HomeKit Certification Test Cases

Release R11.1

For use with HomeKit Accessory Protocol Specification versions

Developer

Table of Contents

Chapter 1

Protocol Test Cases

1.1 Functional

TCF001: Accessory must support a MFi authentication mechanism.

TCF002: Accessory must contain a method to reset the accessory to factory settings (e.g., button or UI element).

TCF003: When submitting materials for Accessory Compliance Verification, the Licensee must provide instructions for performing a firmware update on the Accessory Compliance Questionnaire. Licensee must also provide samples that can receive a firmware update and meet all other requirements specified by the Packing List.

TCF004: Verify accessory information (such as accessory name, advertisement information, manufacturer name, model name, characteristic values, etc.) does not contain any Apple trademarks. Refer to the Apple Trademark List (https://www.apple.com/legal/intellectu property/trademark/appletmlist.html) for more information.

TCF005: Accessory setup code must appear inside accessory's product packaging. If space allows, accessory setup code should be printed on accessory.

TCF006: Verify setup code label meets the requirements as indicated in the Works with Apple HomeKit Identity Guidelines document available in the MFi Portal.

TCF007: QR code or NFC code must be able to be scanned and successfully added to the home.

TCF008: Setup code must be non-trivial, with 8 digits.

TCF012: Accessory must reject pair-setup from a new controller when it is already paired.

TCF013: Accessory should not send notifications when the characteristic's value has not changed.

TCF015: The accessory must always successfully unregister event notifications on every characteristic when a controller deregisters for notifications after having subscribed multiple times.

TCF020: If the accessory has services that include "target state" and "current state" characteristics, the current state characteristic value must be updated when the target state is reached.

TCF022: For accessories that support a service that includes the "on" characteristic. Verify that the correct state is reflected on the controller after power cycling the accessory.

TCF023: Accessory must always be reachable.

TCF032: If accessory utilizes services containing characteristics that expose a transient state between target and current characteristic, changing the target characteristic state must not cause the current state characteristic to misrepresent physical state of accessory or prevent eventual matching of final set target characteristic state (given no mechanical failure).

TCF033: Accessory must handle multiple admin controllers.

TCF034: Admin controller should be able to modify controller permissions.

TCF036: Setup codes must be unique and not derived from public information. If accessory can display a dynamic setup code, it must not be manufactured with a setup code in any form (i.e., inside accessory packaging or on accessory itself), it must generate a new setup code each time it is needed and it must generate setup codes from a cryptographically secure random number generator.

TCF039: The information encoded in the setup payload is a URI with the following format:
X-HM://<Setup Payload encoding>
For R15 or earlier specifications:
The setup payload encoding contains the following information:
<VersionCategoryFlagsAndSetupCode><SetupID>
For R16 or later specifications:
The setup payload header encoding contains the following information:
<VersionCategoryFlagsAndSetupCode>
The payload data encoding contains the following information:
<SetupIDProductNumber><EUIProductNumber>

WAC, IP, and BLE flags must correctly indicate the capabilities of the accessory.

TCF040: Setup payload should be scannable from the NFC tag when the accessory is not powered on.

TCF041: The Setup ID is persistent across reboots and factory reset of the accessory. This identifier must be different than the DeviceID, serial number, model or accessory name and must be random and unique for each accessory instance manufactured by an accessory manufacturer.

TCF042: Accessories must not allow payloads to be written to their NFC tags via NDEF.

TCF043: Programmable NFC tags must leave pairing mode after 5 minutes of inactivity, and move to "Not in pairing mode". The accessory must require users to explicitly trigger NFC pairing mode via a physical interaction on the accessory. Pair-setup can still be allowed after the 5 minute timer expires. Accessories implementing programmable NFC must not implicitly be ready to pair once all pairings have been removed.

TCF044: Accessories must not support both QR and NFC setup payload retrieval modes.

TCF045: The accessory must only have a single valid setup code at a time across NFC, QR, and 8-digit setup modes.

TCF046: If the connection is closed after pair-setup M2, a subsequent pair-setup must succeed.

TCF047: If the connection is closed after pair-setup M4, a subsequent pair-setup must succeed.

TCF053: Accessory must be able to boot and work with HomeKit without Internet access. If Internet access is blocked for an accessory, it can rely on the presence of a local NTP server advertised via DHCP.

TCF054: Accessories must stop advertising within 10 minutes of the last pairing attempt while still in unpaired state. Accessories must require a user action such as a power cycle, reset, button press or other explicit user action for the accessory to re-enter pairing mode and advertise as an unpaired accessory.

TCF001 **Accessory must support a MFi authentication mechanism.**

Applies to all accessories. Perform this test case automatically with HCA. Below are steps that can be used to perform this test case manually in HAT.

1. Pair and discover accessory.
2. For HAP over Wi-Fi or Ethernet accessories: in the Discovery view of the trace window, find the Bonjour Record for the accessory and verify the "Pairing Feature Flags" section.
3. For HAP over BLE accessories: in the HAP Procedures view of the trace window, find the Read Response for the "Pairing Features" characteristic that occurs during pair-setup. Please note: this characteristic does not support Paired Read and is read prior to establishing a secure session when performing pair-setup within HAT.
4. For HAP over Thread accessories: in the Thread Discovery view of the trace window, find the Bonjour Record for the accessory and verify the "Pairing Feature Flags" section.
5. Verify Pairing Features is set to 0x01 for Supports MFi Auth IC or 0x02 for Supports Software Authentication.
6. For software token-based authentication, select the accessory server in the "Controllers" window. In the "Pairing" section, click "Get Authentication Token" and then "Get Server Information." In the "Events" trace view, verify both operations completed successfully.

TCF002 **Accessory must contain a method to reset the accessory to factory settings (e.g., button or UI element).**

Applies to all accessories. Perform this test case with HAT using the steps below.

1. Pair and discover accessory with HAT.
2. In the left sidebar of the Controllers window, note the Device ID. (e.g., AA:BB:CC:DD:EE:FF)
3. Verify the accessory provides a way to perform a factory reset. (e.g., a button or UI element)
4. Factory reset the accessory and wait for the accessory to begin advertising again. If applicable, perform user action such as button press, reboot etc., for accessory to begin advertising as unpaired.
5. Verify accessory advertises using a different Device ID other than the one found in step 2.
6. For accessories that do not support WAC, select the new accessory instance, Pair and Discover, and verify the operations complete successfully. For HAP over Wi-Fi or Ethernet accessories that support WAC, verify the Device ID in the WAC advertisement is different than the Device ID found in step 2. Select the new accessory instance and then successfully complete the WAC procedure.
7. Verify HAT is no longer able to communicate with the accessory using the old accessory instance in the left sidebar, which had the original Device ID found in step 2.

TCF003 **When submitting materials for Accessory Compliance Verification, the Licensee must provide instructions for performing a firmware update on the Accessory Compliance Questionnaire. Licensee must also**

provide samples that can receive a firmware update and meet all other requirements specified by the Packing List.

Applies to all accessories.

TCF004 **Verify accessory information (such as accessory name, advertisement information, manufacturer name, model name, characteristic values, etc.) does not contain any Apple trademarks. Refer to the Apple Trademark List (https://www.apple.com/legal/intellectual-property/trademark/appletmlist.html) for more information.**

Applies to all accessories. Perform this test case with HAT using the steps below.

1. Pair and discover accessory.
2. In left sidebar of Controllers window, select the accessory server.
3. Verify accessory advertisement does not contain any Apple trademarks.
4. Select each characteristic and verify the "Manufacturer Description" does not contain Apple trademarks.
5. Read each characteristic that supports "string" format and verify it does not contain Apple trademarks.

TCF005 **Accessory setup code must appear inside accessory's product packaging. If space allows, accessory setup code should be printed on accessory.**

Applies to all accessories. Applies to accessories that use static setup codes.

1. Verify setup code is printed inside accessory's product packaging or on in-box literature such as manuals or instructions.
2. If space allows, verify setup code is printed on accessory.
3. Verify only one numerical HomeKit setup code is printed on the accessory.

TCF006 **Verify setup code label meets the requirements as indicated in the Works with Apple HomeKit Identity Guidelines document available in the MFi Portal.**

Applies to accessories that support setup via NFC or QR code.

TCF007 **QR code or NFC code must be able to be scanned and successfully added to the home.**

Applies to accessories that support setup via NFC or QR code.

1. Perform this test case using an app.
2. Create home.
3. Add accessory.
4. Scan QR code or NFC code.

5. Verify accessory can be successfully added to home using feature-complete app or Home app to scan QR code or NFC code.
6. If multiple scan codes are present, attempt with each (e.g., QR code label on packaging, QR code label on accessory).

TCF008 **Setup code must be non-trivial, with 8 digits.**

Applies to all accessories.

1. Verify setup code does not contain 1 repeating digit (e.g., 11111111).
2. Verify setup code does not contain numbers in ascending or descending order (e.g., 12345678 or 987654321).

TCF012 **Accessory must reject pair-setup from a new controller when it is already paired.**

Applies to all accessories. Perform this test case automatically with HCA. Below are steps that can be used to perform this test case manually in HAT.

1. Pair and discover accessory with Controller 1.
2. Select the "+" at the bottom of left sidebar and create new Controller 2.
3. In left sidebar, select the new Controller 2.
4. Select the accessory name under Controller 2 and start pairing.
5. For HAP over BLE accessories, look in BLE HAP Transactions traffic view. For HAP over Wi-Fi or Ethernet accessories, look in HTTP traffic view. For HAP over Thread accessories, look in HAP traffic view. Verify that accessory rejects pairing setup from Controller 2 when it is already paired with Controller 1. Verify that accessory rejects the pairing request with the following TLV8 error: Type 0x07 (Error), Value: 6 (kTLVError_Unavailable). <070106>

TCF013 **Accessory should not send notifications when the characteristic's value has not changed.**

Applies to accessories that use HAP over Ethernet or Wi-Fi.

1. Pair and discover accessory.
2. Enable event notifications on the characteristics that support it.
3. In the Controllers window, under "Add Additional Controllers" panel, select "Controller 2" as the Controller, select 'on' for Admin, then select the "Add Controller" button.
4. In the left sidebar, select Controller 2, and the accessory name, then select the "Discover" button.
5. Select the accessory name under Controller 2 and enable notifications on all characteristics that support them.
6. From Controller 2, write to characteristics supporting paired write and notification permissions using the characteristic's current value (e.g. if the value is currently 11, write 11).

7. Verify that the accessory does not send notifications to Controller 1 when state changes have not occurred.

TCF015 **The accessory must always successfully unregister event notifications on every characteristic when a controller deregisters for notifications after having subscribed multiple times.**

Applies to all accessories. Perform this test case with HAT using the steps below.

1. Pair and discover accessory with HAT.
2. Select the Enable (event notification) button 3 times on the characteristics that support event notifications.
3. Disable event notifications for each characteristic that was subscribed multiple times in step 2.
4. Manually change the state of an accessory by physically altering the characteristic (e.g., manually turn the light off).
5. Verify that the accessory does not send any notifications to controller.

TCF020 **If the accessory has services that include "target state" and "current state" characteristics, the current state characteristic value must be updated when the target state is reached.**

Applies to accessories with the Lock Mechanism service. Applies to accessories with the Garage Door Opener service. Perform this test case with HAT using the steps below.

1. Pair and discover accessory.
2. Enable notifications on the "target state" and "current state" characteristics.
3. Write to the "target state" characteristic.
4. Verify the accessory sends a notification for the "current state" characteristic when the "current state" characteristic value is equal to that of the "target state" characteristic.
5. In the Summary panel, select the "Disconnect" button.
6. Power cycle accessory.
7. Read the "current state" characteristic and verify the value has not changed.

TCF022 **For accessories that support a service that includes the "on" characteristic. Verify that the correct state is reflected on the controller after power cycling the accessory.**

Applies to accessories with an "on" characteristic. Perform this test case with HAT using the steps below.

1. Pair and discover accessory.
2. In the Summary panel, select the "Disconnect" button.
3. Power cycle the accessory.
4. Discover the accessory.

5. Read the "on" characteristic.
6. Verify that the "on" characteristic matches the state of the accessory.

TCF023 **Accessory must always be reachable.**

Applies to accessories that use HAP over Ethernet or Wi-Fi. Perform this test case with HAT using the steps below.

1. Pair and discover accessory with HAT.
2. Perform a read to the accessory's primary functional characteristics and verify the accessory responds.
3. Perform a write to the accessory's primary functional characteristics and verify accessory responds.
4. In the Summary panel, select the "Disconnect" button.
5. Remove power from the accessory and power on again. i.e unplug power cord and plug in again, or remove and reinstall batteries.
6. After the accessory begins advertising again via Bonjour, select "Discover" button in the Pairing panel.
7. Verify pair-verify completes successfully.
8. Perform a read to the accessory's primary functional characteristics and verify the accessory responds.
9. Perform a write to the accessory's primary functional characteristics and verify accessory responds.
10. Power cycle router (AP).
11. After the accessory begins advertising again via Bonjour, select "Discover" button in the Pairing panel.
12. Verify pair-verify completes successfully.
13. Perform a read to the accessory's primary functional characteristics and verify the accessory responds.
14. Perform a write to the accessory's primary functional characteristics and verify accessory responds.

TCF032 **If accessory utilizes services containing characteristics that expose a transient state between target and current characteristic, changing the target characteristic state must not cause the current state characteristic to misrepresent physical state of accessory or prevent eventual matching of final set target characteristic state (given no mechanical failure).**

Applies to accessories with "target state" and "current state" characteristics. Perform this test case using the Home app on iOS.

1. Pair and discover accessory.
2. For each service that contains "target state" and "current state" characteristics, rapidly write different values to the "target state" characteristic 10 times using the Home app.
3. Verify the "current state" characteristic always matches the physical state of the accessory.
4. Verify the "target state" characteristic value is eventually reflected in both the "current state" characteristic and the physical accessory state.

Repeat this test multiple times with multiple units.

TCF033 **Accessory must handle multiple admin controllers.**

Applies to all accessories. Perform this test case automatically with HCA. Below are steps that can be used to perform this test case manually in HAT.

