Bytes of Company ID]> [in HEX]
Note	The "Publish Address" needs to be set before issuing this command while testing with PTS.

	All values to be populated in HEX as prompted by the PTS test tool during the particular test execution.

10. Health Client Attention GET

Follow the below CLI command to send the Health Client Fault GET message to the Mesh Network.

Command Syntax	root->model->modelc->health->attentionget
Sample Command	root->model->modelc->health->attentionget
Dependent Parameter	NA
Note	The "Publish Address" needs to be set before issuing this command while testing with PTS.

11. Health Client Attention SET

Follow the below CLI command to send the Health Client Attention SET ACKNOWLEDGED message to the Mesh Network.

Command Syntax	root->model->modelc->health->attentionset
Sample Command	root->model->modelc->health->attentionset 1
Dependent Parameter	<Attention : [1 Byte of Attention]> [in HEX]
Note	The "Publish Address" needs to be set before issuing this command while testing with PTS. All values to be populated in HEX as prompted by the PTS test tool during the particular test execution.

12. Health Client Attention SET Unacknowledged

Follow the below CLI command to send the Health Client Attention SET UNACKNOWLEDGED message to the Mesh Network.

Command Syntax	root->model->modelc->health->attentionsetun
Sample	root->model->modelc->health->attentionsetun 1

Command	
Dependent Parameter	<Attention : [1 Byte of Attention]> [in HEX]
Note	The "Publish Address" needs to be set before issuing this command while testing with PTS. All values to be populated in HEX as prompted by the PTS test tool during the particular test execution.

4.4.2. Health Client PTS Test Procedures

To test the above-mentioned Foundation Model Health Client, follow the below given procedures:

1. Reset and power cycle/ Re-launch the IUT.
2. Setup and Initialize the IUT [if needed by the platform]
3. Register the Foundation Model Servers.
4. Register the specific Foundation Model Client, which is of interest. For example, to register the Health model client follow the instructions from here.
5. Setup the IUT [Board/executable] as a "New Device" which is to be provisioned to a Mesh Network.
6. Run the corresponding "Health Model" test section from PTS software. For example, in this case run the "MESH->Health Model->MESH/CL/HM/<xyz>" tests from PTS.
7. Depending upon the individual test case procedures, one might have to use the specific Health Client related procedures as prompted by PTS. For example, in Health model client to issue a Period GET message follow the procedures as mentioned here.

5. Model Server Testing

This section captures the CLI command sequences needed to setup IUT as a Mesh Model Server and run tests against PTS.

5.1. Model Server Setup Procedures

This section covers the CLI commands needed to setup IUT as a Model server and test the corresponding Model Server testcases with the PTS test tool.

5.1.1. Generic ONOFF Model Registration

After registering with the Foundation Model Servers [as mentioned here], follow the below CLI command to register for Generic ONOFF Model Server and its other associated models.

Command Syntax	root->model->models->generics->onoff
Sample Command	root->model->models->generics->onoff
Dependent Parameter	NA
Note	NA

5.1.2. Generic Level Model Registration

After registering with the Foundation Model Servers [as mentioned here], follow the below CLI command to register for Generic Level Model Server and its other associated models

Command Syntax	root->model->models->generics->level
Sample Command	root->model->models->generics->level
Dependent Parameter	NA
Note	NA

5.1.3. Generic Default Transition Time Model Registration

After registering with the Foundation Model Servers [as mentioned here], follow the below CLI command to register for Generic Default Transition Time Model Server and its other associated models

Command Syntax	root->model->models->generics->transitiontime
Sample Command	root->model->models->generics->transitiontime
Dependent Parameter	NA
Note	NA

5.1.4. Generic Power ONOFF Model Registration

After registering with the Foundation Model Servers [as mentioned here], follow the below CLI command to register for Generic Power ONOFF Model Server and its other associated models

Command Syntax	root->model->models->generics->poweronoff
Sample Command	root->model->models->generics->poweronoff
Dependent Parameter	NA
Note	NA

5.1.5. Generic Power Level Model Registration

After registering with the Foundation Model Servers [as mentioned here], follow the below CLI command to register for Generic Power Level Model Server and its other associated models

Command Syntax	root->model->models->generics->powerlevel
Sample Command	root->model->models->generics->powerlevel
Dependent Parameter	NA
Note	NA

5.1.6. Generic Battery Model Registration

After registering with the Foundation Model Servers [as mentioned here], follow the below CLI command to register for Generic Battery Model Server and its other associated models

Command Syntax	root->model->models->generics->battery
Sample Command	root->model->models->generics->battery
Dependent Parameter	NA
Note	NA

5.1.7. Generic Location Model Registration

After registering with the Foundation Model Servers [as mentioned here], follow the below CLI command to register for Generic Location Model Server and its other associated models

Command Syntax	root->model->models->generics->location
Sample Command	root->model->models->generics->location
Dependent Parameter	NA
Note	NA

5.1.8. Generic Property Model Registration

After registering with the Foundation Model Servers [as mentioned here], follow the below CLI command to register for Generic Property (User, Admin, Manufacturer and Client) Model Server and its other associated models

Command Syntax	root->model->models->generics->property
Sample Command	root->model->models->generics->property
Dependent Parameter	NA
Note	NA

5.1.9. Scene Model Registration

After registering with the Foundation Model Servers [as mentioned here], follow the below CLI command to register for Scene Model Server and Scene Setup Model Server.

Command Syntax	root->model->models->scene
Sample Command	root->model->models->scene
Dependent Parameter	NA
Note	NA

5.1.10. Light Lightness Model Registration

After registering with the Foundation Model Servers [as mentioned here], follow the below CLI command to register for Light Lightness Model Server and its other associated models.

Command Syntax	root->model->models->light->lightness
Sample Command	root->model->models->light->lightness
Dependent Parameter	NA
Note	NA

5.1.11. Light CTL Model Registration

After registering with the Foundation Model Servers [as mentioned here], follow the below CLI command to register for Light CTL Model Server and its other associated models.

Command Syntax	root->model->models->light->ctl
Sample Command	root->model->models->light->ctl
Dependent Parameter	NA
Note	NA

5.1.12. Light HSL Model Registration

After registering with the Foundation Model Servers [as mentioned here], follow the below CLI command to register for Light HSL Model Server and its other associated models.

Command Syntax	root->model->models->light->hsl
Sample Command	root->model->models->light->hsl
Dependent Parameter	NA
Note	NA

5.1.13. Light xyL Model Registration

After registering with the Foundation Model Servers [as mentioned here], follow the below CLI command to register for Light xyL Model Server and its other associated models.

Command Syntax	root->model->models->light->xyl
Sample Command	root->model->models->light->xyl
Dependent Parameter	NA
Note	NA

5.1.14. Light LC Model Registration

After registering with the Foundation Model Servers [as mentioned here], follow the below CLI command to register for Light LC Model Server and its other associated models.

Command Syntax	root->model->models->light->lc
Sample Command	root->model->models->light->lc
Dependent Parameter	NA
Note	NA

5.1.15. Sensor Model Registration

After registering with the Foundation Model Servers [as mentioned here], follow the below CLI command to register for Sensor Model Server and its other associated models.

Command Syntax	root->model->models->sensor
Sample Command	root->model->models->sensor
Dependent Parameter	NA
Note	NA

5.1.16. Time Model Registration

After registering with the Foundation Model Servers [as mentioned here], follow the below CLI command to register for Time Model Server and its other associated models.

Command Syntax	root->model->models->time
Sample Command	root->model->models->time
Dependent Parameter	NA
Note	NA

5.1.17. Scheduler Model Registration

After registering with the Foundation Model Servers [as mentioned here], follow the below CLI command to register for Scheduler Model Server and its other associated models.

Command Syntax	root->model->models->scheduler
Sample Command	root->model->models->scheduler
Dependent Parameter	NA
Note	NA

5.2.Model Server Test Procedures

This section covers the CLI commands needed to test the corresponding Model Server testcases with the PTS test tool.

5.2.1. Model server test steps

To test each of the above mentioned Model Servers individually, follow the below given procedures:

1. Reset and power cycle/ Re-launch the IUT.

2. Setup and Initialize the IUT [if needed by the platform]

3. Register the Foundation Model Servers.

4. Register the specific Model Server which is of interest. For example, to register the Generic ONOFF model server follow the instructions from here.

5. Setup the IUT [Board/executable] as a "New Device" which is to be provisioned to a Mesh Network.

6. Run the corresponding "MMDL" test section from PTS software. For example, for the Generic ONOFF model server run the "MMDL->Server->Generic OnOff" test group from PTS.

NOTE: In most of the Model Server PTS testing, the test procedures after the above steps are automatically handled. But, there could be a few testcases pertaining to scenarios where IUT [Board/executable] needs to be power cycled. In the event of power cycle, the corresponding model that is being tested needs to be registered, i.e. follow the procedure mentioned above from "Step 2" through "Step 4". The "step 1" and "step 5" will not be needed as the IUT is already provisioned hence storage clear and restart as new device is not required.

6. Model Client Testing

This section captures the CLI command sequences needed to setup IUT as a Mesh Model Client and run tests against PTS.

6.1. Model Client Procedures

This section covers the CLI commands needed to setup IUT as a Model Client and test the corresponding Model Client testcases with the PTS test tool.

6.1.1. Generic ONOFF Model

1. Generic ONOFF Model Registration

After registering with the Foundation Model Servers [as mentioned here], follow the below CLI command to register for Generic ONOFF Model Client.

Command Syntax	root->model->modelc->onoff->setup
Sample Command	root->model->modelc->onoff->setup
Dependent Parameter	NA
Note	NA

2. Generic ONOFF Model Set Publish Address

Follow the below CLI command to set the desired Publish address to the Generic ONOFF Model Client from the application.

Command Syntax	root->model->modelc->onoff->publishaddr
Sample Command	root->model->modelc->onoff-> publishaddr 1 0
Dependent Parameter	<Publish Address : [2 Bytes of Address]> <App Key ID : [2 Byte of App Key Index to used]>
Note	In the PTS testing of Model Client, the Publish Address is usually the Unicast Address of the PTS device(which is the Provisioner and the Configuration Client) and its address is typically 0x0001(Unicast Address of Provisioner/PTS).

3. Generic ONOFF Set

Follow the below CLI command to send the SET-ACKNOWLEDGED message to the Mesh Network.

Command Syntax	root->model->modelc->onoff->set
Sample Command	root->model->modelc->onoff->set 1 0 54 5
Dependent Parameter	Mandatory Parameters - <ONOFF Val : [1 Bytes of ON OFF State Value]> [in HEX] <TID : [1 Byte of TID]> [in HEX] Optional Parameters – <Transition Time : [1 Byte of Transition Time]> [in HEX] <Delay : [1 Byte of Delay]> [in HEX]
Note	The "Publish Address" needs to be set before issuing this command while testing with PTS. All values to be populated in HEX as prompted by the PTS test tool during the particular test execution.

4. Generic ONOFF Set Unacknowledged

Follow the below CLI command to send the SET-UNACKNOWLEDGED message to the Mesh Network.

Command Syntax	root->model->modelc->onoff->setun
Sample Command	root->model->modelc->onoff->setun 1 0 54 5
Dependent Parameter	Mandatory Parameters - <ONOFF Val : [1 Bytes of ON OFF State Value]> [in HEX] <TID : [1 Byte of TID]> [in HEX] Optional Parameters – <Transition Time : [1 Byte of Transition Time]> [in HEX] <Delay : [1 Byte of Delay]> [in HEX]
Note	The "Publish Address" needs to be set before issuing this command while testing with PTS. All values to be populated in HEX as prompted by the PTS test tool during the particular test execution.

5. Generic ONOFF Get

Follow the below CLI command to send the GET message to the Mesh Network.

Command Syntax	root->model->modelc->onoff->get
Sample Command	root->model->modelc->onoff->get
Dependent Parameter	NA
Note	The "Publish Address" needs to be set before issuing this command while testing with PTS.

6.1.2. Generic Level Model

1. Generic Level Model Registration

After registering with the Foundation Model Servers [as mentioned here], follow the below CLI command to register for Generic Level Model Client.

Command Syntax	root->model->modelc->level->setup
Sample Command	root->model->modelc->level->setup
Dependent Parameter	NA
Note	NA

2. Generic Level Model Set Publish Address

Follow the below CLI command to set the desired Publish address to the Generic Level Model Client from the application.

Command Syntax	root->model->modelc->level->publishaddr
Sample Command	root->model->modelc->level-> publishaddr 1 0
Dependent Parameter	<Publish Address : [2 Bytes of Address]> <App Key ID : [2 Byte of App Key Index to used]>
Note	In the PTS testing of Model Client, the Publish Address is usually the Unicast Address of the PTS device(which is the Provisioner and the Configuration Client) and its address is

	typically 0x0001(Unicast Address of Provisioner/PTS).

3. Generic Level Set

Follow the below CLI command to send the Generic Level SET-ACKNOWLEDGED message to the Mesh Network.

