Zigbee Test Manager

User Guide

System Level Solutions

System Level Solutions, Inc. (USA)
511 N. Washington Avenue,
Marshall Texas 75670
(408) 852 - 0067

https://www.slscorp.com

Document Version:	1.1
Document Date:	November 2020

ug_szbtm_1.1

About this Guide

Introduction

This guide helps users to know about the basics of Zigbee Test Manager software tool.

Table below shows the revision history of this user guide.

Version	Date	Description
1.1	November 2020	Added Custom Script Operation and separate PICS Operation form Automation Operation in Chp. 2
1.0	August 2019	Initial Release

How to find Information

- The Adobe Acrobat Find feature allows you to search the contents of a PDF file. Use Ctrl + F to open the Find dialog box. Use Shift + Ctrl + N to open to the Go To Page dialog box.
- Bookmarks serve as an additional table of contents.
- Thumbnail icons, which provide miniature preview of each page, provide a link to the pages.
- Numerous links shown in Navy Blue color allow you to jump to related information.

How to Contact SLS

For the most up-to-date information about SLS products, go to the SLS worldwide website at https://www.slscorp.com. For additional information about SLS products, consult the source shown below.

Information Type	E-mail
Product literature services, SLS literature services, Non-technical customer services, Technical support	support@slscorp.com

Typographic Conventions

The user guide uses the typographic conventions as shown below:

Visual Cue	Meaning
Bold Type with Initial Capital Letters	All headings and sub headings titles in a document are displayed in bold type with initial capital letters; Example: **Manual Test Mode.**
Bold Type with Italic Letters	All definitions, figure and table headings are displayed in italics. Examples: **Figure 2-1. Zigbee Test Manager Architecture**
1., 2.	Numbered steps are used in a list of items, when the sequence of items is important. such as steps listed in procedure.
•	Bullets are used in a list of items when the sequence of items is not important.
☞	The hand points to special information that requires special attention
⚠ CAUTION	The caution indicates required information that needs special consideration and understanding and should be read prior to starting or continuing with the procedure or process.
⚠ WARNING	The warning indicates information that should be read prior to starting or continuing the procedure or processes.
👣	The feet direct you to more information on a particular topic.

Contents

1. Introduction

In this IoT era, Zigbee has become a widely used technology for many applications, for example, Home Automation, Smart Energy, Smart Medical devices, etc. due to small, low-power digital radios and low cost. This is turn helps the manufacturers to develop the new products using the profiles defined by Zigbee Alliance. To get the products certified by Zigbee, they have to undergo through the Zigbee certification process. This increase the transition of getting the product into the market. In order to reduce the transition, SLS have come with the solution which helps the manufacturer to validate their products for Zigbee compliance.

The solution is called as Zigbee Test Manger (ZTM) which helps in verifying the product as per the Zigbee Profile Specification standards defined by Zigbee Alliance. It has a simple, yet powerful GUI and provides automatic and manual mode of testing with comprehensive test report of the product. This tool is a complete reference implementation of each of the device types, clusters, and security functions, enabling easy customization by manufacturers to allow rapid development and certification of their products. This tool supports Zigbee profiles such as,

- Smart Energy 1.2b
- Smart Energy 1.4
- Home Automation 1.2
- Zigbee 3.0

ZTM helps the manufacturer to get their product faster time to market.

Benefits

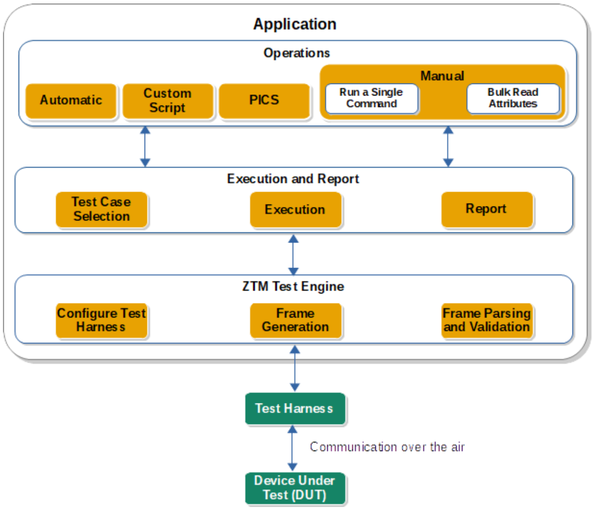
Following is the list of the benefits of using Zigbee Test Manager.

- Enables in-house pre-testing by the manufacturer
- Reduces overall cost and speeds up time to market
- Provides detail of test and data performance
- Supports manual and auto operations for testing
- Emulates Zigbee devices such as light, temperature sensor, dimmer switch, etc.
- Act as a Zigbee logical device such as end device, coordinator or router
- Select the clusters automatically and highlights mandatory test case based on device type
- Shows test statistics in real time

Architecture

Figure 1-1. shows the high level architecture of the ZTM tool.

Figure 1-1. High Level Architecture of ZTM Tool



The application is divided in four major operations, Manual, Automatic, PICS and Custom Script. Based on the selection, the test cases are executed over the selected DUT with the help of Test Harness.

Manual Operation

The manual operation is used to test an individual commands for the Zigbee profile set over the Device Under Test (DUT). It's quick and easy way to validate the commands. Following are the features:

- Support Zigbee general commands, ZDO commands, and cluster specific commands
- Read bulk attributes
- Minimal user input

Automatic Operation

Automatic operation is used when user is looking for validating response of the commands on single click as per Zigbee Alliance test specification for various profiles. This mode is widely used by Zigbee test labs to reduce the turn around time. This mode allows the user to configure the DUT type, select the cluster and associated test cases with the selected clusters. It sequentially executes all test cases and matches it's response with Zigbee Alliance test specification and reports the result. Following are the features:

- Provides extensive details of tests for product diagnostics
- Supports various Zigbee application profiles and parsing
- Auto Cluster and PICS based test cases for selected DUT type
- Project management with test case and reports
- Re-run functionality on test case failure
- Supports packet capturing using external tools, such as Ubiqua and SLS Packet Inspector
- Tests an entire profile in matter of hours
- Import/export functionality for projects

The automatic operation includes multiple sections which helps user to execute the test scenario seamlessly. Following are the sections of the automatic operation:

- Project Management
- Cluster Management

Project Management

Project management allows manufacturer to manage their test and it's result based on DUT profile. It has multiple functions such as create, edit, delete, clone, archive/ unarchive, filter, import and export which helps to manage the projects.

Cluster Management

Cluster management allows to select the cluster based on the profile chosen. It sets the clusters based on the device-type and allows to select the test cases to execute. For example, for metering device in Smart Energy profile, Metering cluster is mandatory, so it will be automatically selected in cluster selection window.

Protocol Implementation Conformation Statement (PICS) Operation

PICS operation allows to select the test cases based on the PICS. It allows to select and configure the PICS directly or by importing the excel file, which generates an endpoint wise cluster list and related test cases. These test cases are configurable and executed as per the selection.

It also supports "Negative Testing" which will logically reverse the PICS selection. This will result in following manners:

- Expecting success response for non selected PICS items

- Expecting other than success response for selected PICS items

Custom Script Operation

Custom script operation allows to create a script for configuring Test Harness and validating the DUT. It provides the list of prefix and API to write a custom script with required parameters. It also supports to export the pre-defined test cases in a script and modify as per requirement. It sequentially executes all test cases and matches it's response with Zigbee Alliance test specification and reports the result. Following are the features:

- Provides an editor to write a script
- Export the test cases in a script
- Provides support for modifying PIXIT value available in exported test case

Execution and Report

This block includes 3 stages - Test Case Selection, Execution and Report. Based on the operation test mode, the test cases are prepared. Mandatory test cases are highlighted with yellow background to differentiate. Select the test cases required to be executed over DUT. Selected test case/s are automatically execute one after another and validated as per Zigbee Alliance test specification. Different prompts are shown to get required information from the user related to test steps. Test case rerun options is provided to rerun any test case on completion or failure. On completion of test cases, a comprehensive HTML report will be generated.

ZTM Test Engine

This block generate the commands based on test case execution to validate the DUT using Test Harness. Based on the operation mode, the test harness is getting configured, and generating, parsing and validating the frames based on the commands and test cases executed.

Test Harness

This is a device/product using which the DUT is getting tested and validated. This will be configured based on the settings made in the tool.

Device Under Test (DUT)

This is a device/product which is under test for Zigbee Profile Standard specification as per Zigbee Alliance.

2. Getting Started

System Requirements

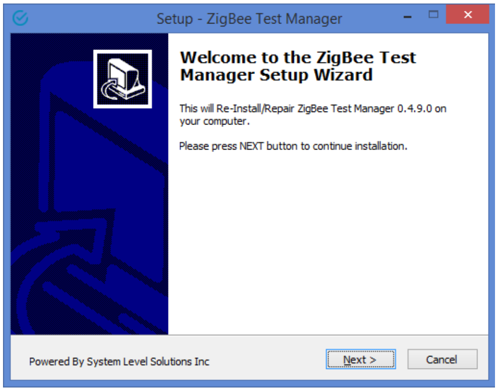
Following is the minimum system requirement to run the ZTM tool.

- **Operating System:** Windows 7 or above
- **RAM:** 4 GB or higher
- **Software:** .NET Framework Version 4.5
- **Device:** Test Harness

Software Installation

The ZTM tool is provided as an EXE file. Double click on the EXE file to install the setup. See Figure 2-1.

Figure 2-1. Software Installation - Welcome Screen



Setup - ZigBee Test Manager

Welcome to the ZigBee Test Manager Setup Wizard

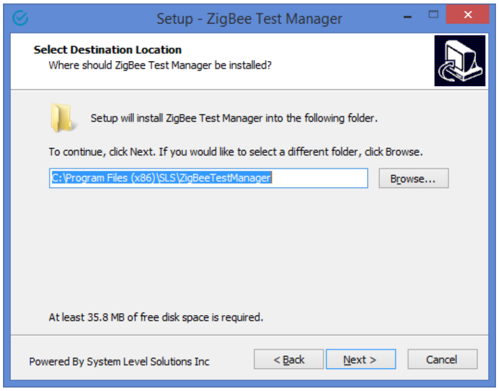
This will Re-Install/Repair ZigBee Test Manager 0.4.9.0 on your computer.

Please press NEXT button to continue installation.

Powered By System Level Solutions Inc

This page will give the information about the software and it's version details. Click Next to select the destination path. See Figure 2-2.

Figure 2-2. Software Installation - Destination Path Selection



Setup - ZigBee Test Manager

Select Destination Location

Where should ZigBee Test Manager be installed?

Setup will install ZigBee Test Manager into the following folder.

To continue, click Next. If you would like to select a different folder, click Browse.

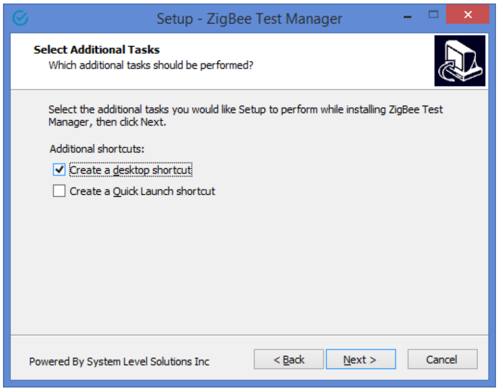
C:\Program Files (x86)\SLS\ZigBeeTestManager

At least 35.8 MB of free disk space is required.

Powered By System Level Solutions Inc

Click on Browse to select the path of installation. By default, it will select "C:\Program Files(x86)\SLS\ZigBeeTestManager". After selecting the path, click Next. See Figure 2-3.

Figure 2-3. Software Installation - Additional Settings



Setup - ZigBee Test Manager

Select Additional Tasks

Which additional tasks should be performed?

Select the additional tasks you would like Setup to perform while installing ZigBee Test Manager, then click Next.

Additional shortcuts:

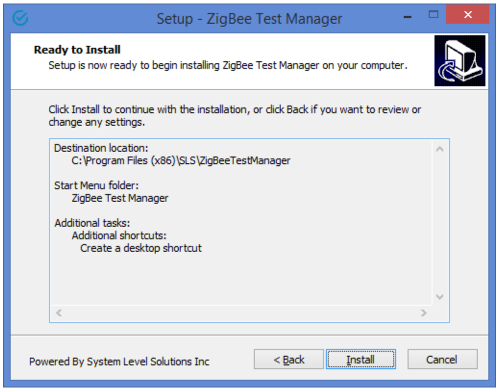
☑ Create a desktop shortcut

□ Create a Quick Launch shortcut

Powered By System Level Solutions Inc

This page ask for additional settings of the software i.e. creating the shortcuts. Check the options appropriate and click Next. See Figure 2-4.

Figure 2-4. Software Installation - Summary



Ready to Install

Setup is now ready to begin installing ZigBee Test Manager on your computer.

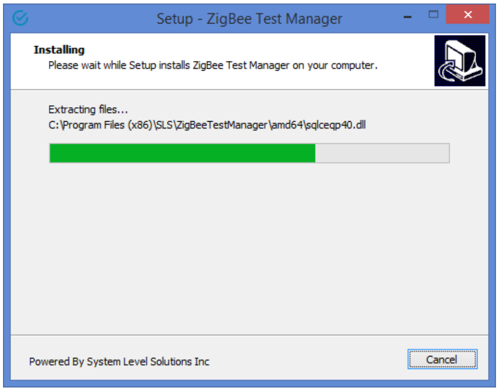
Click Install to continue with the installation, or click Back if you want to review or change any settings.

```
Destination location:
    C:\Program Files (x86)\SLS\ZigBeeTestManager

Start Menu folder:
    ZigBee Test Manager

Additional tasks:
    Additional shortcuts:
      Create a desktop shortcut
```

Powered By System Level Solutions Inc

This gives the summary of the destination path and additional task selected. Click **Install** if there is no change else click **Back**. See Figure 2-5.

Figure 2-5. Software Installation - Installation Progress



Installing

Please wait while Setup installs ZigBee Test Manager on your computer.

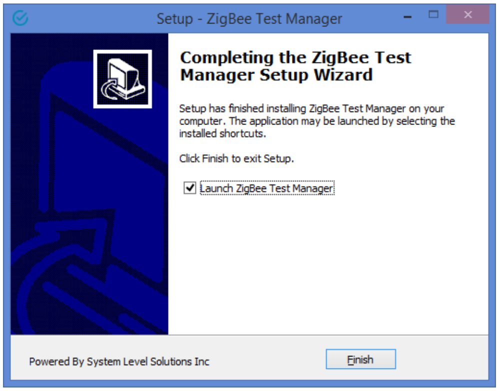
Extracting files...

C:\Program Files (x86)\SLS\ZigBeeTestManager\amd64\sqlceqp40.dll

Powered By System Level Solutions Inc

This will start installing the software in the system and displays completing installation message. See Figure 2-6.

Figure 2-6.   Software Installation - Completing Installation



Setup - ZigBee Test Manager

Completing the ZigBee Test Manager Setup Wizard

Setup has finished installing ZigBee Test Manager on your computer. The application may be launched by selecting the installed shortcuts.

Click Finish to exit Setup.

☑ Launch ZigBee Test Manager

Powered By System Level Solutions Inc

Check the "**Launch ZigBee Test Manager**" to launch the software else uncheck the box and click **Finish** to close the installation window. On closing the window, will create the shortcut path on the Desktop, if selected, along with the program menu shortcut.

Invoking ZTM

Double click on the icon on desktop or open from the path mentioned below to invoke the ZTM tool.

Start Menu > SLS > ZTM Tool

When the tool started, it display the splash screen as shown in Figure 2-7.

Figure 2-7.   Splash Screen

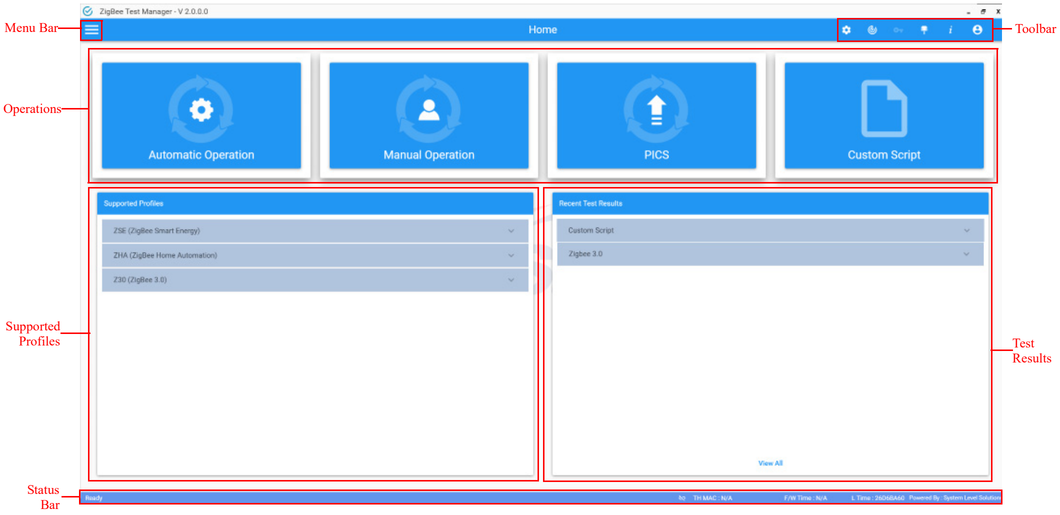
Version 0.04.09.0 - Beta

Updating profile ZSE 1.2b (0 %)

Home Page

This section gives you understanding of ZTM Tool environment that enables the usage seamless and familiarize with the options available. Figure 2-8. shows the ZTM Tool environment i.e. Home Page.

Figure 2-8. Home Page

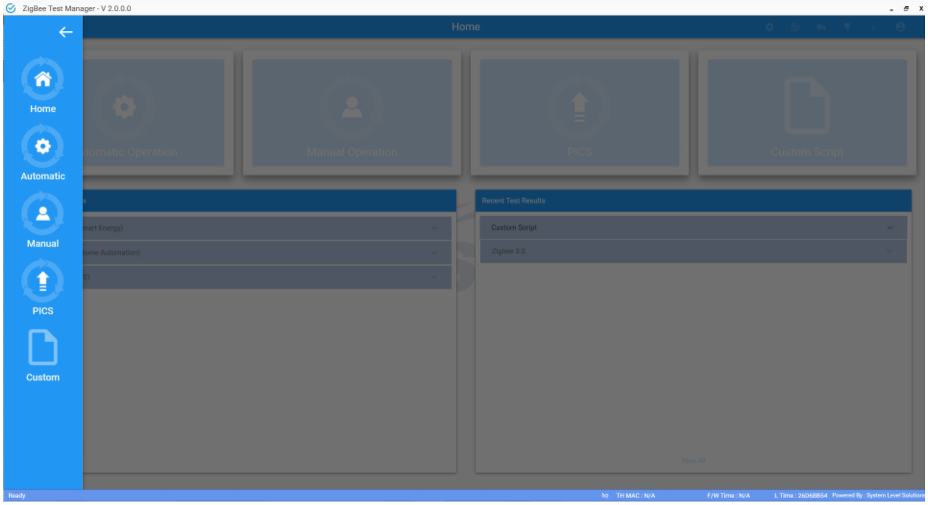


TH MAC : N/A F/W Time : N/A L Time : 260684A60 Powered By : System Level Solutions

Menu Bar Options

The Menu bar displays as 3 lines icon on the top left corner. On clicking the ☰ icon, it expand the Menu bar as shown in Figure 2-9.

Figure 2-9. Menu Bar Options



TH MAC : N/A F/W Time : N/A L Time : 26068854 Powered By : System Level Solutions

Table 2-1 describes the functionality of each icon in the Menu bar.

Table 2-1. Menu Bar Options

Icon	Name	Description
🏠	Home	Displays the home page
⚙	Automatic	Displays the project management dashboard for Automatic operation
👤	Manual	Displays the manual operation dashboard
⬆	PICS	Displays the PICS operation dashboard
📄	Custom Script	Displays the Custom Script dashboard
←	Back	Hide the menu bar

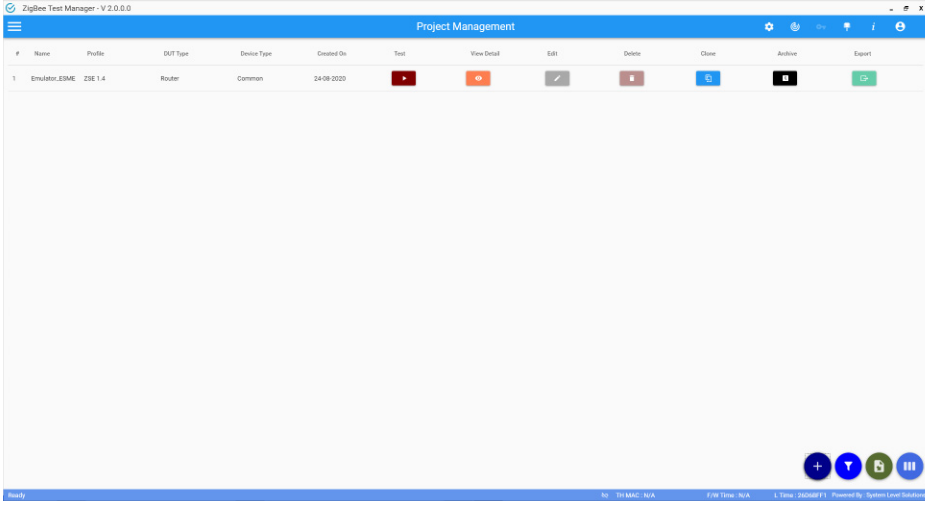
Automatic Operation

Automatic operation performs the pre-defined test cases over the DUT. It is a combination of Project Management, Cluster and PICS Test Execution, and Results View.

Project Management

The Project Management allows to manage the project for executing test cases over DUT. It allows to create, edit, delete, clone, archive/unarchive, import, export, run the test cases and view the results of the project. Figure 2-10. shows the Project Management dashboard.

Figure 2-10. Project Management Dashboard



ZigBee Test Manager - V 2.0.0.0

Project Management

#	Name	Profile	DUT Type	Device Type	Created On	Test	View Detail	Edit	Delete	Clone	Archive	Export
1	Emulator_ESME	ZSE 1.4	Router	Common	24-08-2020	▶	👁	✎	🗑	⧉	■	⇥

Ready TH MAC : N/A F/W Time : N/A L Time : 26D6BFF1 Powered By : System Level Solutions

Table 2-2 describes the options provided in Project Management dashboard.

Table 2-2. Project Management Options

Icon	Name	Description
+	Add New Project	Creates the project for testing
▼ ➤	Project Filter	List the archived project or all projects
🗎	Import Project	Imports the previously saved project
▥ ☰	View List	Displays the projects in list or thumbnail view
▶	Test Using Cluster or PICS Selection	Execute the test cases based on the Cluster or PICS selection

Table 2-2. Project Management Options

Icon	Name	Description
✎	Edit	Allows to edit the project details
🗑	Delete	Delete the project from the list
👁	View	Displays the project details
⧉	Clone	Allows to clone the project with the same settings
🗄	Archive	Archives the project
⇥	Export Project	Export the project

Add New Project

Create Project allows to enter the details of the project, DUT and Client in order to prepare the reports accordingly. The software support Zigbee Smart Energy v1.2b and v1.4 along with Zigbee Home Automation v1.2 profiles to validate the DUT. It also allows to configure the DUT type and Device type to execute the test and measure the results accordingly. Company information are used to record the details as per the company. Figure 2-11. shows the create project window.