1. Pair and discover accessory with Controller 1.
2. In Controllers window, select "+" to create a new IP/BLE/Thread Controller 2.
3. Repeat step 2 to create 5 controllers.
4. Under Controller 1, select the accessory name, under "Add Additional Controllers" panel, select "Controller 2" as Controller, check the box 'on' for Admin and select the "Add Controller" button.
5. On the left pane of the Controllers window, select the accessory name under Controller 2, select "Start" button and select "Discover" button.
6. Under Controller 2, select the accessory name, under "Add Additional Controllers" panel, select "Controller 3" as Controller, check the box 'on' for Admin and select the "Add Controller" button.
7. On the left pane of the Controllers window, select the accessory name under Controller 3, select "Start" button and select "Discover" button.
8. In the Controllers window, under the "Pairing" panel, select the "List Pairings" button. See accessory's response to List Pairings completed. Verify accessory has 3 pairings. Verify in the details of List Pairings Completed that all 3 Controllers have the admin bit set (Permissions set to 1).
9. Under Controller 3, select the accessory name, under "Add Additional Controllers" panel, select "Controller 4" as Controller and select the "Add Controller" button.
10. In the Controllers window, under the "Pairing" panel, select the "List Pairings" button. See accessory's response to List Pairings completed. Verify accessory has 4 pairings and Controller 4 Permissions is set to 0.
11. On the left pane of the Controllers window, select the accessory name under Controller 4, select "Start" button, and select "Discover" button.
12. In the Controllers window, under the "Pairing" panel, select the "List Pairings" button.
13. Verify that accessory rejects the list pairing request with the following: TLV8 error: Type 0x07 (Error), Value: 2 (kTLVError_Authentication) <070102>. For HAP over Wi-Fi or Ethernet accessories, go to the HTTP trace view and see the HTTP response details. For HAP over BLE accessories, go to the HAP Procedures trace view and see the Write response details. For HAP over Thread accessories, go to the HAP Traffic trace view and see the Write response details.
14. For HAP over BLE and HAP over Thread accessories, verify accessory responds with 0x00 (Success). For HAP over BLE accessories, go to the HAP Transactions trace view and see the Write response details. For HAP over Thread accessories, go to the HAP Traffic trace view and see the Write response details.
15. Under Controller 4, select the accessory name. Under "Add Additional Controllers" panel, select "Controller 5" as Controller. Check the box 'on' for Admin and select the "Add Controller" button.

16. Verify that accessory rejects the pairing request with the following: TLV8 error: Type 0x07 (Error), Value: 2 (kTLVError_Authentication) <070102>. For HAP over Wi-Fi or Ethernet accessories, go to the HTTP trace view and see the HTTP response details. For HAP over BLE accessories, go to the HAP Procedures trace view and see the Write response details. For HAP over Thread accessories, go to the HAP Traffic trace view and see the Write response details.
17. For HAP over BLE and HAP over Thread accessories, verify accessory responds with 0x00 (Success). For HAP over BLE accessories, go to the HAP Transactions trace view and see the Write response details. For HAP over Thread accessories, go to the HAP Traffic trace view and see the Write response details.
18. Under Controller 4, select the accessory name. Under "Remove Additional Controllers" panel, select "Controller 1" as Controller and then select the "Remove Controller" button.
19. Verify that accessory rejects the remove pairing request with the following: TLV8 error: Type 0x07 (Error), Value: 2 (kTLVError_Authentication) <070102>. For HAP over Wi-Fi or Ethernet accessories, go to the HTTP trace view and see the HTTP response details. For HAP over BLE accessories, go to the HAP Procedures trace view and see the Write response details. For HAP over Thread accessories, go to the HAP Traffic trace view and see the Write response details.
20. For HAP over BLE and HAP over Thread accessories, verify accessory responds with 0x00 (Success). For HAP over BLE accessories, go to the HAP Transactions trace view and see the Write response details. For HAP over Thread accessories, go to the HAP Traffic trace view and see the Write response details.
21. Under Controller 4, select the accessory name. In the Controllers window, under the "Pairing" panel, select "Remove Pairing" button.
22. Verify that accessory rejects the remove pairing request with the following: TLV8 error: Type 0x07 (Error), Value: 2 (kTLVError_Authentication) <070102>. For HAP over Wi-Fi or Ethernet accessories, go to the HTTP trace view and see the HTTP response details. For HAP over BLE accessories, go to the HAP Procedures trace view and see the Write response details. For HAP over Thread accessories, go to the HAP Traffic trace view and see the Write response details.
23. For HAP over BLE and HAP over Thread accessories, verify accessory responds with 0x00 (Success). For HAP over BLE accessories, go to the HAP Transactions trace view and see the Write response details. For HAP over Thread accessories, go to the HAP Traffic trace view and see the Write response details.
24. Under Controller 3, select the accessory name. Under "Remove Additional Controllers" panel, select "Controller 1" as Controller and select the "Remove Controller" button.
25. Verify Controller 3 was able to successfully remove Controller 1. Status flag should display 0x00, paired.
26. Under Controller 3, select the accessory name, under "Remove Additional Controllers" panel, select "Controller 2" as Controller and select the "Remove Controller" button.
27. Verify Controller 3 was able to successfully remove Controller 2. Status flag should display: 0x00, paired.
28. In the Controllers window, under the "Pairing" panel, select the "List Pairings" button.
29. In the Events view, select the response to the request, and then select "Details" to show the details.

30. Verify the accessory has 2 pairings.
31. Under Controller 3, select the accessory name. Under the "Pairing" panel, select the "Remove Controller" button.
32. Verify Controller 3's Status Flag updates to 1 on the accessory's most recent advertisement. If applicable, perform user action such as button press, reboot etc., for accessory to begin advertising as unpaired. For HAP over Wi-Fi or Ethernet accessories, use the Bonjour Discovery traffic view. For HAP over BLE accessories, use the BLE Discovery traffic view. For HAP over Thread accessories, use the Thread Discovery view.
33. Verify Controller 4 cannot communicate (i.e., cannot write or read and display error response seen in Events traffic view from Controller 4). Verify that accessory rejects the request with the following TVL8 error: Type 0x07 (Error), Value: 2 (kTLVError_Authentication). <070102>. Verify that Status Flag updates to 1 on Controller 4. If applicable, perform user action such as button press, reboot etc., for accessory to begin advertising as unpaired.
34. For HAP over BLE and HAP over Thread accessories, verify accessory responds with 0x00 (Success). For HAP over BLE accessories, go to the HAP Transactions trace view and see the Write response details. For HAP over Thread accessories, go to the HAP Traffic trace view and see the Write response details.

TCF034 **Admin controller should be able to modify controller permissions.**

Applies to all accessories. Perform this test case automatically with HCA. Below are steps that can be used to perform this test case manually in HAT.

1. Pair and discover accessory with Controller 1.
2. In Controllers window, select "+" to create a new controller.
3. Under Controller 1, select the accessory name, under "Add Additional Controllers" panel, select "Controller 2" as Controller and select the "Add Controller" button.
4. In the Controllers window, under the "Pairing" panel, select the "List Pairings" button. See accessory's response to List Pairings completed. Verify in the details of List Pairings Completed that Controller 2 does not have the admin bit set (i.e., permissions set to 0).
5. On the left pane of the Controllers window, select the accessory name under Controller 2, select "Start" button, and select "Discover" button.
6. In the Controllers window, under the "Pairing" panel, select the "List Pairings" button.
7. Verify that accessory rejects the list pairing request with the following: TLV8 error: Type 0x07 (Error), Value: 2 (kTLVError_Authentication) <070102>. For HAP over Wi-Fi or Ethernet accessories, go to the HTTP trace view and see the HTTP response details. For HAP over BLE accessories, go to the HAP Procedures trace view and see the Write response details. For HAP over Thread accessories, go to the HAP Traffic trace view and see the Write response details.
8. For HAP over BLE and HAP over Thread accessories, verify accessory responds with 0x00 (Success). For HAP over BLE accessories, go to the HAP Transactions trace view and see the Write response details. For HAP over Thread accessories, go to the HAP Traffic trace view and see the Write response details.

9. Under Controller 1, select the accessory name. Under "Add Additional Controllers" panel, select "Controller 2" as Controller. Check the box 'on' for Admin and select the "Add Controller" button.
10. In the Controllers window, under the "Pairing" panel, select the "List Pairings" button. See accessory's response to List Pairings completed. Verify in the details of List Pairings Completed that Controller 2 does have the admin bit set (i.e., permissions set to 1).
11. On the left pane of the Controllers window, select the accessory name under Controller 2. Under the "Pairing" panel, select the "List Pairings" button.
12. Verify List Pairing is completed. Verify in the details of List Pairings Completed that both Controller 1 and Controller 2 do have the admin bit set (i.e., permissions set to 1).
13. Under Controller 1, select the accessory name. Under "Add Additional Controller" panel, select Controller 2 as Controller. Uncheck the the Admin box and select the "Add Controller" button.
14. In the Controllers window, under the "Pairing" panel, select the "List Pairings" button. See accessory's response to List Pairings completed. Verify in the details of List Pairings Completed that Controller 2 does not have the admin bit set (i.e., permissions set to 0).
15. From Controller 2, under the "Pairing" panel, select the "List Pairings" button.
16. Verify that accessory rejects the list pairing request with the following: TLV8 error: Type 0x07 (Error), Value: 2 (kTLVError_Authentication) <070102>. For For HAP over Wi-Fi or Ethernet accessories, go to the HTTP trace view and see the HTTP response details. For HAP over BLE accessories, go to the HAP Procedures trace view and see the Write response details.
17. For HAP over BLE and HAP over Thread accessories, verify accessory responds with 0x00 (Success). For HAP over BLE accessories, go to the HAP Transactions trace view and see the Write response details. For HAP over Thread accessories, go to the HAP Traffic trace view and see the Write response details.

TCF036 **Setup codes must be unique and not derived from public information. If accessory can display a dynamic setup code, it must not be manufactured with a setup code in any form (i.e., inside accessory packaging or on accessory itself), it must generate a new setup code each time it is needed and it must generate setup codes from a cryptographically secure random number generator.**

Applies to accessories that have a mechanism to display a dynamic setup code and do not support setup via QR Code or NFC. Perform this test case with HAT using the steps below.

1. If an accessory can display a dynamic setup code, then perform these steps:
2. Select Start Pairing.
3. If the accessory has generated a setup code, then it must present the setup code to the user, e.g. display it on the accessory's screen.
4. Select "Stop Pairing" button in the Controllers window.
5. In the Summary panel, select the "Disconnect" button.
6. Select Start Pairing.
7. Verify that the new displayed setup code differs from step 3.

8. Verify that accessory has not been manufactured with a pairing code fixed to the physical device or printed on any documentation or packaging.

TCF039 **The information encoded in the setup payload is a URI with the following format:**
X-HM://<Setup Payload encoding>
For R15 or earlier specifications:
The setup payload encoding contains the following information:
<VersionCategoryFlagsAndSetupCode><SetupID>
For R16 or later specifications:
The setup payload header encoding contains the following information:
<VersionCategoryFlagsAndSetupCode>
The payload data encoding contains the following information:
<SetupIDProductNumber><EUIProductNumber>

WAC, IP, and BLE flags must correctly indicate the capabilities of the accessory.

Applies to accessories that support setup via NFC or QR code. Perform this test case using HCA.

* Programmable NFC tags must zero out the setup code and setup ID fields from the setup payload while paired.

TCF040 **Setup payload should be scannable from the NFC tag when the accessory is not powered on.**

Applies to accessories that support setup via NFC. Perform this test case using the Home app on iOS.

1. Remove power from the accessory.
2. Using the Home app, add the accessory to the home by scanning the NFC tag.
3. After scanning the tag, restore power to the accessory.
4. Verify pairing completes successfully, and accessory is added to the home.

TCF041 **The Setup ID is persistent across reboots and factory reset of the accessory. This identifier must be different than the DeviceID, serial number, model or accessory name and must be random and unique for each accessory instance manufactured by an accessory manufacturer.**

Applies to accessories that support setup via NFC or QR code. Perform this test case with HAT using the steps below.

1. Begin pairing with accessory in HAT using Companion app.
2. After scanning the payload in Companion, note the last 4 digits of the setup payload. Verify the Setup ID differs from the DeviceID, serial number, model or accessory name, and is random for each accessory instance manufactured by accessory manufacturer.
3. Select Continue to send the payload to HAT.
4. Verify pair-setup completes successfully.
5. Remove pairing from the accessory.

6. If applicable, perform user action such as button press, reboot etc., for accessory to begin advertising as unpaired.
7. Begin pairing with accessory in HAT using Companion app.
8. After scanning the payload in Companion, verify that last 4 digits of the setup payload match the 4 digits noted in step 2.
9. Select Continue to send the payload to HAT.
10. Verify pair-setup completes successfully.
11. Factory reset accessory.
12. If applicable, perform user action such as button press, reboot etc., for accessory to begin advertising as unpaired.
13. Begin pairing with accessory in HAT using Companion app.
14. After scanning the payload in Companion, verify that last 4 digits of the setup payload match the 4 digits noted in step 2.
15. Select Continue to send the payload to HAT.
16. Verify pair setup completes successfully.

TCF042 **Accessories must not allow payloads to be written to their NFC tags via NDEF.**

Applies to accessories that support setup via NFC. Perform this test case using HCA.

TCF043 **Programmable NFC tags must leave pairing mode after 5 minutes of inactivity, and move to "Not in pairing mode". The accessory must require users to explicitly trigger NFC pairing mode via a physical interaction on the accessory. Pair-setup can still be allowed after the 5 minute timer expires. Accessories implementing programmable NFC must not implicitly be ready to pair once all pairings have been removed.**

Applies to accessories that support setup via programmable NFC. Perform this test case with HAT using the steps below.

1. Factory reset accessory and do not place the accessory into pairing mode.
2. Using the Home App, attempt to add accessory to the home by scanning NFC tag.
3. Verify the Home App shows the "accessory is not ready" screen.
4. Place the accessory into pairing mode and wait 5 minutes. Do not scan the NFC tag.
5. After 5 minutes, attempt to add accessory to the home again by scanning NFC tag.
6. Verify the Home App shows the "accessory is not ready" screen.
7. Place the accessory into pairing mode again, and add accessory to the home by scanning NFC tag.
8. Complete pair-setup and verify accessory is successfully added to the home.
9. Remove accessory from the home. Do not place the accessory into pairing mode.

10. Using the Home App, attempt to add accessory to the home by scanning NFC tag.
11. Verify the Home App shows the "accessory is not ready" screen.
12. Using HAT, place the accessory into pairing mode and begin pair setup using the Companion app. Do not select Continue in the Companion app after scanning the NFC tag.
13. Wait 5 minutes. After 5 minutes, select "Continue" in the Companion app and verify pair-setup completes successfully.

TCF044 **Accessories must not support both QR and NFC setup payload retrieval modes.**

Applies to all accessories.

1. Verify the accessory does not use both QR code and NFC setup.

TCF045 **The accessory must only have a single valid setup code at a time across NFC, QR, and 8-digit setup modes.**

Applies to all accessories. Perform this test case with HAT using the steps below.

1. Select Start Pairing on the accessory.
2. Select HomeKit Companion button on the pop-up window..
3. Launch the Companion app on iOS device.
4. Find your Companion app and select either QR Code or NFC.
5. For NFC, hold iOS device near accessory. For QR code, scan the QR code.
6. Select Continue.
7. In the Events traffic view, see Details in the "Try Setup Payload" event.
8. Verify Setup Code from the Payload matches the Setup Code on the label.