Command Syntax	root->model->modelc->level->levelset
Sample Command	root->model->modelc->level->levelset 1FF 0 54 5
Dependent Parameter	Mandatory Parameters - <Level Val : [2 Bytes of Level State Value]> [in HEX] <TID : [1 Byte of TID]> [in HEX] Optional Parameters – <Transition Time : [1 Byte of Transition Time]> [in HEX] <Delay : [1 Byte of Delay]> [in HEX]
Note	The "Publish Address" needs to be set before issuing this command while testing with PTS. All values to be populated in HEX as prompted by the PTS test tool during the particular test execution.

4. Generic Level Set Unacknowledged

Follow the below CLI command to send the Generic Level SET-UNACKNOWLEDGED message to the Mesh Network.

Command Syntax	root->model->modelc->level->levelsetun
Sample Command	root->model->modelc->level->levelsetun 1FF 0 54 5
Dependent Parameter	Mandatory Parameters - <Level Val : [2 Bytes of Level State Value]> [in HEX] <TID : [1 Byte of TID]> [in HEX] Optional Parameters – <Transition Time : [1 Byte of Transition Time]> [in HEX] <Delay : [1 Byte of Delay]> [in HEX]

Note	The "Publish Address" needs to be set before issuing this command while testing with PTS. All values to be populated in HEX as prompted by the PTS test tool during the particular test execution.

5. Generic Level Get

Follow the below CLI command to send the Generic Level GET message to the Mesh Network.

Command Syntax	root->model->modelc->level->levelget
Sample Command	root->model->modelc->level->levelget
Dependent Parameter	NA
Note	The "Publish Address" needs to be set before issuing this command while testing with PTS.

6. Generic Delta Set

Follow the below CLI command to send the Generic Delta SET-ACKNOWLEDGED message to the Mesh Network.

Command Syntax	root->model->modelc->level->deltaset
Sample Command	root->model->modelc->level->deltaset 1FF 0 54 5
Dependent Parameter	Mandatory Parameters - <Delta Level : [4 Bytes of Delta Level Value]> [in HEX] <TID : [1 Byte of TID]> [in HEX] Optional Parameters – <Transition Time : [1 Byte of Transition Time]> [in HEX] <Delay : [1 Byte of Delay]> [in HEX]
Note	The "Publish Address" needs to be set before issuing this command while testing with PTS. All values to be populated in HEX as prompted by the PTS test tool during the particular test execution.

7. Generic Delta Set Unacknowledged

Follow the below CLI command to send the Generic Delta SET-UNACKNOWLEDGED message to the Mesh Network.

Command Syntax	root->model->modelc->level->deltasetun
Sample Command	root->model->modelc->level-> deltasetun 1FF 0 54 5
Dependent Parameter	Mandatory Parameters - <Delta Level : [4 Bytes of Delta Level Value]> [in HEX] <TID : [1 Byte of TID]> [in HEX] Optional Parameters – <Transition Time : [1 Byte of Transition Time]> [in HEX] <Delay : [1 Byte of Delay]> [in HEX]
Note	The "Publish Address" needs to be set before issuing this command while testing with PTS. All values to be populated in HEX as prompted by the PTS test tool during the particular test execution.

8. Generic Move Set

Follow the below CLI command to send the Generic Move SET-ACKNOWLEDGED message to the Mesh Network.

Command Syntax	root->model->modelc->level->moveset
Sample Command	root->model->modelc->level->moveset 1FF 0
Dependent Parameter	<Delta Level : [2 Bytes of Delta Level Value]> [in HEX] <TID : [1 Byte of TID]> [in HEX]
Note	The "Publish Address" needs to be set before issuing this command while testing with PTS. All values to be populated in HEX as prompted by the PTS test tool during the particular test execution.

9. Generic Move Set Unacknowledged

Follow the below CLI command to send the Generic Move SET-UNACKNOWLEDGED message to the Mesh Network.

Command Syntax	root->model->modelc->level->movesetun
Sample Command	root->model->modelc->level->movesetun 1FF 0
Dependent Parameter	<Delta Level : [2 Bytes of Delta Level Value]> [in HEX] <TID : [1 Byte of TID]> [in HEX]
Note	The "Publish Address" needs to be set before issuing this command while testing with PTS. All values to be populated in HEX as prompted by the PTS test tool during the particular test execution.

6.1.3. Generic Default Transition Time Model

1. Generic Default Transition Time Model Registration

After registering with the Foundation Model Servers [as mentioned here], follow the below CLI command to register for Generic Default Transition Time Model Client.

Command Syntax	root->model->modelc->transitiontime->setup
Sample Command	root->model->modelc->transitiontime->setup
Dependent Parameter	NA
Note	NA

2. Generic Default Transition Time Model Set Publish Address

Follow the below CLI command to set the desired Publish address to the Generic Default Transition Time Model Client from the application.

Command Syntax	root->model->modelc->transitiontime->publishaddr
Sample Command	root->model->modelc->transitiontime->publishaddr 1 0

Dependent Parameter	<Publish Address : [2 Bytes of Address]> <App Key ID : [2 Byte of App Key Index to used]>
Note	In the PTS testing of Model Client, the Publish Address is usually the Unicast Address of the PTS device(which is the Provisioner and the Configuration Client) and its address is typically 0x0001(Unicast Address of Provisioner/PTS).

3. Generic Default Transition Time Get

Follow the below CLI command to send the Generic Default Transition Time GET message to the Mesh Network.

Command Syntax	root->model->modelc->transitiontime->get
Sample Command	root->model->modelc->transitiontime->get
Dependent Parameter	NA
Note	The "Publish Address" needs to be set before issuing this command while testing with PTS.

4. Generic Default Transition Time Set

Follow the below CLI command to send the Generic Default Transition Time SET-ACKNOWLEDGED message to the Mesh Network.

Command Syntax	root->model->modelc->transitiontime->set
Sample Command	root->model->modelc->transitiontime->set 1 1
Dependent Parameter	<Number of Steps : [1 Byte of Transition Number of Steps]> [in HEX] <Step Resolution : [1 Byte of Step Resolution]> [in HEX]
Note	The "Publish Address" needs to be set before issuing this command while testing with PTS. All values to be populated in HEX as prompted by the PTS test tool during the particular test execution.

5. Generic Default Transition Time Set Unacknowledged

Follow the below CLI command to send the Generic Default Transition Time SET-UNACKNOWLEDGED message to the Mesh Network.

Command Syntax	root->model->modelc->transitiontime->setun
Sample Command	root->model->modelc->transitiontime->setun 1 1
Dependent Parameter	<Number of Steps : [1 Byte of Transition Number of Steps]> [in HEX] <Step Resolution : [1 Byte of Step Resolution]> [in HEX]
Note	The "Publish Address" needs to be set before issuing this command while testing with PTS. All values to be populated in HEX as prompted by the PTS test tool during the particular test execution.

6.1.4. Generic Power ONOFF Model

1. Generic Power ONOFF Model Registration

After registering with the Foundation Model Servers [as mentioned here], follow the below CLI command to register for Generic Power ONOFF Model Client.

Command Syntax	root->model->modelc->poweronoff->setup
Sample Command	root->model->modelc->poweronoff->setup
Dependent Parameter	NA
Note	NA

2. Generic Power ONOFF Model Set Publish Address

Follow the below CLI command to set the desired Publish address to the Generic Power ONOFF Model Client from the application.

Command Syntax	root->model->modelc->poweronoff->publishaddr
Sample Command	root->model->modelc->poweronoff->publishaddr 1 0
Dependent Parameter	<Publish Address : [2 Bytes of Address]> <App Key ID : [2 Byte of App Key Index to used]>
Note	In the PTS testing of Model Client, the Publish Address is usually the Unicast Address of the PTS device(which is the Provisioner and the Configuration Client) and its address is

	typically 0x0001(Unicast Address of Provisioner/PTS).

3. Generic Power ONOFF on Power up Get

Follow the below CLI command to send the Generic Power ONOFF on Power up GET message to the Mesh Network.

Command Syntax	root->model->modelc->poweronoff->get
Sample Command	root->model->modelc->poweronoff->get
Dependent Parameter	NA
Note	The "Publish Address" needs to be set before issuing this command while testing with PTS.

4. Generic Power ONOFF on Power up Set

Follow the below CLI command to send the Generic Power ONOFF on Power up SET-ACKNOWLEDGED message to the Mesh Network.

Command Syntax	root->model->modelc->poweronoff->set
Sample Command	root->model->modelc->poweronoff->set 1
Dependent Parameter	<On PowerUP : [1 Byte of On Power UP]> [in HEX]
Note	The "Publish Address" needs to be set before issuing this command while testing with PTS. All values to be populated in HEX as prompted by the PTS test tool during the particular test execution.

5. Generic Power ONOFF on Power up Set Unacknowledged

Follow the below CLI command to send the Generic Power ONOFF on Power up SET-UNACKNOWLEDGED message to the Mesh Network.

Command Syntax	root->model->modelc->poweronoff->setun
Sample Command	root->model->modelc->poweronoff->setun 1

Dependent Parameter	<On PowerUP : [1 Byte of On Power UP]> [in HEX]
Note	The "Publish Address" needs to be set before issuing this command while testing with PTS. All values to be populated in HEX as prompted by the PTS test tool during the particular test execution.

6.1.5. Generic Power Level Model

1. Generic Power Level Model Registration

After registering with the Foundation Model Servers [as mentioned here], follow the below CLI command to register for Generic Power Level Model Client.

Command Syntax	root->model->modelc->powerlevel->setup
Sample Command	root->model->modelc->powerlevel->setup
Dependent Parameter	NA
Note	NA

2. Generic Power Level Model Set Publish Address

Follow the below CLI command to set the desired Publish address to the Generic Power Level Model Client from the application.

Command Syntax	root->model->modelc->powerlevel->publishaddr
Sample Command	root->model->modelc->powerlevel->publishaddr 1 0
Dependent Parameter	<Publish Address : [2 Bytes of Address]> <App Key ID : [2 Byte of App Key Index to used]>
Note	In the PTS testing of Model Client, the Publish Address is usually the Unicast Address of the PTS device(which is the Provisioner and the Configuration Client) and its address is typically 0x0001(Unicast Address of Provisioner/PTS).

3. Generic Power Level Get

Follow the below CLI command to send the Generic Power Level GET message to the Mesh Network.

Command Syntax	root->model->modelc->powerlevel->levelget
Sample Command	root->model->modelc->powerlevel->levelget
Dependent Parameter	NA
Note	The "Publish Address" needs to be set before issuing this command while testing with PTS.

4. Generic Power Level Set

Follow the below CLI command to send the Generic Power Level SET-ACKNOWLEDGED message to the Mesh Network.

Command Syntax	root->model->modelc->powerlevel->levelset
Sample Command	root->model->modelc->powerlevel->levelset 1 1 0 0
Dependent Parameter	Mandatory Parameters - <Power : [2 Bytes of Power Level]> [in HEX] <TID : [1 Byte of TID]> [in HEX] Optional Parameters - <Transition Time : [1 Byte of Transition Time]> [in HEX] <Delay : [1 Byte of Delay]> [in HEX]
Note	The "Publish Address" needs to be set before issuing this command while testing with PTS. All values to be populated in HEX as prompted by the PTS test tool during the particular test execution.

5. Generic Power Level Set Unacknowledged

Follow the below CLI command to send the Generic Power Level SET-UNACKNOWLEDGED message to the Mesh Network.

Command Syntax	root->model->modelc->powerlevel->levelsetun
Sample Command	root->model->modelc->powerlevel->levelsetun 1 1 0 0
Dependent	Mandatory Parameters -

Parameter	<Power : [2 Bytes of Power Level]> [in HEX] <TID : [1 Byte of TID]> [in HEX] Optional Parameters - <Transition Time : [1 Byte of Transition Time]> [in HEX] <Delay : [1 Byte of Delay]> [in HEX]
Note	The "Publish Address" needs to be set before issuing this command while testing with PTS. All values to be populated in HEX as prompted by the PTS test tool during the particular test execution.

6. Generic Power Level Default Get

Follow the below CLI command to send the Generic Power Level Default GET message to the Mesh Network.

Command Syntax	root->model->modelc->powerlevel->defaultget
Sample Command	root->model->modelc->powerlevel->defaultget
Dependent Parameter	NA
Note	The "Publish Address" needs to be set before issuing this command while testing with PTS.

7. Generic Power Level Default Set

Follow the below CLI command to send the Generic Power Level Default SET-ACKNOWLEDGED message to the Mesh Network.

Command Syntax	root->model->modelc->powerlevel->defaultset
Sample Command	root->model->modelc->powerlevel->defaultset 1
Dependent Parameter	<Power : [2 Bytes Power Level]> [in HEX]
Note	The "Publish Address" needs to be set before issuing this command while testing with PTS. All values to be populated in HEX as prompted by the PTS test tool during the particular test execution.

8. Generic Power Level Default Set Unacknowledged

Follow the below CLI command to send the Generic Power Level Default SET-UNACKNOWLEDGED message to the Mesh Network.