Figure 2-11. Create Project

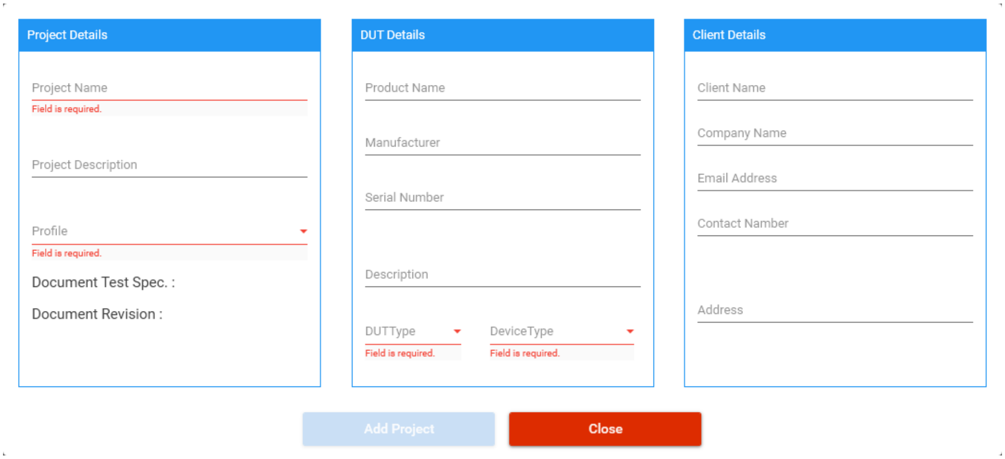
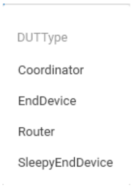
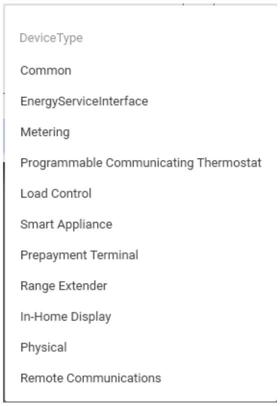


Table 2-3 describes all fields displayed in create project window.

Table 2-3. Create Project Fields

Field	Description
Project Name*	Provide the project name
Project Description	Provide the project description in brief
Profile*	Select the profile to perform the test on DUT such as ZSE v1.2b, 1.4, ZHA v1.2, Zigbee 3.0
Product Name	Provide the product name which is under test
Manufacturer	Provide the manufacturer name of the product
Serial Number	Provide the serial number of the product
Product Description	Provide the brief product description to identify the product

Table 2-3. Create Project Fields

Field	Description
DUT Type*	Select the DUT type such as Coordinator, EndDevice, Router or SleepyEndDevice.  DUTType Coordinator EndDevice Router SleepyEndDevice
Device Type*	Select the device type used to perform the test on DUT. Following is the list of supported device type.  DeviceType Common EnergyServiceInterface Metering Programmable Communicating Thermostat Load Control Smart Appliance Prepayment Terminal Range Extender In-Home Display Physical Remote Communications
Client Name	Provide the client name
Company Name	Provide the company name
Email Address	Provide the E-mail address of the client
Contact Number	Provide the contact details of the client
Address	Provide the address of the client

Note: * Indicates compulsory fields

Table 2-4 describes the buttons displayed in Add New Project window.

Table 2-4. Add New Project Options

Icon	Option	Description
Add Project	Add Project	Creates the project with the provided details
Close	Close	Close the add new project window

Project Filter

Project filter displays the project on dashboard based on the filter selection. There are two options provided to list the projects: archived and all. Table 2-5 describes the options available in project filter.

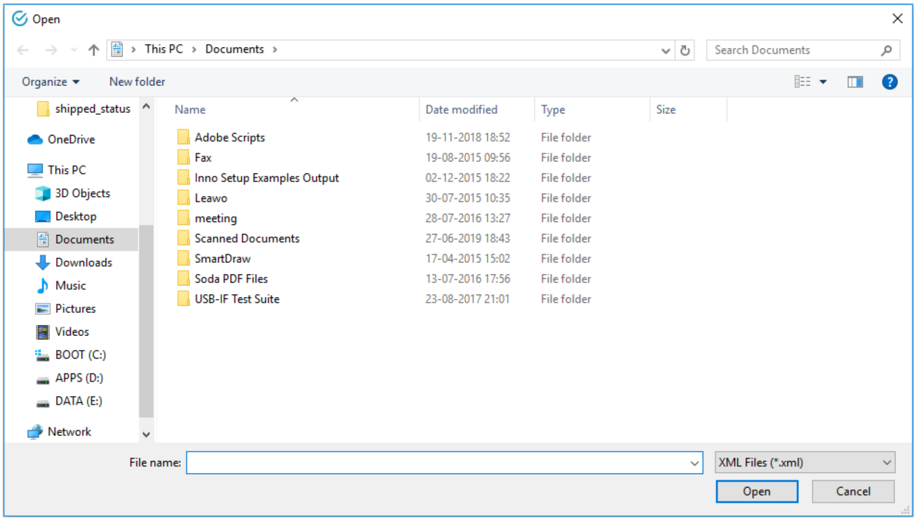
Table 2-5. Project Filter Options

Icon	Option	Description
🔴	View Archived Project	View the archived project on the dashboard
🟢	View All Project	View all the projects available on the dashboard

Import Project

Import project allows to import the project in the project management with their previous settings and details. It allows to import only ZTM exported XML file based projects. Figure 2-12. shows the imported project window.

Figure 2-12. Import Project



Name	Date modified	Type	Size
Adobe Scripts	19-11-2018 18:52	File folder	
Fax	19-08-2015 09:56	File folder	
Inno Setup Examples Output	02-12-2015 18:22	File folder	
Leawo	30-07-2015 10:35	File folder	
meeting	28-07-2016 13:27	File folder	
Scanned Documents	27-06-2019 18:43	File folder	
SmartDraw	17-04-2015 15:02	File folder	
Soda PDF Files	13-07-2016 17:56	File folder	
USB-IF Test Suite	23-08-2017 21:01	File folder	

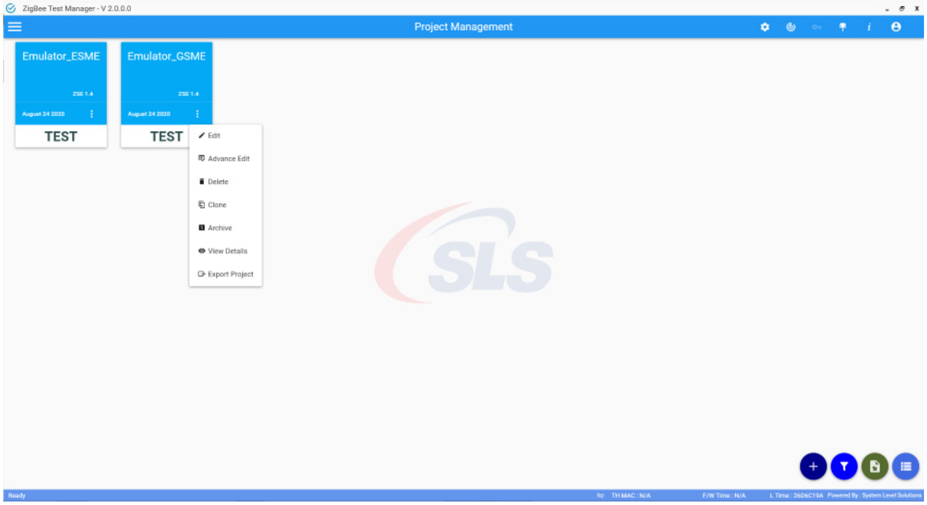
View List

View list allows to view the projects either in list or thumbnail form. On clicking the icon changes the view list on project dashboard. Table 2-6 describes the options available in view list.

Table 2-6. Dashboard View Options

Icon	Option	Description
☰	View List	Displays the projects and it’s operations in list mode. Refer Figure 2-10.
▥	View Large Thumbnail	Displays the projects and it’s operations in thumbnail mode. Refer Figure 2-13.

Figure 2-13. Thumbnail View

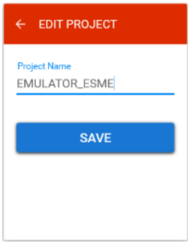


View Large Thumbnail

This options displays the projects in thumbnail view on the project management dashboard. It will display all the project operations by clicking on the Expand ⋮ icon.

To run the test using cluster or PICS selection, click on Test TEST icon. The advance edit option allows to edit the project details, while edit option allows to update the project name. See Figure 2-14.

Figure 2-14. Edit Option in Thumbnail View



Project Name
EMULATOR_ESME

Click on Save Save button to save the updated project name and click on back ⬅ icon to cancel.

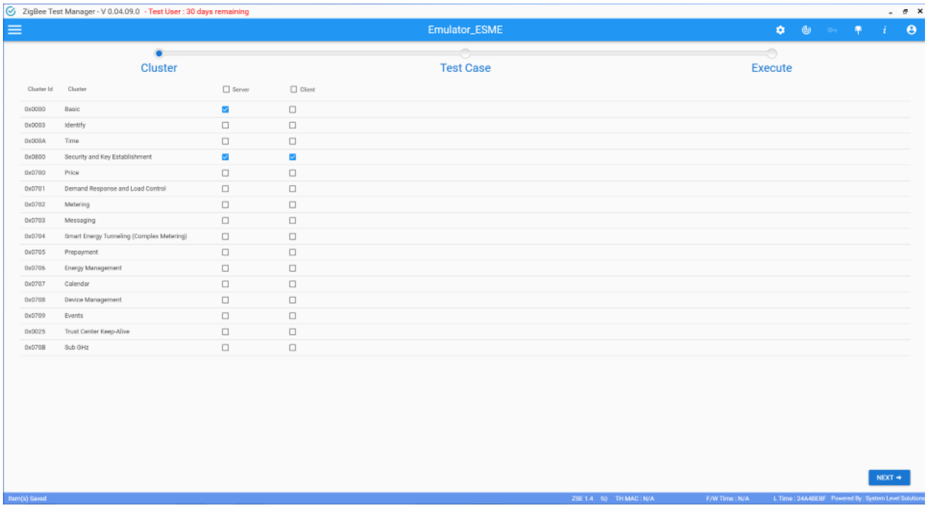
Test Using Cluster Selection

Test using cluster selection mode is selected to perform the test cases based on the cluster clauses. This test is divided in 3 stages - Cluster, Test Case and Execution. Let's understand them in details.

Cluster

The Cluster is a first stage which is displayed on clicking Test using Cluster Selection icon as shown in Figure 2-15. Based on the Device Type and DUT profile of the project, the mandatory clusters are automatically selected. Remaining clusters can be selected based on the requirement. It allows to test both server and client based clusters to verify the DUT.

Figure 2-15. Test Using Cluster Selection Window - Cluster



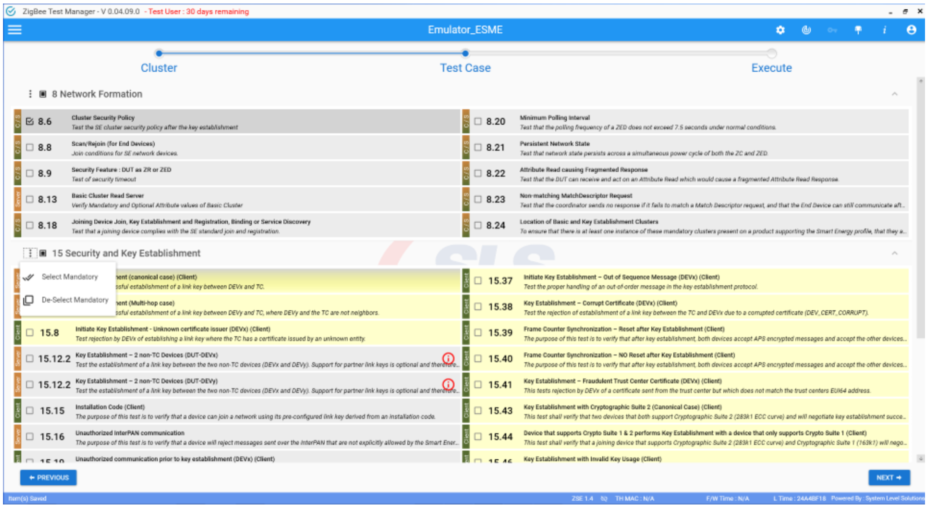
Cluster Id	Cluster	□ Server	□ Client
0x0000	Basic	☑	□
0x0003	Identify	□	□
0x000A	Time	□	□
0x0800	Security and Key Establishment	☑	☑
0x0700	Price	□	□
0x0701	Demand Response and Load Control	□	□
0x0702	Metering	□	□
0x0703	Messaging	□	□
0x0704	Smart Energy Tunneling (Complex Metering)	□	□
0x0705	Prepayment	□	□
0x0706	Energy Management	□	□
0x0707	Calendar	□	□
0x0708	Device Management	□	□
0x0709	Events	□	□
0x0025	Trust Center Keep-Alive	□	□
0x070B	Sub GHz	□	□

After selecting the clusters, click on Next NEXT ➔ button at the bottom of the page to select the test cases.

Test Case

Test case page will allows to select the test cases based on the cluster clause-wise and perform over DUT. See Figure 2-16.

Figure 2-16. Test Using Cluster Selection Window - Test Case



ZigBee Test Manager - V 0.04.09.0 - Test User : 30 days remaining

Emulator_ESME

8 Network Formation

☑ 8.6 Cluster Security Policy — *Test the SE cluster security policy after the key establishment*
□ 8.8 Scan/Rejoin (for End Devices) — *Join conditions for SE network devices.*
□ 8.9 Security Feature : DUT as ZR or ZED — *Test of security timeout*
□ 8.13 Basic Cluster Read Server — *Verify Mandatory and Optional Attribute values of Basic Cluster*
□ 8.18 Joining Device Join, Key Establishment and Registration, Binding or Service Discovery — *Test that a joining device complies with the SE standard join and registration.*
□ 8.20 Minimum Polling Interval — *Test that the polling frequency of a ZED does not exceed 7.5 seconds under normal conditions.*
□ 8.21 Persistent Network State — *Test that network state persists across a simultaneous power cycle of both the ZC and ZED.*
□ 8.22 Attribute Read causing Fragmented Response — *Test that the DUT can receive and act on an Attribute Read which would cause a fragmented Attribute Read Response.*
□ 8.23 Non-matching MatchDescriptor Request — *Test that the coordinator sends no response if it fails to match a Match Descriptor request, and that the End Device can still communicate aft...*
□ 8.24 Location of Basic and Key Establishment Clusters — *To ensure that there is at least one instance of these mandatory clusters present on a product supporting the Smart Energy profile, that they a...*

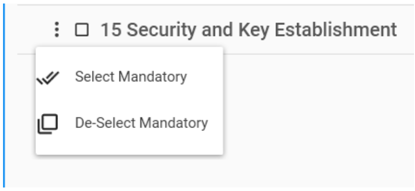
15 Security and Key Establishment

...nent (canonical case) (Client) — *...ssful establishment of a link key between DEVx and TC.*
...nent (Multi-hop case) — *...ssful establishment of a link key between DEVy and TC, where DEVy and the TC are not neighbors.*
□ 15.8 Initiate Key Establishment - Unknown certificate issuer (DEVx) (Client) — *Test rejection by DEVx of establishing a link key where the TC has a certificate issued by an unknown entity.*
□ 15.12.2 Key Establishment – 2 non-TC Devices (DUT-DEVx) — *Test the establishment of a link key between the two non-TC devices (DEVx and DEVy). Support for partner link keys is optional and therefore...*
□ 15.12.2 Key Establishment – 2 non-TC Devices (DUT-DEVy) — *Test the establishment of a link key between the two non-TC devices (DEVx and DEVy). Support for partner link keys is optional and therefore...*
□ 15.15 Installation Code (Client) — *The purpose of this test is to verify that a device can join a network using its pre-configured link key derived from an installation code.*
□ 15.16 Unauthorized InterPAN communication — *The purpose of this test is to verify that a device will reject messages sent over the InterPAN that are not explicitly allowed by the Smart Ener...*
□ 15.19 Unauthorized communication prior to key establishment (DEVx) (Client)
□ 15.37 Initiate Key Establishment – Out of Sequence Message (DEVx) (Client) — *Test the proper handling of an out-of-order message in the key establishment protocol.*
□ 15.38 Key Establishment – Corrupt Certificate (DEVx) (Client) — *Test the rejection of establishment of a link key between the TC and DEVx due to a corrupted certificate (DEV_CERT_CORRUPT).*
□ 15.39 Frame Counter Synchronization – Reset after Key Establishment (Client) — *The purpose of this test is to verify that after key establishment, both devices accept APS encrypted messages and accept the other devices...*
□ 15.40 Frame Counter Synchronization – NO Reset after Key Establishment (Client) — *The purpose of this test is to verify that after key establishment, both devices accept APS encrypted messages and accept the other devices...*
□ 15.41 Key Establishment – Fraudulent Trust Center Certificate (DEVx) (Client) — *This tests rejection by DEVx of a certificate sent from the trust center but which does not match the trust centers EUI64 address.*
□ 15.43 Key Establishment with Cryptographic Suite 2 (Canonical Case) (Client) — *This test shall verify that two devices that both support Cryptographic Suite 2 (283k1 ECC curve) and will negotiate key establishment succe...*
□ 15.44 Device that supports Crypto Suite 1 & 2 performs Key Establishment with a device that only supports Crypto Suite 1 (Client) — *This test shall verify that a joining device that supports Cryptographic Suite 2 (283k1 ECC curve) and Cryptographic Suite 1 (163k1) will nego...*
□ 15.46 Key Establishment with Invalid Key Usage (Client)

Item(s) Saved   ZSE 1.4   TH MAC : N/A   F/W Time : N/A   L Time : 24A4BF18   Powered By : System Level Solutions

Click on the expand ⋮ icon to select the mandatory test cases cluster clause-wise. It will display two options as shown in Figure 2-17.

Figure 2-17. Test Case Selection Options on Expand Click



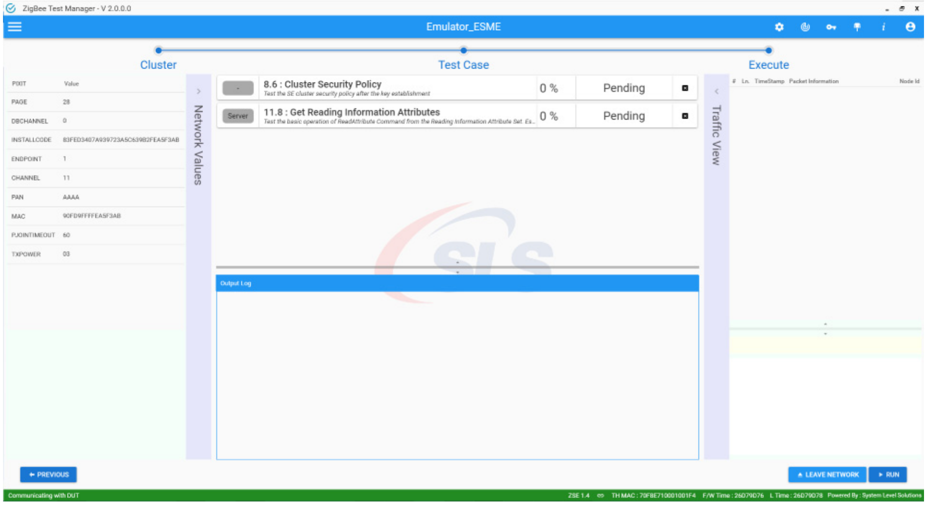
⋮ □ 15 Security and Key Establishment

Click on the cluster clause name to view the list of test cases. See Figure 2-16. The test cases which are mandatory are highlighted with yellow background else the test cases will be highlighted with grey background. To make changes in the cluster selection, click on Previous ← PREVIOUS button at the bottom of the page. After selecting the test cases, click on Next button at the bottom of the page. Next button remains disabled until single test case is not selected.

Execute

Execute window allows to run the test cases based on the test case selection over DUT and list their result in the output window. This is a final stage which list the network values, test case details, output logs and traffic view as shown in Figure 2-18.

Figure 2-18. Test Using Cluster Selection Window - Execute



PIXIT	Value
PAGE	28
DBCHANNEL	0
INSTALLCODE	83FED3407A939723A5C63982FEA5F3AB
ENDPOINT	1
CHANNEL	11
PAN	AAAA
MAC	90FD9FFFFEA5F3AB
PJOINTIMEOUT	60
TXPOWER	03

-	8.6 : Cluster Security Policy *Test the SE cluster security policy after the key establishment*	0 %	Pending	■
Server	11.8 : Get Reading Information Attributes *Test the basic operation of ReadAttribute Command from the Reading Information Attribute Set. Es...*	0 %	Pending	■

#	Ln.	TimeStamp	Packet Information	Node Id

Communicating with DUT   ZSE 1.4   TH MAC : 70F8E710001001F4   F/W Time : 26D79D76   L Time : 26D79D78   Powered By : System Level Solutions

The network values panel displays the details of the network to form/ join during the test case execution. This panel can be minimize or expand by clicking on the name. The traffic view panel displays the network packet details byte-wise to verify. Click on the traffic view panel name to minimize and expand.

Table 2-7 describes the buttons displayed in execute page.

Table 2-7. Execute Page Options

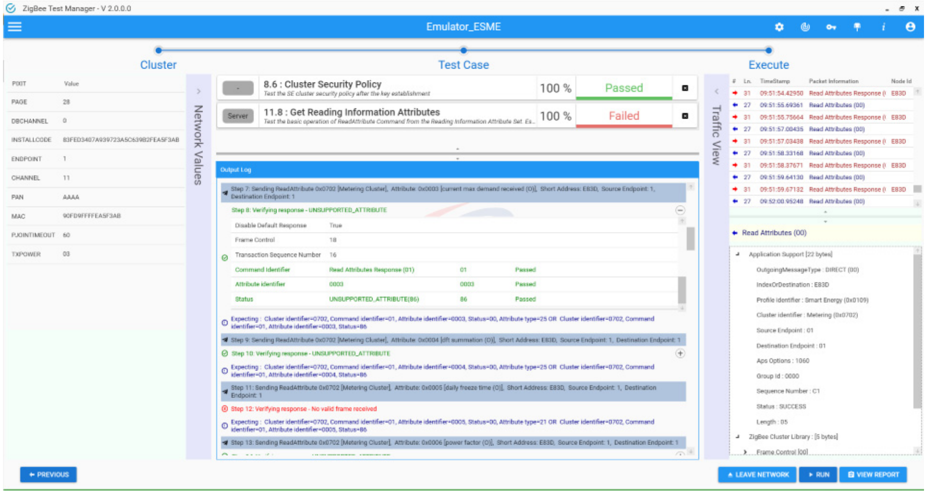
Button	Name	Description
⏏ LEAVE NETWORK	Leave Network	Allows to leave the existing network
← PREVIOUS	Previous	Takes to the test case page

Table 2-7. Execute Page Options

Button	Name	Description
▶ RUN	Run	Execute the test cases
■ STOP	Stop	Stops executing the test cases
↻ RERUN TEST	Rerun Test	Re-execute the test cases
⇄ CONTINUE UNINTERRUPTED	Continue Uninterrupted	Executes the test cases sequentially automatically
▶I NEXT TEST	Next Test	Allows to execute the test cases one by one manually

On clicking Run button, it will execute the test cases sequentially and displays its result on the output log window. The test case window displays clause, test name, status, result and status indicators. If the test case has successfully passed then it will be highlighted with green, on failure highlighted with red and on inconclusive highlighted with yellow background. Based on the status, the indicators displayed on the test case window. It has output log window which displays real-time test step execution details and results. Different prompts are shown to get required information from user related to test steps. See Figure 2-19.