TCF046 **If the connection is closed after pair-setup M2, a subsequent pair-setup must succeed.**

Applies to accessories that use hardware authentication. Applies to accessories that use software certificate authentication. Does not apply to accessories while using the HAP over Thread transport. Perform this test case using HCA.

1. Select the accessory server from the left sidebar of the Controllers window.
2. In the Controllers window, under Pairing panel, select the "Start Pairing" button.
3. When the setup code pairing window appears, select the Stop button.
4. Select the "Disconnect" button.
5. In Pairing panel, select the Start Pairing button.

6. Enter the setup code and select the send button or pair using the Companion app.
7. Verify pair-setup completes successfully.
8. In the summary panel, select the Discover button.
9. Verify pair-verify completes successfully.

TCF047 **If the connection is closed after pair-setup M4, a subsequent pair-setup must succeed.**

Applies to accessories that use hardware authentication. Applies to accessories that use software certificate authentication. Does not apply to accessories while using the HAP over Thread transport. Perform this test case using HCA.

TCF053 **Accessory must be able to boot and work with HomeKit without Internet access. If Internet access is blocked for an accessory, it can rely on the presence of a local NTP server advertised via DHCP.**

Applies to accessories that use HAP over Ethernet or Wi-Fi. Perform this test case with HAT using the steps below.

1. Pair and discover accessory with HAT.
2. Perform a read to the accessory's primary functional characteristics and verify the accessory responds.
3. Perform a write to the accessory's primary functional characteristics and verify the accessory responds.
4. In the Summary panel, select the "Disconnect" button.
5. Remove the ethernet cable connected to the router's WAN port, disconnect modem, or remove coaxial cable from modem/router combination to disable internet access.
6. Power cycle the accessory.
7. After the accessory begins advertising again via Bonjour, select "Discover" button in the Pairing panel.
8. Verify pair-verify completes successfully.
9. Perform a read to the accessory's primary functional characteristics and verify the accessory responds.
10. Perform a write to the accessory's primary functional characteristics and verify the accessory responds.
11. Restore internet access to the router.
12. Perform a read to the accessory's primary functional characteristics and verify the accessory responds.
13. Perform a write to the accessory's primary functional characteristics and verify the accessory responds.

TCF054 **Accessories must stop advertising within 10 minutes of the last pairing attempt while still in unpaired state. Accessories must require a user action such as a power cycle, reset, button press or other explicit user action for the accessory to re-enter pairing mode and advertise as an unpaired accessory.**

Applies to accessories that support HomeKit Accessory Protocol specification R16 or later. Perform this test case with HAT using the steps below.

1. Factory reset accessory.
2. In the bottom left corner of the Controllers window, select the "+" icon to create a new controller.
3. Select the controller and then select the "Start" button to begin discovering accessory servers.
4. Using the Trace window, verify the accessory begins advertising. For HAP over BLE accessoires, use the "BLE Discovery" view. For HAP over Wi-Fi or Ethernet, use the "IP Discovery" view. If the accessory supports WAC2, use the "WAC Discovery" view.
5. Wait 10 minutes.
6. Select the controller and then select the "Stop" button to stop discovering accessory servers. Then select "Start" to begin again.
7. Using the Trace window, verify the accessory is no longer advertising.
8. Place the accessory back into pairing mode by performing a power cycle, reset, button press or other explicit user action.
9. Using the Trace window, verify the accessory begins advertising again.
10. For accessories that support WAC2, select "Join Access Point" and wait for the controller to connect to the accessory's network. Once on the network, wait for the accessory to begin advertising via Bonjour.
11. In the left sidebar of the Controllers window, select the accessory and then select "Start Pairing".
12. After the pairing prompt appears, select "Stop" to dismiss the pairing prompt and to abort the pairing process.
13. Select the "Disconnect" button.
14. Wait 10 minutes.
15. Select the controller and then select the "Stop" button to stop discovering accessory servers. Then select "Start" to begin again.
16. Using the Trace window, verify the accessory is no longer advertising.
17. Place the accessory back into pairing mode by performing a power cycle, reset, button press or other explicit user action.
18. For accessories that support WAC2, select "Join Access Point" and wait for the controller to connect to the accessory's network. Once on the network, wait for the accessory to begin advertising via Bonjour.
19. In the left sidebar of the Controllers window, select the accessory and then select "Start Pairing".
20. Complete the pair-setup process, and then select "Discover".
21. After the Discover operation completes, select "Remove Pairing".
22. Wait 10 minutes.
23. Select the controller and then select the "Stop" button to stop discovering accessory servers. Then select "Start" to begin again.

24. Using the Trace window, verify the accessory is no longer advertising.
25. Place the accessory back into pairing mode by performing a power cycle, reset, button press or other explicit user action.
26. Using the Trace window, verify the accessory begins advertising again.
27. For accessories that support WAC2, select the accessory, select "Join Access Point" and wait for the controller to connect to the accessory's network. Once on the network, wait for the accessory to begin advertising via Bonjour.
28. In the Wi-Fi Accessory Configuration panel, enter the Wi-Fi SSID and Wi-Fi PSK and select the "Send WAC Configuration" button.
29. Ensure your Mac is on the network you expect the accessory to join, otherwise re-join the expected network.
30. Once the accessory begins advertising via Bonjour on the newly joined network, select the "Confirm WAC Configuration" button.
31. After the accessory successfully responds to the /Configured request, select the "Discover" button.
32. After the Discover operation completes, select "Remove Pairing".
33. Wait 10 minutes.
34. Select the controller and then select the "Stop" button to stop discovering accessory servers. Then select "Start" to begin again.
35. Using the Trace window, verify the accessory is no longer advertising.

1.2 HomeKit-enabled Wi-Fi routers

TCSR001: Verify HomeKit can be enabled on the router accessory via the manufacture's app while the iOS device is connected to the router's main Wi-Fi network.

TCSR002: For routers that support satellites to create mesh networks, verify HomeKit can be enabled on the router using the accessory's app when the iOS device is connected to one of the satellites on the main network.

TCSR003: Verify when Ownership Proof token is required, the accessory provides the Ownership Proof Token pairingTypeFlag in pair-setup M2. If the accessory does not receive a valid token from the controller during pair-setup, the accessory must respond with <kTLVError_OwnershipFailure>.

TCSR004: Verify the ability to add firewall rules when paired to the router, and firewall rules are removed after unpairing from the router.

TCSR005: Verify Router does not advertise HAP Wi-Fi router service when the router is in Bridge mode.

TCSR006: For routers that support "guest" networks or other networks other than the main network, verify router only broadcast a _hap._tcp bonjour record on the main network and not on any other network.

TCSR007: Verify router can be added when the HomeKit controller is connected via Wi-Fi on a satellite node.

TCSR008: Verify router does not broadcast a _hap._tcp bonjour record on the WAN port of the router.

TCSR009: Verify accessory's app checks for resident device availability (e.g. supported Apple TV, iPad, or HomePod using the same iCloud account) before pairing with the router. If no resident device is present, verify app does not complete router pairing.

TCSR010: Verify router can successfully add accessories to the network using PPSK credentials.

TCSR011: Verify firewall does not allow incoming traffic from WAN to any of the LAN clients in the restricted group.

TCSR012: Verify the behavior of a WAN TCP rule.

TCSR013: Verify the behavior of a WAN UDP rule.

TCSR014: Verify the behavior of a WAN ICMP rule.

TCSR015: Verify the behavior of LAN UDP Inbound Multicast Bridging rule.

TCSR016: Verify the behavior of LAN UDP Outbound Multicast Bridging rule.

TCSR017: Verify the behavior of LAN TCP Inbound Static port rule with individual port.

TCSR018: Verify the behavior of LAN TCP Inbound Static port rule with port ranges.

TCSR019: Verify the behavior of LAN UDP Inbound Static port rule with individual port.

TCSR020: Verify the behavior of LAN UDP Inbound Static port rule with port ranges.

TCSR021: Verify the behavior of LAN ICMP Inbound rule.

TCSR022: Verify the behavior of LAN TCP Outbound Static port rule with individual port.

TCSR023: Verify the behavior of LAN TCP Outbound Static port rule with port ranges.

TCSR024: Verify the behavior of a LAN UDP Outbound Static port rule with an individual port.

TCSR025: Verify the behavior of a LAN UDP Outbound Static port rule with port ranges.

TCSR026: Verify the behavior of a LAN ICMP Outbound Static rule.

TCSR027: Verify the behavior of a LAN Inbound UDP DNS-SD Dynamic port rule.

TCSR028: Verify the behavior of a LAN Inbound UDP SSDP Dynamic port rule.

TCSR029: Verify the behavior of a a LAN Outbound UDP DNS-SD Dynamic port rule.

TCSR030: Verify the behavior of a LAN Outbound UDP SSDP Dynamic port rule.

TCSR031: Verify that all DNS requests made by accessories with rules must go through the router's DNS server. Verify that the DNS query response TTL is clamped to 10sec.

TCSR032: Verify that recursive DNS requests are also handled by the router. (Router has to send the DNS query to the DNS server with resolve recursively bit enabled.)

TCSR033: Verify that all DNS requests made by client with full access WAN rules and LAN rules should not be clamped by the router's DNS server when using external DNS server.

TCSR034: Verify that a client on the restricted group CANNOT utilize UPNP to open ports up on the firewall.

TCSR035: Verify that a clients part of Main group can utilize UPNP to open ports up on the firewall.

TCSR036: Verify the behavior of an outbound TCP rule to DNS name, verify that when DNS changes to new IP address rule follow name, not Dest IP.

TCSR037: Verify that TCP Inbound/Outbound rules do not get effected when accessory with network declaration changes the IP address.

TCSR038: Verify that UDP Inbound/Outbound rules do not get effected when accessory with network declaration changes the IP address.

TCSR039: Verify that the HAT tool and Home app should discover both Gateway node and the Satellite node if the Satellite node is already setup in the mesh network.

TCSR040: Verify that the HAT tool and Home app should discover newly added satellite node to the mesh network.

TCSR041: Verify that the HomeKit accessory is controllable when moved to the HK LAN.

TCSR042: Verify that traffic is blocked between Restricted group and Main group.

TCSR043: Any Wi-Fi Router services must include the required characteristics.

TCSR044: Verify that the accessory can successfully perform a Network Client Profile Control "Add" operation.

TCSR045: Verify the Network client profile control List operation.

TCSR046: Verify the Network client profile control Read operation.

TCSR047: Verify the Network client profile control Remove operation.

TCSR048: Verify the Network client profile control Update operation.

TCSR049: Verify the Network client status control operation.

TCSR050: Verify the Network Access Violation list operation.

TCSR051: Verify the Network Access Violation reset operation.

TCSR052: Verify that the router disconnects all the established connections for a client when the client configuration is modified (Move client from No Restriction to Auto).

TCSR053: Verify the Traffic between clients within a restricted group.

TCSR054: Verify the Network Access Violation Control Event Notifications.

TCSR001 **Verify HomeKit can be enabled on the router accessory via the manufacture's app while the iOS device is connected to the router's main Wi-Fi network.**

Applies to HomeKit-enabled Wi-Fi router accessories. Perform this test case using an iOS device running the accessory app and the Home app.

1. Use the accessory's app to setup and create a new wireless network.
2. Connect the iOS device to the newly created network.
3. Using the accessory's app, navigate to the HomeKit integration settings.
4. Verify that the user is able to enable HomeKit and complete pair-setup, adding the accessory to the home from within the manufacture's app (Use default settings during the setup flow).
5. After adding the accessory to the home, verify the accessory is visible in the Home App in the home's settings under "Wi-Fi Network and Routers".
6. Open the manufacture's iOS app and verify that the app no longer shows information about enabling HomeKit.
7. Select the router in the Home App in the home's settings under "Wi-Fi Network and Routers".
8. Select the main router (gateway) and select "Remove Accessory" to remove the router from the home.
9. Verify the router is no longer shown in the Home App under "Wi-Fi Network and Routers".
10. Open the manufacture's iOS app and verify that the app shows information about enabling HomeKit again.
11. If applicable, use the accessory's app to create a new wireless "guest" network.
12. Connect the iOS device to the newly created "guest" network.
13. Verify user is unable to enable HomeKit while on the "guest" network.

TCSR002 **For routers that support satellites to create mesh networks, verify HomeKit can be enabled on the router using the accessory's app when the iOS device is connected to one of the satellites on the main network.**

Applies to HomeKit-enabled Wi-Fi router accessories. Perform this test case using an iOS device running the accessory app and the Home app.

1. Use the accessory's app to setup and create a new wireless network, and add a satellite or beacon.
2. Ensure the iOS device is connected to one of the satellites, and not the main gateway. (To achieve this, a shielding cloth can be used to cover the main gateway to cause satellite to have a stronger signal, or satellite and iOS device can be moved to a more isolated area where the signal strength from the satellite to the iOS device is strongest.)

3. Using the accessory's app, navigate to the HomeKit integration settings.
4. Verify that the user is able to enable HomeKit and complete pair-setup, adding the accessory to the home from within the manufacture's app.
5. After adding the accessory to the home, verify the accessory is visible in the Home App in the home's settings under "Wi-Fi Network and Routers".
6. Open the manufacture's iOS app and verify that the app no longer shows information about enabling HomeKit.
7. Select the router in the Home App in the home's settings under "Wi-Fi Network and Routers".
8. Select the main router and select "Remove Accessory" to remove the router from the home.
9. Verify the router is no longer shown in the Home App.
10. Open the manufacture's iOS app and verify that the app shows information about enabling HomeKit again.

TCSR003 **Verify when Ownership Proof token is required, the accessory provides the Ownership Proof Token pairingTypeFlag in pair-setup M2. If the accessory does not receive a valid token from the controller during pair-setup, the accessory must respond with <kTLVError_OwnershipFailure>.**

Applies to HomeKit-enabled Wi-Fi router accessories. Perform this test case using HAT and an iOS device running the accessory app.

1. Use the accessory's app to setup and create a new wireless network.
2. Connect a Mac running HAT to the newly created network.
3. Open HAT, and in the left sidebar under "IP Controllers", select "Controller 1".
4. Using "Controller 1", click the "Start" button under the "Discovery" section on the main screen.
5. In the left sidebar, select the router accessory.
6. Leave Ownership Proof Token field empty and select "Start Pairing" and attempt to complete pair-setup.
7. Using the Events view in the trace, select the Pair-Setup - M2 event and verify it includes PairingTypeFlags with Bit 30 set (Ownership Proof Token Required).
8. Select the Pair-Setup - M6 event and verify it contains error TLV 070108 <kTLVError_OwnershipFailure> (Ownership Proof is either incorrect or not provided).
9. From the main controllers window, enter an invalid Ownership Proof Token, select the "Start Pairing", and attempt to complete pair-setup again.
10. Using the Events view in the trace, select the Pair-Setup - M6 event and verify it contains error TLV 070108 <kTLVError_OwnershipFailure> (Ownership Proof is either incorrect or not provided).
11. From the main controllers window, enter the valid Ownership Proof Token, select the "Start Pairing" button, and complete pair-setup.
12. Verify pairing competes successfully.