Command Syntax	root->model->modelc->powerlevel->defaultsetun
Sample Command	root->model->modelc->powerlevel->defaultsetun 1
Dependent Parameter	<Power : [2 Bytes Power Level]> [in HEX]
Note	The "Publish Address" needs to be set before issuing this command while testing with PTS. All values to be populated in HEX as prompted by the PTS test tool during the particular test execution.

9. Generic Power Level Last Get

Follow the below CLI command to send the Generic Power Level Last GET message to the Mesh Network.

Command Syntax	root->model->modelc->powerlevel->lastget
Sample Command	root->model->modelc->powerlevel->lastget
Dependent Parameter	NA
Note	The "Publish Address" needs to be set before issuing this command while testing with PTS.

10. Generic Power Level Range Get

Follow the below CLI command to send the Generic Power Level Range GET message to the Mesh Network.

Command Syntax	root->model->modelc->powerlevel->rangeget
Sample Command	root->model->modelc->powerlevel->rangeget
Dependent Parameter	NA
Note	The "Publish Address" needs to be set before issuing this command while testing with PTS.

11. Generic Power Level Range Set

Follow the below CLI command to send the Generic Power Level Range SET-ACKNOWLEDGED message to the Mesh Network.

Command Syntax	root->model->modelc->powerlevel->rangeset
Sample Command	root->model->modelc->powerlevel->rangeset 1 4
Dependent Parameter	<Range Minumum : [2 Bytes of minimum range]> [in HEX] <Range Maximum : [2 Bytes of maximum range]> [in HEX]
Note	The "Publish Address" needs to be set before issuing this command while testing with PTS. All values to be populated in HEX as prompted by the PTS test tool during the particular test execution.

12. Generic Power Level Range Set Unacknowledged

Follow the below CLI command to send the Generic Power Level Range SET-UNACKNOWLEDGED message to the Mesh Network.

Command Syntax	root->model->modelc->powerlevel->rangesetun
Sample Command	root->model->modelc->powerlevel->rangesetun 1 4
Dependent Parameter	<Range Minumum : [2 Bytes of minimum range]> [in HEX] <Range Maximum : [2 Bytes of maximum range]> [in HEX]
Note	The "Publish Address" needs to be set before issuing this command while testing with PTS. All values to be populated in HEX as prompted by the PTS test tool during the particular test execution.

6.1.6. Generic Battery Model

1. Generic Battery Model Registration

After registering with the Foundation Model Servers [as mentioned here], follow the below CLI command to register for Generic Battery Model Client.

Command Syntax	root->model->modelc->battery->setup

Sample Command	root->model->modelc->battery->setup
Dependent Parameter	NA
Note	NA

2. Generic Battery Model Set Publish Address

Follow the below CLI command to set the desired Publish address to the Generic Battery Model Client from the application.

Command Syntax	root->model->modelc->battery->publishaddr
Sample Command	root->model->modelc->battery->publishaddr 1 0
Dependent Parameter	<Publish Address : [2 Bytes of Address]> <App Key ID : [2 Byte of App Key Index to used]>
Note	In the PTS testing of Model Client, the Publish Address is usually the Unicast Address of the PTS device(which is the Provisioner and the Configuration Client) and its address is typically 0x0001(Unicast Address of Provisioner/PTS).

3. Generic Battery Get

Follow the below CLI command to send the Generic Battery GET message to the Mesh Network.

Command Syntax	root->model->modelc->battery->get
Sample Command	root->model->modelc->battery->get
Dependent Parameter	NA
Note	The "Publish Address" needs to be set before issuing this command while testing with PTS.

6.1.7. Generic Location Model

1. Generic Location Model Registration

After registering with the Foundation Model Servers [as mentioned here], follow the below CLI command to register for Generic Location Model Client.

Command Syntax	root->model->modelc->location->setup
Sample Command	root->model->modelc->location->setup
Dependent Parameter	NA
Note	NA

2. Generic Location Model Set Publish Address

Follow the below CLI command to set the desired Publish address to the Generic Location Model Client from the application.

Command Syntax	root->model->modelc->location->publishaddr
Sample Command	root->model->modelc->location->publishaddr 1 0
Dependent Parameter	<Publish Address : [2 Bytes of Address]> <App Key ID : [2 Byte of App Key Index to used]>
Note	In the PTS testing of Model Client, the Publish Address is usually the Unicast Address of the PTS device(which is the Provisioner and the Configuration Client) and its address is typically 0x0001(Unicast Address of Provisioner/PTS).

3. Generic Location Global Get

Follow the below CLI command to send the Generic Location Global GET message to the Mesh Network.

Command Syntax	root->model->modelc->location->globalget
Sample Command	root->model->modelc->location->globalget
Dependent Parameter	NA
Note	The "Publish Address" needs to be set before issuing this command while testing with PTS.

4. Generic Location Global Set

Follow the below CLI command to send the Generic Location Global SET-ACKNOWLEDGED message to the Mesh Network.

Command Syntax	root->model->modelc->location->globalset
Sample Command	root->model->modelc->location->globalset 0 0 0
Dependent Parameter	<Global Latitude : [4 Bytes of Global Latitude]> [in HEX] <Global Longitude : [4 Bytes of Global Longtitude]> [in HEX] <Global Altitude : [2 Bytes of Global Altitude]> [in HEX]
Note	The "Publish Address" needs to be set before issuing this command while testing with PTS. All values to be populated in HEX as prompted by the PTS test tool during the particular test execution.

5. Generic Location Global Set Unacknowledged

Follow the below CLI command to send the Generic Location Global SET-UNACKNOWLEDGED message to the Mesh Network.

Command Syntax	root->model->modelc->location->globalsetun
Sample Command	root->model->modelc->location->globalsetun 0 0 0
Dependent Parameter	<Global Latitude : [4 Bytes of Global Latitude]> [in HEX] <Global Longitude : [4 Bytes of Global Longtitude]> [in HEX] <Global Altitude : [2 Bytes of Global Altitude]> [in HEX]
Note	The "Publish Address" needs to be set before issuing this command while testing with PTS. All values to be populated in HEX as prompted by the PTS test tool during the particular test execution.

6. Generic Location Local Get

Follow the below CLI command to send the Generic Location Local GET message to the Mesh Network.

Command Syntax	root->model->modelc->location->localget
Sample Command	root->model->modelc->location->localget
Dependent Parameter	NA
Note	The "Publish Address" needs to be set before issuing this command while testing with PTS.

7. Generic Location Local Set

Follow the below CLI command to send the Generic Location Local SET-ACKNOWLEDGED message to the Mesh Network.

Command Syntax	root->model->modelc->location->localset
Sample Command	root->model->modelc->location->localset
Dependent Parameter	<Local North : [2 Bytes of Local North]> [in HEX] <Local East : [2 Bytes of Local East]> [in HEX] <Local Altitude : [2 Bytes of Local Altitude]> [in HEX] <Floor Number : [1 Byte of Floor Number]> [in HEX] <Uncertainty : [2 Bytes of Uncertainty]> [in HEX]
Note	The "Publish Address" needs to be set before issuing this command while testing with PTS. All values to be populated in HEX as prompted by the PTS test tool during the particular test execution.

8. Generic Location Local Set Unacknowledged

Follow the below CLI command to send the Generic Location Local SET-UNACKNOWLEDGED message to the Mesh Network.

Command Syntax	root->model->modelc->location->localsetun
Sample Command	root->model->modelc->location->localsetun
Dependent Parameter	<Local North : [2 Bytes of Local North]> [in HEX] <Local East : [2 Bytes of Local East]> [in HEX] <Local Altitude : [2 Bytes of Local Altitude]> [in HEX] <Floor Number : [1 Byte of Floor Number]> [in HEX] <Uncertainty : [2 Bytes of Uncertainty]> [in HEX]
Note	The "Publish Address" needs to be set before issuing this command while testing with PTS. All values to be populated in HEX as prompted by the PTS test tool during the particular test execution.

6.1.8. Generic Property Model

1. Generic Property Model Registration

After registering with the Foundation Model Servers [as mentioned here], follow the below CLI command to register for Generic Property Model Client.

Command Syntax	root->model->modelc->property->setup
Sample Command	root->model->modelc->property->setup
Dependent Parameter	NA
Note	NA

2. Generic Property Model Set Publish Address

Follow the below CLI command to set the desired Publish address to the Generic Property Model Client from the application.

Command Syntax	root->model->modelc->property->publishaddr
Sample Command	root->model->modelc->property->publishaddr 1 0
Dependent Parameter	<Publish Address : [2 Bytes of Address]> <App Key ID : [2 Byte of App Key Index to used]>
Note	In the PTS testing of Model Client, the Publish Address is usually the Unicast Address of the PTS device(which is the Provisioner and the Configuration Client) and its address is typically 0x0001(Unicast Address of Provisioner/PTS).

3. Generic Admin Properties Get

Follow the below CLI command to send the Generic Admin Properties GET message to the Mesh Network.

Command Syntax	root->model->modelc->property->adminpropertiesget
Sample Command	root->model->modelc->property->adminpropertiesget
Dependent	NA

Parameter	
Note	The "Publish Address" needs to be set before issuing this command while testing with PTS.

4. Generic Admin Property Get

Follow the below CLI command to send the Generic Admin Property GET message to the Mesh Network.

Command Syntax	root->model->modelc->property->adminpropertyget
Sample Command	root->model->modelc->property->adminpropertyget
Dependent Parameter	<Property ID : [2 Bytes of Admin Property ID]> [in HEX]
Note	The "Publish Address" needs to be set before issuing this command while testing with PTS. All values to be populated in HEX as prompted by the PTS test tool during the particular test execution.

5. Generic Admin Property Set

Follow the below CLI command to send the Generic Admin Property SET-ACKNOWLEDGED message to the Mesh Network.

Command Syntax	root->model->modelc->property->adminpropertyset
Sample Command	root->model->modelc->property->adminpropertyset
Dependent Parameter	<Admin Property ID : [2 Bytes of Admin Property ID]> [in HEX] <Admin User Access : [1 Byte of Admin User Access]> [in HEX] <Admin Property Value : [N Byte of Admin Property Value]> [in HEX]
Note	The "Publish Address" needs to be set before issuing this command while testing with PTS. All values to be populated in HEX as prompted by the PTS test tool during the particular test execution.

6. Generic Admin Property Set Unacknowledged

Follow the below CLI command to send the Generic Admin Property SET-UNACKNOWLEDGED message to the Mesh Network.

Command Syntax	root->model->modelc->property->adminpropertysetun
Sample Command	root->model->modelc->property->adminpropertysetun
Dependent Parameter	<Admin Property ID : [2 Bytes of Admin Property ID]> [in HEX] <Admin User Access : [1 Byte of Admin User Access]> [in HEX] <Admin Property Value : [N Byte of Admin Property Value]> [in HEX]
Note	The "Publish Address" needs to be set before issuing this command while testing with PTS. All values to be populated in HEX as prompted by the PTS test tool during the particular test execution.

7. Generic Manufacturer Properties Get

Follow the below CLI command to send the Generic Manufacturer Properties GET message to the Mesh Network.

Command Syntax	root->model->modelc->property->manufacturerpropertiesget
Sample Command	root->model->modelc->property->manufacturerpropertiesget
Dependent Parameter	NA
Note	The "Publish Address" needs to be set before issuing this command while testing with PTS.

8. Generic Manufacturer Property Get

Follow the below CLI command to send the Generic Manufacturer Property GET message to the Mesh Network.

Command Syntax	root->model->modelc->property->manufacturerpropertyget
Sample	root->model->modelc->property->manufacturerpropertyget

Command	
Dependent Parameter	<Manufacture Property ID : [2 Bytes of User Property ID]> [in HEX]
Note	The "Publish Address" needs to be set before issuing this command while testing with PTS. All values to be populated in HEX as prompted by the PTS test tool during the particular test execution.

9. Generic Manufacturer Property Set

Follow the below CLI command to send the Generic Manufacturer Property SET-ACKNOWLEDGED message to the Mesh Network.

Command Syntax	root->model->modelc->property->manufacturerpropertyset
Sample Command	root->model->modelc->property->manufacturerpropertyset
Dependent Parameter	<Manufacturer Property ID : [2 Bytes of Manufacturer Property ID]> [in HEX] <Manufacturer User Access : [1 Byte of Manufacturer User Access]> [in HEX]
Note	The "Publish Address" needs to be set before issuing this command while testing with PTS. All values to be populated in HEX as prompted by the PTS test tool during the particular test execution.

10. Generic Manufacturer Property Set Unacknowledged

Follow the below CLI command to send the Generic Manufacturer Property SET-UNACKNOWLEDGED message to the Mesh Network.

Command Syntax	root->model->modelc->property->manufacturerpropertysetun
Sample Command	root->model->modelc->property->manufacturerpropertysetun
Dependent Parameter	<Manufacturer Property ID : [2 Bytes of Manufacturer Property ID]> [in HEX] <Manufacturer User Access : [1 Byte of Manufacturer User Access]>

	[in HEX]
Note	The "Publish Address" needs to be set before issuing this command while testing with PTS. All values to be populated in HEX as prompted by the PTS test tool during the particular test execution.