Figure 2-19. Executed Test Case Window



PIXIT	Value
PAGE	28
DBCHANNEL	0
INSTALLCODE	83FED3407A939723A5C639B2FEA5F3AB
ENDPOINT	1
CHANNEL	11
PAN	AAAA
MAC	90FD9FFFFEA5F3AB
PJOINTIMEOUT	60
TXPOWER	03

-	8.6 : Cluster Security Policy Test the SE cluster security policy after the key establishment	100 %	Passed
Server	11.8 : Get Reading Information Attributes Test the basic operation of ReadAttribute Command from the Reading Information Attribute Set. Es...	100 %	Failed

```
Step 7: Sending ReadAttribute 0x0702 [Metering Cluster], Attribute: 0x0003 [current max demand received (O)], Short Address: E83D, Source Endpoint: 1, Destination Endpoint: 1
Step 8: Verifying response - UNSUPPORTED_ATTRIBUTE
    Disable Default Response     True
    Frame Control                18
    Transaction Sequence Number  16
    Command Identifier           Read Attributes Response (01)   01     Passed
    Attribute identifier         0003                            0003   Passed
    Status                       UNSUPPORTED_ATTRIBUTE(86)       86     Passed
Expecting : Cluster identifier=0702, Command identifier=01, Attribute identifier=0003, Status=00, Attribute type=25 OR Cluster identifier=0702, Command identifier=01, Attribute identifier=0003, Status=86
Step 9: Sending ReadAttribute 0x0702 [Metering Cluster], Attribute: 0x0004 [dft summation (O)], Short Address: E83D, Source Endpoint: 1, Destination Endpoint: 1
Step 10: Verifying response - UNSUPPORTED_ATTRIBUTE
Expecting : Cluster identifier=0702, Command identifier=01, Attribute identifier=0004, Status=00, Attribute type=25 OR Cluster identifier=0702, Command identifier=01, Attribute identifier=0004, Status=86
Step 11: Sending ReadAttribute 0x0702 [Metering Cluster], Attribute: 0x0005 [daily freeze time (O)], Short Address: E83D, Source Endpoint: 1, Destination Endpoint: 1
Step 12: Verifying response - No valid frame received
Expecting : Cluster identifier=0702, Command identifier=01, Attribute identifier=0005, Status=00, Attribute type=21 OR Cluster identifier=0702, Command identifier=01, Attribute identifier=0005, Status=86
Step 13: Sending ReadAttribute 0x0702 [Metering Cluster], Attribute: 0x0006 [power factor (O)], Short Address: E83D, Source Endpoint: 1, Destination Endpoint: 1
```

#	Ln.	TimeStamp	Packet Information	Node Id
→	31	09:51:54.42950	Read Attributes Response ((	E83D
←	27	09:51:55.69361	Read Attributes (00)	
→	31	09:51:55.75664	Read Attributes Response ((	E83D
←	27	09:51:57.00435	Read Attributes (00)	
→	31	09:51:57.03438	Read Attributes Response ((	E83D
←	27	09:51:58.33168	Read Attributes (00)	
→	31	09:51:58.37671	Read Attributes Response ((	E83D
←	27	09:51:59.64130	Read Attributes (00)	
→	31	09:51:59.67132	Read Attributes Response ((	E83D
←	27	09:52:00.95248	Read Attributes (00)	

```
← Read Attributes (00)
  Application Support [22 bytes]
      OutgoingMessageType : DIRECT (00)
      IndexOrDestination : E83D
      Profile identifier : Smart Energy (0x0109)
      Cluster identifier : Metering (0x0702)
      Source Endpoint : 01
      Destination Endpoint : 01
      Aps Options : 1060
      Group Id : 0000
      Sequence Number : C1
      Status : SUCCESS
      Length : 05
  ZigBee Cluster Library : [5 bytes]
    > Frame Control [00]
```

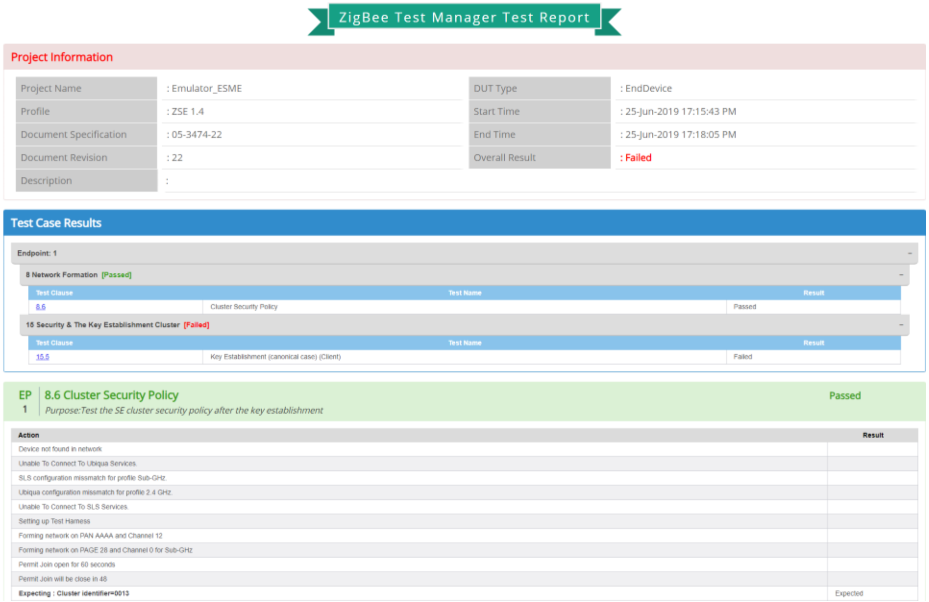
Table 2-8 describes the status indicators.

Table 2-8. ***Status Indicators***

Indicator	Name	Description
✓	Passed	Indicates the test case is passed
✗	Failed	Indicates the test case is failed
■	Inconclusive	Indicates the test case is inconclusive

On completion of the execution of the test cases, the report will be displayed in the default browser automatically. See Figure 2-20.

Figure 2-20. Report Summary



ZigBee Test Manager Test Report

Project Information

Project Name	: Emulator_ESME	DUT Type	: EndDevice
Profile	: ZSE 1.4	Start Time	: 25-Jun-2019 17:15:43 PM
Document Specification	: 05-3474-22	End Time	: 25-Jun-2019 17:18:05 PM
Document Revision	: 22	Overall Result	: Failed
Description	:		

Test Case Results

Endpoint: 1

8 Network Formation [Passed]

Test Clause	Test Name	Result
8.6	Cluster Security Policy	Passed

15 Security & The Key Establishment Cluster [Failed]

Test Clause	Test Name	Result
15.5	Key Establishment (canonical case) (Client)	Failed

EP 1 8.6 Cluster Security Policy Passed

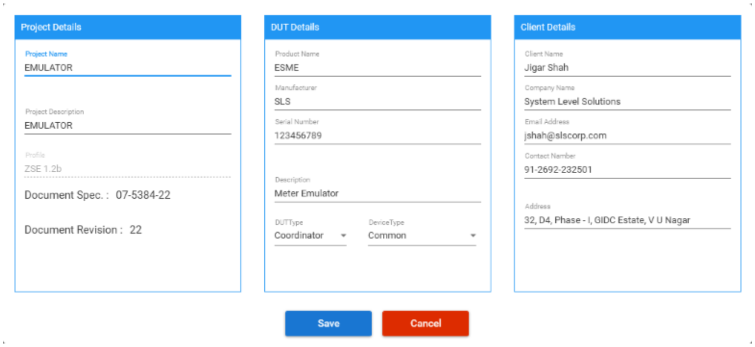
Purpose:Test the SE cluster security policy after the key establishment

Action	Result
Device not found in network	
Unable To Connect To Ubiqua Services.	
SLS configuration missmatch for profile Sub-GHz.	
Ubiqua configuration missmatch for profile 2.4 GHz.	
Unable To Connect To SLS Services.	
Setting up Test Harness	
Forming network on PAN AAAA and Channel 12	
Forming network on PAGE 28 and Channel 0 for Sub-GHz	
Permit Join open for 60 seconds	
Permit Join will be close in 48	
Expecting : Cluster identifier=0013	Expected

Edit

Edit allows to update the project details except the Zigbee profile. Figure 2-21. shows the edit window.

Figure 2-21. Edit Window



Project Details

Project Name: EMULATOR

Project Description: EMULATOR

Profile: ZSE 1.2b

Document Spec. : 07-5384-22

Document Revision : 22

DUT Details

Product Name: ESME

Manufacturer: SLS

Serial Number: 123456789

Description: Meter Emulator

DUTType: Coordinator

DeviceType: Common

Client Details

Client Name: Jigar Shah

Company Name: System Level Solutions

Email Address: jshah@slscorp.com

Contact Namber: 91-2692-232501

Address: 32, D4, Phase - I, GIDC Estate, V U Nagar

Table 2-9 describes the buttons displayed in edit project window.

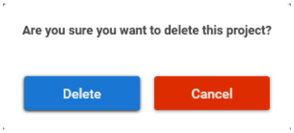
Table 2-9. Edit Project Window Options

Button	Name	Description
Save	Save	Saves the project details and close the window
Cancel	Cancel	Close the dialog box and display the project management dashboard

Delete

Delete allows to delete the project created. It displays the dialog box as shown in Figure 2-22.

Figure 2-22. Delete Dialog Box



Are you sure you want to delete this project?

Table 2-10 describes the buttons displayed in delete dialog box.

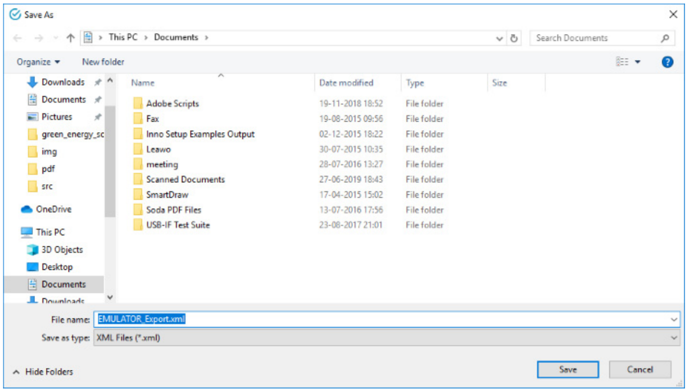
Table 2-10. Delete Dialog Box Options

Button	Name	Description
Delete	Delete	Delete the project
Cancel	Cancel	Close the dialog box and display the project management dashboard

Export

Export allows to export the whole project in XML format. It displays the Save As window as shown in Figure 2-23.

Figure 2-23. Export Save As Window



Name	Date modified	Type	Size
Adobe Scripts	19-11-2018 18:52	File folder	
Fax	19-08-2015 09:56	File folder	
Inno Setup Examples Output	02-12-2015 18:22	File folder	
Leawo	30-07-2015 10:35	File folder	
meeting	28-07-2016 13:27	File folder	
Scanned Documents	27-06-2019 18:43	File folder	
SmartDraw	17-04-2015 15:02	File folder	
Soda PDF Files	13-07-2016 17:56	File folder	
USB-IF Test Suite	23-08-2017 21:01	File folder	

File name: EMULATOR_Export.xml

Save as type: XML Files (*.xml)

Table 2-11 describes the buttons displayed in export window.

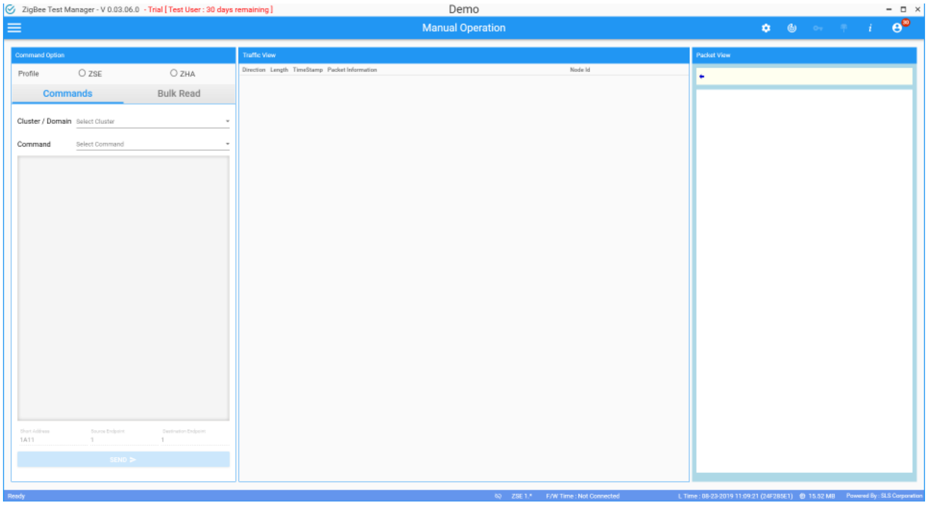
Table 2-11. Export Window Options

Button	Name	Description
Save	Save	Save the project at the selected path
Cancel	Cancel	Close the dialog box and display the project management dashboard

Manual Operation

Manual operation allows to execute individual commands over DUT and validate the frames quickly. It’s easy to use and execute the commands. Figure 2-24. displays the Manual operation window.

Figure 2-24. Manual Operation



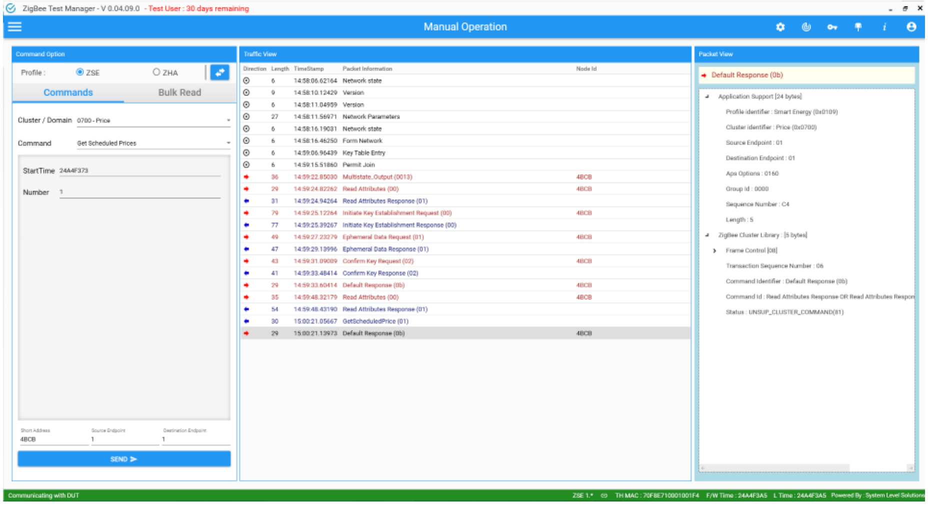
ZigBee Test Manager - V 0.03.06.0 - Trial [Test User : 30 days remaining]

Short Address: 1A11
Source Endpoint: 1
Destination Endpoint: 1

Ready   ZSE 1.*   F/W Time : Not Connected   L Time : 08-23-2019 11:09:21 (24F285E1)   15.52 MB   Powered By : SLS Corporation

There are 3 section in the window - Command Option, Traffic View and Packet View. Command option allows to select the individual command or read multiple attributes based on the profile selected. For individual command, select Command tab and for read multiple attribute, select Bulk Read tab. Based on the profile, the cluster details will be available. Choose the Cluster / Domain from the drop down list and it will display the available command for the selected cluster. Choose the command and it displays the required fields to be filled for the command to execute. Enter the Short Address, Source Endpoint and Destination Endpoint details. It will enable the Send SEND > button which will send the command with added information and showcase the response in the traffic view and packet view sections. See Figure 2-25.

Figure 2-25. Executing Command in Manual Operation



ZigBee Test Manager - V 0.04.09.0 - Test User : 30 days remaining

Manual Operation

Command Option

Profile : ● ZSE ○ ZHA

Commands | Bulk Read

Cluster / Domain: 0700 - Price

Command: Get Scheduled Prices

StartTime: 24A4F373

Number: 1

Short Address: 4BCB

Source Endpoint: 1

Destination Endpoint: 1

SEND

Traffic View

Direction	Length	TimeStamp	Packet Information	Node Id
⊙	6	14:58:06.62164	Network state	
⊙	9	14:58:10.12429	Version	
⊙	6	14:58:11.04959	Version	
⊙	27	14:58:11.56971	Network Parameters	
⊙	6	14:58:16.19031	Network state	
⊙	6	14:58:16.46250	Form Network	
⊙	6	14:59:06.96439	Key Table Entry	
⊙	6	14:59:15.51860	Permit Join	
→	36	14:59:22.85030	Multistate_Output (0013)	4BCB
→	29	14:59:24.82262	Read Attributes (00)	4BCB
←	31	14:59:24.94264	Read Attributes Response (01)	
→	79	14:59:25.12264	Initiate Key Establishment Request (00)	4BCB
←	77	14:59:25.39267	Initiate Key Establishment Response (00)	
→	49	14:59:27.23279	Ephemeral Data Request (01)	4BCB
←	47	14:59:29.13996	Ephemeral Data Response (01)	
→	43	14:59:31.09009	Confirm Key Request (02)	4BCB
←	41	14:59:33.48414	Confirm Key Response (02)	
→	29	14:59:33.60414	Default Response (0b)	4BCB
→	35	14:59:48.32179	Read Attributes (00)	4BCB
←	54	14:59:48.43190	Read Attributes Response (01)	
←	30	15:00:21.05667	GetScheduledPrice (01)	
→	29	15:00:21.13973	Default Response (0b)	4BCB

Packet View

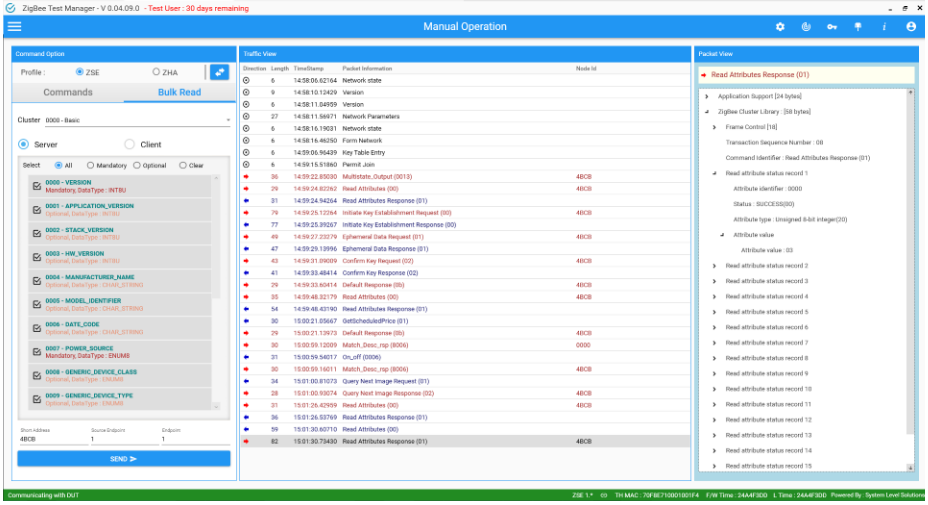
→ Default Response (0b)

- Application Support [24 bytes]
  - Profile identifier : Smart Energy (0x0109)
  - Cluster identifier : Price (0x0700)
  - Source Endpoint : 01
  - Destination Endpoint : 01
  - Aps Options : 0160
  - Group Id : 0000
  - Sequence Number : C4
  - Length : 5
- ZigBee Cluster Library : [5 bytes]
  - Frame Control [08]
  - Transaction Sequence Number : 06
  - Command Identifier : Default Response (0b)
  - Command Id : Read Attributes Response OR Read Attributes Respon
  - Status : UNSUP_CLUSTER_COMMAND(81)

Communicating with DUT   ZSE 1.*   TH MAC : 70F8E710001001F4   F/W Time : 24A4F3A5   L Time : 24A4F3A5   Powered By : System Level Solutions

In the same way, choose Bulk Read tab to read the multiple attributes at a time and check the response of the DUT. Select the Cluster and it will list out all the attributes available for either Server or Client. It allows to send a read attribute command with All, Mandatory, Optional attributes over the selected Short Address, Source Endpoint and Destination Endpoint. The results can be viewed in the traffic view and packet view. See Figure 2-26.

Figure 2-26. Bulk Read in Manual Operation



ZigBee Test Manager - V 0.04.09.0 - Test User : 30 days remaining

Manual Operation

Command Option

Profile : ● ZSE ○ ZHA

Commands | Bulk Read

Cluster 0000 - Basic

● Server ○ Client

Select ● All ○ Mandatory ○ Optional ○ Clear

- ☑ 0000 - VERSION
Mandatory, DataType : INT8U
- ☑ 0001 - APPLICATION_VERSION
Optional, DataType : INT8U
- ☑ 0002 - STACK_VERSION
Optional, DataType : INT8U
- ☑ 0003 - HW_VERSION
Optional, DataType : INT8U
- ☑ 0004 - MANUFACTURER_NAME
Optional, DataType : CHAR_STRING
- ☑ 0005 - MODEL_IDENTIFIER
Optional, DataType : CHAR_STRING
- ☑ 0006 - DATE_CODE
Optional, DataType : CHAR_STRING
- ☑ 0007 - POWER_SOURCE
Mandatory, DataType : ENUM8
- ☑ 0008 - GENERIC_DEVICE_CLASS
Optional, DataType : ENUM8
- ☑ 0009 - GENERIC_DEVICE_TYPE
Optional, DataType : ENUM8

Short Address: 4BCB
Source Endpoint: 1
Endpoint: 1

SEND

Traffic View

Direction	Length	TimeStamp	Packet Information	Node Id
⊙	6	14:58:06.62164	Network state	
⊙	9	14:58:10.12429	Version	
⊙	6	14:58:11.04959	Version	
⊙	27	14:58:11.56971	Network Parameters	
⊙	6	14:58:16.19031	Network state	
⊙	6	14:58:16.46250	Form Network	
⊙	6	14:59:06.96439	Key Table Entry	
⊙	6	14:59:15.51860	Permit Join	
→	36	14:59:22.85030	Multistate_Output (0013)	4BCB
→	29	14:59:24.82262	Read Attributes (00)	4BCB
←	31	14:59:24.94264	Read Attributes Response (01)	
→	79	14:59:25.12264	Initiate Key Establishment Request (00)	4BCB
←	77	14:59:25.39267	Initiate Key Establishment Response (00)	
→	49	14:59:27.23279	Ephemeral Data Request (01)	4BCB
←	47	14:59:29.13996	Ephemeral Data Response (01)	
→	43	14:59:31.09009	Confirm Key Request (02)	4BCB
←	41	14:59:33.48414	Confirm Key Response (02)	
→	29	14:59:33.60414	Default Response (0b)	4BCB
→	35	14:59:48.32179	Read Attributes (00)	4BCB
←	54	14:59:48.43190	Read Attributes Response (01)	
←	30	15:00:21.05667	GetScheduledPrice (01)	
→	29	15:00:21.13973	Default Response (0b)	4BCB
→	30	15:00:59.12009	Match_Desc_rsp (8006)	0000
←	31	15:00:59.54017	On_off (0006)	
→	30	15:00:59.16011	Match_Desc_rsp (8006)	4BCB
←	34	15:01:00.81073	Query Next Image Request (01)	
→	28	15:01:00.93074	Query Next Image Response (02)	4BCB
→	31	15:01:26.42959	Read Attributes (00)	4BCB
←	36	15:01:26.53769	Read Attributes Response (01)	
←	59	15:01:30.60710	Read Attributes (00)	
→	82	15:01:30.73430	Read Attributes Response (01)	4BCB

Packet View

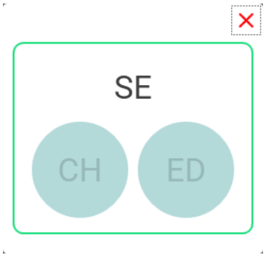
→ Read Attributes Response (01)

```
> Application Support [24 bytes]
⌟ ZigBee Cluster Library : [58 bytes]
    > Frame Control [18]
      Transaction Sequence Number : 08
      Command Identifier : Read Attributes Response (01)
    ⌟ Read attribute status record 1
        Attribute identifier : 0000
        Status : SUCCESS(00)
        Attribute type : Unsigned 8-bit integer(20)
      ⌟ Attribute value
          Attribute value : 03
    > Read attribute status record 2
    > Read attribute status record 3
    > Read attribute status record 4
    > Read attribute status record 5
    > Read attribute status record 6
    > Read attribute status record 7
    > Read attribute status record 8
    > Read attribute status record 9
    > Read attribute status record 10
    > Read attribute status record 11
    > Read attribute status record 12
    > Read attribute status record 13
    > Read attribute status record 14
    > Read attribute status record 15
```

Communicating with DUT   ZSE 1.*   TH MAC : 70F8E710001001F4   F/W Time : 24A4F3DD   L Time : 24A4F3DD   Powered By : System Level Solutions

To change the device type, click on the device type ⇄ icon. It will display the device type window as shown in Figure 2-27.