TCSR004 **Verify the ability to add firewall rules when paired to the router, and firewall rules are removed after unpairing from the router.**

Applies to HomeKit-enabled Wi-Fi router accessories. Perform this test case using HAT, the accessory app, a Raspberry Pi running an ADK IP accessory, and the test network declaration JSON files found on the MFi Portal.

1. Use the accessory's app to setup and create a new wireless network.
2. Setup an ADK device as a HomeKit accessory.
3. On the ADK terminal, run `#sudo raspi-config` and follow Network Options to join the network created in step 1.
4. Join the newly created network on a Mac running HAT.
5. Pair and Discover router accessory using HAT.
6. In the left sidebar, select "Managed Network Enable".
7. On the main screen, navigate to the "Prepare and Execute Timed Write [Unsigned Integer]" section.
8. Enter a value of "1" in the text field and click the "Timed Write" button.
9. In the left sidebar, select the name of the accessory that was added in step 3.
10. On the main accessory server view, navigate to the "Network Declarations" pane and select the "Select File" button next to "Import JSON".
11. Browse and select the TCSR004.json file, downloaded from the MFi Portal.
12. Select "Apply rules" button in the "Network Declarations" section.
13. On the screen that appears, ensure "Restricted" is selected at top of page, and select "Apply" to apply the rules to the router for this accessory.
14. Verify the rules are correctly reflected on the router by selecting the "Show Configured" button.
15. Once rules are applied, start a TCP iperf server on the ADK terminal by running the following command on ADK terminal: `iperf -s -p 2234 -i 1`.
16. On the Mac which is running HAT, open a new Terminal window.
17. Use iperf to connect to the server started in above step using the following command: `iperf -c <ip address of ADK accessory> -p 2234 -i 1`.
18. Verify connection is not made in iperf and traffic is not sent.
19. Stop iperf on the Mac and ADK using CTRL + C on the terminal.
20. On the main screen of the HAT tool, click the "Remove Pairing" button under the "Pairing" section for the router accessory. When successfully unpaired, router accessory should remove all the firewall rules.
21. On ADK terminal start a TCP iperf server by running the following command on ADK terminal: `iperf -s -p 2235 -i 1`.

22. On the Mac which is running HAT, open the Terminal application.
23. Use iperf to connect to the server started in above step using the following command: `iperf -c <ip address of ADK accessory> -p 2235 -i 1`.
24. Verify connection is made in iperf and traffic is sent successfully.

TCSR005 **Verify Router does not advertise HAP Wi-Fi router service when the router is in Bridge mode.**

Applies to HomeKit-enabled Wi-Fi router accessories. Perform this test case using HAT and an iOS device running the accessory app.

1. Use the accessory's app to setup and create a new wireless network.
2. Open HAT, and in the left sidebar under "IP Controllers", click on "Controller 1"
3. With "Controller 1" selected on the left, click the "Start" button under the "Discovery" section on the main screen.
4. In the left sidebar, Verify if the router is seen in the list of accessories.
5. Go the router app and select Network settings.
6. From the WAN settings of the router, select Bridge as the WAN setting for the router. Note: The router may need some time to apply the settings, and may reboot.
7. Once the router is back online, Go to HAT tool.
8. Open HAT, and in the left sidebar under "IP Controllers", select "Controller 1".
9. With "Controller 1" selected on the left, click the "Stop" button and then click "Start" button under the "Discovery" section on the main screen.
10. In the left sidebar, Verify router should not be listed in the list of accessories advertised.

TCSR006 **For routers that support "guest" networks or other networks other than the main network, verify router only broadcast a _hap._tcp bonjour record on the main network and not on any other network.**

Applies to HomeKit-enabled Wi-Fi router accessories. Perform this test case using HAT and an iOS device running the accessory app.

1. Use the accessory's app to setup and create a new wireless network.
2. Join the newly created network on a Mac running HAT.
3. Open HAT, and in the left sidebar under "IP Controllers", select "Controller 1".
4. With "Controller 1" selected on the left, click the "Start" button under the "Discovery" section on the main screen.
5. In the left sidebar, select the router.
6. In the left sidebar, verify there is entry for the router under test.
7. Create a "guest" wireless network.

8. Join the network from step 7 on a Mac running HAT.
9. Open HAT, and in the left sidebar under “IP Controllers”, select “Controller 1”.
10. With “Controller 1” selected on the left, first click “Stop” then click on“Start” button under the “Discovery” section on the main screen.
11. In the left sidebar, verify there is no entry for the router under test.

TCSR007 **Verify router can be added when the HomeKit controller is connected via Wi-Fi on a satellite node.**

Applies to HomeKit-enabled Wi-Fi router accessories. Perform this test case using HAT and an iOS device running the accessory app.

1. Use the accessory’s app to setup and create a new wireless network.
2. Add a satellite node to create a mesh network.
3. Join the newly created network on a Mac running HAT. Ensure the Mac is connected to one of the satellites, and not the main gateway. (To achieve this, a shielding cloth can be used to cover the main gateway to cause satellite to have a stronger signal, or satellite and Mac can be moved to a more isolated area where the signal strength from the satellite to the Mac is strongest.)
4. Open HAT, and in the left sidebar under “IP Controllers”, select “Controller 1”.
5. With “Controller 1” selected on the left, click the “Start” button under the “Discovery” section on the main screen.
6. In the left sidebar, select the router.
7. On the main screen, enter the Ownership Proof Token if needed, then click the “Start Pairing” button under the “Pairing” section.
8. In the popup, enter the setup payload or setup code of the router.
9. Verify pair setup completes successfully.

TCSR008 **Verify router does not broadcast a _hap._tcp bonjour record on the WAN port of the router.**

Applies to HomeKit-enabled Wi-Fi router accessories. Perform this test case using HAT and an iOS device running the accessory app.

1. Connect one side of the ethernet cable to WAN port of the router and connect the other end to a switch.
2. Connect the WAN connection cable from the Modem to the above switch.
3. To the same switch connect another ethernet cable to the switch and other end connect to the Mac client running HAT.
4. Use the accessory’s app to setup and create a new wireless network.
5. With the Mac client connected to the switch. Open HAT, in the left sidebar under “IP Controllers”, click on “Controller 1”.

6. With “Controller 1” selected on the left, click the “Start” button under the “Discovery” section on the main screen.
7. In the left sidebar, verify there is no entry for the router under test.

TCSR009 **Verify accessory’s app checks for resident device availability (e.g. supported Apple TV, iPad, or HomePod using the same iCloud account) before pairing with the router. If no resident device is present, verify app does not complete router pairing.**

Applies to HomeKit-enabled Wi-Fi router accessories. Perform this test case using an iOS device running the accessory app.

1. Setup router and create a network.
2. Connect an iOS device signed into iCloud to the newly created network.
3. Enable Home Hub on a supported iPad or Apple TV, or use a HomePod, running the latest version of iOS using the same iCloud account.
4. Using the Home App, navigate to the home’s settings and select “Hubs and Bridges” to verify the resident device shows as “Connected”.
5. Using the accessory’s app, navigate to the HomeKit integration settings.
6. Verify that the user is able to enable HomeKit and complete pair-setup, adding the accessory to the home from within the manufacture’s app (Use default settings during the setup flow).
7. After adding the accessory to the home, verify the accessory is visible in the Home App in the home’s settings under ”Wi-Fi Network and Routers”.
8. Open the manufacture’s iOS app and verify that the app no longer shows information about enabling HomeKit.
9. Select the router in the Home App in the home’s settings under “Wi-Fi Network and Routers”
10. Select the main router (gateway) and select “Remove Accessory” to remove the router from the home.
11. Verify the router is no longer shown in the Home App under “Wi-Fi Network and Routers”.
12. Open the manufacture’s iOS app and verify that the app shows information about enabling HomeKit again.
13. Using the resident device, navigate to the Settings app and select “Home”, then disable Home Hub. For HomePod, use an iOS device to remove HomePod from the home.
14. Open the manufacture’s iOS app and verify that the app shows information about enabling HomeKit again.
15. Attempt to enable HomeKit, and verify that the app stops the setup process with a warning message stating that a resident device is needed to pair the router to home.

TCSR010 **Verify router can successfully add accessories to the network using PPSK credentials.**

Applies to HomeKit-enabled Wi-Fi router accessories. Perform this test case using HAT and an iOS device running the accessory app.

1. Use the accessory's app to setup and create a new wireless network.
2. Prepare a ADK device as a HomeKit accessory.
3. Join the newly created network with a Mac client running HAT.
4. Open HAT, and in the left sidebar under "IP Controllers", select "Controller 1".
5. With "Controller 1" selected on the left, click the "Start" button under the "Discovery" section on the main screen.
6. In the left sidebar, select the router.
7. On the main screen, enter the Ownership Proof Token if needed, then click the "Start Pairing" button under the "Pairing" section, and complete pair setup.
8. In the main screen, click the "Discover" button in the "Summary" section.
9. In the left sidebar, select "Managed Network Enable".
10. On the main screen, navigate to the "Prepare and Execute Timed Write [Unsigned Integer]" section.
11. Enter a value of "1" in the text field and click the "Timed Write" button.
12. In the left sidebar, select "Network Client Profile Control".
13. Under the "Write Options" section, ensure "Write With Response" is enabled.
14. On the main screen, navigate to the "Prepare and Execute Timed Write [TLV8 Packet]" section.
15. Click on the "Build TLV" button.
16. In the "Network Client Control" TLV builder, enter the following values:
    - Set the "Operation" to "Add".
    - Leave the "Client Profile Identifier" field blank.
    - Enter "1" into the "Client Group Identifier" text field.
    - Enter "31 31 31 31 31 31 31 31" (HEX converted value of "11111111") into the Credential Data field.
    - Select "PSK" for the Credential Type.
    - Enter "010100" into the "WAN Firewall Config" field.
    - Enter "010100" in the LAN Firewall Config field.
17. Click the "Add" button to add rule to the TLV builder table.
18. Click the "Build TLV" button in the bottom right corner.
19. In the "Prepare and Execute Timed Write [TLV8 Packet]" section, click the "Timed Write" button to write this TLV to the router. Note: Note: If the trace view shows a warning, this can be ignored.
20. On the ADK terminal, run `#sudo raspi-config` and follow Network Options to join the network created in step 1 with with "11111111" as the Wi-Fi password and enter the same network name as the one created in step 1.
21. Verify the ADK accessory connected to the network successfully.

TCSR011 **Verify firewall does not allow incoming traffic from WAN to any of the LAN clients in the restricted group.**

Applies to HomeKit-enabled Wi-Fi router accessories. Perform this test case using HAT and an iOS device running the accessory app.

1. Use the accessory's app to setup and create a new wireless network.
2. Setup an iOS device and join the network created in step 1.
3. Join the newly created network on a Mac running HAT.
4. Open HAT, and in the left sidebar under "IP Controllers", select "Controller 1"..
5. With "Controller 1" selected on the left, click the "Start" button under the "Discovery" section on the main screen.
6. In the left sidebar, select the router.
7. On the main screen, enter the Ownership Proof Token if needed, then click the "Start Pairing" button under the "Pairing" section, and complete pair setup.
8. In the main screen, click the "Discover" button in the "Summary" section.
9. In the left sidebar, select "Managed Network Enable".
10. On the main screen, navigate to the "Prepare and Execute Timed Write [Unsigned Integer]" section.
11. Enter a value of "1" in the text field and click the "Timed Write" button.
12. In the left sidebar, select "Network Client Profile Control".
13. Under the "Write Options" section, ensure "Write With Response" is enabled.
14. On the main screen, navigate to the "Prepare and Execute Timed Write [TLV8 Packet]" section.
15. Click on the "Build TLV" button.
16. In the "Network Client Profile Control" TLV builder, set the following parameters:
    - Set the "Operation" to "Add".
    - Leave the "Client Profile Identifier" field blank.
    - Enter "1" into the "Client Group Identifier" text field.
    - Enter "31 31 31 31 31 31 31 31" (HEX converted value of "11111111") into the Credential Data field.
    - Select "PSK" for the Credential Type.
    - Enter "0101010200" into the WAN Firewall Config field.
    - Enter "010100" into the LAN Firewall Config field.
17. Click the "Add" button to add rule to the TLV builder table.
18. Click the "Build TLV" button in the bottom right corner.
19. In the "Prepare and Execute Timed Write [TLV8 Packet]" section, click the "Timed Write" button to write this TLV to the router. Note: If the trace view shows a warning, this can be ignored.
20. On the ADK terminal, run `#sudo raspi-config` and follow Network Options to join the network created in step 1, using "11111111" as the Wi-Fi password and enter the same network name as the one created in step 1.

21. Verify the ADK accessory connects to the network successfully.
22. Connect another Mac client on the WAN side of the router using a switch.
23. On the ADK accessory terminal run iperf in server mode with command `iperf –s –p 2234 –i 1`.
24. On the Mac connected on the WAN side, open Terminal and run iperf to connect to the server running on the ADK accessory using: `iperf –c <ip address of ADK accessory> –p 2234 –i 1`.
25. Verify iperf traffic does not transmit traffic and fail.

TCSR012 **Verify the behavior of a WAN TCP rule.**

Applies to HomeKit-enabled Wi-Fi router accessories. Perform this test case using HAT, the accessory app, a Raspberry Pi running an ADK IP accessory, a Mac client running iperf, and the test network declaration JSON files found on the MFi Portal.

1. Use the accessory's app to setup and create a new wireless network.
2. Setup an ADK device as a HomeKit accessory.
3. On the ADK terminal, run `#sudo raspi-config` and follow Network Options to join the wireless network created in step 1.
4. Join the newly created network on a Mac running HAT.
5. Pair and discover router accessory.
6. In the left sidebar, select "Managed Network Enable".
7. On the main screen, navigate to the "Prepare and Execute Timed Write [Unsigned Integer]" section.
8. Enter a value of "1" in the text field and click the "Timed Write" button.
9. In the left sidebar, select the name of the accessory that was added in Step 3.
10. On the main accessory server view, navigate to the "Network Declarations" pane and select the "Select File" button next to "Import JSON".
11. Browse and select TCSR012.json file, downloaded from the MFi Portal.
12. Select "Apply rules" button in the "Network Declarations" section.
13. On the next screen, ensure "No Restrictions" is selected at the import rules line at top of page, and press "Apply" to apply those rules to the router for this accessory.
14. Verify the rules are correctly reflected on the router by selecting the "Show Configured" button.
15. On ADK terminal, try to access any web server by running the command `#curl –v <web address>`, Make sure the https server is connected.
16. Setup a Mac client on the WAN side of the router with any WAN IP. Open the terminal on the Mac client, run a TCP server on any valid port in the range 0-65535 with the command `#iperf –s –p <Port Number> –i 1 –u`
17. On the ADK terminal run `iperf –c <Mac WAN IP as above> –p <Port Number as above> –i 1 –u`, verify the traffic is successfully sent without being blocked.