11. Generic User Properties Get

Follow the below CLI command to send the Generic User Properties GET message to the Mesh Network.

Command Syntax	root->model->modelc->property->userpropertiesget
Sample Command	root->model->modelc->property->userpropertiesget
Dependent Parameter	NA
Note	The "Publish Address" needs to be set before issuing this command while testing with PTS.

12. Generic User Property Get

Follow the below CLI command to send the Generic User Property GET message to the Mesh Network.

Command Syntax	root->model->modelc->property-> userpropertyget
Sample Command	root->model->modelc->property-> userpropertyget
Dependent Parameter	<User Property ID : [2 Bytes of User Property ID]> [in HEX]
Note	The "Publish Address" needs to be set before issuing this command while testing with PTS. All values to be populated in HEX as prompted by the PTS test tool during the particular test execution.

13. Generic User Property Set

Follow the below CLI command to send the Generic User Property SET-ACKNOWLEDGED message to the Mesh Network.

Command Syntax	root->model->modelc->property->userpropertyset

Sample Command	root->model->modelc->property->userpropertyset
Dependent Parameter	<User Property ID : [2 Bytes of User Property ID]> [in HEX] <User Property Value : [N Byte of User Property Value]> [in HEX]
Note	The "Publish Address" needs to be set before issuing this command while testing with PTS. All values to be populated in HEX as prompted by the PTS test tool during the particular test execution.

14. Generic User Property Set Unacknowledged

Follow the below CLI command to send the Generic User Property SET-UNACKNOWLEDGED message to the Mesh Network.

Command Syntax	root->model->modelc->property->userpropertysetun
Sample Command	root->model->modelc->property->userpropertysetun
Dependent Parameter	<User Property ID : [2 Bytes of User Property ID]> [in HEX] <User Property Value : [N Byte of User Property Value]> [in HEX]
Note	The "Publish Address" needs to be set before issuing this command while testing with PTS. All values to be populated in HEX as prompted by the PTS test tool during the particular test execution.

15. Generic Client Properties Get

Follow the below CLI command to send the Generic User Properties GET message to the Mesh Network.

Command Syntax	root->model->modelc->property->clientpropertiesget
Sample Command	root->model->modelc->property->clientpropertiesget
Dependent Parameter	<Client Property ID : [2 Bytes of Client Property ID]> [in HEX]
Note	The "Publish Address" needs to be set before issuing this command while testing with PTS.

	All values to be populated in HEX as prompted by the PTS test tool during the particular test execution.

6.1.9. Scene Model

1. Scene Model Registration

After registering with the Foundation Model Servers [as mentioned here], follow the below CLI command to register for Scene Model Client.

Command Syntax	root->model->modelc->scene->setup
Sample Command	root->model->modelc->scene->setup
Dependent Parameter	NA
Note	NA

2. Scene Model Set Publish Address

Follow the below CLI command to set the desired Publish address to the Scene Model Client from the application.

Command Syntax	root->model->modelc->scene->publishaddr
Sample Command	root->model->modelc->scene->publishaddr 1 0
Dependent Parameter	<Publish Address : [2 Bytes of Address]> <App Key ID : [2 Byte of App Key Index to used]>
Note	In the PTS testing of Model Client, the Publish Address is usually the Unicast Address of the PTS device(which is the Provisioner and the Configuration Client) and its address is typically 0x0001(Unicast Address of Provisioner/PTS).

3. Scene Get

Follow the below CLI command to send the Scene GET message to the Mesh Network.

Command Syntax	root->model->modelc->scene->get
Sample	root->model->modelc->scene->get

Command	
Dependent Parameter	NA
Note	The "Publish Address" needs to be set before issuing this command while testing with PTS.

4. Scene Register Get

Follow the below CLI command to send the Scene Register GET message to the Mesh Network.

Command Syntax	root->model->modelc->scene->registerget
Sample Command	root->model->modelc->scene->registerget
Dependent Parameter	NA
Note	The "Publish Address" needs to be set before issuing this command while testing with PTS.

5. Scene Store

Follow the below CLI command to send the Scene Store ACKNOWLEDGED message to the Mesh Network.

Command Syntax	root->model->modelc->scene->store
Sample Command	root->model->modelc->scene->store
Dependent Parameter	<Scene Number : [2 Bytes of Scene Number]> [in HEX]
Note	The "Publish Address" needs to be set before issuing this command while testing with PTS. All values to be populated in HEX as prompted by the PTS test tool during the particular test execution.

6. Scene Store Unacknowledged

Follow the below CLI command to send the Scene Store UNACKNOWLEDGED message to the Mesh Network.

Command Syntax	root->model->modelc->scene->storeun

Sample Command	root->model->modelc->scene->storeun
Dependent Parameter	<Scene Number : [2 Bytes of Scene Number]> [in HEX]
Note	The "Publish Address" needs to be set before issuing this command while testing with PTS. All values to be populated in HEX as prompted by the PTS test tool during the particular test execution.

7. Scene Recall

Follow the below CLI command to send the Scene Recall ACKNOWLEDGED message to the Mesh Network.

Command Syntax	root->model->modelc->scene->recall
Sample Command	root->model->modelc->scene->recall
Dependent Parameter	Mandatory Parameters - <Scene Number : [2 Bytes of Scene Number]> [in HEX] <TID : [1 Byte of TID]> [in HEX] Optional Parameters - <Transition Time : [1 Byte of Transition Time]> [in HEX] <Delay : [1 Byte of Delay]> [in HEX]
Note	The "Publish Address" needs to be set before issuing this command while testing with PTS. All values to be populated in HEX as prompted by the PTS test tool during the particular test execution.

8. Scene Recall Unacknowledged

Follow the below CLI command to send the Scene Recall UNACKNOWLEDGED message to the Mesh Network.

Command Syntax	root->model->modelc->scene->recallun
Sample Command	root->model->modelc->scene->recallun
Dependent	Mandatory Parameters -

Parameter	<Scene Number : [2 Bytes of Scene Number]> [in HEX] <TID : [1 Byte of TID]> [in HEX] Optional Parameters - <Transition Time : [1 Byte of Transition Time]> [in HEX] <Delay : [1 Byte of Delay]> [in HEX]
Note	The "Publish Address" needs to be set before issuing this command while testing with PTS. All values to be populated in HEX as prompted by the PTS test tool during the particular test execution.

9. Scene Delete

Follow the below CLI command to send the Scene Delete ACKNOWLEDGED message to the Mesh Network.

Command Syntax	root->model->modelc->scene->delete
Sample Command	root->model->modelc->scene->delete
Dependent Parameter	<Scene Number : [2 Bytes of Scene Number]> [in HEX]
Note	The "Publish Address" needs to be set before issuing this command while testing with PTS. All values to be populated in HEX as prompted by the PTS test tool during the particular test execution.

10. Scene Delete Unackowledged

Follow the below CLI command to send the Scene Delete UNACKNOWLEDGED message to the Mesh Network.

Command Syntax	root->model->modelc->scene->deleteun
Sample Command	root->model->modelc->scene->deleteun
Dependent Parameter	<Scene Number : [2 Bytes of Scene Number]> [in HEX]
Note	The "Publish Address" needs to be set before issuing this command while testing with PTS. All values to be populated in HEX as prompted by the PTS test tool during the particular

	test execution.

6.1.10. Light Lightness Model

1. Light Lightness Model Registration

After registering with the Foundation Model Servers [as mentioned here], follow the below CLI command to register for Light Lightness Model Client.

Command Syntax	root->model->modelc->lightness->setup
Sample Command	root->model->modelc->lightness->setup
Dependent Parameter	NA
Note	NA

2. Light Lightness Model Set Publish Address

Follow the below CLI command to set the desired Publish address to the Light Lightness Model Client from the application.

Command Syntax	root->model->modelc->lightness->publishaddr
Sample Command	root->model->modelc->lightness->publishaddr 1 0
Dependent Parameter	<Publish Address : [2 Bytes of Address]> <App Key ID : [2 Byte of App Key Index to used]>
Note	In the PTS testing of Model Client, the Publish Address is usually the Unicast Address of the PTS device(which is the Provisioner and the Configuration Client) and its address is typically 0x0001(Unicast Address of Provisioner/PTS).

3. Light Lightness Get

Follow the below CLI command to send the Light Lightness GET message to the Mesh Network.

Command Syntax	root->model->modelc->lightness->lightnessget
Sample Command	root->model->modelc->lightness->lightnessget

Dependent Parameter	NA
Note	The "Publish Address" needs to be set before issuing this command while testing with PTS.

4. Light Lightness Set

Follow the below CLI command to send the Light Lightness SET-ACKNOWLEDGED message to the Mesh Network.

Command Syntax	root->model->modelc->lightness->lightnessset
Sample Command	root->model->modelc->lightness->lightnessset 03E8 1 54 5
Dependent Parameter	<Lightness Level : [2 Bytes of Lightness Level Value]> [in HEX] <TID : [1 Byte of TID]> [in HEX] <Transition Time : [1 Byte of Transition Time]> [in HEX] <Delay : [1 Byte of Delay]> [in HEX]
Note	The "Publish Address" needs to be set before issuing this command while testing with PTS.

5. Light Lightness Set Unacknowledged

Follow the below CLI command to send the Light Lightness SET-UNACKNOWLEDGED message to the Mesh Network.

Command Syntax	root->model->modelc->lightness->lightnesssetun
Sample Command	root->model->modelc->lightness->lightnesssetun 03E8 1 54 5
Dependent Parameter	<Lightness Level : [2 Bytes of Lightness Level Value]> [in HEX] <TID : [1 Byte of TID]> [in HEX] <Transition Time : [1 Byte of Transition Time]> [in HEX] <Delay : [1 Byte of Delay]> [in HEX]
Note	The "Publish Address" needs to be set before issuing this command while testing with PTS.

6. Light Lightness Liner Get

Follow the below CLI command to send the Light Lightness Linear GET message to the Mesh Network.

Command Syntax	root->model->modelc->lightness->linearget
Sample Command	root->model->modelc->lightness->linearget
Dependent Parameter	NA
Note	The "Publish Address" needs to be set before issuing this command while testing with PTS.

7. Light Lightness Linear Set

Follow the below CLI command to send the Light Lightness Linear SET-ACKNOWLEDGED message to the Mesh Network.

Command Syntax	root->model->modelc->lightness->linearset
Sample Command	root->model->modelc->lightness->linearset 03E8 1 54 5
Dependent Parameter	<Lightness Level : [2 Bytes of Lightness Level Value]> [in HEX] <TID : [1 Byte of TID]> [in HEX] <Transition Time : [1 Byte of Transition Time]> [in HEX] <Delay : [1 Byte of Delay]> [in HEX]
Note	The "Publish Address" needs to be set before issuing this command while testing with PTS. Currently, the dependent parameters are not segregated as "Mandatory" and "Optional". All the parameters are categorized as "Mandatory".

8. Light Lightness Linear Set Unacknowledged

Follow the below CLI command to send the Light Lightness Linear SET-UNACKNOWLEDGED message to the Mesh Network.

Command Syntax	root->model->modelc->lightness->linearsetun
Sample Command	root->model->modelc->lightness->linearset 7530 1 54 5
Dependent Parameter	<Lightness Level : [2 Bytes of Lightness Level Value]> [in HEX] <TID : [1 Byte of TID]> [in HEX] <Transition Time : [1 Byte of Transition Time]> [in HEX] <Delay : [1 Byte of Delay]> [in HEX]

Note	The "Publish Address" needs to be set before issuing this command while testing with PTS. Currently, the dependent parameters are not segregated as "Mandatory" and "Optional". All the parameters are categorized as "Mandatory".

9. Light Lightness Default Get

Follow the below CLI command to send the Light Lightness Default GET message to the Mesh Network.

Command Syntax	root->model->modelc->xyl->defaultget
Sample Command	root->model->modelc->xyl->defaultget
Dependent Parameter	NA
Note	The "Publish Address" needs to be set before issuing this command while testing with PTS.

10. Light Lightness Default Set

Follow the below CLI command to send the Light Lightness Default SET-ACKNOWLEDGED message to the Mesh Network.

Command Syntax	root->model->modelc->lightness->defaultset
Sample Command	root->model->modelc->lightness->defaultset 1388
Dependent Parameter	<Lightness Level : [2 Bytes of Lightness Level Value]> [in HEX]
Note	The "Publish Address" needs to be set before issuing this command while testing with PTS.

11. Light Lightness Default Set Unacknowledged

Follow the below CLI command to send the Light Lightness Default SET-UNACKNOWLEDGED message to the Mesh Network.

Command Syntax	root->model->modelc->lightness->defaultsetun
Sample Command	root->model->modelc->lightness->defaultsetun 2710
Dependent	<Lightness Level : [2 Bytes of Lightness Level Value]> [in HEX]

Parameter	
Note	The "Publish Address" needs to be set before issuing this command while testing with PTS.

12. Light Lightness Range Get

Follow the below CLI command to send the Light Lightness Range GET message to the Mesh Network.