Figure 2-27. Test Harness Device Type Selection Window



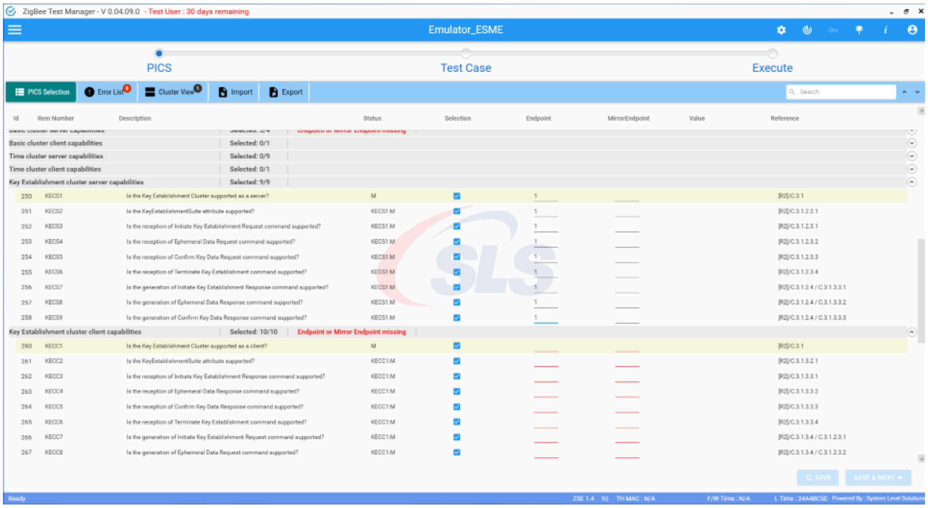
PICS Operation

PICS operation allows to execute the test cases based on the PICS specified by Zigbee Alliance with more user friendly inputs. Based on PICS selection, endpoint wise cluster list will be generated and related test cases will be loaded. Applicable test cases are automatically selected with an option to select/deselect by user as per requirement. This test is divided in 3 stages - PICS, Test Case and Execute. Let’s see them in detail.

PICS

PICS allows to make the selection of PICS based on the project profile. This page displays PICS selection, Error List, Cluster View, Import and Export options. See Figure 2-28.

Figure 2-28. Test Using PICS Selection - PICS



ZigBee Test Manager - V 0.04.09.0 - Test User : 30 days remaining

Emulator_ESME

Id	Item Number	Description	Status	Selection	Endpoint	MirrorEndpoint	Value	Reference
Basic cluster server capabilities			Selected: 3/4	Endpoint or Mirror Endpoint missing				
Basic cluster client capabilities			Selected: 0/1					
Time cluster server capabilities			Selected: 0/9					
Time cluster client capabilities			Selected: 0/1					
Key Establishment cluster server capabilities			Selected: 9/9					
250	KECS1	Is the Key Establishment Cluster supported as a server?	M	☑	1			[R2]/C.3.1
251	KECS2	Is the KeyEstablishmentSuite attribute supported?	KECS1:M	☑	1			[R2]/C.3.1.2.2.1
252	KECS3	Is the reception of Initiate Key Establishment Request command supported?	KECS1:M	☑	1			[R2]/C.3.1.2.3.1
253	KECS4	Is the reception of Ephemeral Data Request command supported?	KECS1:M	☑	1			[R2]/C.3.1.2.3.2
254	KECS5	Is the reception of Confirm Key Data Request command supported?	KECS1:M	☑	1			[R2]/C.3.1.2.3.3
255	KECS6	Is the reception of Terminate Key Establishment command supported?	KECS1:M	☑	1			[R2]/C.3.1.2.3.4
256	KECS7	Is the generation of Initiate Key Establishment Response command supported?	KECS1:M	☑	1			[R2]/C.3.1.2.4 / C.3.1.3.3.1
257	KECS8	Is the generation of Ephemeral Data Response command supported?	KECS1:M	☑	1			[R2]/C.3.1.2.4 / C.3.1.3.3.2
258	KECS9	Is the generation of Confirm Key Data Response command supported?	KECS1:M	☑	1			[R2]/C.3.1.2.4 / C.3.1.3.3.3
Key Establishment cluster client capabilities			Selected: 10/10	Endpoint or Mirror Endpoint missing				
260	KECC1	Is the Key Establishment Cluster supported as a client?	M	☑				[R2]/C.3.1
261	KECC2	Is the KeyEstablishmentSuite attribute supported?	KECC1:M	☑				[R2]/C.3.1.3.2.1
262	KECC3	Is the reception of Initiate Key Establishment Response command supported?	KECC1:M	☑				[R2]/C.3.1.3.3.1
263	KECC4	Is the reception of Ephemeral Data Response command supported?	KECC1:M	☑				[R2]/C.3.1.3.3.2
264	KECC5	Is the reception of Confirm Key Data Response command supported?	KECC1:M	☑				[R2]/C.3.1.3.3.3
265	KECC6	Is the reception of Terminate Key Establishment command supported?	KECC1:M	☑				[R2]/C.3.1.3.3.4
266	KECC7	Is the generation of Initiate Key Establishment Request command supported?	KECC1:M	☑				[R2]/C.3.1.3.4 / C.3.1.2.3.1
267	KECC8	Is the generation of Ephemeral Data Request command supported?	KECC1:M	☑				[R2]/C.3.1.3.4 / C.3.1.2.3.2

Ready ZSE 1.4 TH MAC : N/A F/W Time : N/A L Time : 24A4BC5D Powered By : System Level Solutions

Table 2-12 describes the options available in PICS window.

Table 2-12. PICS Selection Test Case Options

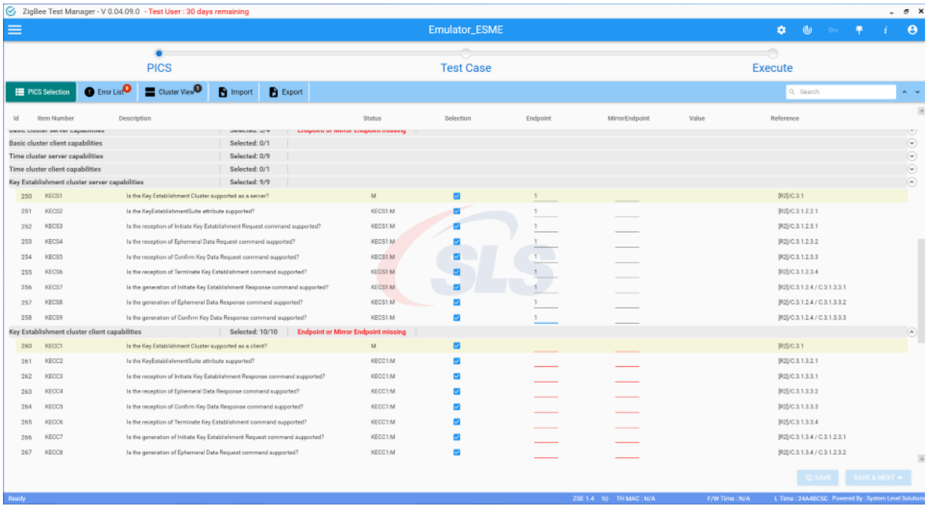
Options	Name	Description
PICS Selection	PICS Selection	Allows to select the PICS and provide the require details
Error List 0	Error Log	Displays the errors generated during in PICS selection
Cluster View 0	Cluster View	Displays the cluster based on the information provided in the PICS selection option
Import	Import	Imports the excel file of the PICS

Table 2-12. PICS Selection Test Case Options

Options	Name	Description
Export	Export	Exports the PICS in excel file
🔍 Search	Search	Searches the PICS item number and description from the list
⌃ ⌄	Next PICS Selection	Takes to next / previous PICS
NEXT ➜	Next	Allows to go to next page in the PICS test
⌄	Expand	Expands the list of the PICS

The PICS are listed based on the Zigbee specification with their Item Number, Description, Status, Selection, Endpoint, MirrorEndpoint, Value and Reference Numbers. Expand each PICS to select an individual test case to run and add their remaining fields such as Endpoint, Mirror Endpoint and Value, if any. See Figure 2-29.

Figure 2-29. PICS Selection - Filling the Details



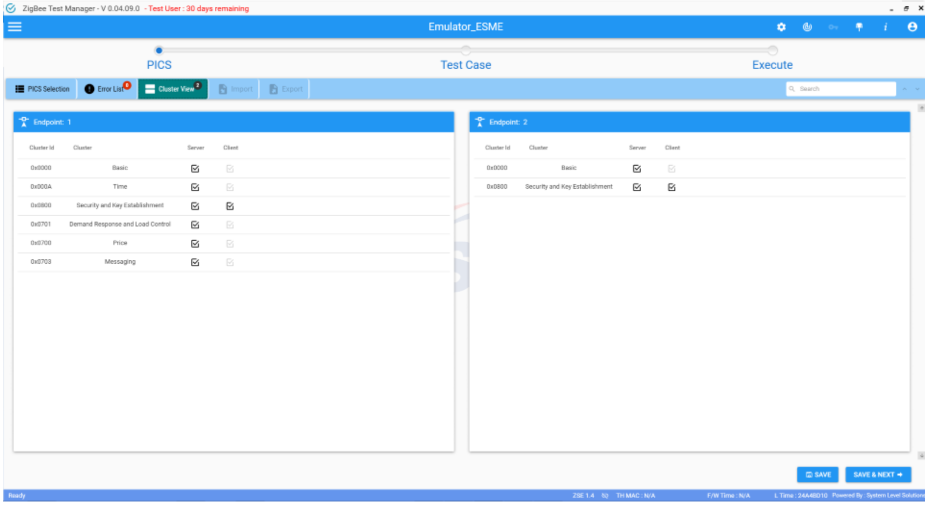
ZigBee Test Manager - V 0.04.09.0 - Test User : 30 days remaining

Id	Item Number	Description	Status	Selection	Endpoint	MirrorEndpoint	Value	Reference
Basic cluster server capabilities	Selected: 3/4	Endpoint or Mirror Endpoint missing						
Basic cluster client capabilities	Selected: 0/1							
Time cluster server capabilities	Selected: 0/9							
Time cluster client capabilities	Selected: 0/1							
Key Establishment cluster server capabilities	Selected: 9/9							
250	KECS1	Is the Key Establishment Cluster supported as a server?	M	☑	1			[R2]/C.3.1
251	KECS2	Is the KeyEstablishmentSuite attribute supported?	KECS1:M	☑	1			[R2]/C.3.1.2.2.1
252	KECS3	Is the reception of Initiate Key Establishment Request command supported?	KECS1:M	☑	1			[R2]/C.3.1.2.3.1
253	KECS4	Is the reception of Ephemeral Data Request command supported?	KECS1:M	☑	1			[R2]/C.3.1.2.3.2
254	KECS5	Is the reception of Confirm Key Data Request command supported?	KECS1:M	☑	1			[R2]/C.3.1.2.3.3
255	KECS6	Is the reception of Terminate Key Establishment command supported?	KECS1:M	☑	1			[R2]/C.3.1.2.3.4
256	KECS7	Is the generation of Initiate Key Establishment Response command supported?	KECS1:M	☑	1			[R2]/C.3.1.2.4 / C.3.1.3.3.1
257	KECS8	Is the generation of Ephemeral Data Response command supported?	KECS1:M	☑	1			[R2]/C.3.1.2.4 / C.3.1.3.3.2
258	KECS9	Is the generation of Confirm Key Data Response command supported?	KECS1:M	☑	1			[R2]/C.3.1.2.4 / C.3.1.3.3.3
Key Establishment cluster client capabilities	Selected: 10/10	Endpoint or Mirror Endpoint missing						
260	KECC1	Is the Key Establishment Cluster supported as a client?	M	☑				[R2]/C.3.1
261	KECC2	Is the KeyEstablishmentSuite attribute supported?	KECC1:M	☑				[R2]/C.3.1.3.2.1
262	KECC3	Is the reception of Initiate Key Establishment Response command supported?	KECC1:M	☑				[R2]/C.3.1.3.3.1
263	KECC4	Is the reception of Ephemeral Data Response command supported?	KECC1:M	☑				[R2]/C.3.1.3.3.2
264	KECC5	Is the reception of Confirm Key Data Response command supported?	KECC1:M	☑				[R2]/C.3.1.3.3.3
265	KECC6	Is the reception of Terminate Key Establishment command supported?	KECC1:M	☑				[R2]/C.3.1.3.3.4
266	KECC7	Is the generation of Initiate Key Establishment Request command supported?	KECC1:M	☑				[R2]/C.3.1.3.4 / C.3.1.2.3.1
267	KECC8	Is the generation of Ephemeral Data Request command supported?	KECC1:M	☑				[R2]/C.3.1.3.4 / C.3.1.2.3.2

Ready ZSE 1.4 TH MAC : N/A F/W Time : N/A L Time : 24A4BC5D Powered By : System Level Solutions

Based on the entered value, the final cluster is listed with endpoints in the Cluster View option. See Figure 2-30.

Figure 2-30. PICS Selection - Cluster View



Endpoint: 1

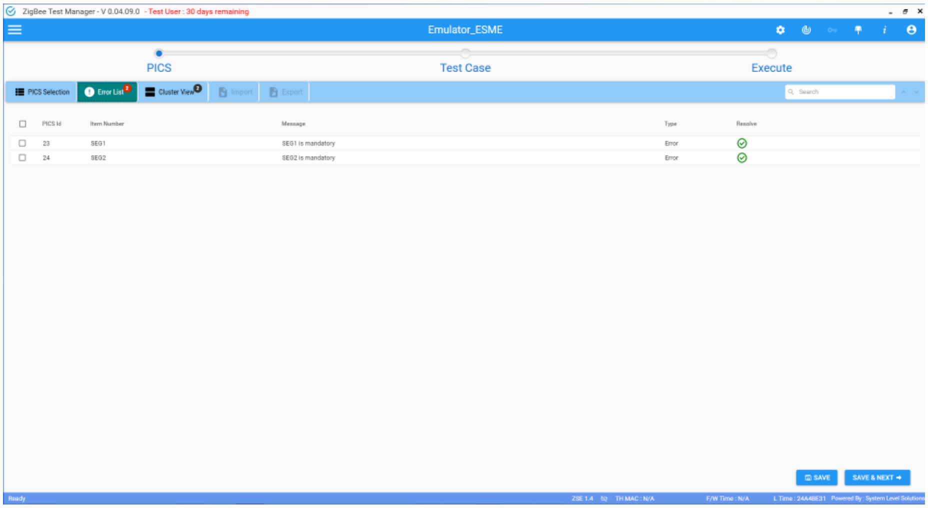
Cluster Id	Cluster	Server	Client
0x0000	Basic	☑	☐
0x000A	Time	☑	☐
0x0800	Security and Key Establishment	☑	☑
0x0701	Demand Response and Load Control	☑	☐
0x0700	Price	☑	☐
0x0703	Messaging	☑	☐

Endpoint: 2

Cluster Id	Cluster	Server	Client
0x0000	Basic	☑	☐
0x0800	Security and Key Establishment	☑	☑

While filling the details of the PICS, if there is any errors then it will be listed under Error View. See Figure 2-31.

Figure 2-31. PICS Selection - Error View



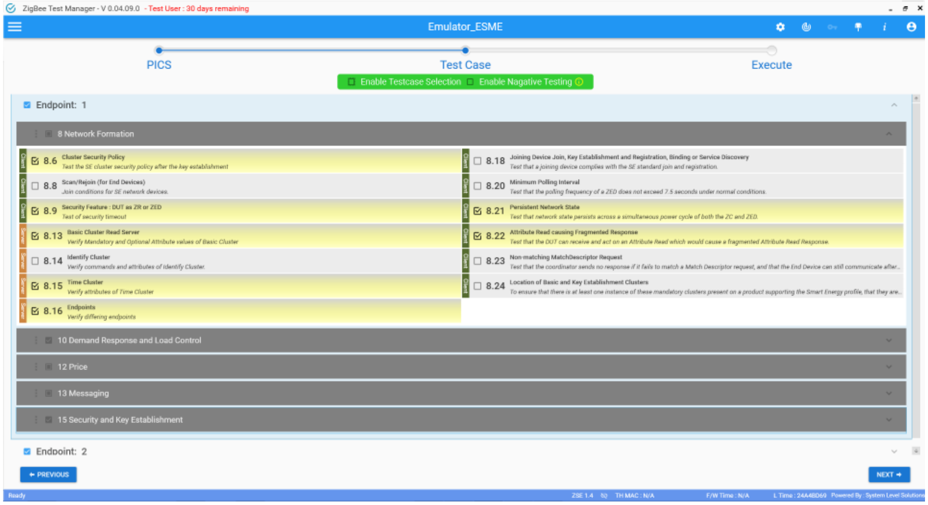
☐	PICS Id	Item Number	Message	Type	Resolve
☐	23	SEG1	SEG1 is mandatory	Error	✓
☐	24	SEG2	SEG2 is mandatory	Error	✓

After selecting the PICS, click on Next NEXT ➔ button at the bottom of the page to move to test case stage.

Test Case

Based on the PICS selection, the test cases are prepared endpoint wise and displayed as shown in Figure 2-32.

Figure 2-32. Test Using PICS Selection - Test Case



ZigBee Test Manager - V 0.04.09.0 - Test User : 30 days remaining

□ Enable Testcase Selection □ Enable Nagative Testing

☑ Endpoint: 1

8 Network Formation

- Client ☑ 8.6 Cluster Security Policy — *Test the SE cluster security policy after the key establishment*
- Client □ 8.8 Scan/Rejoin (for End Devices) — *Join conditions for SE network devices.*
- Client ☑ 8.9 Security Feature : DUT as ZR or ZED — *Test of security timeout*
- Server ☑ 8.13 Basic Cluster Read Server — *Verify Mandatory and Optional Attribute values of Basic Cluster*
- Server □ 8.14 Identify Cluster — *Verify commands and attributes of Identify Cluster.*
- Server ☑ 8.15 Time Cluster — *Verify attributes of Time Cluster*
- Server ☑ 8.16 Endpoints — *Verify differing endpoints*
- Client □ 8.18 Joining Device Join, Key Establishment and Registration, Binding or Service Discovery — *Test that a joining device complies with the SE standard join and registration.*
- Client □ 8.20 Minimum Polling Interval — *Test that the polling frequency of a ZED does not exceed 7.5 seconds under normal conditions.*
- Client ☑ 8.21 Persistent Network State — *Test that network state persists across a simultaneous power cycle of both the ZC and ZED.*
- Client ☑ 8.22 Attribute Read causing Fragmented Response — *Test that the DUT can receive and act on an Attribute Read which would cause a fragmented Attribute Read Response.*
- Client □ 8.23 Non-matching MatchDescriptor Request — *Test that the coordinator sends no response if it fails to match a Match Descriptor request, and that the End Device can still communicate after...*
- Client □ 8.24 Location of Basic and Key Establishment Clusters — *To ensure that there is at least one instance of these mandatory clusters present on a product supporting the Smart Energy profile, that they are...*

10 Demand Response and Load Control

12 Price

13 Messaging

15 Security and Key Establishment

☑ Endpoint: 2

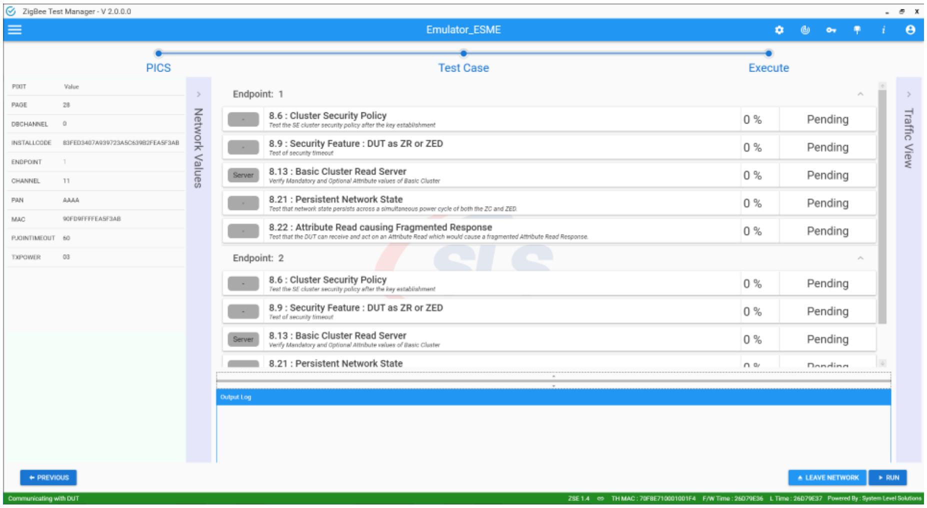
Ready ZSE 1.4 TH MAC : N/A F/W Time : N/A L Time : 24A4BD69 Powered By : System Level Solutions

Mandatory test cases are highlighted with yellow background. Select/de-select the test cases based on the requirement. To change the test case list, click on Previous ← PREVIOUS button else click Next NEXT ➔ button to move to Execute stage.

Execute

It lists all selected test cases to execute over DUT along with the Network Values, Traffic View, and Output Log. See Figure 2-33.

Figure 2-33. Test Using PICS Selection - Execute



PIXIT	Value
PAGE	28
DBCHANNEL	0
INSTALLCODE	83FED3407A939723A5C639B2FEA5F3AB
ENDPOINT	1
CHANNEL	11
PAN	AAAA
MAC	90FD9FFFFEA5F3AB
PJOINTIMEOUT	60
TXPOWER	03

Endpoint: 1

-	8.6 : Cluster Security Policy *Test the SE cluster security policy after the key establishment*	0 %	Pending
-	8.9 : Security Feature : DUT as ZR or ZED *Test of security timeout*	0 %	Pending
Server	8.13 : Basic Cluster Read Server *Verify Mandatory and Optional Attribute values of Basic Cluster*	0 %	Pending
-	8.21 : Persistent Network State *Test that network state persists across a simultaneous power cycle of both the ZC and ZED.*	0 %	Pending
-	8.22 : Attribute Read causing Fragmented Response *Test that the DUT can receive and act on an Attribute Read which would cause a fragmented Attribute Read Response.*	0 %	Pending

Endpoint: 2

-	8.6 : Cluster Security Policy *Test the SE cluster security policy after the key establishment*	0 %	Pending
-	8.9 : Security Feature : DUT as ZR or ZED *Test of security timeout*	0 %	Pending
Server	8.13 : Basic Cluster Read Server *Verify Mandatory and Optional Attribute values of Basic Cluster*	0 %	Pending
	8.21 : Persistent Network State	0 %	Pending

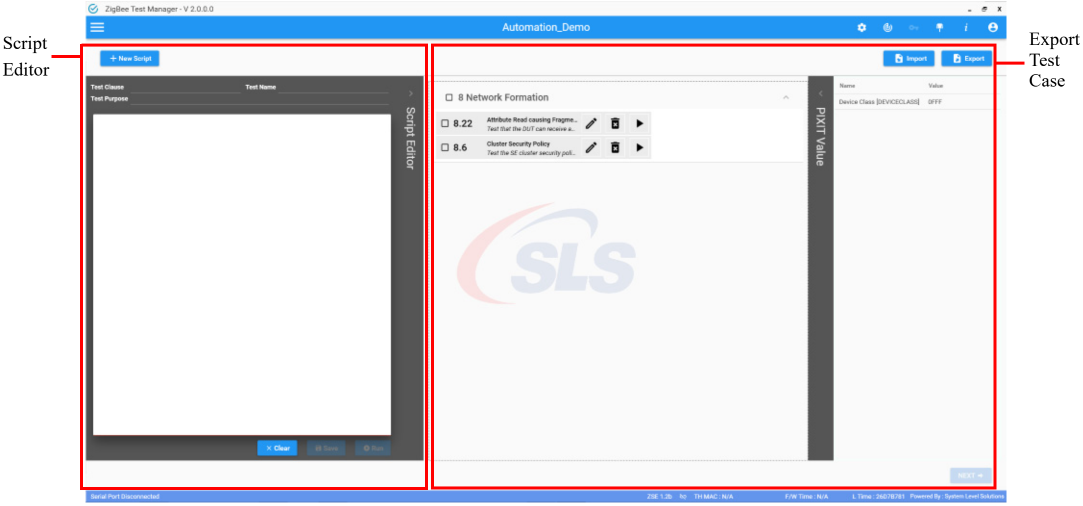
Communicating with DUT   ZSE 1.4   TH MAC : 70F8E71000100 1F4   F/W Time : 26D79E36   L Time : 26D79E37   Powered By : System Level Solutions

Click on Run button to start executing the test cases sequentially. For uninterrupted execution, click on Continue Uninterrupted button at the bottom of the page. All the button and it’s functionality and test case status indication remains same as mentioned in Table 2-7 and Table 2-8 respectively. After successful running the test cases, it displays the test results in HTML report in default browser. See Figure 2-20.