18. On ADK terminal, run the command `#ping <Mac WAN IP as above>`, make sure ping command is success with host reachable.

19. On ADK accessory, run a TCP server on any valid port in the range 0-65535 with the command `#iperf –s –p <Port Number> –i 1`

20. On the Mac running HAT, open terminal and run the command `#iperf –c < Ip of the ADK client running iperf s as above> –p <Port Number as above> –i 1` Make sure the TCP traffic goes through and not get blocked.

21. On the Mac, run a UDP server on any valid port in the range 0-65535 with the command `#iperf –s –p <Port Number> –i 1` -u

22. On ADK client, open terminal and run the command iperf -c < Ip of the Mac client running iperf s as above> -p <Port Number as above> -i 1 -u Make sure the UDP traffic goes through and not get blocked.

23. On Mac running HAT client, open terminal and run the command `# ping <ip of the ADK accessory>`. Verify the ping success with host reachable.

24. In the left sidebar, select the name of the accessory that was added in step 3.

25. On the main screen under heading "Network Declarations" choose button "Select File" next to "Import JSON"

26. Browse and select the TCSR012.json file, downloaded from the MFi Portal.

27. Select "Apply rules" button in the "Network Declarations" section.

28. On the next screen, ensure "Auto" is selected at the top of page, and select "Apply" to apply those rules to the router for this accessory.

29. Verify the rules are correctly reflected on the router by selecting the "Show Configured" button.

30. SSH into ADK accessory using ADK user and IP or Hostname.

31. On ADK terminal, run the command `#curl –v https://www.engadget.com`, Make sure the https server is connected.

32. On ADK terminal, run the command `#curl –v www.facebook.com`, make sure the http server is not connected.

33. On ADK terminal, run the command `#curl –v https://www.gizmodo.com`, Make sure the https server is connected.

34. On ADK terminal, run the command `#curl –v https://www.techcrunch.com`, Make sure the https server is connected

35. On ADK terminal, run the command `#curl –v https://aws.amazon.com`, Make sure the https server is connected

36. On ADK terminal, run the command `#curl –v <any website except mentioned above>`, Make sure the server is not connected

37. (Note: IP address in the below teststeps assume that the current network subnet is 192.168.20.x, If it is different in your setup please modify it according to your setup. Also same ip address needs to be modified in the provided json file aswell.)Setup a Mac client on the WAN side of the router with WAN IP 192.168.20.10. Open the terminal on the Mac client and run `iperf –s –p 8000 –i 1`.

38. On the ADK terminal, run `iperf -c 192.168.20.10 -p 8000 -i 1`, verify the traffic is successfully sent without being blocked.
39. Terminate the iperf session on both Mac client and ADK accessory.
40. Setup a Mac client on the WAN side of the router with WAN IP 192.168.20.11. Open the terminal on the Mac client and run `iperf -s -p 8101 -i 1`.
41. On the ADK terminal, run `iperf -c 192.168.20.11 -p 8101 -i 1`, verify the traffic is successfully sent without being blocked.
42. Terminate the iperf session on both Mac client and ADK accessory.
43. Setup a Mac client on the WAN side of the router with WAN IP 192.168.20.11. Open the terminal on the Mac client and run `iperf -s -p 8180 -i 1`.
44. On the ADK terminal run `iperf -c 192.168.20.11 -p 8180 -i 1`, verify the traffic is successfully sent without being blocked.
45. Terminate the iperf session on both Mac client and ADK accessory.
46. Setup a Mac client on the WAN side of the router with WAN IP 192.168.20.16. Open the terminal on the Mac client and run `iperf -s -p 8200 -i 1`.
47. On the ADK terminal, run `iperf -c 192.168.20.16 -p 8200 -i 1`, verify the traffic is successfully sent without being blocked.
48. Terminate the iperf session on both Mac client and ADK accessory.
49. Setup a Mac client on the WAN side of the router with WAN IP 192.168.20.20. Open the terminal on the Mac client and run `iperf -s -p 8200 -i 1`.
50. On the ADK terminal run `iperf -c 192.168.20.20 -p 8200 -i 1`, verify the traffic is successfully sent without being blocked.
51. Terminate the iperf session on both Mac client and ADK accessory.
52. Setup a Mac client on the WAN side of the router with WAN IP 192.168.20.25. Open the terminal on the Mac client and run `iperf -s -p 8300 -i 1`.
53. On the ADK terminal run `iperf -c 192.168.20.25 -p 8300 -i 1`, verify the traffic is successfully sent without being blocked.
54. Terminate the iperf session on both Mac client and ADK accessory.
55. Setup a Mac client on the WAN side of the router with WAN IP 192.168.20.27. Open the terminal on the Mac client and run `iperf -s -p 8300 -i 1`.
56. On the ADK terminal run `iperf -c 192.168.20.27 -p 8300 -i 1`, verify the traffic is successfully sent without being blocked.
57. Terminate the iperf session on both Mac client and ADK accessory.
58. Setup a Mac client on the WAN side of the router with WAN IP 192.168.20.30. Open the terminal on the Mac client and run `iperf -s -p 11000 -i 1`.
59. On the ADK terminal run `iperf -c 192.168.20.30 -p 11000 -i 1`, verify the traffic is successfully sent without being blocked.

60. Terminate the iperf session on both Mac client and ADK accessory.
61. Setup a Mac client on the WAN side of the router with any other WAN IP except mentioned above. Open the terminal on the Mac client and run a TCP server on any valid port in the range 0-65535 with the command "iperf -s -p <Port Number> -i 1"
62. On the ADK terminal run `iperf –c <Mac WAN IP as above> –p <Port Number as above> –i 1`, verify the traffic is blocked.
63. Terminate the iperf session on both Mac client and ADK accessory.
64. In the left sidebar, select the name of the accessory that was added in Step 3.
65. On the main accessory server view, navigate to the "Network Declarations" pane and select the "Select File" button next to "Import JSON".
66. Browse and select TCSR012.json file, downloaded from the MFi Portal.
67. Select "Apply rules" button in the "Network Declarations" section.
68. On the next screen, ensure "No Restrictions" is selected at the import rules line at top of page, and press "Apply" to apply those rules to the router for this accessory.
69. Verify the rules are correctly reflected on the router by selecting the "Show Configured" button.
70. On ADK terminal, try to access any web server by running the command `#curl –v <web address>`, Make sure the https server is connected.
71. Setup a Mac client on the WAN side of the router with any WAN IP. Open the terminal on the Mac client, run a UDP server on any valid port in the range 0-65535 with the command `#iperf –s –p <Port Number> –i 1 –u`
72. On the ADK terminal run `iperf –c <Mac WAN IP as above> –p <Port Number as above> –i 1 –u`, verify the traffic is successfully sent without being blocked.
73. On ADK terminal, run the command `#ping <Mac WAN IP as above>`, make sure ping command is success with host reachable.
74. On ADK accessory, run a TCP server on any valid port in the range 0-65535 with the command `#iperf –s –p <Port Number> –i 1`
75. On the Mac running HAT, open terminal and run the `iperf –c < Ip of the ADK client running iperf s as above> –p <Port Number as above> –i 1`, Make sure the TCP traffic goes through and not get blocked.
76. On the Mac, run a UDP server on any valid port in the range 0-65535 with the command `#iperf –s –p <Port Number> –i 1` **-u**
77. On ADK client, open terminal and run the `iperf –c < Ip of the Mac client running iperf s as above> –p <Port Number as above> –i 1 –u`,Make sure the UDP traffic goes through and not get blocked.
78. On Mac running HAT client, open terminal and run the `ping <ip of the ADK accessory>`. Verify the ping success with host reachable.
79. Setup another ADK device as a HomeKit accessory, or use the same ADK accessory as above.

80. Connect the ADK device to the router's ethernet port (Make sure ADK accessory is only connected using the ethernet port and not connected wirelessly).

81. If the router supports ethernet, repeat the steps from step 9 to step 78 on the ADK accessory connected using ethernet.

82. Setup another ADK device as a HomeKit accessory, or use the same ADK accessory as above.

83. Connect the ADK device to the network.

84. Ensure the ADK is connected to one of the satellites, and not the main gateway. (To achieve this, a shielding cloth can be used to cover the main gateway to cause satellite to have a stronger signal, or satellite and iOS device can be moved to a more isolated area where the signal strength from the satellite to the ADK device is strongest.)

85. Repeat the steps from step 9 to step 78 on the ADK accessory connected wirelessly to the mesh node.

TCSR013 **Verify the behavior of a WAN UDP rule.**

Applies to HomeKit-enabled Wi-Fi router accessories. Perform this test case using HAT, the accessory app, a Raspberry Pi running an ADK IP accessory, a Mac client running iperf, and the test network declaration JSON files found on the MFi Portal.

1. Use the accessory's app to setup and create a new wireless network.
2. Setup an ADK device as a HomeKit accessory.
3. On the ADK terminal, run `#sudo raspi-config` and follow Network Options to join the network created in step 1.
4. Join the newly created network on a Mac running HAT.
5. Pair and discover router accessory.
6. In the left sidebar, select "Managed Network Enable".
7. On the main screen, navigate to the "Prepare and Execute Timed Write [Unsigned Integer]" section.
8. Enter a value of "1" in the text field and click the "Timed Write" button.
9. In the left sidebar, select the name of the accessory that was added in step 3.
10. On the main screen under heading "Network Declarations" choose button "Select File" next to "Import JSON"
11. Browse and select the TCSR013.json file, downloaded from the MFi Portal.
12. Select "Apply rules" button in the "Network Declarations" section.
13. On the next screen, ensure "No Restrictions" is selected at the import rules line at top of page, and press "Apply" to apply those rules to the router for this accessory.
14. Verify the rules are correctly reflected on the router by selecting the "Show Configured" button.
15. On ADK terminal, try to access any web server by running the command `#curl –v <web address>`, Make sure the https server is connected.

16. Setup a Mac client on the WAN side of the router with any WAN IP. Open the terminal on the Mac client, run a TCP server on any valid port in the range 0-65535 with the command `#iperf –s –p <Port Number> –i 1 –u`
17. On the ADK terminal run `#iperf –c <Mac WAN IP as above> –p <Port Number as above> –i 1 –u`, verify the traffic is successfully sent without being blocked.
18. On ADK terminal, run the command `#ping <Mac WAN IP as above>`, make sure ping command is success with host reachable.
19. On ADK accessory, run a TCP server on any valid port in the range 0-65535 with the command `#iperf –s –p <Port Number> –i 1`
20. On the Mac running HAT, open terminal and run the command `#iperf –c < Ip of the ADK client running iperf s as above> –p <Port Number as above> –i 1` Make sure the TCP traffic goes through and not get blocked.
21. On the Mac, run a UDP server on any valid port in the range 0-65535 with the command `#iperf –s –p <Port Number> –i 1` -u
22. On ADK client, open terminal and run the command iperf -c < Ip of the Mac client running iperf s as above> -p <Port Number as above> -i 1 -u Make sure the UDP traffic goes through and not get blocked.
23. On Mac running HAT client, open terminal and run the command `# ping <ip of the ADK accessory>`. Verify the ping success with host reachable.
24. In the left sidebar, select the name of the accessory that was added in step 3.
25. On the main screen under heading "Network Declarations" choose button "Select File" next to "Import JSON"
26. Browse and select the TCSR013.json file, downloaded from the MFi Portal.
27. Select "Apply rules" button in the "Network Declarations" section.
28. On the next screen, ensure "Auto" is selected at the top of page, and select "Apply" to apply those rules to the router for this accessory.
29. Verify the rules are correctly reflected on the router by selecting the "Show Configured" button.
30. Setup a Mac client on the WAN side of the router with WAN IP 192.168.20.10. Open the terminal on the Mac client and run "iperf -s -p 8000 -i 1 -u"
31. (Note: IP address in the below teststeps assume that the current network subnet is 192.168.20.x, If it is different in your setup please modify it according to your setup. Also same ip address needs to be modified in the provided json file aswell.)On the ADK terminal run "iperf -c 192.168.20.10 -p 8000 -i 1 -u", verify the traffic is successfully sent without being blocked.
32. On the ADK terminal run `iperf –c 192.168.20.10 –p 8000 –i 1 –u`, verify the traffic is successfully sent without being blocked.
33. Terminate the iperf session on both Mac client and ADK accessory.
34. Setup a Mac client on the WAN side of the router with WAN IP 192.168.20.11. Open the terminal on the Mac client and run `iperf –s –p 8101 –i 1 –u`.