Command Syntax	root->model->modelc->lightness->rangeget
Sample Command	root->model->modelc->lightness->rangeget
Dependent Parameter	NA
Note	The "Publish Address" needs to be set before issuing this command while testing with PTS.

13. Light Lightness Range Set

Follow the below CLI command to send the Light Lightness Range SET-ACKNOWLEDGED message to the Mesh Network.

Command Syntax	root->model->modelc->lightness ->rangeset
Sample Command	root->model->modelc->lightness ->rangeset 0064 F000
Dependent Parameter	<Range Min : [2 Bytes of Minimum Range]> [in HEX] <Range Max : [2 Bytes of Maximum Range]> [in HEX]
Note	The "Publish Address" needs to be set before issuing this command while testing with PTS.

14. Light Lightness Range Set Unacknowledged

Follow the below CLI command to send the Light Lightness Range SET-UNACKNOWLEDGED message to the Mesh Network.

Command Syntax	root->model->modelc->lightness->rangesetun
Sample Command	root->model->modelc->lightness->rangesetun 0064 F000
Dependent	<Range Min : [2 Bytes of Minimum Range]> [in HEX]

Parameter	<Range Max : [2 Bytes of Maximum Range]> [in HEX]
Note	The "Publish Address" needs to be set before issuing this command while testing with PTS.

15. Light Lightness Last Get

Follow the below CLI command to send the Light Lightness Last GET message to the Mesh Network.

Command Syntax	root->model->modelc->lightness->lastget
Sample Command	root->model->modelc->lightness->lastget
Dependent Parameter	NA
Note	The "Publish Address" needs to be set before issuing this command while testing with PTS.

6.1.11. Light CTL Model

1. Light CTL Model Registration

After registering with the Foundation Model Servers [as mentioned here], follow the below CLI command to register for Light CTL Model Client.

Command Syntax	root->model->modelc->ctl->setup
Sample Command	root->model->modelc->ctl->setup
Dependent Parameter	NA
Note	NA

2. Light CTL Model Set Publish Address

Follow the below CLI command to set the desired Publish address to the Light CTL Model Client from the application.

Command Syntax	root->model->modelc->ctl->publishaddr
Sample Command	root->model->modelc->ctl->publishaddr 1 0

Dependent Parameter	<Publish Address : [2 Bytes of Address]> <App Key ID : [2 Byte of App Key Index to used]>
Note	In the PTS testing of Model Client, the Publish Address is usually the Unicast Address of the PTS device(which is the Provisioner and the Configuration Client) and its address is typically 0x0001(Unicast Address of Provisioner/PTS).

3. Light CTL Get

Follow the below CLI command to send the Light CTL GET message to the Mesh Network.

Command Syntax	root->model->modelc->ctl-> ctlget
Sample Command	root->model->modelc->ctl-> ctlget
Dependent Parameter	NA
Note	The "Publish Address" needs to be set before issuing this command while testing with PTS.

4. Light CTL Set

Follow the below CLI command to send the Light CTL SET-ACKNOWLEDGED message to the Mesh Network.

Command Syntax	root->model->modelc->ctl-> ctlset
Sample Command	root->model->modelc->ctl-> ctlset 2710 4000 7530 1 root->model->modelc->ctl-> ctlset 2710 4000 7530 1 c0 0
Dependent Parameter	Mandatory Parameters - <CTL Lightness Level : [2 Bytes of CTL Lightness Level Value]> [in HEX] <CTL Temp Level : [2 Bytes of CTL Temperature Level Value]> [in HEX] <CTL delta uv : [2 Bytes of Delta UV Level Value]> [in HEX] <TID : [1 Byte of TID]> [in HEX] Optional Parameters – <Transition Time : [1 Byte of Transition Time]> [in HEX] <Delay : [1 Byte of Delay]> [in HEX]

Note	The "Publish Address" needs to be set before issuing this command while testing with PTS.

5. Light CTL Set unacknowledged

Follow the below CLI command to send the Light CTL SET-UNACKNOWLEDGED message to the Mesh Network.

Command Syntax	root->model->modelc->ctl-> ctlsetun
Sample Command	root->model->modelc->ctl-> ctlsetun 2710 4000 7530 1 root->model->modelc->ctl-> ctlset 2710 4000 7530 1 c0 0
Dependent Parameter	Mandatory Parameters - <CTL Lightness Level : [2 Bytes of CTL Lightness Level Value]> [in HEX] <CTL Temp Level : [2 Bytes of CTL Temperature Level Value]> [in HEX] <CTL delta uv : [2 Bytes of Delta UV Level Value]> [in HEX] <TID : [1 Byte of TID]> [in HEX] Optional Parameters – <Transition Time : [1 Byte of Transition Time]> [in HEX] <Delay : [1 Byte of Delay]> [in HEX]
Note	The "Publish Address" needs to be set before issuing this command while testing with PTS.

6. Light CTL Default Get

Follow the below CLI command to send the Light CTL default GET message to the Mesh Network.

Command Syntax	root->model->modelc->ctl-> defaultget
Sample Command	root->model->modelc->ctl-> defaultget
Dependent Parameter	NA
Note	The "Publish Address" needs to be set before issuing this command while testing with PTS.

7. Light CTL Default Set

Follow the below CLI command to send the Light CTL default SET-ACKNOWLEDGED message to the Mesh Network.

Command Syntax	root->model->modelc->ctl-> defaultset
Sample Command	root->model->modelc->ctl-> defaultset 2710 3000 2710
Dependent Parameter	Mandatory Parameters - <CTL Lightness Level : [2 Bytes of CTL Lightness Level Value]> [in HEX] <CTL Temp Level : [2 Bytes of CTL Temperature Level Value]> [in HEX] <CTL delta uv : [2 Bytes of Delta UV Level Value]> [in HEX]
Note	The "Publish Address" needs to be set before issuing this command while testing with PTS.

8. Light CTL Default Set unacknowledged

Follow the below CLI command to send the Light CTL default SET-UNACKNOWLEDGED message to the Mesh Network.

Command Syntax	root->model->modelc->ctl-> defaultsetun
Sample Command	root->model->modelc->ctl-> defaultsetun 2710 3000 2710
Dependent Parameter	Mandatory Parameters - <CTL Lightness Level : [2 Bytes of CTL Lightness Level Value]> [in HEX] <CTL Temp Level : [2 Bytes of CTL Temperature Level Value]> [in HEX] <CTL delta uv : [2 Bytes of Delta UV Level Value]> [in HEX]
Note	The "Publish Address" needs to be set before issuing this command while testing with PTS.

9. Light CTL Temperature Get

Follow the below CLI command to send the Light CTL temperature GET message to the Mesh Network.

Command Syntax	root->model->modelc->ctl-> tempget
Sample Command	root->model->modelc->ctl-> tempget
Dependent Parameter	NA
Note	The "Publish Address" needs to be set before issuing this command while testing with PTS.

10. Light CTL Temperature Set

Follow the below CLI command to send the Light CTL temperature SET-ACKNOWLEDGED message to the Mesh Network.

Command Syntax	root->model->modelc->ctl-> tempset
Sample Command	root->model->modelc->ctl-> tempset 3000 7530 1 root->model->modelc->ctl-> tempset 3000 7530 1 54 5
Dependent Parameter	Mandatory Parameters – <CTL Temp Level : [2 Bytes of CTL Temperature Level Value]> [in HEX] <CTL delta uv : [2 Bytes of Delta UV Level Value]> [in HEX] <TID : [1 Byte of TID]> [in HEX] Optional Parameters – <Transition Time : [1 Byte of Transition Time]> [in HEX] <Delay : [1 Byte of Delay]> [in HEX]
Note	The "Publish Address" needs to be set before issuing this command while testing with PTS.

11. Light CTL Temperature Set unacknowledged

Follow the below CLI command to send the Light CTL temperature SET-UNACKNOWLEDGED message to the Mesh Network.

Command Syntax	root->model->modelc->ctl-> tempsetun

Sample Command	root->model->modelc->ctl-> tempsetun 3000 7530 1 root->model->modelc->ctl-> tempsetun 3000 7530 1 54 5
Dependent Parameter	Mandatory Parameters – <CTL Temp Level : [2 Bytes of CTL Temperature Level Value]> [in HEX] <CTL delta uv : [2 Bytes of Delta UV Level Value]> [in HEX] <TID : [1 Byte of TID]> [in HEX] Optional Parameters – <Transition Time : [1 Byte of Transition Time]> [in HEX] <Delay : [1 Byte of Delay]> [in HEX]
Note	The "Publish Address" needs to be set before issuing this command while testing with PTS.

12. Light CTL Temperature Range Get

Follow the below CLI command to send the Light CTL temperature range GET message to the Mesh Network.

Command Syntax	root->model->modelc->ctl-> temprangeget
Sample Command	root->model->modelc->ctl-> temprangeget
Dependent Parameter	NA
Note	The "Publish Address" needs to be set before issuing this command while testing with PTS.

13. Light CTL Temperature range Set

Follow the below CLI command to send the Light CTL temperature range SET-ACKNOWLEDGED message to the Mesh Network.

Command Syntax	root->model->modelc->ctl-> temprangeset
Sample Command	root->model->modelc->ctl-> temprangeset 0600 4000
Dependent Parameter	Mandatory Parameters –

	<Range Min : [2 Bytes of Minimum Temperature Range]> [in HEX] <Range Max : [2 Bytes of Maximum Temperature Range]> [in HEX]
Note	The "Publish Address" needs to be set before issuing this command while testing with PTS.

14. Light CTL Temperature Range set unacknowledged

Follow the below CLI command to send the Light CTL temperature range SET-UNACKNOWLEDGED message to the Mesh Network.

Command Syntax	root->model->modelc->ctl-> temprangesetun
Sample Command	root->model->modelc->ctl-> temprangesetun 0600 4000
Dependent Parameter	Mandatory Parameters – <Range Min : [2 Bytes of Minimum Temperature Range]> [in HEX] <Range Max : [2 Bytes of Maximum Temperature Range]> [in HEX]
Note	The "Publish Address" needs to be set before issuing this command while testing with PTS.

6.1.12. Light HSL Model

1. Light HSL Model Registration

After registering with the Foundation Model Servers [as mentioned here], follow the below CLI command to register for Light HSL Model Client.

Command Syntax	root->model->modelc->hsl->setup
Sample Command	root->model->modelc->hsl->setup
Dependent Parameter	NA
Note	NA

2. Light HSL Model Set Publish Address

Follow the below CLI command to set the desired Publish address to the Light HSL Model Client from the application.

Command	root->model->modelc->hsl->publishaddr

Syntax	
Sample Command	root->model->modelc->hsl->publishaddr 1 0
Dependent Parameter	<Publish Address : [2 Bytes of Address]> <App Key ID : [2 Byte of App Key Index to used]>
Note	In the PTS testing of Model Client, the Publish Address is usually the Unicast Address of the PTS device(which is the Provisioner and the Configuration Client) and its address is typically 0x0001(Unicast Address of Provisioner/PTS).

3. Light HSL Get

Follow the below CLI command to send the Light HSL GET message to the Mesh Network.

Command Syntax	root->model->modelc->hsl-> hslget
Sample Command	root->model->modelc->hsl-> hslget
Dependent Parameter	NA
Note	The "Publish Address" needs to be set before issuing this command while testing with PTS.

4. Light HSL Set

Follow the below CLI command to send the Light HSL SET-ACKNOWLEDGED message to the Mesh Network.

Command Syntax	root->model->modelc->hsl-> hslset
Sample Command	root->model->modelc->hsl-> hslset 0D80 1ED2 3039 1 root->model->modelc->hsl-> hslset 04D2 162E 2334 1 C0 0
Dependent Parameter	Mandatory Parameters - <HSL Lightness Level : [2 Bytes of HSL Lightness Level Value]> [in HEX] <HSL Hue Level : [2 Bytes of HSL Hue Level Value]> [in HEX] <HSL Saturation : [2 Bytes of HSL Saturation Level Value]> [in HEX] <TID : [1 Byte of TID]> [in HEX] Optional Parameters –

	<Transition Time : [1 Byte of Transition Time]> [in HEX] <Delay : [1 Byte of Delay]> [in HEX]
Note	The "Publish Address" needs to be set before issuing this command while testing with PTS.

5. Light HSL Set unacknowledged

Follow the below CLI command to send the Light HSL SET-UNACKNOWLEDGED message to the Mesh Network.

Command Syntax	root->model->modelc->hsl-> hslsetun
Sample Command	root->model->modelc->hsl-> hslsetun 0D80 1ED2 3039 1 root->model->modelc->hsl-> hslsetun 04D2 162E 2334 1 C0 0
Dependent Parameter	Mandatory Parameters - <HSL Lightness Level : [2 Bytes of HSL Lightness Level Value]> [in HEX] <HSL Hue Level : [2 Bytes of HSL Hue Level Value]> [in HEX] <HSL Saturation : [2 Bytes of HSL Saturation Level Value]> [in HEX] <TID : [1 Byte of TID]> [in HEX] Optional Parameters – <Transition Time : [1 Byte of Transition Time]> [in HEX] <Delay : [1 Byte of Delay]> [in HEX]
Note	The "Publish Address" needs to be set before issuing this command while testing with PTS.