Custom Script Operation

Custom script allow to create new custom script along with DUT commands, and execute. It allows to export profile specific written test scripts by Zigbee Test Manager tool. Figure 2-34. shows the custom script operation window.

Figure 2-34. Custom Script Operation Window



There are 2 section in the window - Script Editor, and Export Test Case.

Script Editor

The script editor allows to write and execute the custom script. The script commands are available which helps the user to write a script and execute. To create a script, click on New Script + New Script button. Fill the Test Clause, Test Name, and Test Purpose in the script. The editor have 3 buttons - Clear, Save and Run.

Table 2-13 shows the prefix list to write a custom script.

Table 2-13. Custom Script Prefix List

Prefix	Syntax	Description
Print	print {<information>}	Print the information
Prompt Wait	prompt wait <No of seconds> prompt wait <No of seconds> {<information>}	Helps to wait for specific time while executing the command
Prompt Continue	prompt continue {<information>}	Notify information to user and prompts them for a response

Table 2-13. Custom Script Prefix List

Prefix	Syntax	Description

Prompt Check	prompt check {<information>}
	[YES]
	{
	<Scripts>
	}
	[NO]
	{
	<Scripts>
	}
		Notify information to the user and prompts them for a response either YES/NO, based on response script block will be executed

Command	Command <Command Name> {<Parameter 1>...<Parameter n>}	Write specific command to execute

Expect	expect {<Command Payload Name 1>=<Expected value 1>,…,<Command Payload Name n>=<Exacted value n>}
	Multiple Expect:
	expect {<Command Payload Name 1>=<Expected value 1>,…,<Command Payload Name n>=<Exacted value n>} || expect {<Command Payload Name 1>=<Expected value 1>,…,<Command Payload Name n>=<Exacted value n>}
		Validate the response data, based on the expected value result will be generated
		Multiple expect statement are added by using "||" operator.

Table 2-14 shows the commands used in custom script.

Table 2-14. Custom Script Command List

Command	Syntax	Description
Global Commands		
ApsSecurity	ApsSecurity {<bool value>}	Set APS security for outgoing APS frames
ClearBindingTable	ClearBindingTable	Clear binding table entry's in test harness
ClearKeys	ClearKeys	Clear key table entry's in test harness
RadioOff	RadioOff {<bool value>}	Turn ON/Off the test harness radio
NetworkSecurity	NetworkSecurity {<bool value>}	Set the network layer security
ApsLayerSecurity	ApsLayerSecurity {<byte value>}	Set APS security for outgoing APS frames

Table 2-14. Custom Script Command List

Command	Syntax	Description
Direction	Direction {<int direction>}	Set ZCL layer frame control direction bit for outgoing frames
TimeSync	TimeSync {<ushort shortAddress> <byte srcEndpoint> <byte dstEndpoint>}	Sync the time between DUT and test harness
SetLocalTime	SetLocalTime	Set local machine to time to test harness
DefaultResponseBit	DefaultResponseBit {<byte bitValue>}	Set ZCL layer frame control DefaultResponse bit for outgoing frames
Network Command		
NetworkForm	NetworkForm {<ushort panId> <int radioTXpower> <int channel>}	Forming a new network
AddEndDeviceInstallCode	AddEndDeviceInstallCode {<int index> <byte[] EUI64> <byte[] installCode>}	Whitelisting the device to join the network
PermitJoin	PermitJoin {<int time>}	Enabling permit join time of specific time
NetworkFindAndJoin	NetworkFindAndJoin	Perform the find and join operation for joining
NetworkJoin	NetworkJoin {<ushort panId> <int radioTXpower> <int channel>}	Send specific join request
NetworkLeave	NetworkLeave	Add the test harness network
AddLinkKey	AddLinkKey {<int index> <byte[] EUI64> <byte[] linkKey>}	Add link in link key table
ChangeNWKKey	ChangeNWKKey {<byte[] nwkKey>}	Change network key
SetRadioChannel	SetRadioChannel {<int channel>}	Switch current channel to new channel
ChangeChannel	ChangeChannel {<int newChannel>}	Change current channel with new channel
FindUnusedPanIdAndForm Network	FindUnusedPanIdAndFormNetwork	Search and form the network on unused PAN ID
SetExtendedPanId	SetExtendedPanId {byte[] extendedPANId}	Set extended PAN ID in the test harness

Table 2-14. Custom Script Command List

Command	Syntax	Description
General Commands		
ReadAttribute	ReadAttribute {<ushort clusterId> <ushort attributeId> <ushort shortAddress> <byte srcEndpoint> <byte dstEndpoint>}	Send read attribute request to DUT
WriteAttribute	WriteAttribute {<ushort clusterId> <ushort attributeId> <byte dataType> <byte[] data> <ushort shortAddress> <byte srcEndpoint> <byte dstEndpoint>}	Send write attribute request to DUT
Write	Write {<byte endpoint> <ushort clusterId> <ushort attributeId> <boolisServerAttribute> <byte dataType> <byte[] data>}	Write attribute data in the test harness
Report	Report {<byte srcEndpoint> <ushort clusterId> <ushort attributeId> <byte mask> <ushort shortAddress> <byte dstEndpoint>}	Send read report command to DUT
Raw	Raw {<ushort shortAddress> <byte srcEndpoint> <byte dstEndpoint> <ushort clusterId> <string message>}	Send any ZCL cluster specific command to the DUT
ZDO Command		
Bind	Bind {<ushort shortAddress> <byte srcEndpoint> <byte dstEndpoint> <ushort clusterId> <byte[] destEUI>}	Send bind request to DUT
ClearInOutCluster	ClearInOutCluster {<string type>}	Clear InOutCluster list for match descriptor request
AddInOutCluster	AddInOutCluster(<string type> <ushort clusterId>)	Add cluster InOutCluster list for match descriptor request
MatchDescriptorRequest	MatchDescriptorRequest {<ushort nodeId> <ushort profileId>}	Send match descriptor request to DUT
Active	Active {<ushort nodeId>}	Send active endpoint request to DUT
SimpleDescriptorRequest	SimpleDescriptorRequest {<ushort nodeId> <byte endpoint>}	Send simple descriptor request to DUT
ZdoRouteRequest	ZdoRouteRequest {<ushort nodeId> <int index>}	Send route request to DUT
ZdoMgmtLqi	ZdoMgmtLqi {<ushort nodeId> <int statrIndex>}	Send ZDO Mgmt_Lqi_req command to DUT
ZdoMgmtBind	ZdoMgmtBind {<ushort nodeId> <int statrIndex>}	Send ZDO Mgmt_Bind_req command to DUT
NodeDescriptor	NodeDescriptor {<ushort nodeId>}	Send node descriptor request to The DUT

Table 2-14. Custom Script Command List

Command	Syntax	Description
OTA Cluster Command		
ImageNotify	ImageNotify {<ushort destAddress> <byte endpoint> <byte payloadType> <byte queryJitter> <ushort manufacturerId> <ushort imageTypeId> <uint firmwareVersion>}	Send image notify command to DUT
QueryNextImage	QueryNextImage	Send query next image request command to DUT
UpgradePolicy	UpgradePolicy {<int policyValue>}	Set upgrade policy for upgrade image
UpgradeEndRequest	UpgradeEndRequest	Send upgrade end request command to DUT
UpgradeEndResponse	UpgradeEndResponse {<ushort destAddress> <byte endpoint> <ushort manufacturerId> <ushort imageTypeId> <uint firmwareVersion>}	Send upgrade end response command to DUT
BlockRequestPolicy	BlockRequestPolicy {<int policyValue>}	Set block request policy
StopOtaClient	StopOtaClient	Stop OTA client process
StartOtaClient	StartOtaClient	Start OTA client process
ImageBlockRequest	ImageBlockRequest {<ushort manufacturerId> <ushort imageTypeId> <uint firmwareVersion>}	Send image block request command to DUT
SetQueryPolicy	SetQueryPolicy {<int policyValue>}	Set query policy when it receives a query request from the client
SetUpgradeTime	SetUpgradeTime {<uint timeValue>}	Set image upgrade time
SetPageRequest	SetPageRequest {<bool isPageRequestOn>}	Set page request ON or OFF for page request command
DiscoverOTAServer	DiscoverOTAServer	Discover OTA server
UPLOADFILETODIRECTORY	UPLOADFILETODIRECTORY {<ushort shortAdderess> <byte dstEndpoint>}	Upload OTA file in to allocated directory
REMOVEDIRECTORYFILES	REMOVEDIRECTORYFILES	Remove OTA file from the allocated directory
Smart Energy Profile Clusters Command **a) Metering**		
RequestFastPollMode	RequestFastPollMode {<byte updatePeriod> <byte duration> <ushort shortAddress> <byte srcEndpoint> <byte dstEndpoint>}	Send request fast poll mode command to DUT

Table 2-14. Custom Script Command List

Command	Syntax	Description
GetSnapshot	GetSnapshot {<uint earliestStartTime> <uint latestEndTime> <byte snapshotOffset> <uint snapshotCause> <ushort shortAddress> <byte srcEndpoint> <byte dstEndpoint>}	Send get snapshot command to DUT
TakeSnapshot	TakeSnapshot {<uint snapshotCause> <ushort shortAddress> <byte srcEndpoint> <byte dstEndpoint>}	Send take snapshot command to DUT
ScheduleSnapshot	ScheduleSnapshot {<uint issuerEventId> <uint commandIndex> <uint commandCount> <uint snapshotScheduleID> <uint snapshotStartTime> <uint snapshotSchedule> <uint snapshotPayloadType> <uint snapshotCause> <ushort shortAddress> <byte srcEndpoint> <byte dstEndpoint>}	Send schedule snapshot command to DUT
RequestMirror	RequestMirror {<ushort shortAddress> <byte srcEndpoint> <byte dstEndpoint>}	Send request mirror command to DUT
ConfigureMirror	ConfigureMirror {<uint issuerEventId> <uint reportingInterval> <bool mirrorNotificationReporting> <uint notificationScheme> <ushort shortAddress> <byte srcEndpoint> <byte dstEndpoint>}	Configure mirror on DUT
CfgNftScheme	CfgNftScheme {<uint issuerEventId> <uint notificationScheme> <uint notificationFlagOrder> <ushort shortAddress> <byte srcEndpoint> <byte dstEndpoint>}	Send configure notification scheme command to DUT
GetNtfyMsg	GetNtfyMsg {<uint notificationScheme> <ushort notificationFlagAttributeID> <uint notificationFlagN> <ushort shortAddress> <byte srcEndpoint> <byte dstEndpoint>}	Send get notified message command to DUT
ChangeSupply	ChangeSupply {<uint providerId> <uint issuerEventId> <uint requestDateTime> <uint implementationDateTime> <byte proposedSupplyStatus> <byte supplyControlBits> <ushort shortAddress> <byte srcEndpoint> <byte dstEndpoint>}	Send change supply command to DUT
RemoveMirror	RemoveMirror {<ushort shortAddress> <byte srcEndpoint> <byte dstEndpoint>}	Send remove mirror command to DUT
StartSampling	StartSampling {<uint issuerId> <uint startTime> <byte sampleType> <ushort sampleRequestInterval> <ushort maxNumberOfSamples> <ushort shortAddress> <byte srcEndpoint> <byte dstEndpoint>}	Send start sampling command to DUT
StartSamplingResponse	StartSamplingResponse {<ushort sampleId> <ushort shortAddress> <byte srcEndpoint> <byte dstEndpoint>}	Send start sampling response command to DUT

Table 2-14. Custom Script Command List

Command	Syntax	Description
GetSampleData	GetSampleData {<ushort sampleId> <uint earliestSampleTime> <byte sampleType> <ushort numberOfSamples> <ushort shortAddress> <byte srcEndpoint> <byte dstEndpoint>}	Send the get sample data command to DUT
ResetLoadLimitControl	ResetLoadLimitControl {<uint providerId> <uint issuerEventId> <ushort shortAddress> <byte srcEndpoint> <byte dstEndpoint>}	Send reset load limit control command to DUT
LocalChangeSupply	LocalChangeSupply {<byte proposedSupplyStatus> <ushort shortAddress> <byte srcEndpoint> <byte dstEndpoint>}	Send local change supply command to DUT
SetSupplyStatus	SetSupplyStatus {<uint issuerEventId> <byte supplyTamperState> <byte supplyDepletionState> <byte supplyUncontrolledFlowState> <byte loadLimitSupplyState> <ushort shortAddress> <byte srcEndpoint> <byte dstEndpoint>}	Send set supply status command to DUT
SetUncontrolledFlowThreshold	SetUncontrolledFlowThreshold {<uint providerId> <uint issuerEventId> <ushort uncontrolledFlowThreshold> <byte unitOfMeasure> <ushort multiplier> <ushort divisor> <byte stabilisationPeriod> <ushort measurementPeriod> <ushort shortAddress> <byte srcEndpoint> <byte dstEndpoint>}	Send set uncontrolled flow threshold command to DUT
GetMeteringProfile	GetMeteringProfile {<byte intervalChanel> <uint endTime> <byte numberofPeriod> <ushort shortAddress> <byte srcEndpoint> <byte dstEndpoint>}	Send the get metering profile command to DUT
b) Price		
PublishPrice	PublishPrice {<uint providerId> <string rateLabel> <uint issuerEventId> <uint currentTime> <byte unitOfMeasure> <ushort currency> <byte priceTrailingDigitAndPricTier> <byte numberOfPriceTiersAndRegisterTier> <uint startTime> <ushort durationInMinutes> <uint price> <byte priceRatio> <uint generationPrice> <byte generationPriceRatio> <uint alternateCostDelivered> <byte alternateCostUnit> <byte alternateCostTrailingDigit> <byte numberOfBlockThresholds> <byte priceControl> <byte numberOfGenerationTiers> <byte generationTier> <byte extendedNumberOfPriceTiers> <byte extendedPriceTier> <byte extendedRegisterTier> <ushort shortAddress> <byte srcEndpoint> <byte dstEndpoint}	Publish price command to DUT
GetScheduledPrices	GetScheduledPrices {<uint startTime> <byte numberofEvents> <ushort shortAddress> <byte srcEndpoint> <byte dstEndpoint>}	Send get scheduled prices to DUT
PriceClear	PriceClear {<byte endpoint>}	Clear local price table value

Table 2-14. Custom Script Command List

Command	Syntax	Description
GetBlockPeriods	GetBlockPeriods {<uint startTime> <byte numberOfEvents> <byte tariffType> <ushort shortAddress> <byte srcEndpoint> <byte dstEndpoint>}	Send get block periods command to DUT
ConfigureBlockPeriod	ConfigureBlockPeriod {<byte endpoint> <uint providerId> <uint issuerEventId> <uint blockPeriodStartTime> <uint blockPeriodDuration> <byte blockPeriodControl> <byte blockPeriodDurationType> <uint thresholdMultiplier> <uint thresholdDivisor> <byte tariffType> <byte tariffResolutionPeriod>}	Configure block period in test harness
GetConversionFactor	GetConversionFactor {<uint earliestStartTime> <uint minIssuerEventId> <ushort numberOfCommands> <ushort shortAddress> <byte srcEndpoint> <byte dstEndpoint>}	Send get conversion factor command to DUT
GetCalorificValue	GetCalorificValue {<uint earliestStartTime> <uint minIssuerEventId> <ushort numberOfCommands> <ushort shortAddress> <byte srcEndpoint> <byte dstEndpoint>}	Send get calorific value command to DUT
GetTariffInfo	GetTariffInfo {<uint earliestStartTime> <uint minIssuerEventId> <ushort numberOfCommands> <byte tarrifType> <ushort shortAddress> <byte srcEndpoint> <byte dstEndpoint>}	Send get tariff info command to DUT
GetPriceMatrix	GetPriceMatrix {<uint issuerTariffId> <ushort shortAddress> <byte srcEndpoint> <byte dstEndpoint>}	Send get price matrix command to DUT
GetBlockThresholds	GetBlockThresholds {<uint issuerTariffId> <ushort shortAddress> <byte srcEndpoint> <byte dstEndpoint>}	Send get block thresholds command to DUT
FillPublishTariffInformation	FillPublishTariffInformation {<uint providerId> <uint issuerEventId> <uint issuerTariffId> <uint startTime> <byte tariffTypeChargingScheme> <string tariffLabel> <byte numberOfPriceTiersInUse> <byte numberOfBlockThresholdsInUse> <byte unitOfMeasure> <ushort currency> <byte priceTrailingDigit> <byte standingCharge> <byte tierBlockMode> <byte blockThresholdMultiplier> <byte blockThresholdDivisor> <byte endpoint> <byte status>}	Configure tariff information to test harness
PublishCppEvent	PublishCppEvent {<uint providerId> <uint issuerEventId> <uint startTime> <ushort duration> <byte tariffType> <byte priceTier> <byte cppAuth> <ushort shortAddress> <byte srcEndpoint> <byte dstEndpoint>}	Send publish CPP event command to DUT
GetTariffCancellation	GetTariffCancellation {<ushort shortAddress> <byte srcEndpoint> <byte dstEndpoint>}	Send get tariff cancellation command to DUT
CancelTariff	CancelTariff {<uint providerId> <uint issuerTariffId> <byte tariffType> <ushort shortAddress> <byte srcEndpoint> <byte dstEndpoint>}	Send cancel tariff command to DUT

Table 2-14. Custom Script Command List

Command	Syntax	Description
PublishCurrencyConversion	PublishCurrencyConversion {<uint providerId> <uint issuerEventId> <uint startTime> <ushort oldCurrency> <ushort newCurrency> <uint conversionFactor> <byte conversionFactorTrailingDigit> <uint currencyChangeControlFlags> <ushort shortAddress> <byte srcEndpoint> <byte dstEndpoint>}	Send publish currency conversion command to DUT
GetCurrencyConversion	GetCurrencyConversion {<ushort shortAddress> <byte srcEndpoint> <byte dstEndpoint>}	Send get currency conversion command to DUT
PublishPriceBuffer	PublishPriceBuffer {<uint providerId> <string rateLabel> <uint issuerEventId> <byte unitOfMeasure> <ushort currency> <byte priceTrailingDigitAndPricTier> <byte numberOfPriceTiersAndRegisterTier> <uint startTime> <ushort durationInMinutes> <uint price> <byte priceRatio> <uint generationPrice> <byte generationPriceRatio> <uint alternateCostDelivered> <byte alternateCostUnit> <byte alternateCostTrailingDigit> <byte numberOfBlockThresholds> <byte priceControl> <byte relaventEndpoint> <byte indexInPriceTable>}	Configure price table to test harness
PublishPriceMetrix	PublishPriceMetrix {<uint providerId> <uint issuerEventId> <uint startTime> <uint issuerTariffId> <byte commandIndex> <byte totalNoOfCmd> <byte subPayloadControl> <byte[] priceMetrixSubPayload> <ushort shortAddress> <byte srcEndpoint> <byte dstEndpoint>}	Send publish price metrix command to DUT
ClearTariffInformation	ClearTariffInformation {<byte endpoint>}	Clear tariff information from test harness
ClearPriceMetrix	ClearPriceMetrix {<byte endpoint>}	Clear price metrix from test harness
ClearBlockThreshold	ClearBlockThreshold {<byte endpoint>}	Clear block threshold from test harness
PublishBlockThreshold	PublishBlockThreshold {<uint providerId> <uint issuerEventId> <uint startTime> <uint issuerTariffId> <byte commandIndex> <byte noOfCmd> <byte subPayloadControl> <byte[] subPayload> <ushort shortAddress> <byte srcEndpoint> <byte dstEndpoint>}	Send publish block threshold to DUT
GetCo2ValueCommand	GetCo2ValueCommand {<uint earliestStartTime> <uint minIssuerEventID> <byte noOfCmd> <byte tariffType> <ushort shortAddress> <byte srcEndpoint> <byte dstEndpoint>}	Send get CO_2 value command to DUT

Table 2-14. Custom Script Command List

Command	Syntax	Description
PublishCo2ValCommand	PublishCo2ValCommand {<uint providerId> <uint issuerEventID> <uint startTime> <byte tariffType> <uint c02Value> <byte c02Unit> <byte c02ValueTrailingDigit> <ushort shortAddress> <byte srcEndpoint> <byte dstEndpoint>}	Send publish CO_2 value command to DUT
ConfigureBlockThresholds	ConfigureBlockThresholds {<byte endpoint> <uint providerId> <uint issuerEventId> <uint startTime> <uint issuerTariffId> <byte commandIndex> <byte numberOfCommands> <byte subpayloadControl> <byte[] payload>}	Configure block thresholds to test harness
ConfigurePriceMatrix	ConfigurePriceMatrix {<byte endpoint> <uint providerId> <uint issuerEventId> <uint startTime> <uint issuerTariffId> <byte commandIndex> <byte numberOfCommands> <byte subpayloadControl> <byte[] priceMetrixSubPayload>}	Configure price matrix to test harness
ConfigureCo2Value	ConfigureCo2Value {<byte endpoint> <uint issuerEventId> <uint startTime> <uint providerId> <byte tariffType> <uint co2Value> <byte co2ValueUnit> <byte co2ValueTrailingDigit>}	Configure CO_2 value to test harness
ClearCo2Value	ClearCo2Value {<byte endpoint>}	Send clear CO_2 value from test harness
GetTierLabel	GetTierLabel {<uint issuerTrafficId> <ushort shortAddress> <byte srcEndpoint> <byte dstEndpoint>}	Send get tier label command to DUT
PublishTierLabels	PublishTierLabels {<uint providerId> <uint issuerEventId> <uint issuerTariffId> <byte commandIndex> <byte noOfCmd> <byte noOfLabels> <byte[] tierPayload> <ushort shortAddress> <byte srcEndpoint> <byte dstEndpoint>}	Send publish tier labels command to DUT
SetTierLabel	SetTierLabel {<byte endpoint> <byte index> <byte valid> <uint providerId> <uint issuerEventId> <uint issuerTariffId> <byte tierId> <byte[] tierLabel>}	Configure tier label value in the test harness
ClearTierLabelsTable	ClearTierLabelsTable {<byte endpoint>}	Clear tier label values from test harness
GetBillingPeriod	GetBillingPeriod {<uint earlieststartTime> <uint minIssuerEventId> <byte noOfCmds> <byte tariffType> <ushort shortAddress> <byte srcEndpoint> <byte dstEndpoint>}	Send get billing period command to DUT
PublishBillingPeriod	PublishBillingPeriod {<uint providerId> <uint issuerEventId> <uint startTime> <uint duration> <byte durationType> <byte tariffType> <ushort shortAddress> <byte srcEndpoint> <byte dstEndpoint>}	Send publish billing period command to DUT