35. On the ADK terminal run `iperf -c 192.168.20.11 -p 8101 -i 1 -u`, verify the traffic is successfully sent without being blocked.
36. Terminate the iperf session on both Mac client and ADK accessory.
37. Setup a Mac client on the WAN side of the router with WAN IP 192.168.20.11. Open the terminal on the Mac client and run `iperf -s -p 8180 -i 1 -u`.
38. On the ADK terminal run `iperf -c 192.168.20.11 -p 8180 -i 1 -u`, verify the traffic is successfully sent without being blocked.
39. Terminate the iperf session on both Mac client and ADK accessory.
40. Setup a Mac client on the WAN side of the router with WAN IP 192.168.20.16. Open the terminal on the Mac client and run `iperf -s -p 8200 -i 1 -u`.
41. On the ADK terminal run `iperf -c 192.168.20.16 -p 8200 -i 1 -u`, verify the traffic is successfully sent without being blocked.
42. Terminate the iperf session on both Mac client and ADK accessory.
43. Setup a Mac client on the WAN side of the router with WAN IP 192.168.20.20. Open the terminal on the Mac client and run `iperf -s -p 8200 -i 1 -u`.
44. On the ADK terminal run `iperf -c 192.168.20.20 -p 8200 -i 1 -u`, verify the traffic is successfully sent without being blocked.
45. Terminate the iperf session on both Mac client and ADK accessory.
46. Setup a Mac client on the WAN side of the router with WAN IP 192.168.20.25. Open the terminal on the Mac client and run `iperf -s -p 8300 -i 1 -u`.
47. On the ADK terminal run `iperf -c 192.168.20.25 -p 8300 -i 1 -u`, verify the traffic is successfully sent without being blocked.
48. Terminate the iperf session on both Mac client and ADK accessory.
49. Setup a Mac client on the WAN side of the router with WAN IP 192.168.20.27. Open the terminal on the Mac client and run `iperf -s -p 8300 -i 1 -u`.
50. On the ADK terminal run `iperf -c 192.168.20.27 -p 8300 -i 1 -u`, verify the traffic is successfully sent without being blocked.
51. Terminate the iperf session on both Mac client and ADK accessory.
52. Setup a Mac client on the WAN side of the router with WAN IP 192.168.20.30. Open the terminal on the Mac client and run `iperf -s -p 11000 -i 1 -u`.
53. On the ADK terminal run `iperf -c 192.168.20.30 -p 11000 -i 1 -u`, verify the traffic is successfully sent without being blocked.
54. Terminate the iperf session on both Mac client and ADK accessory.
55. Setup a Mac client on the WAN side of the router with any other WAN IP except mentioned above. Open the terminal on the Mac client and run a UDP server on any valid port in the range 0-65535 with the command "iperf -s -p <Port Number> -i 1 -u"

56. On the ADK terminal run `iperf –c <Mac WAN IP as above> –p <Port Number as above> –i 1 –u`, verify the traffic is blocked.
57. Terminate the iperf session on both Mac client and ADK accessory.
58. In the left sidebar, select the name of the accessory that was added in step 3.
59. On the main screen under heading “Network Declarations” choose button “Select File” next to “Import JSON”
60. Browse and select the TCSR013.json file, downloaded from the MFi Portal.
61. Select “Apply rules” button in the “Network Declarations” section.
62. On the next screen, ensure “No Restrictions” is selected at the import rules line at top of page, and press “Apply” to apply those rules to the router for this accessory.
63. Verify the rules are correctly reflected on the router by selecting the “Show Configured” button.
64. On ADK terminal, try to access any web server by running the command `#curl –v <web address>`, Make sure the https server is connected.
65. Setup a Mac client on the WAN side of the router with any WAN IP. Open the terminal on the Mac client, run a TCP server on any valid port in the range 0-65535 with the command `#iperf –s –p <Port Number> –i 1 –u`
66. On the ADK terminal run `iperf –c <Mac WAN IP as above> –p <Port Number as above> –i 1 –u`, verify the traffic is successfully sent without being blocked.
67. On ADK terminal, run the command `ping <Mac WAN IP as above`, make sure ping command is success with host reachable.
68. On ADK accessory, run a TCP server on any valid port in the range 0-65535 with the command `#iperf –s –p <Port Number> –i 1`
69. On the Mac running HAT, open terminal and run `iperf –c < Ip of the ADK client running iperf s as above> –p <Port Number as above> –i 1`, Make sure the TCP traffic goes through and not get blocked.
70. On the Mac, run a UDP server on any valid port in the range 0-65535 with this `iperf –s –p <Port Number> –i 1 –u`.
71. On ADK client, open terminal and run the `iperf –c < Ip of the Mac client running iperf s as above> –p <Port Number as above> –i 1 –u`. Make sure the UDP traffic goes through and not get blocked.
72. On Mac running HAT client, open terminal and run the command `ping <ip of the ADK accessory>`. Verify the ping success with host reachable.
73. Setup another ADK device as a HomeKit accessory, or use the same ADK accessory as above.
74. Connect the ADK device to the router’s ethernet port (Make sure ADK accessory is only connected using the ethernet port and not connected wirelessly).
75. If the router supports ethernet, repeat the steps from step 9 to step 71 on the ADK accessory connected using ethernet.

76. Setup another ADK device as a HomeKit accessory, or use the same ADK accessory as above.
77. Connect the ADK device to the network.
78. Ensure the ADK is connected to one of the satellites, and not the main gateway. (To achieve this, a shielding cloth can be used to cover the main gateway to cause satellite to have a stronger signal, or satellite and iOS device can be moved to a more isolated area where the signal strength from the satellite to the ADK device is strongest.)
79. Repeat the steps from step 9 to step 71 on the ADK accessory connected wirelessly to the mesh node.

TCSR014 **Verify the behavior of a WAN ICMP rule.**

Applies to HomeKit-enabled Wi-Fi router accessories. Perform this test case using HAT, the accessory app, a Raspberry Pi running an ADK IP accessory, and the test network declaration JSON files found on the MFi Portal.

1. Use the accessory's app to setup and create a new wireless network.
2. Setup an ADK device as a HomeKit accessory.
3. On the ADK terminal run `#sudo raspi-config` and follow Network Options to join the network created in step 1.
4. Join the newly created network on a Mac running HAT.
5. Pair and Discover router accessory
6. In the left sidebar, select the name of the accessory that was added in step 3.
7. On the main screen under heading "Network Declarations" choose button "Select File" next to "Import JSON"
8. Browse and select the TCSR014.json file, downloaded from the MFi Portal.
9. Select "Apply rules" button in the "Network Declarations" section.
10. On the next screen, ensure "No Restrictions" is selected at the import rules line at top of page, and press "Apply" to apply those rules to the router for this accessory.
11. Verify the rules are correctly reflected on the router by selecting the "Show Configured" button.
12. On ADK terminal, try to access any web server by running the command `#curl –v <web address>`, Make sure the https server is connected.
13. Setup a Mac client on the WAN side of the router with any WAN IP. Open the terminal on the Mac client, run a TCP server on any valid port in the range 0-65535 with the command `#iperf –s –p <Port Number> –i 1 –u`
14. On the ADK terminal run `#iperf –c <Mac WAN IP as above> –p <Port Number as above> –i 1 –u`, verify the traffic is successfully sent without being blocked.
15. On ADK terminal, run the command `#ping <Mac WAN IP as above>`, make sure ping command is success with host reachable.

16. On ADK accessory, run a TCP server on any valid port in the range 0-65535 with the command `#iperf –s –p <Port Number> –i 1`
17. On the Mac running HAT, open terminal and run the command `#iperf –c < Ip of the ADK client running iperf s as above> –p <Port Number as above> –i 1` Make sure the TCP traffic goes through and not get blocked.
18. On the Mac, run a UDP server on any valid port in the range 0-65535 with the command `#iperf –s –p <Port Number> –i 1` -u
19. On ADK client, open terminal and run the command iperf -c < Ip of the Mac client running iperf s as above> -p <Port Number as above> -i 1 -u Make sure the UDP traffic goes through and not get blocked.
20. On Mac running HAT client, open terminal and run the command `# ping <ip of the ADK accessory>`. Verify the ping success with host reachable.
21. In the left sidebar, select the name of the accessory that was added in step 3.
22. On the main screen under heading "Network Declarations" choose button "Select File" next to "Import JSON"
23. Browse and select the TCSR014.json file, downloaded from the MFi Portal.
24. Select "Apply rules" button in the "Network Declarations" section.
25. On the next screen, ensure "Auto" is selected at the top of page, and press "Apply" to apply those rules to the router for this accessory.
26. Verify the rules are correctly reflected on the router by selecting the "Show Configured" button.
27. On ADK terminal, run the command `#ping www.engadget.com`, make sure ping command is success with host reachable.
28. On ADK terminal, run the command `#ping www.cnn.com`, make sure ping command is success with host reachable.
29. On ADK terminal, run the command `#ping www.gizmodo.com`, make sure ping command is success with host reachable.
30. On ADK terminal, run the command `#ping www.techcrunch.com`, make sure ping command is success with host reachable.
31. On ADK terminal, run the command `#ping store.amazon.com`, make sure ping command is success with host reachable.
32. On ADK terminal, run the command `#ping <any website except mentioned above>`, make sure ping command fails with host unreachable.
33. (Note: IP address in the below teststeps assume that the current network subnet is 192.168.20.x, If it is different in your setup please modify it according to your setup. Also same ip address needs to be modified in the provided json file aswell.)Setup a Mac client on the WAN side of the router with IP 192.168.20.10.
34. On ADK terminal, run the command `#ping 192.168.20.10`, make sure ping command is success with host reachable.
35. Setup a Mac client on the WAN side of the router with WAN IP 192.168.20.20

36. On ADK terminal, run the command `#ping 192.168.20.20`, make sure ping command fails with host unreachable.
37. In the left sidebar, select the name of the accessory that was added in step 3.
38. On the main screen under heading “Network Declarations” choose button “Select File” next to “Import JSON”
39. Browse and select the TCSR014.json file, downloaded from the MFi Portal.
40. Select “Apply rules” button in the “Network Declarations” section.
41. On the next screen, ensure “No Restrictions” is selected at the import rules line at top of page, and press “Apply” to apply those rules to the router for this accessory.
42. Verify the rules are correctly reflected on the router by selecting the “Show Configured” button.
43. On ADK terminal, try to access any web server by running the command `#curl –v <web address>` , Make sure the https server is connected.
44. Setup a Mac client on the WAN side of the router with any WAN IP. Open the terminal on the Mac client, run a TCP server on any valid port in the range 0-65535 with the command `#iperf –s –p <Port Number> -i 1 –u`
45. On the ADK terminal run `#iperf –c <Mac WAN IP as above> –p <Port Number as above> –i 1 –u`, verify the traffic is successfully sent without being blocked.
46. On ADK terminal, run the command `#ping <Mac WAN IP as above>`, make sure ping command is success with host reachable.
47. On ADK accessory, run a TCP server on any valid port in the range 0-65535 with the command `#iperf –s –p <Port Number> –i 1`
48. On the Mac running HAT, open terminal and run `#iperf –c < Ip of the ADK client running iperf s as above> –p <Port Number as above> –i 1`, Make sure the TCP traffic goes through and not get blocked.
49. On the Mac, run a UDP server on any valid port in the range 0-65535 with the command `#iperf –s –p <Port Number> –i 1 –u`.
50. On ADK client, open terminal and run the `#iperf –c < Ip of the Mac client running iperf s as above> –p <Port Number as above> –i 1 –u`, Make sure the UDP traffic goes through and not get blocked.
51. On Mac running HAT client, open terminal and run the command `#ping <ip of the ADK accessory>`. Verify the ping success with host reachable.
52. Setup another ADK device as a HomeKit accessory, or use the same ADK accessory as above.
53. Connect the ADK device to the router’s ethernet port (Make sure ADK accessory is only connected using the ethernet port and not conencted wirelessly).
54. (If Applicable)Repeat the steps from step 6 to step 51 on the ADK accessory connected using ethernet.
55. Setup another ADK device as a HomeKit accessory, or use the same ADK accessory as above..
56. Connect the ADK device to the network.

57. Ensure the ADK is connected to one of the satellites, and not the main gateway. (To achieve this, a shielding cloth can be used to cover the main gateway to cause satellite to have a stronger signal, or satellite and iOS device can be moved to a more isolated area where the signal strength from the satellite to the ADK device is strongest.)

58. Repeat the steps from step 6 to step 51 on the ADK accessory connected wirelessly to the mesh node.

TCSR015 **Verify the behavior of LAN UDP Inbound Multicast Bridging rule.**

Applies to HomeKit-enabled Wi-Fi router accessories. Perform this test case using HAT, the accessory app, a Raspberry Pi running an ADK IP accessory, a Mac client running iperf, and the test network declaration JSON files found on the MFi Portal.

1. Use the accessory's app to setup and create a new wireless network.
2. Setup an ADK device as a HomeKit accessory.
3. On the ADK terminal run `#sudo raspi-config` and follow Network Options to join the network created in step 1.
4. Join the newly created network on a Mac running HAT.
5. Pair and discover router accessory.
6. In the left sidebar, select "Managed Network Enable".
7. On the main screen, navigate to the "Prepare and Execute Timed Write [Unsigned Integer]" section.
8. Enter a value of "1" in the text field and click the "Timed Write" button.
9. In the left sidebar, select the name of the accessory that was added in step 3.
10. On the main accessory server view, navigate to the "Network Declarations" pane and select the "Select File" button next to "Import JSON".
11. Browse and select the TCSR015.json file, downloaded from the MFi Portal.
12. Select "Apply rules" button in the "Network Declarations" section.
13. On the next screen, ensure "No Restrictions" is selected at the import rules line at top of page, and press "Apply" to apply those rules to the router for this accessory.
14. Verify the rules are correctly reflected on the router by selecting the "Show Configured" button.
15. On ADK terminal, try to access any web server by running the command `#curl –v <web address>`, Make sure the https server is connected.
16. Setup a Mac client on the WAN side of the router with any WAN IP. Open the terminal on the Mac client, run a TCP server on any valid port in the range 0-65535 with the command `#iperf –s –p <Port Number> –i 1 –u`
17. On the ADK terminal run `#iperf –c <Mac WAN IP as above> –p <Port Number as above> –i 1 –u`, verify the traffic is successfully sent without being blocked.
18. On ADK terminal, run the command `#ping <Mac WAN IP as above>`, make sure ping command is success with host reachable.

19. On ADK accessory, run a TCP server on any valid port in the range 0-65535 with the command `#iperf –s –p <Port Number> –i 1`.
20. On the Mac running HAT, open terminal and run the command `#iperf –c < Ip of the ADK client running iperf s as above> –p <Port Number as above> –i 1`, Make sure the TCP traffic goes through and not get blocked.
21. On the Mac, run a UDP server on any valid port in the range 0-65535 with the command `#iperf –s –p <Port Number> –i 1 –u`.
22. On ADK client, open terminal and run the command `#iperf –c < Ip of the Mac client running iperf s as above> –p <Port Number as above> –i 1 –u`. Make sure the UDP traffic goes through and not get blocked.
23. On Mac running HAT client, open terminal and run the command `#ping <ip of the ADK accessory>`. Verify the ping success with host reachable.
24. In the left sidebar, select the name of the accessory that was added in step 3.
25. On the main screen under heading "Network Declarations" choose button "Select File" next to "Import JSON"
26. Browse and select the TCSR015.json file, downloaded from the MFi Portal.
27. Select "Apply rules" button in the "Network Declarations" section.
28. On the next screen, ensure "Auto" is selected at the import rules line at top of page, and press "Apply" to apply those rules to the router for this accessory.
29. Verify the rules are correctly reflected on the router by selecting the "Show Configured" button.
30. On the ADK accessory, Login into the terminal of the ADK accessory and start a udp multicast server as below.
31. Run command on the ADK accessory: `#iperf –s –B 239.255.1.3 –u –f m –i 5 –p 6000`, which will start a udp multicast server.
32. On the Mac client running the HAT tool run the `#iperf –c 239.255.1.3 –u –b 990m –f m –i 5 –t 30 –p 6000` command which will start the multicast client.
33. Verify the multicast traffic is successfully sent without being blocked.
34. On the ADK accessory, Login into the terminal of the ADK accessory and start a udp multicast server on any other valid multicast IP address except above and any valid port in the range 0-65535 as below.
35. Run command on the ADK accessory: `#iperf –s –B <Multicast IP address from above> –u –f m –i 5 –p <Port Number as above>`, which will start a udp multicast server.
36. On the Mac client running the HAT tool run the `#iperf –c <Multicast IP address from above> –u –b 990m –f m –i 5 –t 30 –p <Port Number as above> command which will start the multicast client.`
37. Verify the multicast traffic is being blocked.
38. On the Mac client running HAT, start a udp multicast server as below.
39. Run the command on the Mac client `#iperf –s –B 239.255.1.3 –u –f m –i 5 –p 6000`. This will start a udp multicast server.