6. Light HSL Hue Get

Follow the below CLI command to send the Light HSL Hue GET message to the Mesh Network.

Command Syntax	root->model->modelc->hsl-> hueget
Sample Command	root->model->modelc->hsl-> hueget
Dependent Parameter	NA
Note	The "Publish Address" needs to be set before issuing this command while testing with PTS.

7. Light HSL Hue Set

Follow the below CLI command to send the Light HSL Hue SET-ACKNOWLEDGED message to the Mesh Network.

Command Syntax	root->model->modelc->hsl-> hueset
Sample Command	root->model->modelc->hsl-> hueset 3E8 1 root->model->modelc->hsl-> hueset 2710 1 c0 0
Dependent Parameter	Mandatory Parameters - <HSL Hue Level : [2 Bytes of HSL Hue Level Value]> [in HEX] <TID : [1 Byte of TID]> [in HEX] Optional Parameters – <Transition Time : [1 Byte of Transition Time]> [in HEX] <Delay : [1 Byte of Delay]> [in HEX]
Note	The "Publish Address" needs to be set before issuing this command while testing with PTS.

8. Light HSL Hue Set unacknowledged

Follow the below CLI command to send the Light HSL Hue SET-UNACKNOWLEDGED message to the Mesh Network.

Command Syntax	root->model->modelc->hsl-> huesetun
Sample Command	root->model->modelc->hsl-> huesetun BB8 1 root->model->modelc->hsl-> huesetun 4E20 1 C0 0
Dependent Parameter	Mandatory Parameters - <HSL Hue Level : [2 Bytes of HSL Hue Level Value]> [in HEX] <TID : [1 Byte of TID]> [in HEX] Optional Parameters – <Transition Time : [1 Byte of Transition Time]> [in HEX] <Delay : [1 Byte of Delay]> [in HEX]
Note	The "Publish Address" needs to be set before issuing this command while testing with PTS.

9. Light HSL range Get

Follow the below CLI command to send the Light HSL range GET message to the Mesh Network.

Command Syntax	root->model->modelc->hsl-> rangeget
Sample Command	root->model->modelc->hsl-> rangeget
Dependent Parameter	NA
Note	The "Publish Address" needs to be set before issuing this command while testing with PTS.

10. Light HSL range Set

Follow the below CLI command to send the Light HSL range SET-ACKNOWLEDGED message to the Mesh Network.

Command Syntax	root->model->modelc->hsl-> rangeset
Sample Command	root->model->modelc->hsl-> rangeset 12c f000 64 c000
Dependent Parameter	Mandatory Parameters - <Hue Range Min : [2 Bytes of Hue Min]> [in HEX] <Hue Range Max : [2 Bytes of Hue Max]> [in HEX] <Saturation Range Min : [2 Bytes of Hue Min]> [in HEX] <Saturation Range Max : [2 Bytes of Hue Max]> [in HEX]
Note	The "Publish Address" needs to be set before issuing this command while testing with PTS.

11. Light HSL range Set Unacknowledged

Follow the below CLI command to send the Light HSL range SET-UNACKNOWLEDGED message to the Mesh Network.

Command Syntax	root->model->modelc->hsl-> rangesetun
Sample Command	root->model->modelc->hsl-> rangesetun 12c f000 64 c000
Dependent	Mandatory Parameters -

Parameter	<Hue Range Min : [2 Bytes of Hue Min]> [in HEX] <Hue Range Max : [2 Bytes of Hue Max]> [in HEX] <Saturation Range Min : [2 Bytes of Hue Min]> [in HEX] <Saturation Range Max : [2 Bytes of Hue Max]> [in HEX]
Note	The "Publish Address" needs to be set before issuing this command while testing with PTS.

12. Light HSL default Get

Follow the below CLI command to send the Light HSL default GET message to the Mesh Network.

Command Syntax	root->model->modelc->hsl-> defaultget
Sample Command	root->model->modelc->hsl-> defaultget
Dependent Parameter	NA
Note	The "Publish Address" needs to be set before issuing this command while testing with PTS.

13. Light HSL default Set

Follow the below CLI command to send the Light HSL default SET-ACKNOWLEDGED message to the Mesh Network.

Command Syntax	root->model->modelc->hsl-> defaultset
Sample Command	root->model->modelc->ctl-> defaultset 64 7B 3039
Dependent Parameter	Mandatory Parameters - <Lightness level : [2 Bytes of Hue Min]> [in HEX] <Hue Level : [2 Bytes of Hue Max]> [in HEX] <Saturation Level : [2 Bytes of Hue Min]> [in HEX]
Note	The "Publish Address" needs to be set before issuing this command while testing with PTS.

14. Light HSL default Set Unacknowledged

Follow the below CLI command to send the Light HSL default SET-UNACKNOWLEDGED message to the Mesh Network.

Command Syntax	root->model->modelc->hsl-> defaultsetun
Sample Command	root->model->modelc->hsl-> defaultsetun 64 7B 3039
Dependent Parameter	Mandatory Parameters - <Lightness level : [2 Bytes of Hue Min]> [in HEX] <Hue Level : [2 Bytes of Hue Max]> [in HEX] <Saturation Level : [2 Bytes of Hue Min]> [in HEX]
Note	The "Publish Address" needs to be set before issuing this command while testing with PTS.

15. Light HSL Saturation Get

Follow the below CLI command to send the Light HSL saturation GET message to the Mesh Network.

Command Syntax	root->model->modelc->hsl-> saturget
Sample Command	root->model->modelc->hsl-> hueget
Dependent Parameter	NA
Note	The "Publish Address" needs to be set before issuing this command while testing with PTS.

16. Light HSL Saturation Set

Follow the below CLI command to send the Light HSL Saturation SET-ACKNOWLEDGED message to the Mesh Network.

Command Syntax	root->model->modelc->hsl-> saturset
Sample Command	root->model->modelc->hsl-> saturset 3E8 1 root->model->modelc->hsl-> saturset 2710 1 c0 0
Dependent Parameter	Mandatory Parameters -

	<HSL Saturation Level : [2 Bytes of HSL Hue Level Value]> [in HEX] <TID : [1 Byte of TID]> [in HEX] Optional Parameters – <Transition Time : [1 Byte of Transition Time]> [in HEX] <Delay : [1 Byte of Delay]> [in HEX]
Note	The "Publish Address" needs to be set before issuing this command while testing with PTS.

17. Light HSL Saturation Set Unacknowledged

Follow the below CLI command to send the Light HSL Saturation SET-UNACKNOWLEDGED message to the Mesh Network.

Command Syntax	root->model->modelc->hsl-> satursetun
Sample Command	root->model->modelc->hsl-> satursetun 3E8 1 root->model->modelc->hsl-> satursetun 2710 1 c0 0
Dependent Parameter	Mandatory Parameters - <HSL Saturation Level : [2 Bytes of HSL Hue Level Value]> [in HEX] <TID : [1 Byte of TID]> [in HEX] Optional Parameters – <Transition Time : [1 Byte of Transition Time]> [in HEX] <Delay : [1 Byte of Delay]> [in HEX]
Note	The "Publish Address" needs to be set before issuing this command while testing with PTS.

18. Light HSL Target Get

Follow the below CLI command to send the Light HSL target GET message to the Mesh Network.

Command Syntax	root->model->modelc->hsl-> targetget
Sample Command	root->model->modelc->hsl-> targetget
Dependent Parameter	NA

Note	The "Publish Address" needs to be set before issuing this command while testing with PTS.

6.1.13. Light xyL Model

1. Light xyL Model Registration

After registering with the Foundation Model Servers [as mentioned here], follow the below CLI command to register for Light xyL Model Client.

Command Syntax	root->model->modelc->xyl->setup
Sample Command	root->model->modelc->xyl->setup
Dependent Parameter	NA
Note	NA

2. Light xyL Model Set Publish Address

Follow the below CLI command to set the desired Publish address to the Light xyL Model Client from the application.

Command Syntax	root->model->modelc->xyl->publishaddr
Sample Command	root->model->modelc->xyl->publishaddr 1 0
Dependent Parameter	<Publish Address : [2 Bytes of Address]> <App Key ID : [2 Byte of App Key Index to used]>
Note	In the PTS testing of Model Client, the Publish Address is usually the Unicast Address of the PTS device(which is the Provisioner and the Configuration Client) and its address is typically 0x0001(Unicast Address of Provisioner/PTS).

3. Light xyL Get

Follow the below CLI command to send the Light xyL GET message to the Mesh Network.

Command Syntax	root->model->modelc->xyl->xylget
Sample Command	root->model->modelc->xyl->xylget

Dependent Parameter	NA
Note	The "Publish Address" needs to be set before issuing this command while testing with PTS.

4. Light xyL Set

Follow the below CLI command to send the Light xyL SET-ACKNOWLEDGED message to the Mesh Network.

Command Syntax	root->model->modelc->xyl->xylset
Sample Command	root->model->modelc->xyl->xylset 2710 bb8 fa0 1 c0 0
Dependent Parameter	Mandatory Parameters - <xyL Lightness Level : [2 Bytes of Lightness Level Value]> [in HEX] <xyL X Level : [2 Bytes of xyL X Level Value]> [in HEX] <xyL y Level : [2 Bytes of xyL Y Level Value]> [in HEX] <TID : [1 Byte of TID]> [in HEX] Optional Parameters – <Transition Time : [1 Byte of Transition Time]> [in HEX] <Delay : [1 Byte of Delay]> [in HEX]
Note	The "Publish Address" needs to be set before issuing this command while testing with PTS.

5. Light xyL Set Unacknowledged

Follow the below CLI command to send the Light xyL SET-UNACKNOWLEDGED message to the Mesh Network.

Command Syntax	root->model->modelc->xyl->xylsetun
Sample Command	root->model->modelc-> xyl->xylsetun 2710 bb8 fa0 1 c0 0
Dependent Parameter	Mandatory Parameters - <xyL Lightness Level : [2 Bytes of Lightness Level Value]> [in HEX] <xyL X Level : [2 Bytes of xyL X Level Value]> [in HEX] <xyL y Level : [2 Bytes of xyL Y Level Value]> [in HEX] <TID : [1 Byte of TID]> [in HEX]

	Optional Parameters – <Transition Time : [1 Byte of Transition Time]> [in HEX] <Delay : [1 Byte of Delay]> [in HEX]
Note	The "Publish Address" needs to be set before issuing this command while testing with PTS.

6. Light xyL Default Get

Follow the below CLI command to send the Light xyL default GET message to the Mesh Network.

Command Syntax	root->model->modelc->xyl->defaultget
Sample Command	root->model->modelc->xyl->defaultget
Dependent Parameter	NA
Note	The "Publish Address" needs to be set before issuing this command while testing with PTS.

7. Light xyL Default Set

Follow the below CLI command to send the Light xyL Default SET-ACKNOWLEDGED message to the Mesh Network.

Command Syntax	root->model->modelc->xyl->defaultset
Sample Command	root->model->modelc->xyl->defaultset 4d2 162e 2334
Dependent Parameter	<xyL Lightness Level : [2 Bytes of Lightness Level Value]> [in HEX] <xyL X Level : [2 Bytes of xyL X Level Value]> [in HEX] <xyL y Level : [2 Bytes of xyL Y Level Value]> [in HEX]
Note	The "Publish Address" needs to be set before issuing this command while testing with PTS.

8. Light xyL Default Set Unacknowledged

Follow the below CLI command to send the Light xyL Default SET-UNACKNOWLEDGED message to the Mesh Network.

Command	root->model->modelc->xyl->defaultsetun

Syntax	
Sample Command	root->model->modelc-> xyl->defaultsetun 4d2 162e 2334
Dependent Parameter	<xyL Lightness Level : [2 Bytes of Lightness Level Value]> [in HEX] <xyL X Level : [2 Bytes of xyL X Level Value]> [in HEX] <xyL y Level : [2 Bytes of xyL Y Level Value]> [in HEX]
Note	The "Publish Address" needs to be set before issuing this command while testing with PTS.

9. Light xyL Range Get

Follow the below CLI command to send the Light xyL Range GET message to the Mesh Network.

Command Syntax	root->model->modelc->xyl->rangeget
Sample Command	root->model->modelc->xyl->rangeget
Dependent Parameter	NA
Note	The "Publish Address" needs to be set before issuing this command while testing with PTS.

10. Light xyL Range Set

Follow the below CLI command to send the Light xyL Range SET-ACKNOWLEDGED message to the Mesh Network.

Command Syntax	root->model->modelc->xyl->rangeset
Sample Command	root->model->modelc->xyl->rangeset 64 2710 64 2710
Dependent Parameter	<xyL X Range Min : [2 Bytes of xyL X Minimum Range]> [in HEX] <xyL X Range Max : [2 Bytes of xyL X Maximum Range]> [in HEX] <xyL Y Range Min : [2 Bytes of xyL Y Minimum Range]> [in HEX] <xyL Y Range Max : [2 Bytes of xyL Y Maximum Range]> [in HEX]
Note	The "Publish Address" needs to be set before issuing this command while testing with PTS.