Table 2-14. Custom Script Command List

Command	Syntax	Description
ConfigureBillingPeriod	ConfigureBillingPeriod {<byte endpoint> <uint startTime> <uint issuerEventId> <uint providerId> <uint billingPeriodDuration> <byte billingPeriodDurationType> <byte tariffType>}	Configure billing period in the test harness
ClearBillingPeriodTableEntry	ClearBillingPeriodTableEntry {<byte endpoint>}	Clear billing period table entry from test harness
GetConsolidatedBill	GetConsolidatedBill {<uint earlieststartTime> <uint minIssuerEventId> <byte noOfCmds> <byte tariffType> <ushort shortAddress> <byte srcEndpoint> <byte dstEndpoint>}	Send get consolidated bill command to DUT
PublishConsolidatedBill	PublishConsolidatedBill {<uint providerId> <uint issuerEventId> <uint startTime> <uint duration> <byte durationType> <byte tariffType> <uint consolidatedBill> <ushort currency> <byte billTrailingDigit> <ushort shortAddress> <byte srcEndpoint> <byte dstEndpoint>}	Send publish consolidated bill command to DUT
ConfigureConsolidatedBill	ConfigureConsolidatedBill {<byte endpoint> <uint startTime> <uint issuerEventId> <uint providerId> <uint billingPeriodDuration> <byte billingPeriodDurationType> <byte tariffType> <uint consolidatedBill> <ushort currency> <byte billTrailingDigit>}	Configure consolidated bill in the test harness
ClearConsolidatedBillTable	ClearConsolidatedBillTable {<byte endpoint>}	Clear consolidated bill table from test harness
GetCreditPayment	GetCreditPayment {<uint latestEndTime> <byte noOfRecords> <ushort shortAddress> <byte srcEndpoint> <byte dstEndpoint>}	Send get credit payment command to DUT
PublishCreditPayment	PublishCreditPayment {<uint providerId> <uint issuerEventId> <uint creditPaymentDueDate> <uint creditPaymentOverdueAmount> <byte creditPaymentStatus> <uint creditPayment> <uint creditPaymentDate> <byte[] creditPaymentRef> <ushort shortAddress> <byte srcEndpoint> <byte dstEndpoint>}	Send publish credit payment command to DUT
ConfigureCreditPayment	ConfigureCreditPayment {<byte endpoint> <byte index> <byte valid> <uint providerId> <uint issuerEventId> <uint creditPaymentDueDate> <uint creditPaymentOverdueAmount> <byte creditPaymentStatus> <uint creditPayment> <uint creditPaymentDate> <byte[] creditPaymentRef>}	Configure credit payment in the test harness
GetCurrentPrice	GetCurrentPrice {<byte commandOptions> <ushort destination> <byte inSrcEndpoint> <byte inDstEndpoint>}	Send get current price command to DUT

Table 2-14. Custom Script Command List

Command	Syntax	Description
ConfigureCurrencyConversion	ConfigureCurrencyConversion {<byte endpoint> <byte valid> <uint providerId> <uint issuerEventId> <uint startTime> <ushort oldCurrency> <ushort newCurrency> <uint conversionFactor> <byte conversionFactorTrailingDigit> <uint currencyChangeControlFlags>}	Configure currency conversion command in the test harness
CalorificValueAdd	CalorificValueAdd {<byte endpoint> <uint issuerEventId> <uint startTime> <uint calorificValue> <byte calorificValueUnit> <byte calorificValueTrailingDigit>}	Send configure calorific value on test harness
ClearCalorific	ClearCalorific {<byte endpoint>}	Clear calorific value from the test harness
ConversionFactorAdd	ConversionFactorAdd {<byte endpoint> <uint issuerEventId> <uint startTime> <uint conversionFactor> <byte conversionFactorTrailingDigit>}	Configure conversion factor value in test harness
ClearConversionFactor	ClearConversionFactor {<byte endpoint>}	Clear conversion factor value from the test harness
c) Tunneling		
RequestTunnel	RequestTunnel {<byte protocolId> <ushort manufactureCode> <byte flowControlSupport> <ushort maxIncommingTransferSize> <ushort shortAddress> <byte srcEndpoint> <byte dstEndpoint>}	Send request tunnel command to DUT
ToggleTunnelFull	ToggleTunnelFull	Set/unset tunnel status FULL in the test harness
GetSupportedTunnelProtocols	GetSupportedTunnelProtocols {<byte protocolOffset> <ushort shortAddress> <byte srcEndpoint> <byte dstEndpoint>}	Send get supported tunnel protocols command to DUT
ToggleTunnelBusy	ToggleTunnelBusy	Set/unset tunnel status BUSY in the test harness
TransferDataToClient	TransferDataToClient {<ushort tunnelId> <byte[] data> <ushort shortAddress> <byte srcEndpoint> <byte dstEndpoint>}	Send transfer data command to client from the test harness
TransferDataToServer	TransferDataToServer {<ushort tunnelId> <byte[] data> <ushort shortAddress> <byte srcEndpoint> <byte dstEndpoint>}	Send transfer data command to server from the test harness
CloseTunnel	CloseTunnel {<ushort tunnelId> <ushort shortAddress> <byte srcEndpoint> <byte dstEndpoint>}	Send close tunnel command to DUT

Table 2-14. Custom Script Command List

Command	Syntax	Description
d) Prepayment		
PrepaymentChgPmtMode	PrepaymentChgPmtMode {<uint providerId> <uint issuerEventId> <uint implementationDate> <ushort proposedPaymentControlConfiguration> <uint cutOffValue> <ushort shortAddress> <byte srcEndpoint> <byte dstEndpoint>}	Send change payment mode command to DUT
PrepaymentEmCredSetup	PrepaymentEmCredSetup {<uint issuerEventId> <uint startTime> <uint emCreditLimit> <uint emCreditThreshold> <ushort shortAddress> <byte srcEndpoint> <byte dstEndpoint>}	Send emergency credit setup command to DUT
PrepaymentCredAdj	PrepaymentCredAdj {<uint issuerEventId> <uint startTime> <byte creditAdjType> <uint creditAdjValue> <ushort shortAddress> <byte srcEndpoint> <byte dstEndpoint>}	Send credit adjustment command to DUT
PrepaymentConsTopUp	PrepaymentConsTopUp {<byte originatingDevice> <byte[] topUpCode> <ushort shortAddress> <byte srcEndpoint> <byte dstEndpoint>}	Send consumer top up command to DUT
PrepaymentGetTopUpLog	PrepaymentGetTopUpLog {<uint latestEndTime> <byte numberofRecords> <ushort shortAddress> <byte srcEndpoint> <byte dstEndpoint>}	Send get top up log command to DUT
PrepaymentGetPrepaySnap shot	PrepaymentGetPrepaySnapshot {<uint earliestStartTime> <uint latestEndTime> <byte snapshotOffset> <uint snapshotCause> <ushort shortAddress> <byte srcEndpoint> <byte dstEndpoint>}	Send get prepay snapshot command to DUT
PrepaymentSelAvEmCred	PrepaymentSelAvEmCred {<uint commandIssueDate> <byte originatingDevice> <ushort shortAddress> <byte srcEndpoint> <byte dstEndpoint>}	Send select available emergency credit command to DUT
PrepaymentSetLowCredWn gLvl	PrepaymentSetLowCredWngLvl {<uint lowCreditWarningLevel> <ushort shortAddress> <byte srcEndpoint> <byte dstEndpoint>}	Send set low credit warning level command to DUT
PrepaymentSetMaxCredLmt	PrepaymentSetMaxCredLmt {<uint providerID> <uint issuerEventID> <uint implementationDate> <uint maxCreditLevel> <uint maxCreditPerTopUp> <ushort shortAddress> <byte srcEndpoint> <byte dstEndpoint>}	Send set maximum credit limit command to dUT
PrepaymentSetOaDebtCap	PrepaymentSetOaDebtCap {<uint providerID> <uint issuerEventID> <uint implementationDate> <uint OverallDebtCap> <ushort shortAddress> <byte srcEndpoint> <byte dstEndpoint>}	Send set overall debt cap command to DUT

Table 2-14. Custom Script Command List

Command	Syntax	Description
PrepaymentChgDebt	PrepaymentChgDebt {<uint issuerEventId> <byte[] debtLabel> <int debtAmt> <byte debtRecoveryMethod> <byte debtAmountType> <uint debtRecoveryStartTime> <ushort debtRecoveryCollectionTime> <byte debtRecoveryFrequency> <int debtRecoveryAmt> <ushort debtRecoveryBalancePercentage> <ushort shortAddress> <byte srcEndpoint> <byte dstEndpoint>}	Send change debt command to DUT
PrepaymentGetDebtRepmtLog	PrepaymentGetDebtRepmtLog {<uint latestEndTime> <byte numberOfDebts> <byte debtType> <ushort shortAddress> <byte srcEndpoint> <byte dstEndpoint>}	Send get debt repayment log command to DUT
e) Device Management		
GetChangeOfTenancy	GetChangeOfTenancy {<ushort shortAddress> <byte srcEndpoint> <byte dstEndpoint>}	Send get change of tenancy to DUT
ConfigurePublishChangeOfTenancy	ConfigurePublishChangeOfTenancy {<uint providerId> <uint issuerEventId> <byte tariffType> <uint implementationDateTime> <uint proposedTenancyChangeControl> <byte pendingupdate>}	Configure change of tenancy in the test harness
GetChangeOfSupplier	GetChangeOfSupplier {<ushort shortAddress> <byte srcEndpoint> <byte dstEndpoint>}	Send get change of supplier command to DUT
ConfigurePublishChangeOfSupplier	ConfigurePublishChangeOfSupplier {<byte endpoint> <uint providerId> <uint issuerEventId> <byte tariffType> <uint ProposedProviderID> <uint ProviderChangeImplementationTime> <uint providerChangeControl> <string proposedProviderName> <string proposedProviderContactDetails>}	Configure change of supplier in the test harness
RequestNewPasswordResponse	RequestNewPasswordResponse {<uint issuerEventId> <uint implementationDateTime> <ushort durationInMinutes> <byte passwordType> <string password> <ushort shortAddress> <byte srcEndpoint> <byte dstEndpoint>}	Send request new password response command to DUT
DmServerconfigureRequestNewPasswordResponse	DmServerconfigureRequestNewPasswordResponse {<uint implementationDateTime> <ushort durationInMinutes> <byte passwordType> <string newpassword>}	Configure request new password response in the test harness
RequestNewPassword	RequestNewPassword {<byte passwordType> <ushort shortAddress> <byte srcEndpoint> <byte dstEndpoint>}	Send request new password command to DUT
SetCIN	SetCIN {<uint issuerEventId> <uint implementationDateTime> <uint providerId> <string CIDN>}	Set CIN in the test harness
UpdateCIN	UpdateCIN {<ushort shortAddress> <byte inSrcEndpoint> <byte inDstEndpoint>}	Send update CIN command to the DUT

Table 2-14. Custom Script Command List

Command	Syntax	Description
PendingUpdates	PendingUpdates {<uint pendingUpdatesMask>}	Configure pending updates in the test harness
GetCIN	GetCIN {<ushort shortAddress> <byte srcEndpoint> <byte dstEndpoint>}	Send get CIN command to DUT
SetSiteId	SetSiteId {<uint issuerEventId> <uint implementationDateTime> <uint providerId> <string Siteid>}	Set site ID in the test harness
UpdateSiteId	UpdateSiteId {<ushort shortAddress> <byte inSrcEndpoint> <byte inDstEndpoint>}	Send update site ID to the DUT
GetSiteId	GetSiteId {<ushort shortAddress> <byte srcEndpoint> <byte dstEndpoint>}	Send get site ID command to DUT
GeteventConfig	GeteventConfig {<ushort eventId> <ushort shortAddress> <byte srcEndpoint> <byte dstEndpoint>}	Send get event configuration command to DUT
SetEventConfigure	SetEventConfigure {<uint issuerEventId> <uint startDateTime> <byte eventConfiguration> <byte configurationControl> <byte[] eventConfigurationPayload> <ushort shortAddress> <byte srcEndpoint> <byte dstEndpoint>}	Send set event configure command to DUT
PublishChangeOfTenancy	PublishChangeOfTenancy {<ushort shortAddress> <byte srcEndpoint> <byte dstEndpoint>}	Send publish change of tenancy command to DUT
PublishChangeOfSupplier	PublishChangeOfSupplier {<ushort shortAddress> <byte srcEndpoint> <byte dstEndpoint>}	Send publish change of supplier command to DUT
ConfigureAndPublishChangeOfTenancy	ConfigureAndPublishChangeOfTenancy {<uint providerId> <uint issuerEventId> <byte tariffType> <uint implementationDateTime> <uint proposedTenancyChangeControl> <byte pendingupdate> <ushort shortAddress> <byte srcEndpoint> <byte dstEndpoint>}	Configure and publish change of tenancy command to DUT
ConfigureAndPublishChangeOfSupplier	ConfigureAndPublishChangeOfSupplier {<byte endpoint> <uint providerId> <uint issuerEventId> <byte tariffType> <uint ProposedProviderID> <uint ProviderChangeImplementationTime> <uint providerChangeControl> <string proposedProviderName> <string proposedProviderContactDetails> <ushort shortAddress> <byte srcEndpoint> <byte dstEndpoint>}	Configure and publish change of supplier command to DUT

Table 2-14. Custom Script Command List

Command	Syntax	Description
f) Event		
GetEventLog	GetEventLog {<byte eventControlLogId> <ushort eventId> <uint startsTime> <uint endTime> <byte numberOfEvents> <ushort eventOffset> <ushort shortAddress> <byte srcEndpoint> <byte dstEndpoint>}	Send get event log command to DUT
SetEventLog	SetEventLog {<byte logId> <byte index> <ushort eventId> <uint eventStartTime> <string eventData>}	Configure event logs in the test harness
ClearEventLog	ClearEventLog {<byte logId> <ushort shortAddress> <byte srcEndpoint> <byte dstEndpoint>}	Send clear event log command to DUT
g) Calender		
GetCalendar	GetCalendar {<uint earliestStartTime> <uint minIssuerEventID> <byte noOfCalendars> <byte calendarType> <uint providerId> <ushort shortAddress> <byte srcEndpoint> <byte dstEndpoint>}	Send get calendar command to DUT
GetDayProfile	GetDayProfile {<uint providerId> <uint calendarId> <byte startDayId> <byte numberOfDays> <ushort shortAddress> <byte srcEndpoint> <byte dstEndpoint>}	Send get day profile command to DUT
GetWeekProfile	GetWeekProfile {<uint providerId> <uint calendarId> <byte startWeekId> <byte numberOfWeeks> <ushort shortAddress> <byte srcEndpoint> <byte dstEndpoint>}	Send get week profile command to DUT
GetSeasons	GetSeasons {<uint providerId> <uint calendarId> <ushort shortAddress> <byte srcEndpoint> <byte dstEndpoint>}	Send get seasons command to DUT
GetSpecialDays	GetSpecialDays {<uint startTime> <byte numberOfEvents> <byte calendarType> <uint providerId> <uint calendarId> <ushort shortAddress> <byte srcEndpoint> <byte dstEndpoint>}	Send get special days command to DUT
LoadFlatCalendarD3	LoadFlatCalendarD3 {<byte index> <uint providerId> <uint issuerEventId> <uint calendarId> <uint activationTime> <byte calendarType> <string calendarName>}	Configure flat calendar as per Appendix D.3 (as per test specification) in the test harness
LoadEnhancedCalendarD2	LoadEnhancedCalendarD2 {<byte index> <uint providerId> <uint issuerEventId> <uint calendarId> <uint activationTime> <byte calendarType> <string calendarName>}	Configure flat calendar as per Appendix D.2 (as per test specification) in the test harness
PublishCalendar	PublishCalendar {<ushort shortAddress> <byte srcEndpoint> <byte dstEndpoint> <byte calendarIndex>}	Publish loaded calender to DUT
GetCalendarCancellation	GetCalendarCancellation {<ushort shortAddress> <byte srcEndpoint> <byte dstEndpoint>}	Send get calendar cancellation command to DUT

Table 2-14. Custom Script Command List

Command	Syntax	Description
h) Demand Response and Load Control		
CECommand	CECommand {<byte endpoint> <byte index> <uint issuerEventId> <ushort deviceClass> <byte utilityEnrollmentGroup> <uint startTime> <ushort duration> <byte criticalityLevel> <byte coolingTempOffset> <byte heatingTempOffset> <ushort coolingTempSetPoint> <ushort heatingTempSetPoint> <byte avgLoadPercentage> <byte dutyCycle> <byte eventControl>}	Configure LCE command in the test harness
SendLCEMessage	SendLCEMessage {<ushort shortAddress> <byte srcEndpoint> <byte dstEndpoint> <byte index>}	Send configured LCE command to the DUT
ClearScheduledLoadControl Events	ClearScheduledLoadControlEvents {<byte endpoint>}	Clear scheduled load control events from the test harness
CancelLCECommand	CancelLCECommand {<uint issuerEventId> <ushort deviceClass> <byte utilityEnrollmentGroup> <byte cancelControl> <uint effectiveTime> <ushort shortAddress> <byte srcEndpoint> <byte dstEndpoint>}	Send cancel load control event command to DUT
SetLoadControlEventOptIn OrOut	SetLoadControlEventOptInOrOut {<byte endpoint> <uint eventId> <bool optIn>}	Set load control event optinorout option in the test harness
DrlcCancelAllCommand	DrlcCancelAllCommand {<byte cancelControl> <ushort shortAddress> <byte srcEndpoint> <byte dstEndpoint>}	Send cancel all load control events command to DUT
GetScheduledEventComma nd	GetScheduledEventCommand {<uint startTime> <int numberOfEvent> <ushort shortAddress> <byte srcEndpoint> <byte dstEndpoint>}	Send get scheduled event command to DUT
GetScheduledEventComma nd1	GetScheduledEventCommand1 {<uint startTime> <int numberOfEvent> <uint issuerEventId> <ushort shortAddress> <byte srcEndpoint> <byte dstEndpoint>}	Send get scheduled event command with issuerEventId to DUT
i) Messaging		
MsgDisplayCommand	MsgDisplayCommand {<uint msgId> <byte msgControl> <uint startTime> <ushort durationInMinute> <string msg> <byte extendedMsgControl> <ushort shortAddress> <byte srcEndpoint> <byte dstEndpoint>}	Send message display command to DUT
GetMessage	GetMessage {<ushort destination> <byte inSrcEndpoint> <byte inDstEndpoint>}	Send get last message command to DUT
ConfirmMessage	ConfirmMessage {<int endpoint>}	Confirm received message in the test harness

Table 2-14. Custom Script Command List

Command	Syntax	Description
MsgCancelCommand	MsgCancelCommand {<uint msgId> <byte msgControl> <ushort shortAddress> <byte srcEndpoint> <byte dstEndpoint>}	Send cancel message command to DUT
ClearMessage	ClearMessage {<int endpoint>}	Clear messages from the test harness
MsgEnhancedConfirmation	MsgEnhancedConfirmation {<uint msgId> <uint confirmationTime> <byte msgConfirmationControl> <string msgResp> <ushort shortAddress> <byte srcEndpoint> <byte dstEndpoint>}	Send message confirmation command with enhancement to DUT
DispProtectedMsgCommand	DispProtectedMsgCommand {<uint msgId> <byte msgControl> <uint startTime> <ushort durationInMinute> <string msg> <byte extendedMsgControl> <ushort shortAddress> <byte srcEndpoint> <byte dstEndpoint>}	Display protected message command to DUT
CancelAllMessages	CancelAllMessages {<uint implementationDate> <ushort shortAddress> <byte srcEndpoint> <byte dstEndpoint>}	Send cancel all messages command to DUT
GetMessageCancellation	GetMessageCancellation {<uint earliestImplementationDate> <ushort shortAddress> <byte srcEndpoint> <byte dstEndpoint>}	Send get message cancellation command to DUT
j) Key Establishment		
TestHarnessAutoRegistration	TestHarnessAutoRegistration {<bool status>}	Set key establishment process with joining in the test harness
KeyEstablishmentSelectSuite	KeyEstablishmentSelectSuite {<ushort suite>}	Set key establishment suite value in the test harness
InitiateKeyEstablishment	InitiateKeyEstablishment {<ushort nodeId> <byte endpoint>}	Start key establishment process from the test harness
ConfigureCertCorruptByteIndex	ConfigureCertCorruptByteIndex {<byte byteIndex>}	Configure corrupted byte in the test harness certificate
TestHarnessKeyEstablishmentSetMode	TestHarnessKeyEstablishmentSetMode {<int mode>}	Set key establishment status in the test harness
THKeyEstablishmentSetModeDelayCbke	THKeyEstablishmentSetModeDelayCbke {<ushort delay> <ushort advertisedDelay>}	Change the advertised delays by the local device for CBKE
THConfigureCertLength	THConfigureCertLength {<int length>}	Configure mangles length of the test harness certification

Table 2-14. Custom Script Command List

Command	Syntax	Description
PartnerLinkKeyExchange	PartnerLinkKeyExchange {<ushort remoteNodeId> <int endPoint>}	Perform partner link key exchange process with non-TC device
SetOutofSequenceMode	SetOutofSequenceMode {<uint commandId>}	Set out of sequence of key establishment commands
ConfigureCertSubject	ConfigureCertSubject {<byte[] eui64>}	Configure subject in the test harness certificate
ConfigureCertChangeByte	ConfigureCertChangeByte {<byte byteIndex> <byte newByte>}	Change the byte in the test harness certificate
KeyEstablishmentKeyMangleCommand	KeyEstablishmentKeyMangleCommand {<int keyLength>}	Mangles the length of the ephemeral key
ConfigureIssuer	ConfigureIssuer {<byte[] issuer>}	Configure issuer in the test harness certificate
SendCommandInterPan	SendCommandInterPan {<uint startTime> <int numberOfEvent> <ushort panId> <ushort profileId> <ushort options> <byte[] macAddress>}	Send inter PAN command to DUT with the get scheduled event command
EzspRequestKey	EzspRequestKey {<byte[] partnerId>}	Send APS request key command to DUT
SetPartnerLinkkeyExchangeFlag	SetPartnerLinkkeyExchangeFlag {<bool value>}	Set partner link key exchange flag in the test harness, whether to perform partner link key exchange process or not
UpdateKeyState	UpdateKeyState {<int index> <bool keyIsAuthorized>}	Update link key state in the Test harness
UpdateTrustCenterLinkKey	UpdateTrustCenterLinkKey {<bool keyIsAuthorized>}	Update TrustCenter link key state in the test harness
SetApsFrameCounterFlag	SetApsFrameCounterFlag	Reset APS frame counter value to zero after key establishment in the test harness
SetApsAdvFrameCounterFlag	SetApsAdvFrameCounterFlag	Reset APS frame counter value to 1000 after key establishment in the test harness
CbkeAllowPartner	CbkeAllowPartner {<bool allowCbke>}	Set flag in the test harness for it is allowing to perform CBKE process with non-TC device

Table 2-14. Custom Script Command List

Command	Syntax	Description
k) Trust Center swap-out		
TCExportCommand	TCExportCommand	Export test harness network details in to the file
TCImportCommand	TCImportCommand	Import other trust center network details file into test harness
BroadcastNetworkKeyUpdate	BroadcastNetworkKeyUpdate	Broadcast NetworkKeyUpdate command on the Network
l) Sub-GHz		
StartMultiPhy	StartMultiPhy {<int page> <int channel> <int power>}	Start the sub-GHz radio and form the network on sub-GHz network in the test harness
StopMultiPhy	StopMultiPhy	Stop sub-GHz radio in the test harness
SetBandMode	SetBandMode {<uint mode>}	Set channel scanning mode for joining
GetSuspendZCLMessageStatus	GetSuspendZCLMessageStatus {<ushort nodeId> <int endpoint>}	Send get suspend ZCL messages status command status to DUT
IgnoreSuspendZclMessages	IgnoreSuspendZclMessages {<bool status>}	Ignore suspend ZCL messages command in the test harness
SendSuspendZclMessages Command	SendSuspendZclMessagesCommand {<ushort nodeId> <int srcEndpoint> <int destEndpoint> <int period>}	Send suspend ZCL messages command to DUT
SendUnsolicitedEnhancedUpdateNotify	SendUnsolicitedEnhancedUpdateNotify {<ushort nodeId> <int channelPage> <int channel> <ushort macTxUcastTotal> <ushort macTxUcastFailures> <ushort macTxUcastRetries> <int period>}	Send Mgmt_NWK_Unsolicited_Enhanced_Update_Notify command to DUT
SetApsAckBit	SetApsAckBit {<int value>}	Set APS layer frame control acknowledgement request bit
DoNotSuspendClient	DoNotSuspendClient {<bool value>}	Ignore client suspension after sending suspend ZCL messages command to DUT