40. On the ADK accessory `#iperf -c 239.255.1.3 -u -b 990m -f m -i 5 -t 30 -p 6000.` command which will start the multicast client.
41. Verify the multicast traffic is being blocked.
42. On ADK terminal start a TCP iperf server by running the following command on ADK terminal: `#iperf -s -p 2234 -i 1`
43. On the Mac which is running HAT, open the Terminal application.
44. Use iperf to connect to the server started in above step using the following command: `#iperf -c <ip address of ADK accessory> -p 2234 -i 1`.
45. Verify connection is not made in iperf and traffic is not sent.
46. In the left sidebar, select the name of the accessory that was added in step 3.
47. On the main accessory server view, navigate to the "Network Declarations" pane and select the "Select File" button next to "Import JSON".
48. Browse and select the TCSR015.json file, downloaded from the MFi Portal.
49. Select "Apply rules" button in the "Network Declarations" section.
50. On the next screen, ensure "No Restrictions" is selected at the import rules line at top of page, and press "Apply" to apply those rules to the router for this accessory.
51. Verify the rules are correctly reflected on the router by selecting the "Show Configured" button.
52. On ADK terminal, try to access any web server by running the command `#curl -v <web address>` , Make sure the https server is connected.
53. Setup a Mac client on the WAN side of the router with any WAN IP. Open the terminal on the Mac client, run a TCP server on any valid port in the range 0-65535 with the command `#iperf -s -p <Port Number> -i 1 -u`.
54. On the ADK terminal run `#iperf -c <Mac WAN IP as above> -p <Port Number as above> -i 1 -u`, verify the traffic is successfully sent without being blocked.
55. On ADK terminal, run the command `#ping <Mac WAN IP as above>`, make sure ping command is success with host reachable.
56. On ADK accessory, run a TCP server on any valid port in the range 0-65535 with the command `#iperf -s -p <Port Number> -i 1`.
57. On the Mac running HAT, open terminal and run the command `#iperf -c < Ip of the ADK client running iperf s as above> -p <Port Number as above> -i 1`. Make sure the TCP traffic goes through and not get blocked.
58. On the Mac, run a UDP server on any valid port in the range 0-65535 with the command `#iperf -s -p <Port Number> -i 1 -u`.
59. On ADK client, open terminal and run the command `#iperf -c < Ip of the Mac client running iperf s as above> -p <Port Number as above> -i 1 -u`. Make sure the UDP traffic goes through and not get blocked.
60. On Mac running HAT client, open terminal and run the command `#ping <ip of the ADK accessory>`. Verify the ping success with host reachable

61. Setup another ADK device as a HomeKit accessory, or use the same ADK accessory as above..
62. Connect the ADK device to the router's ethernet port (Make sure ADK accessory is only connected using the ethernet port and not conencted wirelessly).
63. (If Applicable)Repeat the steps from step 9 to step 60 on the ADK accessory connected using Ethernet.
64. Setup another ADK device as a HomeKit accessory, or use the same ADK accessory as above.
65. Connect the ADK device to the network.
66. Ensure the ADK is connected to one of the satellites, and not the main gateway. (To achieve this, a shielding cloth can be used to cover the main gateway to cause satellite to have a stronger signal, or satellite and iOS device can be moved to a more isolated area where the signal strength from the satellite to the ADK device is strongest.)
67. Repeat the steps from step 9 to step 60 on the ADK accessory connected wirelessly to the mesh node.

TCSR016 **Verify the behavior of LAN UDP Outbound Multicast Bridging rule.**

Applies to HomeKit-enabled Wi-Fi router accessories. Perform this test case using HAT, the accessory app, a Raspberry Pi running an ADK IP accessory, a Mac client running iperf, and the test network declaration JSON files found on the MFi Portal.

1. Use the accessory's app to setup and create a new wireless network.
2. Setup an ADK device as a HomeKit accessory.
3. On the ADK terminal run `#sudo raspi-config` and follow Network Options to join the network created in step 1.
4. Join the newly created network on a Mac running HAT.
5. Pair and Discover router accessory
6. In the left sidebar, select "Managed Network Enable".
7. On the main screen, navigate to the "Prepare and Execute Timed Write [Unsigned Integer]" section.
8. Enter a value of "1" in the text field and click the "Timed Write" button.
9. In the left sidebar, select the name of the accessory that was added in step 3.
10. On the main accessory server view, navigate to the "Network Declarations" pane and select the "Select File" button next to "Import JSON".
11. Browse and select the TCSR016.json file, downloaded from the MFi Portal.
12. Select "Apply rules" button in the "Network Declarations" section.
13. On the next screen, ensure "No Restrictions" is selected at the import rules line at top of page, and press "Apply" to apply those rules to the router for this accessory.
14. Verify the rules are correctly reflected on the router by selecting the "Show Configured" button.
15. On ADK terminal, try to access any web server by running the command `#curl –v <web address>`, Make sure the https server is connected.

16. Setup a Mac client on the WAN side of the router with any WAN IP. Open the terminal on the Mac client, run a TCP server on any valid port in the range 0-65535 with the command `#iperf –s –p <Port Number> –i 1 –u`
17. On the ADK terminal run `#iperf –c <Mac WAN IP as above> –p <Port Number as above> –i 1 –u`, verify the traffic is successfully sent without being blocked.
18. On ADK terminal, run the command `#ping <Mac WAN IP as above>`, make sure ping command is success with host reachable.
19. On ADK accessory, run a TCP server on any valid port in the range 0-65535 with the command `#iperf –s –p <Port Number> –i 1`
20. On the Mac running HAT, open terminal and run the command `#iperf –c < Ip of the ADK client running iperf s as above> –p <Port Number as above> –i 1` Make sure the TCP traffic goes through and not get blocked.
21. On the Mac, run a UDP server on any valid port in the range 0-65535 with the command `#iperf –s –p <Port Number> –i 1` -u
22. On ADK client, open terminal and run the command iperf -c < Ip of the Mac client running iperf s as above> -p <Port Number as above> -i 1 -u Make sure the UDP traffic goes through and not get blocked.
23. On Mac running HAT client, open terminal and run the command `# ping <ip of the ADK accessory>`. Verify the ping success with host reachable.
24. In the left sidebar, select the name of the accessory that was added in step 3.
25. On the main screen under heading "Network Declarations" choose button "Select File" next to "Import JSON"
26. Browse and select the TCSR016.json file, downloaded from the MFi Portal.
27. Select "Apply rules" button in the "Network Declarations" section.
28. On the next screen, ensure "Auto" is selected at the import rules line at top of page, and press "Apply" to apply those rules to the router for this accessory.
29. Verify the rules are correctly reflected on the router by selecting the "Show Configured" button.
30. On the Mac client running HAT, start a udp multicast server as below.
31. Run the command on the Mac client `#iperf –s –B 239.255.1.3 –u –f m –i 5 –p 6000`. This will start a udp multicast server.
32. On the ADK accessory run `iperf –c 239.255.1.3 –u –b 990m –f m –i 5 –t 30 –p 6000` command which will start the multicast client.
33. Verify the multicast traffic is successfully sent without being blocked.
34. On the Mac client running HAT, start a udp multicast server on any other valid multicast IP address except above and any valid port in the range 0-65535 as below.
35. Run the command on the Mac client #iperf -s -B <Multicast IP address from above> -u -f m -i 5 -p <Port Number as above>. This will start a udp multicast server.

36. On the ADK accessory. run the command `#iperf –c <Multicast IP address from above> –u –b 990m –f m –i 5 –t 30 –p <Port Number as above>` , which will start the multicast client.
37. Verify the multicast traffic is being blocked.
38. On the ADK accessory, Login into the terminal of the ADK accessory and start a udp multicast server as below.
39. Run command on the ADK accessory: `#iperf –s –B 239.255.1.3 –u –f m –i 5 –p 6000`, which will start a udp multicast server.
40. On the Mac client running the HAT tool run the iperf -c 239.255.1.3 -u -b 990m -f m -i 5 -t 30 -p 6000 command which will start the multicast client.
41. Verify the multicast traffic is being blocked.
42. On ADK terminal, start a TCP iperf server by running the following command on ADK terminal: `iperf –s –p 2234 –i 1`
43. On the Mac which is running HAT, open Terminal application.
44. Use iperf to connect to the server started in above step using the following command: `iperf –c <ip address of ADK accessory> –p 2234 –i 1`.
45. Verify connection is not made in iperf and traffic is not sent.
46. In the left sidebar, select the name of the accessory that was added in step 3.
47. On the main accessory server view, navigate to the “Network Declarations” pane and select the “Select File” button next to “Import JSON”.
48. Browse and select the TCSR016.json file, downloaded from the MFi Portal.
49. Select “Apply rules” button in the “Network Declarations” section.
50. On the next screen, ensure “No Restrictions” is selected at the import rules line at top of page, and press “Apply” to apply those rules to the router for this accessory.
51. Verify the rules are correctly reflected on the router by selecting the “Show Configured” button.
52. On ADK terminal, try to access any web server by running the command `#curl –v <web address>`, Make sure the https server is connected.
53. Setup a Mac client on the WAN side of the router with any WAN IP. Open the terminal on the Mac client, run a TCP server on any valid port in the range 0-65535 with the command `#iperf –s –p <Port Number> –i 1 –u`
54. On the ADK terminal run `#iperf –c <Mac WAN IP as above> –p <Port Number as above> –i 1 –u`, verify the traffic is successfully sent without being blocked.
55. On ADK terminal, run the command `#ping <Mac WAN IP as above>`, make sure ping command is success with host reachable.
56. On ADK accessory, run a TCP server on any valid port in the range 0-65535 with the command `#iperf –s –p <Port Number> –i 1`

57. On the Mac running HAT, open terminal and run the command `#iperf –c < Ip of the ADK client running iperf s as above> –p <Port Number as above> –i 1` Make sure the TCP traffic goes through and not get blocked.
58. On the Mac, run a UDP server on any valid port in the range 0-65535 with the command `#iperf –s –p <Port Number> –i 1` -u
59. On ADK client, open terminal and run the command iperf -c < Ip of the Mac client running iperf s as above> -p <Port Number as above> -i 1 -u Make sure the UDP traffic goes through and not get blocked.
60. On Mac running HAT client, open terminal and run the command `# ping <ip of the ADK accessory>`. Verify the ping success with host reachable.
61. Setup another ADK device as a HomeKit accessory, or use the same ADK accessory as above.
62. Connect the ADK device to the router's ethernet port (Make sure ADK accessory is only connected using the ethernet port and not connected wirelessly).
63. (If Applicable)Repeat the steps from step 9 to step 60 on the ADK accessory connected using Ethernet.
64. Setup another ADK device as a HomeKit accessory, or use the same ADK accessory as above.
65. Connect the ADK device to the network.
66. Ensure the ADK is connected to one of the satellites, and not the main gateway. (To achieve this, a shielding cloth can be used to cover the main gateway to cause satellite to have a stronger signal, or satellite and iOS device can be moved to a more isolated area where the signal strength from the satellite to the ADK device is strongest.)
67. Repeat the steps from step 9 to step 60 on the ADK accessory connected wirelessly to the mesh node.

TCSR017 **Verify the behavior of LAN TCP Inbound Static port rule with individual port.**

Applies to HomeKit-enabled Wi-Fi router accessories. Perform this test case using HAT, the accessory app, a Raspberry Pi running an ADK IP accessory, a Mac client running iperf, and the test network declaration JSON files found on the MFi Portal.

1. Use the accessory's app to setup and create a new wireless network.
2. Setup an ADK device as a HomeKit accessory.
3. On the ADK terminal, run `#sudo raspi–config` and follow Network Options to join the network created in step 1.
4. Join the newly created network on a Mac running HAT.
5. Pair and discover router accessory.
6. In the left sidebar, select "Managed Network Enable".
7. On the main screen, navigate to the "Prepare and Execute Timed Write [Unsigned Integer]" section.
8. Enter a value of "1" in the text field and click the "Timed Write" button.
9. In the left sidebar, select the name of the accessory that was added in step 3.

10. On the main screen under heading “Network Declarations” choose button “Select File” next to “Import JSON”
11. Browse and select the TCSR017.json file, downloaded from the MFi Portal.
12. Select “Apply rules” button in the “Network Declarations” section.
13. On the next screen, ensure “No Restrictions” is selected at the import rules line at top of page, and press “Apply” to apply those rules to the router for this accessory.
14. Verify the rules are correctly reflected on the router by selecting the “Show Configured” button.
15. On ADK terminal, try to access any web server by running the command `#curl –v <web address>`, Make sure the https server is connected.
16. Setup a Mac client on the WAN side of the router with any WAN IP. Open the terminal on the Mac client, run a TCP server on any valid port in the range 0-65535 with the command `#iperf –s –p <Port Number> –i 1 –u`
17. On the ADK terminal run `#iperf –c <Mac WAN IP as above> –p <Port Number as above> –i 1 –u`, verify the traffic is successfully sent without being blocked.
18. On ADK terminal, run the command `#ping <Mac WAN IP as above>`, make sure ping command is success with host reachable.
19. On ADK accessory, run a TCP server on any valid port in the range 0-65535 with the command `#iperf –s –p <Port Number> –i 1`
20. On the Mac running HAT, open terminal and run the command `#iperf –c < Ip of the ADK client running iperf s as above> –p <Port Number as above> –i 1` Make sure the TCP traffic goes through and not get blocked.
21. On the Mac, run a UDP server on any valid port in the range 0-65535 with the command `#iperf –s –p <Port Number> –i 1 -u`
22. On ADK client, open terminal and run the command iperf -c < Ip of the Mac client running iperf s as above> -p <Port Number as above> -i 1 -u Make sure the UDP traffic goes through and not get blocked.
23. On Mac running HAT client, open terminal and run the command `# ping <ip of the ADK accessory>`. Verify the ping success with host reachable.
24. In the left sidebar, select the name of the accessory that was added in step 3.
25. On the main screen under heading “Network Declarations” choose button “Select File” next to “Import JSON”
26. Browse and select the TCSR017.json file, downloaded from the MFi Portal.
27. Select “Apply rules” button in the “Network Declarations” section.
28. On the next screen, ensure “Auto” is selected at the import rules line at top of page, and press “Apply” to apply those rules to the router for this accessory.
29. Verify the rules are correctly reflected on the router by selecting the “Show Configured” button.
30. On ADK accessory, run a TCP server on a specific ports with command: `iperf –s –p 9000 –i 1`