11. Light xyL Range Set Unacknowledged

Follow the below CLI command to send the Light xyL Range SET-UNACKNOWLEDGED message to the Mesh Network.

Command Syntax	root->model->modelc->xyl->rangesetun
Sample Command	root->model->modelc-> xyl->rangesetun 64 2710 64 2710
Dependent Parameter	<xyL X Range Min : [2 Bytes of xyL X Minimum Range]> [in HEX] <xyL X Range Max : [2 Bytes of xyL X Maximum Range]> [in HEX] <xyL Y Range Min : [2 Bytes of xyL Y Minimum Range]> [in HEX] <xyL Y Range Max : [2 Bytes of xyL Y Maximum Range]> [in HEX]
Note	The "Publish Address" needs to be set before issuing this command while testing with PTS.

12. Light xyL Target Get

Follow the below CLI command to send the Light xyL Target GET message to the Mesh Network.

Command Syntax	root->model->modelc->xyl->targetget
Sample Command	root->model->modelc->xyl->targetget
Dependent Parameter	NA
Note	The "Publish Address" needs to be set before issuing this command while testing with PTS.

6.1.14. Light LC Model

1. Light LC Model Registration

After registering with the Foundation Model Servers [as mentioned here], follow the below CLI command to register for Light LC Model Client.

Command Syntax	root->model->modelc->lc->setup
Sample Command	root->model->modelc->lc->setup

Dependent Parameter	NA
Note	NA

2. Light LC Model Set Publish Address

Follow the below CLI command to set the desired Publish address to the Light LC Model Client from the application.

Command Syntax	root->model->modelc->lc->publishaddr
Sample Command	root->model->modelc->lc->publishaddr 1 0
Dependent Parameter	<Publish Address : [2 Bytes of Address]> <App Key ID : [2 Byte of App Key Index to used]>
Note	In the PTS testing of Model Client, the Publish Address is usually the Unicast Address of the PTS device(which is the Provisioner and the Configuration Client) and its address is typically 0x0001(Unicast Address of Provisioner/PTS).

3. Light LC Mode Get

Follow the below CLI command to send the Light LC GET message to the Mesh Network.

Command Syntax	root->model->modelc->lc->modeget
Sample Command	root->model->modelc->lc->modeget
Dependent Parameter	NA
Note	The "Publish Address" needs to be set before issuing this command while testing with PTS.

4. Light LC Mode Set

Follow the below CLI command to send the Light LC Mode SET-ACKNOWLEDGED message to the Mesh Network.

Command Syntax	root->model->modelc->lc-> modeset
Sample	root->model->modelc->lc-> modeset 1

Command	
Dependent Parameter	Mandatory Parameters - <mode : [1 Byte of Lightness Level Value]> [in HEX]
Note	The "Publish Address" needs to be set before issuing this command while testing with PTS.

5. Light LC Mode Set Unacknowledged

Follow the below CLI command to send the Light LC mode SET-UNACKNOWLEDGED message to the Mesh Network.

Command Syntax	root->model->modelc->lc-> modesetun
Sample Command	root->model->modelc->lc-> modesetun 1
Dependent Parameter	Mandatory Parameters - <mode : [1 Byte of Lightness Level Value]> [in HEX]
Note	The "Publish Address" needs to be set before issuing this command while testing with PTS.

6. Light LC ONOFF get

Follow the below CLI command to send the Light LC ON-OFF get message to the Mesh Network.

Command Syntax	root->model->modelc->lc-> lightonoffget
Sample Command	root->model->modelc->lc-> lightonoffget
Dependent Parameter	NA
Note	The "Publish Address" needs to be set before issuing this command while testing with PTS.

7. Light LC ONOFF set

Follow the below CLI command to send the Light LC ON-OFF set message to the Mesh Network.

Command Syntax	root->model->modelc->lc-> lightonoffset
Sample Command	root->model->modelc->lc-> lightonoffset 1 1
Dependent Parameter	Mandatory Parameters - <onoff_level : [1 Byte of Lightness Level Value]> [in HEX] <TID : [1 Byte of TID]> [in HEX] Optional Parameters – <Transition Time : [1 Byte of Transition Time]> [in HEX] <Delay : [1 Byte of Delay]> [in HEX]
Note	The "Publish Address" needs to be set before issuing this command while testing with PTS.

8. Light LC ONOFF Set Unacknowledged

Follow the below CLI command to send the Light LC ON-OFF Set Unacknowledged message to the Mesh Network.

Command Syntax	root->model->modelc->lc-> lightonoffsetun
Sample Command	root->model->modelc->lc-> lightonoffsetun 1 1 root->model->modelc->lc-> lightonoffsetun 1 1 c0 0
Dependent Parameter	Mandatory Parameters - <onoff_level : [1 Byte of Lightness Level Value]> [in HEX] <TID : [1 Byte of TID]> [in HEX] Optional Parameters – <Transition Time : [1 Byte of Transition Time]> [in HEX] <Delay : [1 Byte of Delay]> [in HEX]
Note	The "Publish Address" needs to be set before issuing this command while testing with PTS.

9. Light LC Property Get

Follow the below CLI command to send the Light LC property get message to the Mesh Network.

Command Syntax	root->model->modelc->lc-> propertyget
Sample Command	root->model->modelc->lc-> propertyget
Dependent Parameter	NA
Note	The "Publish Address" needs to be set before issuing this command while testing with PTS.

10. Light LC Property Set

Follow the below CLI command to send the Light LC Property Set message to the Mesh Network.

Command Syntax	root->model->modelc->lc-> propertyset
Sample Command	root->model->modelc->lc-> propertyset 2e 20
Dependent Parameter	Mandatory Parameters - <property_id : [2 Bytes Property ID identifying a Light LC Property.]> [in HEX] <property_len : [variable length of property length]> [in HEX]
Note	The "Publish Address" needs to be set before issuing this command while testing with PTS.

11. Light LC Property Set Unacknowledged

Follow the below CLI command to send the Light LC Property Set unacknowledged message to the Mesh Network.

Command Syntax	root->model->modelc->lc-> propertysetun
Sample Command	root->model->modelc->lc-> propertysetun 2e 20
Dependent Parameter	Mandatory Parameters - <property_id : [2 Bytes Property ID identifying a Light LC Property.]> [in HEX] <property_len : [variable length of property length]> [in HEX]

Note	The "Publish Address" needs to be set before issuing this command while testing with PTS.

12. Light LC Occupancy Mode get

Follow the below CLI command to send the Light LC occupancy mode get message to the Mesh Network.

Command Syntax	root->model->modelc->lc-> omget
Sample Command	root->model->modelc->lc-> omget
Dependent Parameter	NA
Note	The "Publish Address" needs to be set before issuing this command while testing with PTS.

13. Light LC Occupancy Mode Set

Follow the below CLI command to send the LC Occupancy Mode set message to the Mesh Network.

Command Syntax	root->model->modelc->lc-> omset
Sample Command	root->model->modelc->lc-> omset 1
Dependent Parameter	Mandatory Parameters - <Mode : [1 Byte of Mode]> [in HEX]
Note	The "Publish Address" needs to be set before issuing this command while testing with PTS.

14. Light LC Occupancy Mode Set Unacknowledged

Follow the below CLI command to send the LC Occupancy Mode Set unacknowledged message to the Mesh Network.

Command Syntax	root->model->modelc->lc-> omsetun
Sample Command	root->model->modelc->lc-> omsetun 1
Dependent Parameter	Mandatory Parameters - <Mode : [1 Byte of Mode]> [in HEX]
Note	The "Publish Address" needs to be set before issuing this command while testing with PTS.

6.1.15. Sensor Model

1. Sensor Model Registration

After registering with the Foundation Model Servers [as mentioned here], follow the below CLI command to register for Sensor Model Client.

Command Syntax	root->model->modelc->sensor->setup
Sample Command	root->model->modelc->sensor->setup
Dependent Parameter	NA
Note	NA

2. Sensor Model Set Publish Address

Follow the below CLI command to set the desired Publish address to the Sensor Model Client from the application.

Command Syntax	root->model->modelc->sensor->publishaddr
Sample Command	root->model->modelc->sensor->publishaddr 1 0
Dependent Parameter	<Publish Address : [2 Bytes of Address]> <App Key ID : [2 Byte of App Key Index to used]>
Note	In the PTS testing of Model Client, the Publish Address is usually the Unicast Address of the PTS device(which is the Provisioner and the Configuration Client) and its address is typically 0x0001(Unicast Address of Provisioner/PTS).

3. Sensor Descriptor Get

Follow the below CLI command to send the Sensor Descriptor GET message to the Mesh Network.

Command Syntax	root->model->modelc->sensor->descriptorget
Sample Command	root->model->modelc->sensor->descriptorget 0001
Dependent	Optional Parameters -

Parameter	<PropertyID: [2 Bytes of PropertyID Value]>
Note	The "Publish Address" needs to be set before issuing this command while testing with PTS.

4. Sensor Cadence Get

Follow the below CLI command to send the Sensor Cadence GET message to the Mesh Network.

Command Syntax	root->model->modelc->sensor->cadenceget
Sample Command	root->model->modelc->sensor->cadenceget 0001
Dependent Parameter	Mandatory Parameters - < PropertyID : [2 Byte of PropertyID Value]> [in HEX]
Note	The "Publish Address" needs to be set before issuing this command while testing with PTS.

5. Sensor Cadence Set

Follow the below CLI command to send the Sensor Cadence SET-ACKNOWLEDGED message to the Mesh Network.

Command Syntax	root->model->modelc->sensor->cadenceset
Sample Command	root->model->modelc-> sensor->cadenceset 0001 01 02 3456 7890 01 2345 6789
Dependent Parameter	Mandatory Parameters - < PropertyID : [2 Bytes of PropertyID Value]> [in HEX] <Fast Cadence Period Divisor : [1 Byte Value{Only 7 Usable Bits}]> [in HEX] <Status Trigger Type : [1 Byte of Trigger Type Value{Only 1 Usable Bits}]> [in HEX] < Status Trigger Delta Down Value : [N Bytes of Status Trigger Delta Down Value]> [in HEX] < Status Trigger Delta Up Value : [N Bytes of Status Trigger Delta Up Value]> [in HEX]

	<Status Min Interval : [1 Byte of Status Min Interval Value]> [in HEX] <Fast Cadence Low Value : [N Bytes of Cadence Low Value]> [in HEX] <Fast Cadence High Value : [N Bytes of Cadence High Value]> [in HEX]
Note	The "Publish Address" needs to be set before issuing this command while testing with PTS.

6. Sensor Cadence Set Unacknowledged

Follow the below CLI command to send the Sensor Cadence SET-UNACKNOWLEDGED message to the Mesh Network.

Command Syntax	root->model->modelc->sensor->cadencesetun
Sample Command	root->model->modelc->sensor->cadencesetun 0001 01 02 3456 7890 01 2345 6789
Dependent Parameter	Mandatory Parameters - < PropertyID : [2 Bytes of PropertyID Value]> [in HEX] <Fast Cadence Period Divisor : [1 Byte Value{Only 7 Usable Bits}]> [in HEX] <Status Trigger Type : [1 Byte of Trigger Type Value{Only 1 Usable Bits}]> [in HEX] < Status Trigger Delta Down Value : [N Bytes of Status Trigger Delta Down Value]> [in HEX] < Status Trigger Delta Up Value : [N Bytes of Status Trigger Delta Up Value]> [in HEX] <Status Min Interval : [1 Byte of Status Min Interval Value]> [in HEX] <Fast Cadence Low Value : [N Bytes of Cadence Low Value]> [in HEX] <Fast Cadence High Value : [N Bytes of Cadence High Value]> [in HEX]

Note	The "Publish Address" needs to be set before issuing this command while testing with PTS.

7. Sensor Setting Get

Follow the below CLI command to send the Sensor Setting GET message to the Mesh Network.

Command Syntax	root->model->modelc->sensor->settingget
Sample Command	root->model->modelc->sensor->settingget 0001 0002
Dependent Parameter	Mandatory Parameters - < PropertyID : [2 Bytes of PropertyID Value]> [in HEX] < Setting PropertyID : [2 Bytes of Setting PropertyID Value]> [in HEX]
Note	The "Publish Address" needs to be set before issuing this command while testing with PTS.

8. Sensor Settings Get

Follow the below CLI command to send the Sensor Settings GET message to the Mesh Network.

Command Syntax	root->model->modelc->sensor->settingsget
Sample Command	root->model->modelc->sensor->settingsget 0001
Dependent Parameter	Mandatory Parameters - < PropertyID : [2 Bytes of PropertyID Value]> [in HEX]
Note	The "Publish Address" needs to be set before issuing this command while testing with PTS.

9. Sensor Setting Set

Follow the below CLI command to send the Sensor Setting SET-ACKNOWLEDGED message to the Mesh Network.