Table 2-14. Custom Script Command List

Command	Syntax	Description
Home Automation Profile Clusters Command a) Global		
ConfigureReporting	ConfigureReporting {<ushort clusterId> <ushort attributeId> <byte attributeType> <ushort Min> <ushort Max> <byte[] message> <ushort shortAddress> <byte srcEndpoint> <byte dstEndpoint>}	Send configure reporting command to DUT
DiscoverCommandReveived	DiscoverCommandReveived {<ushort clusterId> <ushort startCmdIdentifier> <int maxCmdIdentifier> <bool clientToserver> <ushort shortAddress> <byte srcEndpoint> <byte dstEndpoint>}	Send discover commands received command to DUT
DiscoverCommandGenerated	DiscoverCommandGenerated {<ushort clusterId> <ushort startCmdIdentifier> <int maxCmdIdentifier> <bool clientToserver> <ushort shortAddress> <byte srcEndpoint> <byte dstEndpoint>}	Send discover commands generated command to DUT
DiscoverAttributesExtended	DiscoverAttributesExtended {<ushort clusterId> <ushort startCmdIdentifier> <int maxCmdIdentifier> <bool clientToserver> <ushort shortAddress> <byte srcEndpoint> <byte dstEndpoint>}	Send discover attributes extended command to DUT
ReadReportingConfiguration	ReadReportingConfiguration {ushort <clusterId> <byte direction> <ushort attributeId> <ushort shortAddress> <byte srcEndpoint> <byte dstEndpoint>}	Send read reporting configuration command to DUT
ZclGlobalDiscover	ZclGlobalDiscover {<ushort clusterId> <ushort attributeId> <byte maxToReport> <ushort shortAddress> <byte srcEndpoint> <byte dstEndpoint>}	Send discover attributes command to DUT
b) Basic		
ResetToFactoryDefault	ResetToFactoryDefault {<ushort shortAddress> <byte srcEndpoint> <byte dstEndpoint>}	Send reset to factory default command to DUT
c) Identify		
IdentifyId	IdentifyId {<ushort identifyTime> <ushort shortAddress> <byte srcEndpoint> <byte dstEndpoint>}	Send identify command to DUT
IdentifyQuery	IdentifyQuery {<ushort shortAddress> <byte srcEndpoint> <byte dstEndpoint>}	Send identify query command to DUT
d) Groups		
RemoveAllGroups	RemoveAllGroups {<ushort shortAddress> <byte srcEndpoint> <byte dstEndpoint>}	Send remove all groups command to DUT
AddGroup	AddGroup {<ushort groupId> <string groupName> <ushort shortAddress> <byte srcEndpoint> <byte dstEndpoint>}	Send add group command to DUT

Table 2-14. Custom Script Command List

Command	Syntax	Description
GetGroupMembership	GetGroupMembership {<byte groupCount> <ushort[] group> <ushort shortAddress> <byte srcEndpoint> <byte dstEndpoint>}	Send get group membership command to DUT
ViewGroup	ViewGroup {<ushort groupId> <ushort shortAddress> <byte srcEndpoint> <byte dstEndpoint>}	Send view group command to DUT
RemoveGroup	RemoveGroup {<ushort groupId> <ushort shortAddress> <byte srcEndpoint> <byte dstEndpoint>}	Send remove group command to DUT
GroupsAddIfId	GroupsAddIfId {<ushort groupId> <string groupName> <ushort shortAddress> <byte srcEndpoint> <byte dstEndpoint>}	Send add group if identifying command to DUT
e) Scenes		
StoreScene	StoreScene {<ushort groupId> <byte sceneId> <ushort shortAddress> <byte srcEndpoint> <byte dstEndpoint>}	Send store scene command to DUT
RecallScene	RecallScene {<ushort groupId> <byte sceneId> <ushort shortAddress> <byte srcEndpoint> <byte dstEndpoint>}	Send recall scene command to DUT
ViewScene	ViewScene {<ushort groupId> <byte sceneId> <ushort shortAddress> <byte srcEndpoint> <byte dstEndpoint>}	Send view scene command to DUT
GetSceneMembership	GetSceneMembership {<ushort groupId> <ushort shortAddress> <byte srcEndpoint> <byte dstEndpoint>}	Get scene membership command to DUT
ScenesRemoveAll	ScenesRemoveAll {<ushort groupId> <ushort shortAddress> <byte srcEndpoint> <byte dstEndpoint>}	Send remove all scenes command to DUT
ScenesAdd	ScenesAdd {<ushort groupId> <byte sceneId> <ushort transitionTime> <string sceneName> <ushort shortAddress> <byte srcEndpoint> <byte dstEndpoint>}	Send add scene command to DUT
ScenesRemove	ScenesRemove {<ushort groupId> <byte sceneId> <ushort shortAddress> <byte srcEndpoint> <byte dstEndpoint>}	Send remove scene command to DUT
ScenesEnhancedAdd	ScenesEnhancedAdd {<ushort groupId> <byte sceneId> <ushort transitionTime> <string sceneName> <ushort shortAddress> <byte srcEndpoint> <byte dstEndpoint>}	Send enhanced add scene command to DUT
SceneEnhancedView	SceneEnhancedView {<ushort groupId> <byte sceneId> <ushort shortAddress> <byte srcEndpoint> <byte dstEndpoint>}	Send enhanced view scene command to DUT
ScenesCopy	ScenesCopy {<byte mode> <ushort groupIdFrom> <byte scenesIdFrom> <ushort groupIdTo> <byte scenesIdTo> <ushort shortAddress> <byte srcEndpoint> <byte dstEndpoint>}	Send copy scene command to DUT

Table 2-14. Custom Script Command List

Command	Syntax	Description
f) OnOff		
On	On {<ushort shortAddress> <byte srcEndpoint> <byte dstEndpoint>}	Send the ON command to DUT
Off	Off {<ushort shortAddress> <byte srcEndpoint> <byte dstEndpoint>}	Send the OFF command to DUT
Toggle	Toggle {<ushort shortAddress> <byte srcEndpoint> <byte dstEndpoint>}	Send the toggle command to DUT
OffWithEffect	OffWithEffect {<byte effectId> <byte effectVariant> <ushort shortAddress> <byte srcEndpoint> <byte dstEndpoint>}	Send the off with effect command to DUT
OnWithRecallGlobalScene	OnWithRecallGlobalScene {<ushort shortAddress> <byte srcEndpoint> <byte dstEndpoint>}	Send the on with recall global scene command to DUT
OnWithTimedOff	OnWithTimedOff {<byte onOffControl> <ushort onTime> <ushort offWaitTime> <ushort shortAddress> <byte srcEndpoint> <byte dstEndpoint>}	Send the on with timed off command to DUT
g) Level Control		
MoveToLevel	MoveToLevel {<byte level> <ushort transitionTime> <ushort shortAddress> <byte srcEndpoint> <byte dstEndpoint>}	Send the move to level command to DUT
OnOffMoveToLevel	OnOffMoveToLevel {<byte level> <ushort transitionTime> <ushort shortAddress> <byte srcEndpoint> <byte dstEndpoint>}	Send the move to level (with On/Off) command to DUT
Move	Move {<byte moveMode> <byte rate> <ushort shortAddress> <byte srcEndpoint> <byte dstEndpoint>}	Send the move command to DUT
OnOffMove	OnOffMove {<byte moveMode> <byte rate> <ushort shortAddress> <byte srcEndpoint> <byte dstEndpoint>}	Send the move (with On/Off) command to DUT
Stop	Stop {<ushort shortAddress> <byte srcEndpoint> <byte dstEndpoint>}	Send the stop command to DUT
OnOffStop	OnOffStop {<ushort shortAddress> <byte srcEndpoint> <byte dstEndpoint>}	Send the stop (with On/Off) command to DUT
Step	Step {<byte stepMode> <byte stepSize> <ushort transitionTime> <ushort shortAddress> <byte srcEndpoint> <byte dstEndpoint>}	Send the step command to DUT
OnOffStep	OnOffStep {<byte stepMode> <byte stepSize> <ushort transitionTime> <ushort shortAddress> <byte srcEndpoint> <byte dstEndpoint>}	Send the step (with On/Off) command to DUT

Table 2-14. Custom Script Command List

Command	Syntax	Description
h) Thermostat		
SetPointRaiseLower	SetPointRaiseLower {<byte mode> <byte amount> <ushort shortAddress> <byte srcEndpoint> <byte dstEndpoint>}	Send the set point raise/ lower command to DUT
SetWeeklySchedule	SetWeeklySchedule {<byte transitionsForSequence> <byte dayOfWeekForSequence> <byte modeForSequence> <byte[] payload> <ushort shortAddress> <byte srcEndpoint> <byte dstEndpoint>}	Send the set weekly schedule command to DUT
GetWeeklySchedule	GetWeeklySchedule {<byte daysToReturn> <byte modeToReturn> <ushort shortAddress> <byte srcEndpoint> <byte dstEndpoint>}	Send the get weekly schedule command to DUT
GetRelayStatusLog	GetRelayStatusLog {<ushort shortAddress> <byte srcEndpoint> <byte dstEndpoint>}	Send the get relay status log command to DUT
ClearWeeklySchedule	ClearWeeklySchedule {<ushort shortAddress> <byte srcEndpoint> <byte dstEndpoint>}	Send the clear weekly schedule command to DUT
i) Electrical Measurement		
GetProfileInfo	GetProfileInfo {<ushort shortAddress> <byte srcEndpoint> <byte dstEndpoint>}	Send the get profile info command to DUT
GetMeasurementProfile	GetMeasurementProfile {<ushort attributeId> <uint startTime> <byte numberOfIntervals> <ushort shortAddress> <byte srcEndpoint> <byte dstEndpoint>}	Send the get measurement profile command to DUT
j) Appliance Events and Alerts		
ApplianceGetAlerts	ApplianceGetAlerts {<ushort shortAddress> <byte srcEndpoint> <byte dstEndpoint>}	Send the get alerts command to DUT
ApplianceAlertsNotification	ApplianceAlertsNotification {<byte alertsCount> <byte alertStructures> <ushort alertStructuresLen> <ushort shortAddress> <byte srcEndpoint> <byte dstEndpoint>}	Send the alerts notification command to DUT
ApplianceEventsNotification	ApplianceEventsNotification {<byte eventHeader> <byte eventIdentification> <ushort shortAddress> <byte srcEndpoint> <byte dstEndpoint>}	Send the event notification command to DUT
k) Appliance Statistics		
ZclLogQueueRequest	ZclLogQueueRequest {<ushort shortAddress> <byte srcEndpoint> <byte dstEndpoint>}	Send the log queue request command to DUT
l) Poll Control		
SetModeValue	SetModeValue {<byte mode>}	Set the fast polling mode in the test harness

Table 2-14. Custom Script Command List

Command	Syntax	Description
SetTimeoutValue	SetTimeoutValue {<ushort timeout>}	Set the timeout value in the test harness
SetResponseValue	SetResponseValue {<byte mode>}	Set the fast polling mode in the test harness for the response
FastPollStop	FastPollStop {<ushort shortAddress> <byte srcEndpoint> <byte dstEndpoint>}	Send the fast poll stop command to DUT
SetLongPollInterval	SetLongPollInterval {<uint newInterval> <ushort shortAddress> <byte srcEndpoint> <byte dstEndpoint>}	Send the set long poll interval command to DUT
SetShortPollInterval	SetShortPollInterval {<ushort newInterval> <ushort shortAddress> <byte srcEndpoint> <byte dstEndpoint>}	Send set short poll interval command to DUT
m) Door Lock		
DoorLock	DoorLock {<byte[] PIN> <ushort shortAddress> <byte srcEndpoint> <byte dstEndpoint>}	Send the lock door command to DUT
DoorUnLock	DoorUnLock {<byte[] PIN> <ushort shortAddress> <byte srcEndpoint> <byte dstEndpoint>}	Send the unlock door command to DUT
n) IAS Zone		
Enroll	Enroll {<ushort zoneType> <ushort mfCode> <ushort shortAddress> <byte srcEndpoint> <byte dstEndpoint>}	Send the zone enroll request command to DUT
StatusChange	StatusChange {<ushort zoneStatus> <byte extStatus> <byte zoneId> <ushort delay> <ushort shortAddress> <byte srcEndpoint> <byte dstEndpoint>}	Send the zone status change notification command to DUT
o) Color Control		
MoveToColor	MoveToColor {<ushort colorX> <ushort colorY> <ushort transitionTime> <byte optionsMask> <byte optionsoOverride> <ushort shortAddress> <byte srcEndpoint> <byte dstEndpoint>}	Send the move to color command to DUT
ColorControlMoveToSat	ColorControlMoveToSat {<byte saturation> <ushort transitionTime> <ushort shortAddress> <byte srcEndpoint> <byte dstEndpoint>}	Send the move to saturation command to DUT
ColorControlMoveToHue	ColorControlMoveToHue {<byte hue> <byte direction> <ushort transitionTime> <ushort shortAddress> <byte srcEndpoint> <byte dstEndpoint>}	Send the move to hue command to DUT
ColorControlEnhancedMoveToHue	ColorControlEnhancedMoveToHue {<ushort enhancedHue> <byte direction> <ushort transitionTime> <ushort shortAddress> <byte srcEndpoint> <byte dstEndpoint>}	Send the enhanced move to hue command to DUT

Table 2-14. Custom Script Command List

Command	Syntax	Description
ColorControlMoveHue	ColorControlMoveHue {<byte moveMode> <byte rate> <ushort shortAddress> <byte srcEndpoint> <byte dstEndpoint>}	Send the move hue command to DUT
ColorControlEMoveHue	ColorControlEMoveHue {<byte moveMode> <ushort rate> <ushort shortAddress> <byte srcEndpoint> <byte dstEndpoint>}	Send the enhanced move hue command to DUT
ColorControlStepHue	ColorControlStepHue {<byte stepMode> <byte stepSize> <byte transitionTime> <ushort shortAddress> <byte srcEndpoint> <byte dstEndpoint>}	Send the step hue command to DUT
ColorControlEStepHue	ColorControlEStepHue {<byte stepMode> <ushort stepSize> <ushort transitionTime> <ushort shortAddress> <byte srcEndpoint> <byte dstEndpoint>}	Send the enhanced step hue command to DUT
ControlMoveSat	ControlMoveSat {<byte mode> <byte rate> <ushort shortAddress> <byte srcEndpoint> <byte dstEndpoint>}	Send the move saturation command to DUT
ColorControlStepSat	ColorControlStepSat {<byte stepMode> <byte stepSize> <byte transitionTime> <ushort shortAddress> <byte srcEndpoint> <byte dstEndpoint>}	Send the step saturation command to DUT
ColorControlMoveToHueAndSat	ColorControlMoveToHueAndSat {<byte hue> <byte saturation> <ushort transitionTime> <ushort shortAddress> <byte srcEndpoint> <byte dstEndpoint>}	Send the move to hue and saturation command to DUT
ColorControlEMoveToHueAndSat	ColorControlEMoveToHueAndSat {<ushort enhancedHue> <byte saturation> <ushort transitionTime> <ushort shortAddress> <byte srcEndpoint> <byte dstEndpoint>}	Send the enhanced move to hue and saturation command to DUT
ColorControlMoveToColorNew	ColorControlMoveToColorNew {<ushort colorX> <ushort colorY> <ushort transitionTime> <ushort shortAddress> <byte srcEndpoint> <byte dstEndpoint>}	Send the move to color command to DUT
ColorControlMoveColor	ColorControlMoveColor {<ushort rateX> <ushort rateY> <ushort shortAddress> <byte srcEndpoint> <byte dstEndpoint>}	Send the move color command to DUT
ColorControlMoveColorTemp	ColorControlMoveColorTemp {<byte moveMode> <ushort rate> <ushort colorTemperatureMin> <ushort colorTemperatureMax> <ushort shortAddress> <byte srcEndpoint> <byte dstEndpoint>}	Send the move color temperature command to DUT
ColorControlStepColor	ColorControlStepColor {<ushort stepX> <ushort stepY> <ushort transitionTime> <ushort shortAddress> <byte srcEndpoint> <byte dstEndpoint>}	Send the step color command to DUT

Table 2-14. Custom Script Command List		
Command	**Syntax**	**Description**
ColorControlStepColorTemp	ColorControlStepColorTemp {<byte stepMode> <ushort stepSize> <ushort transitionTime> <ushort colorTemperatureMin> <ushort colorTemperatureMax> <ushort shortAddress> <byte srcEndpoint> <byte dstEndpoint>}	Send the step color temperature command to DUT
ColorControlMoveToColorTemperature	ColorControlMoveToColorTemperature {<ushort colorTemperature> <ushort transitionTime> <ushort shortAddress> <byte srcEndpoint> <byte dstEndpoint>}	Send the move to color temperature command to DUT
ColorControlColorLoopSet	ColorControlColorLoopSet {<byte updateFlag> <byte action> <byte direction> <ushort time> <ushort startHue> <ushort shortAddress> <byte srcEndpoint> <byte dstEndpoint>}	Send the color loop set command to DUT
ColorControlStopMoveStep	ColorControlStopMoveStep {<ushort shortAddress> <byte srcEndpoint> <byte dstEndpoint>}	Send the stop move step command to DUT
Zigbee 3.0 Profile Commands **a) Global**		
UnbindUnicastRequest	UnbindUnicastRequest {<ushort target> <byte[] destinationEui64> <byte destinationEndpoint> <ushort clusterId> <byte[] sourceEui64> <byte sourceEndpoint>}	Unicast ZDO unbind command request to the DUT
UnbindGroupRequest	UnbindGroupRequest {<ushort target> <byte[] destinationEui64> <ushort clusterId> <byte sourceEndpoint> <ushort groupAddress>}	ZDO unbind group request to the DUT
UnbindUnicastRequestTHSourceEUI	UnbindUnicastRequestTHSourceEUI {<ushort target> <byte[] destinationEui64> <byte destinationEndpoint> <ushort clusterId> <byte sourceEndpoint>}	Unicast ZDO unbind command request without source EUI to the DUT
BindRequestSourceEUI	BindRequestSourceEUI {<ushort target> <byte sourceEndpoint> <byte destinationEndpoint> <ushort clusterId> <byte[] destinationEUI> <byte[] sourceEui>}	Send ZDO bind command request to the DUT
Z3ReadAttribute	Z3ReadAttribute {<ushort clusterId> <ushort attributeId> <ushort destination> <byte srcEndpoint> <byte dstEndpoint> <ushort profileId>}	Send the read attribute request with profile Id to the DUT
Z3ZdoBindGroup	Z3ZdoBindGroup {<ushort shortAddress> <byte srcEndpoint> <ushort groupId> <ushort cluster> <byte[] dstMAC>}	Send the ZDO bind group command request to the DUT
b) Zigbee 3.0 Network		
NetworkCreatorStart	NetworkCreatorStart {<bool centralizedSecurity>}	Create a new network in the test harness

Table 2-14. Custom Script Command List

Command	Syntax	Description
NetworkCreatorForm	NetworkCreatorForm {<bool centralizedSecurity> <ushort panId> <int radioTxPower> <int channel>}	Create a new network with PAN ID and channel in the test harness
NetworkCreatorChannelMask	NetworkCreatorChannelMask {<int action> <int mask> <uint channelOrNewMask>}	Set channel mask in the test harness
NetworkSteeringStart	NetworkSteeringStart {<uint steeringOptionsMask>}	Start network steering process in the test harness
OpenNetwork	OpenNetwork	Open network for the centralize network in the test harness
NetworkSteeringChannelAdd	NetworkSteeringChannelAdd {<byte maskToAddTo> <int channelToAdd>}	Add channel in the mask for network steering in the test harness
NetworkSteeringChannelSubtract	NetworkSteeringChannelSubtract {<byte maskToAddTo> <int channelToSubtract>}	Subtract channel in the mask for network steering in the test harness
NetworkSteeringChannelSubtract	NetworkSteeringChannelSubtract {<byte maskToAddTo> <int channelToSubtract>}	Set channel in the mask for network steering in the test harness
PermitJoiningRequest	PermitJoiningRequest {<ushort nodeId> <uint duration>}	Send the permit join request to the DUT
SetJoiningLinkKey	SetJoiningLinkKey {<byte[] macAddress> <byte[] linkKey>}	Set link key for the joining
ClearJoiningLinkKey	ClearJoiningLinkKey	Clear set link key from the test harness
FindAndBindTargetStart	FindAndBindTargetStart {<byte endpoint>}	Start a find and bind process (as Target) in the test harness
FindAndBindInitiatorStart	FindAndBindInitiatorStart {<byte endpoint>}	Start find and bind process (as initiator) in the test harness
OpenNetworkWithKey	OpenNetworkWithKey {<byte[] macAddress> <byte[] linkKey>}	Open network with specific link key in the test harness
Mgmtpermitjoin	Mgmtpermitjoin {<ushort nodeId> <ushort permitDuration> <byte options>}	Send Mgmt permit join request to the DUT
InitiateTouchLink	InitiateTouchLink	Start Touchlink procedure from the test harness

Table 2-14. Custom Script Command List

Command	Syntax	Description
c) Zigbee 3.0 ZLL Commissioning		
ScanRequestProcess	ScanRequestProcess {<byte linkInitiator> <uint optionsValue>}	Send the scan request command to the DUT
SetDeviceMode	SetDeviceMode {<byte modeValue>}	Set the device mode in the test harness
NetworkStartRequest	NetworkStartRequest {<ushort nodeId> <ushort freeAddrBegin> <ushort freeAddrEnd> <ushort freeGroupBegin> <ushort freeGroupEnd> <uint option>}	Send the network start request command to the DUT
NetworkJoinRouterRequest	NetworkJoinRouterRequest {<ushort nodeId> <ushort freeAddrBegin> <ushort freeAddrEnd> <ushort freeGroupBegin> <ushort freeGroupEnd> <uint option>}	Send the network join router request command to the DUT
NetworkJoinEndDeviceRequest	NetworkJoinEndDeviceRequest {<ushort nodeId> <ushort freeAddrBegin> <ushort freeAddrEnd> <ushort freeGroupBegin> <ushort freeGroupEnd> <uint option>}	Send the network join enddevice request command to the DUT
ResetToFactoryDefault	ResetToFactoryDefault {<uint option>}	Send the ResetToFactoryDefault command to the DUT
ConfigScanResponse	ConfigScanResponse {<uint option>}	Configure scan response in the test harness
NetworkUpdateRequest	NetworkUpdateRequest {<uint option>}	Send the network update request command to the DUT
FormZLLNetwork	FormZLLNetwork {<byte channel> <int power> <ushort panId>}	Form a new ZLL network in the test harness
Z3MgmtNwkUpdateRequest	Z3MgmtNwkUpdateRequest {<ushort scanChannel> <ushort scanDuration> <byte scanCount> <ushort destination> <uint options>}	Send the Mgmt network update request command to the DUT
Z3MgmtLeave	Z3MgmtLeave {<ushort destination> <bool removeChildren> <bool rejoin> <uint options>}	Send the Mgmt leave command to the DUT
DeviceInformationRequest	DeviceInformationRequest {<byte startIndex> <uint option>}	Send the device information request command to the DUT
StartAsRouter	StartAsRouter {<ushort panId> <uint option>}	Start test harness as router
Z3NwkRejoinRequest	Z3NwkRejoinRequest {<ushort nodeId>}	Send the rejoin request to the DUT
IdentifyRequest	IdentifyRequest {<ushort duration> <uint option>}	Send the identify request to the DUT

Table 2-14. Custom Script Command List

Command	Syntax	Description
Z3ZdoNwkAddrReq	Z3ZdoNwkAddrReq {<byte[] destEui64> <byte requestType> <byte startIndex> <ushort destShort> <uint option>}	Send the network address request to the DUT
Z3ZdoIeeeAddrReq	Z3ZdoIeeeAddrReq {<ushort nwkAddrOfInterest> <byte requestType> <byte startIndex> <ushort destination> <uint option>}	Send the IEEE address request to the DUT
Z3ZdoActiveEndpointRequest	Z3ZdoActiveEndpointRequest {<ushort destination> <ushort nwkAddrOfInterest> <uint option>}	Send the active endpoint request command to the DUT
Z3ZdoSimpleDescReq	Z3ZdoSimpleDescReq {<ushort destination> <byte endpoint> <ushort nwkAddrOfInterest> <uint option>}	Send the simple descriptor request command to the DUT
Z3ZdoMatchDescReq	Z3ZdoMatchDescReq {<ushort destination> <ushort nwkAddrOfInterest> <ushort profileId> <uint option>}	Send the match descriptor request command to the DUT
Z3NwkLeave	Z3NwkLeave {<bool rejoin> <bool request> <bool removeChildren> <ushort destinationShort> <uint option>}	Send the network leave command to the DUT
SetShortAddress	SetShortAddress {<ushort shortAddress>}	Set custom node id in the test harness
SetScanChannel	SetScanChannel {<int channel>}	Set the scan channel used by the ZLL commissioning in the test harness
ResetToFactoryNew	ResetToFactoryNew	Reset the local device to factory new
ZllGroupsIdRequest	ZllGroupsIdRequest {<ushort destinationAdd> <byte srcEndpoint> <byte dstEndpoint> <byte startIndex>}	Send the group identifiers request to the DUT
ZllEndpointListRequest	ZllEndpointListRequest {<ushort destinationAdd> <byte srcEndpoint> <byte dstEndpoint> <byte startIndex>}	Send the get endpoint list request to the DUT

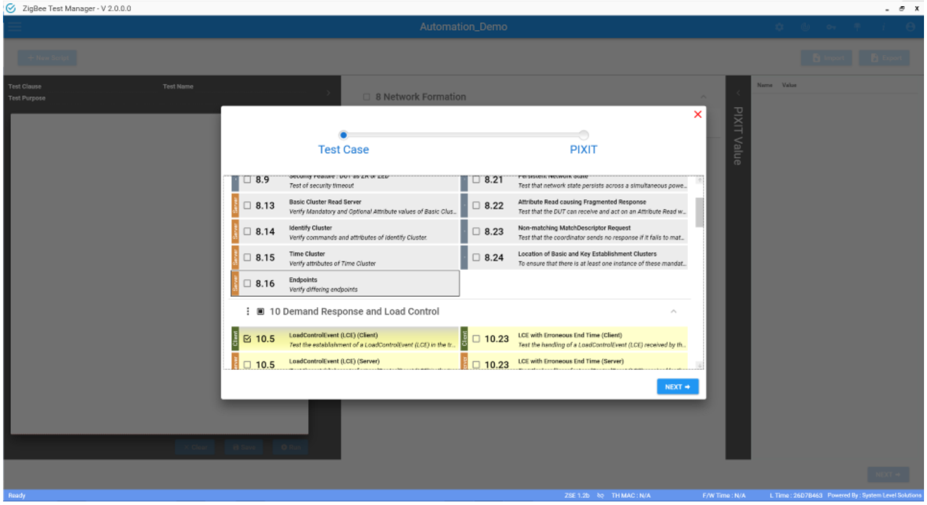
☞ For detailed description and parameters of the prefix and commands, refer “Zigbee Test Manager Custom Script API User Guide”.