Command Syntax	root->model->modelc->sensor->settingset
Sample Command	root->model->modelc->sensor->settingset 0001 0002 1234
Dependent	Mandatory Parameters -

Parameter	< PropertyID : [2 Bytes of PropertyID Value]> [in HEX] < Setting PropertyID : [2 Bytes of Setting PropertyID Value]> [in HEX] < Setting Raw : [N Bytes of Setting Raw Value]> [in HEX]
Note	The "Publish Address" needs to be set before issuing this command while testing with PTS.

10. Sensor Setting Set Unacknowledged

Follow the below CLI command to send the Sensor Cadence SET-UNACKNOWLEDGED message to the Mesh Network.

Command Syntax	root->model->modelc->sensor->sensorsetun
Sample Command	root->model->modelc->sensor->sensorsetun 0001 0002 1234
Dependent Parameter	Mandatory Parameters - < PropertyID : [2 Bytes of PropertyID Value]> [in HEX] < Setting PropertyID : [2 Bytes of Setting PropertyID Value]> [in HEX] < Setting Raw : [N Bytes of Setting Raw Value]> [in HEX]
Note	The "Publish Address" needs to be set before issuing this command while testing with PTS.

11. Sensor Get

Follow the below CLI command to send the Sensor GET message to the Mesh Network.

Command Syntax	root->model->modelc->sensor->get
Sample Command	root->model->modelc->sensor->get 0001
Dependent Parameter	Optional Parameters - <PropertyID: [2 Bytes of PropertyID Value]>
Note	The "Publish Address" needs to be set before issuing this command while testing with PTS.

12. Sensor Column Get

Follow the below CLI command to send the Sensor Column GET message to the Mesh Network.

Command	root->model->modelc->sensor->columnget

Syntax	
Sample Command	root->model->modelc->sensor->columnget 0002 123456789
Dependent Parameter	Mandatory Parameters - < PropertyID : [2 Byte of PropertyID Value]> [in HEX] < Raw Value X : [N Bytes of Raw Value Identifying a column]> [in HEX]
Note	The "Publish Address" needs to be set before issuing this command while testing with PTS.

13. Sensor Series Get

Follow the below CLI command to send the Sensor Series GET message to the Mesh Network.

Command Syntax	root->model->modelc->sensor->seriesget
Sample Command	root->model->modelc->sensor->seriesget 0001 234 5678
Dependent Parameter	Mandatory Parameters - < PropertyID : [2 Bytes of PropertyID Value]> [in HEX] Optional Parameters - < Raw Value X1 : [N Bytes of Raw Value Identifying a starting column]> [in HEX] < Raw Value X2 : [N Bytes of Raw Value Identifying a ending column]> [in HEX]
Note	The "Publish Address" needs to be set before issuing this command while testing with PTS.

6.1.16. Time Model

1. Time Model Registration

After registering with the Foundation Model Servers [as mentioned here], follow the below CLI command to register for Time Model Client.

Command Syntax	root->model->modelc->time ->setup
Sample Command	root->model->modelc->time->setup

Dependent Parameter	NA
Note	NA

2. Time Model Set Publish Address

Follow the below CLI command to set the desired Publish address to the Time Model Client from the application.

Command Syntax	root->model->modelc->time ->publishaddr
Sample Command	root->model->modelc->time->publishaddr 1 0
Dependent Parameter	<Publish Address : [2 Bytes of Address]> <App Key ID : [2 Byte of App Key Index to used]>
Note	In the PTS testing of Model Client, the Publish Address is usually the Unicast Address of the PTS device(which is the Provisioner and the Configuration Client) and its address is typically 0x0001(Unicast Address of Provisioner/PTS).

3. Time Get

Follow the below CLI command to send the Time GET message to the Mesh Network.

Command Syntax	root->model->modelc->time->timeget
Sample Command	root->model->modelc->time->timeget
Dependent Parameter	NA
Note	The "Publish Address" needs to be set before issuing this command while testing with PTS.

4. Time Set

Follow the below CLI command to send the Time SET message to the Mesh Network.

Command Syntax	root->model->modelc->time->timeset
Sample Command	root->model->modelc->time->timeset 100000000 0 0 01 1234 00

Dependent Parameter	Mandatory Parameters - <TAI time in seconds : [5 Bytes of TAI time in second Value]> [in HEX] <Sub-Second Time : [1 Byte Value of sub-second time in units of 1/256th second]> [in HEX] <Uncertainty : [1 Byte of estimated uncertainty in 10 millisecond steps Value]> [in HEX] <Time Authority : [1 Byte with the value = 1 for Time Authority and value = 0 for No Authority]> [in HEX] <TAI UTC Delta : [2 Bytes of current difference between TAI and UTC in seconds]> [in HEX] <Time Zone Offset : [1 Byte of the local time zone offset in 15 minute increments]> [in HEX]
Note	The "Publish Address" needs to be set before issuing this command while testing with PTS.

5. Time Zone Get

Follow the below CLI command to send the Time Zone GET message to the Mesh Network.

Command Syntax	root->model->modelc->time->timezoneget
Sample Command	root->model->modelc->time->timezoneget
Dependent Parameter	NA
Note	The "Publish Address" needs to be set before issuing this command while testing with PTS.

6. Time Zone Set

Follow the below CLI command to send the Time Zone SET-ACKNOWLEDGED message to the Mesh Network.

Command Syntax	root->model->modelc->time->timezoneset
Sample Command	root->model->modelc->time->timezoneset 00 123456789
Dependent	Mandatory Parameters -

Parameter	<Time Zone Offset New : [1 Byte of upcoming local time zone offset Value]> [in HEX] <TAI of Zone change : [5 Bytes of TAI seconds time of the upcoming time zone offset change Value]> [in HEX]
Note	The "Publish Address" needs to be set before issuing this command while testing with PTS.

7. Time TAI UTC Delta Get

Follow the below CLI command to send the Time TAI UTC Delta GET message to the Mesh Network.

Command Syntax	root->model->modelc->time->taideltaget
Sample Command	root->model->modelc->time->taideltaget
Dependent Parameter	NA
Note	The "Publish Address" needs to be set before issuing this command while testing with PTS.

8. Time TAI UTC Delta Set

Follow the below CLI command to send the Time TAI UTC Delta SET-ACKNOWLEDGED message to the Mesh Network.

Command Syntax	root->model->modelc->time->taideltaset
Sample Command	root->model->modelc->time->taideltaset 1234 123456789
Dependent Parameter	Mandatory Parameters - <TAI UTC Delta New : [2 Bytes of upcoming difference between TAI and UTC in seconds{Only 15 Usable Bits}]> [in HEX] <TAI of Delta change : [5 Bytes of TAI seconds time of the upcoming TAI UTC Delta change]> [in HEX]
Note	The "Publish Address" needs to be set before issuing this command while testing with PTS.

9. Time Role Get

Follow the below CLI command to send the Time Role GET message to the Mesh Network.

Command Syntax	root->model->modelc->time->timeroleget
Sample Command	root->model->modelc->time->timeroleget
Dependent Parameter	NA
Note	The "Publish Address" needs to be set before issuing this command while testing with PTS.

10. Time Role Set

Follow the below CLI command to send the Time Role SET-ACKNOWLEDGED message to the Mesh Network.

Command Syntax	root->model->modelc->time->timeroleset
Sample Command	root->model->modelc->time->timeroleset 3
Dependent Parameter	Mandatory Parameters - <Time Role : [1 Byte of Time Role for the element]> [in HEX]
Note	The "Publish Address" needs to be set before issuing this command while testing with PTS.

6.1.17. Scheduler Model

1. Scheduler Model Registration

After registering with the Foundation Model Servers [as mentioned here], follow the below CLI command to register for Scheduler Model Client.

Command Syntax	root->model->modelc->scheduler->setup
Sample Command	root->model->modelc->scheduler->setup
Dependent Parameter	NA
Note	NA

2. Scheduler Model Set Publish Address

Follow the below CLI command to set the desired Publish address to the Scheduler Model Client from the application.

Command Syntax	root->model->modelc->scheduler->publishaddr
Sample Command	root->model->modelc->scheduler->publishaddr 1 0
Dependent Parameter	<Publish Address : [2 Bytes of Address]> <App Key ID : [2 Byte of App Key Index to used]>
Note	In the PTS testing of Model Client, the Publish Address is usually the Unicast Address of the PTS device(which is the Provisioner and the Configuration Client) and its address is typically 0x0001(Unicast Address of Provisioner/PTS).

3. Scheduler Get

Follow the below CLI command to send the Scheduler GET message to the Mesh Network.

Command Syntax	root->model->modelc->scheduler->get
Sample Command	root->model->modelc->scheduler->get
Dependent Parameter	NA
Note	The "Publish Address" needs to be set before issuing this command while testing with PTS.

4. Scheduler Action Get

Follow the below CLI command to send the Scheduler Action GET message to the Mesh Network.

Command Syntax	root->model->modelc->scheduler->actionget
Sample Command	root->model->modelc->scheduler->actionget 1
Dependent Parameter	Mandatory Parameters - <Index : [1 Byte of Index of the schedule register entry to get]> [in HEX]
Note	The "Publish Address" needs to be set before issuing this command while testing with PTS.

5. Scheduler Action Set

Follow the below CLI command to send the Scheduler Action SET-ACKNOWLEDGED message to the Mesh Network.

Command Syntax	root->model->modelc->scheduler->actionset
Sample Command	root->model->modelc->scheduler->actionset 1 1 1 1 1 1 1 1 1 1 1
Dependent Parameter	Mandatory Parameters - <Index : [1 Byte of Index of the schedule register entry to get]> [in HEX] <Year : [1 Byte of scheduled year for the action]> [in HEX] <Month : [2 Bytes of scheduled month for the action]> [in HEX] <Day : [1 Byte of scheduled day of the month for the action]> [in HEX] <Hour : [1 Byte of scheduled hour for the action]> [in HEX] <Minute : [1 Byte of scheduled minute for the action]> [in HEX] <Second : [1 Byte of scheduled second for the action]> [in HEX] <Day of the Week : [1 Byte of scheduled day of the week for the action]> [in HEX] <Action : [1 Byte of action to be performed at the scheduled time]> [in HEX] <Scene Number : [2 Bytes of scene number to be used for the action]> [in HEX] <Transition Time : [1 Byte of Transition time for the action]> [in HEX]
Note	The "Publish Address" needs to be set before issuing this command while testing with PTS.

6. Scheduler Action Set Unacknowledged

Follow the below CLI command to send the Scheduler Action SET-UNACKNOWLEDGED message to the Mesh Network.

Command Syntax	root->model->modelc->scheduler->actionsetun
Sample Command	root->model->modelc->scheduler->actionsetun 1 1 1 1 1 1 1 1 1 1
Dependent Parameter	Mandatory Parameters - <Index : [1 Byte of Index of the schedule register entry to get]> [in HEX] <Year : [1 Byte of scheduled year for the action]> [in HEX] <Month : [2 Bytes of scheduled month for the action]> [in HEX] <Day : [1 Byte of scheduled day of the month for the action]> [in HEX] <Hour : [1 Byte of scheduled hour for the action]> [in HEX] <Minute : [1 Byte of scheduled minute for the action]> [in HEX] <Second : [1 Byte of scheduled second for the action]> [in HEX] <Day of the Week : [1 Byte of scheduled day of the week for the action]> [in HEX] <Action : [1 Byte of action to be performed at the scheduled time]> [in HEX] <Scene Number : [2 Bytes of scene number to be used for the action]> [in HEX] <Transition Time : [1 Byte of Transition time for the action]> [in HEX]
Note	The "Publish Address" needs to be set before issuing this command while testing with PTS.

6.2.Model Client Test Procedures

This section covers the CLI commands needed to test the corresponding Model Client testcases with the PTS test tool.

6.2.1. Model Client test steps

To test each of the above-mentioned Model Clients individually, follow the below given procedures:

1. Reset and power cycle/ Re-launch the IUT.
2. Setup and Initialize the IUT [if needed by the platform]
3. Register the Foundation Model Servers.
4. Register the specific Model Client which is of interest. For example, to register the Generic ONOFF model client follow the instructions from here.
5. Setup the IUT [Board/executable] as a "New Device" which is to be provisioned to a Mesh Network.
6. Run the corresponding "MMDL" test section from PTS software. For example, in this case run the "MMDL->Client->Generic OnOff Client" test group from PTS.
7. Depending upon the individual test case procedures, one might have to use the specific Model Client related procedures as prompted by PTS. For example, in Generic ONOFF model client to issue a GET message follow the procedures as mentioned here.

7. References

Sl.No.	Reference
1.	Bluetooth SIG Mesh Profile Specification
2.	Bluetooth SIG Mesh Model Specification

Mindtree Limited
Contact Information

Email: bluetooth@mindtree.com
Phase 1, Global Village, Behind R V Engineering College,
Mylasandra, Mysore Road
Bangalore – 560059
INDIA
Phone- 91 (80) 67064000
Fax- 91 (80) 67064100
Web- www.mindtree.com