After writing the script, click on Run button available on the bottom of the script editor to execute each command and validate their response. It displays the same buttons and test case status indication as explained in Table 2-7 and Table 2-8 respectively. On clicking Run button, it executes the test case(s) sequentially and displays the result in the HTML report in default browser as shown in Figure 2-20.

Export Test Case

This panel shows options to export and import pre-defined test case/s in the script and define their PIXIT value and execute them. Click on Export Export button to select the test cases. See Figure 2-35.

Figure 2-35. Export Test Cases



☐ 8.9 Security Feature : DUT as ZR or ZED
Test of security timeout

☐ 8.21 Persistent Network State
Test that network state persists across a simultaneous powe...

☐ 8.13 Basic Cluster Read Server
Verify Mandatory and Optional Attribute values of Basic Clus...

☐ 8.22 Attribute Read causing Fragmented Response
Test that the DUT can receive and act on an Attribute Read w...

☐ 8.14 Identify Cluster
Verify commands and attributes of Identify Cluster.

☐ 8.23 Non-matching MatchDescriptor Request
Test that the coordinator sends no response if it fails to mat...

☐ 8.15 Time Cluster
Verify attributes of Time Cluster

☐ 8.24 Location of Basic and Key Establishment Clusters
To ensure that there is at least one instance of these mandat...

☐ 8.16 Endpoints
Verify differing endpoints

10 Demand Response and Load Control

☑ 10.5 LoadControlEvent (LCE) (Client)
Test the establishment of a LoadControlEvent (LCE) in the tr...

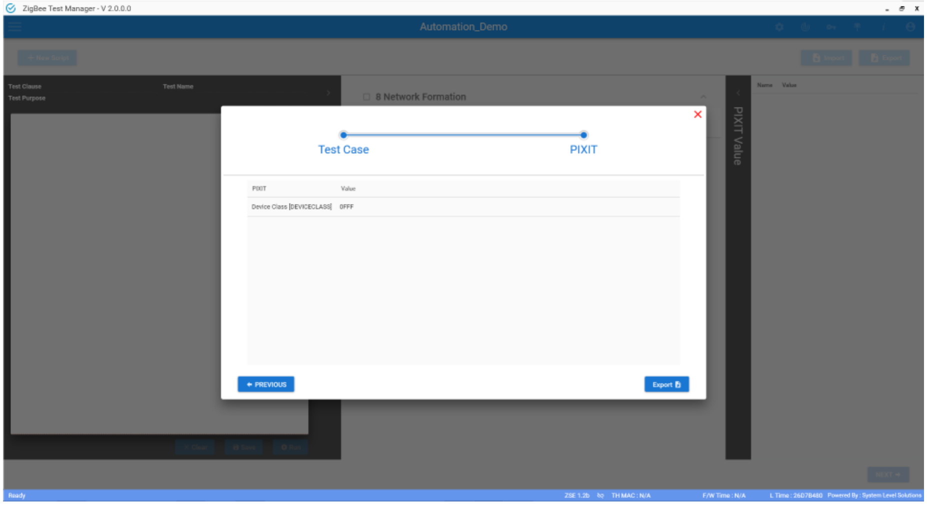
☐ 10.23 LCE with Erroneous End Time (Client)
Test the handling of a LoadControlEvent (LCE) received by th...

☐ 10.5 LoadControlEvent (LCE) (Server)

☐ 10.23 LCE with Erroneous End Time (Server)

Select and click on Next NEXT ➔ button to see available PIXIT value. It lists all PIXIT value with default values and allows to modify as per DUT configuration. Click Previous ← PREVIOUS button to modify the test case/s selection. See Figure 2-36.

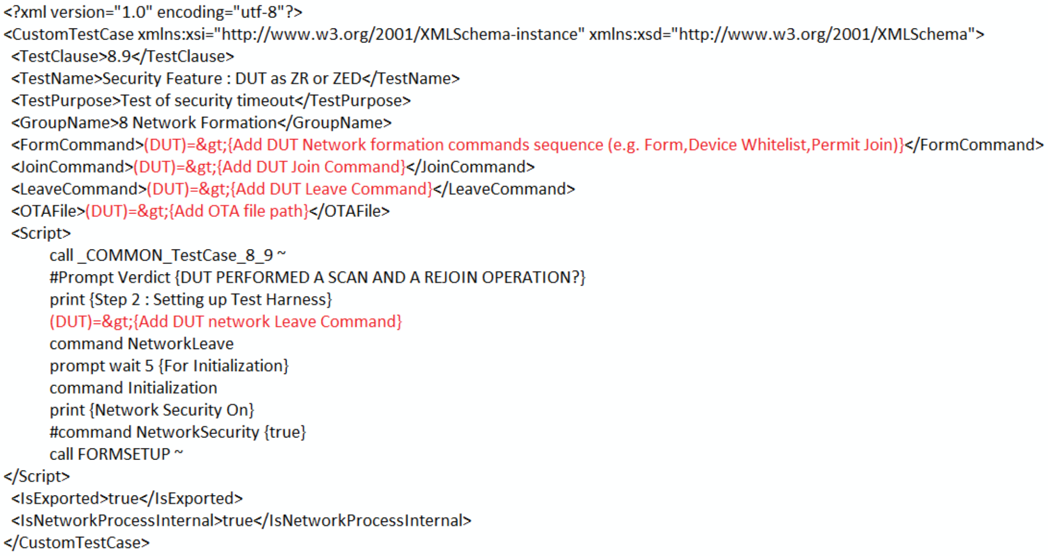
Figure 2-36. Selected Test Case PIXIT Value



PIXIT	Value
Device Class [DEVICECLASS]	0FFF

Click on Export button to export the selected test case/s in a script file. See Figure 2-37.

Figure 2-37. Exported Custom Script



```
<?xml version="1.0" encoding="utf-8"?>
<CustomTestCase xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema">
 <TestClause>8.9</TestClause>
 <TestName>Security Feature : DUT as ZR or ZED</TestName>
 <TestPurpose>Test of security timeout</TestPurpose>
 <GroupName>8 Network Formation</GroupName>
 <FormCommand>(DUT)=&gt;{Add DUT Network formation commands sequence (e.g. Form,Device Whitelist,Permit Join)}</FormCommand>
 <JoinCommand>(DUT)=&gt;{Add DUT Join Command}</JoinCommand>
 <LeaveCommand>(DUT)=&gt;{Add DUT Leave Command}</LeaveCommand>
 <OTAFile>(DUT)=&gt;{Add OTA file path}</OTAFile>
 <Script>
      call _COMMON_TestCase_8_9 ~
      #Prompt Verdict {DUT PERFORMED A SCAN AND A REJOIN OPERATION?}
      print {Step 2 : Setting up Test Harness}
      (DUT)=&gt;{Add DUT network Leave Command}
      command NetworkLeave
      prompt wait 5 {For Initialization}
      command Initialization
      print {Network Security On}
      #command NetworkSecurity {true}
      call FORMSETUP ~
</Script>
 <IsExported>true</IsExported>
 <IsNetworkProcessInternal>true</IsNetworkProcessInternal>
</CustomTestCase>
```

The script can be modified as per DUT configuration using the prefix and commands explained in Table 2-13 and Table 2-14 . Write the DUT command using “(DUT)=>” prefix. For example,

```
(DUT)=>JN,LEAVE
```

Exported file contains some Place Holder which guides the editor to write the relevant command and validate them with respected test specification. The Place Holder can be kept as it is, if there is no requirement to add/modify.

For example, “Prompt Check” command can be modify as per DUT configuration and applicability.

```
prompt check {Does DUT supports receiving unsolicited
messages}
[YES]
{
   command ClearInOutCluster {in}
   command ClearInOutCluster {out}
   command AddInOutCluster {in 0700}
   command AddInOutCluster {in 0701}
   command AddInOutCluster {in 0702}
   command AddInOutCluster {in 0703}
   print {Step 10: performing service discovery}
   command MatchDescriptorRequest {[NWK:SHORTADDRESS] 0109}
   print {Verifying Response}
   expect {Cluster identifier=8006, Status=00}
}
[NO]
{
   print {Step 9: DUT does not support receiving unsolicited
   messages.}
}
```

If DUT capable of receiving unsolicited messages then execute command accordingly, like perform only “YES” block, i.e.

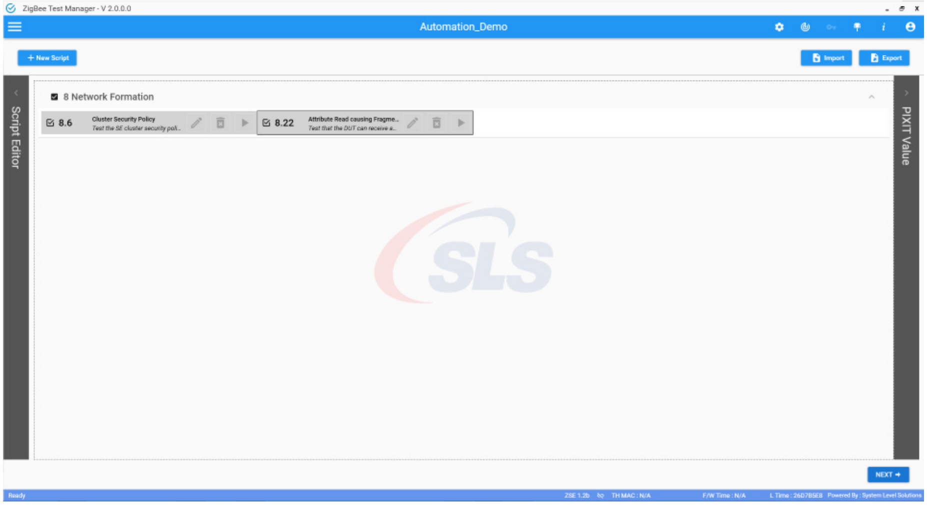
```
command ClearInOutCluster {in}
command ClearInOutCluster {out}
command AddInOutCluster {in 0700}
command AddInOutCluster {in 0701}
command AddInOutCluster {in 0702}
command AddInOutCluster {in 0703}
print {Step 10: performing service discovery}
command MatchDescriptorRequest {[NWK:SHORTADDRESS] 0109}
print {Verifying Response}
expect {Cluster identifier=8006, Status=00}
```

If DUT not capable of receiving unsolicited messages then perform only “NO” block, i.e.

```
print {Step 9: DUT does not support receiving unsolicited
messages.}
```

After customizing the script, click on Import Import button to import the script file with their test case/s. See Figure 2-38.

Figure 2-38. Import Test Cases



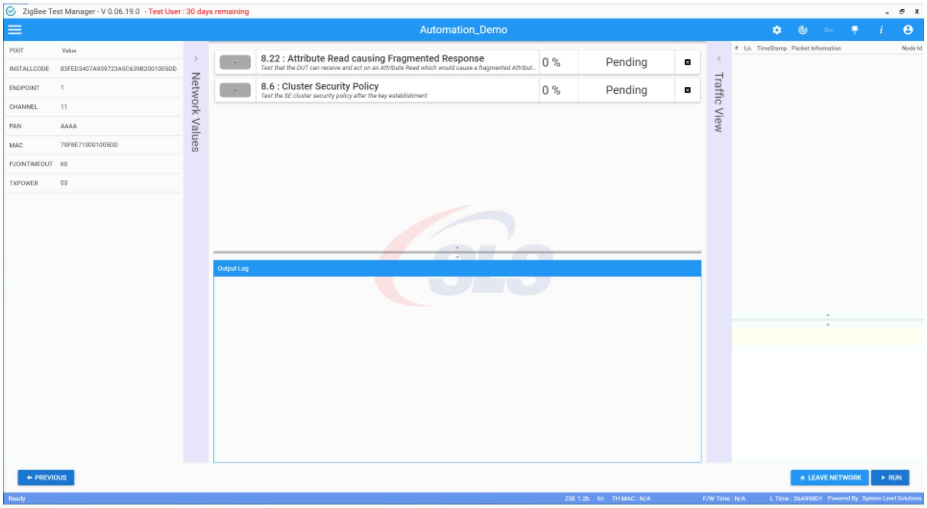
8 Network Formation

☑ 8.6 Cluster Security Policy
Test the SE cluster security poli...

☑ 8.22 Attribute Read causing Fragme...
Test that the DUT can receive a...

It displays all test cases available in script or exported. Select the test case/s to execute on available DUT and verify its operation. Click on Next NEXT ➜ button to execute them. See Figure 2-39.

Figure 2-39. Execute the Test Case



ZigBee Test Manager - V 0.06.19.0 - Test User : 30 days remaining

Automation_Demo

PIXIT	Value
INSTALLCODE	83FED3407A939723A5C639B2001005DD
ENDPOINT	1
CHANNEL	11
PAN	AAAA
MAC	70F8E71000100SDD
PJOINTIMEOUT	60
TXPOWER	03

Test case	Progress	Status
8.22 : Attribute Read causing Fragmented Response *Test that the DUT can receive and act on an Attribute Read which would cause a fragmented Attribut...*	0 %	Pending
8.6 : Cluster Security Policy *Test the SE cluster security policy after the key establishment*	0 %	Pending

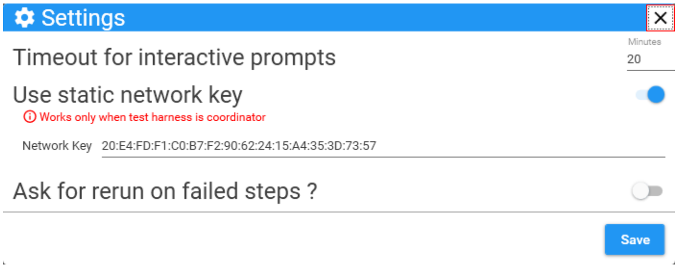
Ready ZSE 1.2b TH MAC : N/A F/W Time : N/A L Time : 26A998D1 Powered By : System Level Solutions

It displays the same buttons and test case status indication as explained in Table 2-7 and Table 2-8 respectively. On clicking Run button, it executes the test case(s) sequentially and displays the result in the HTML report in default browser as shown in Figure 2-20.

Settings

Settings window allows to do the settings for executing the test cases. Click on Settings ⚙ icon available on the top of the window. Figure 2-40. shows the settings window.

Figure 2-40. Settings Window



Settings

Timeout for interactive prompts — Minutes: 20

Use static network key — ●

ⓘ Works only when test harness is coordinator

Network Key 20:E4:FD:F1:C0:B7:F2:90:62:24:15:A4:35:3D:73:57

Ask for rerun on failed steps ? — ○

There are following settings available.

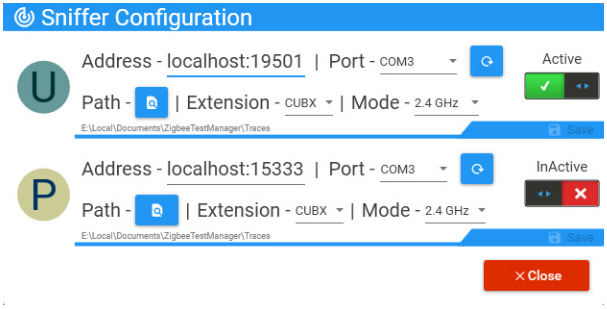
- Timeout for interactive prompts:
  This sets the time-out for prompt screens. It ensures that prompt closes after defined time-out minutes. While closing prompt after time-out it takes negative value of active prompt.
- Use static network key:
  To use the static network key value, enable this option. By default, this setting is disabled. On enabling the this setting, it display the Network Key field to be filled. Enter the 16 byte key.
- Ask for rerun on failed steps:
  To rerun the test step if it gets failed, enable this option. It will ask always on failure of the test steps. By default this setting is enabled.

After setting up the values, click on Save Save button to save and close the window.

Sniffer Configuration

Sniffer Configuration allows to choose the program to log the network transaction while testing the DUT. The tools support Ubiqua and SLS owned, packet inspector. Click on the Sniffer Configuration icon and it will displays the window as shown in Figure 2-41.

Figure 2-41. Sniffer Configuration Window



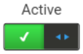
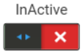
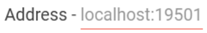
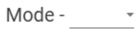
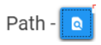
Sniffer Configuration

Address - localhost:19501 | Port - COM3 | Active ✓
Path - | Extension - CUBX | Mode - 2.4 GHz
E:\Local\Documents\ZigbeeTestManager\Traces

Address - localhost:15333 | Port - COM3 | InActive ✕
Path - | Extension - CUBX | Mode - 2.4 GHz
E:\Local\Documents\ZigbeeTestManager\Traces

Table 2-15 describes the button and fields displayed in sniffer configuration window.

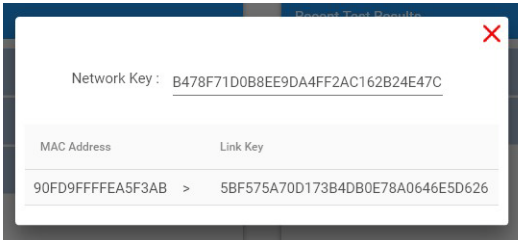
Table 2-15. Sniffer Configuration Window Options

Button/Field	Name	Description
 Active	Active	Indicates the selected sniffer tool is enabled
 InActive	Inactive	Indicates the selected sniffer tool is disabled
 Address - localhost:19501	Address	Allows to enter the address of the connected sniffer tool
Extension - CUBX	Extension	Allows to select the extension type of the connected sniffer tool
Port - COM1	Port	Allows to select the port of the connected sniffer tool
 Mode -	Mode	Allows to select the mode of communication for connected sniffer tool
 Path -	Path	Allows to select the path for saving the logs captured by the connected sniffer tool
	Refresh	Refresh the connection of connected sniffer tool
Close	Close	Closes the sniffer configuration window

View Key/s

The view key allows to view the network key and link key generated after DUT connection with network for testing. Click on view key 🔑 icon to view the keys. See Figure 2-42.

Figure 2-42. Network Key Window



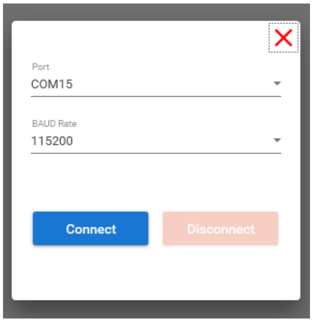
Network Key : B478F71D0B8EE9DA4FF2AC162B24E47C

MAC Address		Link Key
90FD9FFFFEA5F3AB	>	5BF575A70D173B4DB0E78A0646E5D626

Test Harness Connection

Test harness connection allows to connect with test harness over serial port from the application. Click on test harness connection icon to set the connection details. See Figure 2-43.

Figure 2-43. Test Harness Connection Window



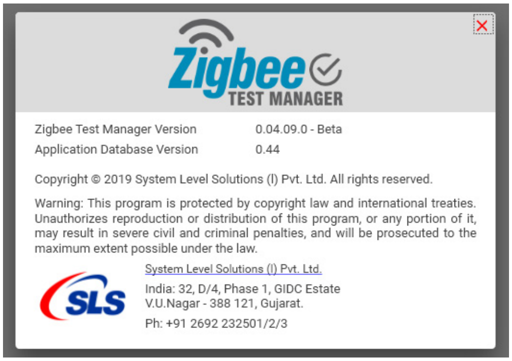
Port: COM15

BAUD Rate: 115200

Information

Information provides the details about the Zigbee test manager tools release details and the it’s developer information. Click on information icon to see the details of the software. See Figure 2-44.

Figure 2-44. Information Window



Zigbee TEST MANAGER

Zigbee Test Manager Version 0.04.09.0 - Beta

Application Database Version 0.44

Copyright © 2019 System Level Solutions (I) Pvt. Ltd. All rights reserved.

Warning: This program is protected by copyright law and international treaties. Unauthorizes reproduction or distribution of this program, or any portion of it, may result in severe civil and criminal penalties, and will be prosecuted to the maximum extent possible under the law.

System Level Solutions (I) Pvt. Ltd.
India: 32, D/4, Phase 1, GIDC Estate
V.U.Nagar - 388 121, Gujarat.
Ph: +91 2692 232501/2/3

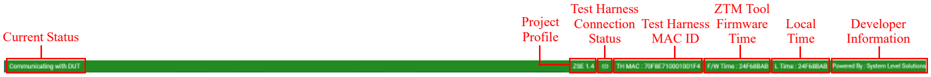
User Profile

User profile displays all the information about the user such as name, address, company name, and contact details. Click on user profile 👤 icon to set the connection details.

Status Bar

Status bar provides the information such as local time, F/W time, current status, project profile, test harness connection status, MAC ID and developer details. See Figure 2-45.

Figure 2-45. Status Bar



Communicating with DUT

ZSE 1.4

TH MAC : 70F8E710001001F4

F/W Time : 24F68BAB

L Time : 24F68BAB

Powered By : System Level Solutions