31. On the Mac running HAT, open terminal and run the command `iperf –c <ip of the ADK client running iperf s as above> –p 9000 –i 1`. Make sure the TCP traffic goes through and not get blocked.
32. On ADK accessory, run a TCP server on any valid port in the range 0-65535 except 9000 with the command `#iperf –s –p <Port Number> –i 1`
33. On the Mac running HAT, open terminal and run the command `#iperf –c < Ip of the ADK client running iperf s as above> –p <Port Number as above> –i 1` Make sure the TCP traffic does not go through and gets blocked.
34. On Mac client, run a TCP server on any valid port in the range 0-65535 with the command `#iperf –s –p <Port Number> –i 1`
35. On ADK accessory, open terminal and run the command iperf -c < Ip of the Mac client running iperf s as above> -p <Port Number as above> -i 1 Make sure the TCP traffic does not go through and gets blocked.
36. In the left sidebar, select the name of the accessory that was added in step 3.
37. On the main screen under heading "Network Declarations" choose button "Select File" next to "Import JSON"
38. Browse and select the TCSR017.json file, downloaded from the MFi Portal.
39. Select "Apply rules" button in the "Network Declarations" section.
40. On the next screen, ensure "No Restrictions" is selected at the import rules line at top of page, and press "Apply" to apply those rules to the router for this accessory.
41. Verify the rules are correctly reflected on the router by selecting the "Show Configured" button.
42. On ADK terminal, try to access any web server by running the command `#curl –v <web address>`, Make sure the https server is connected.
43. Setup a Mac client on the WAN side of the router with any WAN IP. Open the terminal on the Mac client, run a TCP server on any valid port in the range 0-65535 with the command `#iperf –s –p <Port Number> –i 1 –u`
44. On the ADK terminal run `#iperf –c <Mac WAN IP as above> –p <Port Number as above> –i 1 –u`, verify the traffic is successfully sent without being blocked.
45. On ADK terminal, run the command `#ping <Mac WAN IP as above>`, make sure ping command is success with host reachable.
46. On ADK accessory, run a TCP server on any valid port in the range 0-65535 with the command `#iperf –s –p <Port Number> –i 1`
47. On the Mac running HAT, open terminal and run the command `#iperf –c < Ip of the ADK client running iperf s as above> –p <Port Number as above> –i 1` Make sure the TCP traffic goes through and not get blocked.
48. On the Mac, run a UDP server on any valid port in the range 0-65535 with the command `#iperf –s –p <Port Number> –i 1` -u
49. On ADK client, open terminal and run the command iperf -c < Ip of the Mac client running iperf s as above> -p <Port Number as above> -i 1 -u Make sure the UDP traffic goes through and not get blocked.

50. On Mac running HAT client, open terminal and run the command `# ping <ip of the ADK accessory>`. Verify the ping success with host reachable.
51. Setup another ADK device as a HomeKit accessory, or use the same ADK accessory as above.
52. Connect the ADK device to the router's ethernet port (Make sure ADK accessory is only connected using the ethernet port and not conencted wirelessly).
53. If the router supports ethernet, repeat the steps from step 9 to step 50 on the ADK accessory connected using ethernet.
54. Setup another ADK device as a HomeKit accessory, or use the same ADK accessory as above.
55. Connect the ADK device to the network.
56. Ensure the ADK is connected to one of the satellites, and not the main gateway. (To achieve this, a shielding cloth can be used to cover the main gateway to cause satellite to have a stronger signal, or satellite and iOS device can be moved to a more isolated area where the signal strength from the satellite to the ADK device is strongest.)
57. Repeat the steps from step 9 to step 50 on the ADK accessory connected wirelessly to the mesh node.

TCSR018 **Verify the behavior of LAN TCP inbound Static port rule with port ranges.**

Applies to HomeKit-enabled Wi-Fi router accessories. Perform this test case using HAT, the accessory app, a Raspberry Pi running an ADK IP accessory, a Mac client running iperf, and the test network declaration JSON files found on the MFi Portal.

1. Use the accessory's app to setup and create a new wireless network.
2. Setup an ADK device as a HomeKit accessory.
3. On the ADK terminal run `#sudo raspi-config` and follow Network Options to join the network created in step 1.
4. Join the newly created network on a Mac running HAT.
5. Pair and discover router accessory.
6. In the left sidebar, select "Managed Network Enable".
7. On the main screen, navigate to the "Prepare and Execute Timed Write [Unsigned Integer]" section.
8. Enter a value of "1" in the text field and click the "Timed Write" button.
9. In the left sidebar, select the name of the accessory that was added in step 3.
10. On the main screen under heading "Network Declarations" choose button "Select File" next to "Import JSON"
11. Browse and select the TCSR018.json file, downloaded from the MFi Portal.
12. Select "Apply rules" button in the "Network Declarations" section.
13. On the next screen, ensure "No Restrictions" is selected at the import rules line at top of page, and press "Apply" to apply those rules to the router for this accessory.

14. Verify the rules are correctly reflected on the router by selecting the “Show Configured” button.
15. On ADK terminal, try to access any web server by running the command `#curl –v <web address>`, Make sure the https server is connected.
16. Setup a Mac client on the WAN side of the router with any WAN IP. Open the terminal on the Mac client, run a TCP server on any valid port in the range 0-65535 with the command `#iperf –s –p <Port Number> –i 1 –u`
17. On the ADK terminal run `#iperf –c <Mac WAN IP as above> –p <Port Number as above> –i 1 –u`, verify the traffic is successfully sent without being blocked.
18. On ADK terminal, run the command `#ping <Mac WAN IP as above>`, make sure ping command is success with host reachable.
19. On ADK accessory, run a TCP server on any valid port in the range 0-65535 with the command `#iperf –s –p <Port Number> –i 1`
20. On the Mac running HAT, open terminal and run the command `#iperf –c < Ip of the ADK client running iperf s as above> –p <Port Number as above> –i 1` Make sure the TCP traffic goes through and not get blocked.
21. On the Mac, run a UDP server on any valid port in the range 0-65535 with the command `#iperf –s –p <Port Number> –i 1 -u`
22. On ADK client, open terminal and run the command iperf -c < Ip of the Mac client running iperf s as above> -p <Port Number as above> -i 1 -u Make sure the UDP traffic goes through and not get blocked.
23. On Mac running HAT client, open terminal and run the command `# ping <ip of the ADK accessory>`. Verify the ping success with host reachable.
24. In the left sidebar, select the name of the accessory that was added in step 3.
25. On the main screen under heading “Network Declarations” choose button “Select File” next to “Import JSON”
26. Browse and select the TCSR018.json file, downloaded from the MFi Portal.
27. Select “Apply rules” button in the “Network Declarations” section.
28. On the next screen, ensure “Auto” is selected at the import rules line at top of page, and press “Apply” to apply those rules to the router for this accessory.
29. Verify the rules are correctly reflected on the router by selecting the “Show Configured” button.
30. On the ADK accessory, run a TCP server on a specific ports with command: `iperf –s –p 9000 –i 1`.
31. On Mac running HAT client, open terminal and run the command `iperf –c <ip of the ADK client running iperf s as above> –p 9000 –i 1`. Make sure the TCP traffic goes through and not get blocked.
32. On ADK accessory Run a TCP server on a specific ports with command: `iperf –s –p 9050 –i 1`.
33. On Mac running HAT client, open terminal and run the command `iperf –c <ip of the ADK client running iperf s as above> –p 9000 –i 1`. Make sure the TCP traffic goes through and not get blocked.

34. On ADK accessory Run a TCP server on a specific ports with command: `iperf –s –p 9100 –i 1`.
35. On Mac running HAT client, open terminal and run the command `iperf –c <ip of the ADK client running iperf s as above> –p 9100 –i 1`. Make sure the TCP traffic goes through and not get blocked.
36. On ADK accessory, run a TCP server on any valid port in the range 0-65535 except 9000-9100 with the command `#iperf –s –p <Port Number> –i 1`
37. On the Mac running HAT, open terminal and run the command `#iperf –c < Ip of the ADK client running iperf s as above> –p <Port Number as above> –i 1` Make sure the TCP traffic does not go through and gets blocked.
38. On Mac client, run a TCP server on any valid port in the range 0-65535 with the command `#iperf –s –p <Port Number> –i 1`
39. On ADK accessory, open terminal and run the command iperf -c < Ip of the Mac client running iperf s as above> -p <Port Number as above> -i 1 Make sure the TCP traffic does not go through and gets blocked.
40. In the left sidebar, select the name of the accessory that was added in step 3.
41. On the main screen under heading "Network Declarations" choose button "Select File" next to "Import JSON"
42. Browse and select the TCSR018.json file, downloaded from the MFi Portal.
43. Select "Apply rules" button in the "Network Declarations" section.
44. On the next screen, ensure "No Restrictions" is selected at the import rules line at top of page, and press "Apply" to apply those rules to the router for this accessory.
45. Verify the rules are correctly reflected on the router by selecting the "Show Configured" button.
46. On ADK terminal, try to access any web server by running the command `#curl –v <web address>`, Make sure the https server is connected.
47. Setup a Mac client on the WAN side of the router with any WAN IP. Open the terminal on the Mac client, run a TCP server on any valid port in the range 0-65535 with the command `#iperf –s –p <Port Number> –i 1 –u`
48. On the ADK terminal run `#iperf –c <Mac WAN IP as above> –p <Port Number as above> –i 1 –u`, verify the traffic is successfully sent without being blocked.
49. On ADK terminal, run the command `#ping <Mac WAN IP as above>`, make sure ping command is success with host reachable.
50. On ADK accessory, run a TCP server on any valid port in the range 0-65535 with the command `#iperf –s –p <Port Number> –i 1`
51. On the Mac running HAT, open terminal and run the command `#iperf –c < Ip of the ADK client running iperf s as above> –p <Port Number as above> –i 1` Make sure the TCP traffic goes through and not get blocked.
52. On the Mac, run a UDP server on any valid port in the range 0-65535 with the command `#iperf –s –p <Port Number> –i 1` -u

53. On ADK client, open terminal and run the command iperf -c < Ip of the Mac client running iperf s as above> -p <Port Number as above> -i 1 -u Make sure the UDP traffic goes through and not get blocked.
54. On Mac running HAT client, open terminal and run the command `# ping <ip of the ADK accessory>`. Verify the ping success with host reachable.
55. Setup another ADK device as a HomeKit accessory, or use the same ADK accessory as above.
56. Connect the ADK device to the router's ethernet port (Make sure ADK accessory is only connected using the ethernet port and not connected wirelessly).
57. If the router supports ethernet Repeat the steps from step 9 to step 54 on the ADK accessory connected using Ethernet.
58. Setup another ADK device as a HomeKit accessory, or use the same ADK accessory as above.
59. Connect the ADK device to the network.
60. Ensure the ADK is connected to one of the satellites, and not the main gateway. (To achieve this, a shielding cloth can be used to cover the main gateway to cause satellite to have a stronger signal, or satellite and iOS device can be moved to a more isolated area where the signal strength from the satellite to the ADK device is strongest.)
61. Repeat the steps from step 9 to step 54 on the ADK accessory connected wirelessly to the mesh node.

TCSR019 **Verify the behavior of LAN UDP Inbound Static port rule with individual port.**

Applies to HomeKit-enabled Wi-Fi router accessories. Perform this test case using HAT, the accessory app, a Raspberry Pi running an ADK IP accessory, a Mac client running iperf, and the test network declaration JSON files found on the MFi Portal.

1. Use the accessory's app to setup and create a new wireless network.
2. Setup an ADK device as a HomeKit accessory.
3. On the ADK terminal run `#sudo raspi-config` and follow Network Options to join the network created in step 1.
4. Pair and discover router accessory.
5. In the left sidebar, select "Managed Network Enable".
6. On the main screen, navigate to the "Prepare and Execute Timed Write [Unsigned Integer]" section.
7. Enter a value of "1" in the text field and click the "Timed Write" button.
8. In the left sidebar, select the name of the accessory that was added in step 3.
9. On the main screen under heading "Network Declarations" choose button "Select File" next to "Import JSON"
10. Browse and select the TCSR019.json file, downloaded from the MFi Portal.
11. Select "Apply rules" button in the "Network Declarations" section.
12. On the next screen, ensure "No Restrictions" is selected at the import rules line at top of page, and press "Apply" to apply those rules to the router for this accessory.

13. Verify the rules are correctly reflected on the router by selecting the “Show Configured” button.
14. On ADK terminal, try to access any web server by running the command `#curl –v <web address>`, Make sure the https server is connected.
15. Setup a Mac client on the WAN side of the router with any WAN IP. Open the terminal on the Mac client, run a TCP server on any valid port in the range 0-65535 with the command `#iperf –s –p <Port Number> –i 1 –u`
16. On the ADK terminal run `#iperf –c <Mac WAN IP as above> –p <Port Number as above> –i 1 –u`, verify the traffic is successfully sent without being blocked.
17. On ADK terminal, run the command `#ping <Mac WAN IP as above>`, make sure ping command is success with host reachable.
18. On ADK accessory, run a TCP server on any valid port in the range 0-65535 with the command `#iperf –s –p <Port Number> –i 1`
19. On the Mac running HAT, open terminal and run the command `#iperf –c < Ip of the ADK client running iperf s as above> –p <Port Number as above> –i 1` Make sure the TCP traffic goes through and not get blocked.
20. On the Mac, run a UDP server on any valid port in the range 0-65535 with the command `#iperf –s –p <Port Number> –i 1 -u`
21. On ADK client, open terminal and run the command iperf -c < Ip of the Mac client running iperf s as above> -p <Port Number as above> -i 1 -u Make sure the UDP traffic goes through and not get blocked.
22. On Mac running HAT client, open terminal and run the command `# ping <ip of the ADK accessory>`. Verify the ping success with host reachable.
23. In the left sidebar, select the name of the accessory that was added in step 3.
24. On the main screen under heading “Network Declarations” choose button “Select File” next to “Import JSON”
25. Browse and select the TCSR019.json file, downloaded from the MFi Portal.
26. Select “Apply rules” button in the “Network Declarations” section.
27. On the next screen, ensure “Auto” is selected at the import rules line at top of page, and press “Apply” to apply those rules to the router for this accessory.
28. Verify the rules are correctly reflected on the router by selecting the “Show Configured” button.
29. On ADK accessory, run a UDP server on a specific ports with command: `iperf –s –p 9000 –i 1 –u`.
30. On Mac running HAT, open terminal and run the command `iperf –c <ip of the ADK client running iperf s as above> –p 9000 –i 1 –u`. Make sure the UDP traffic goes through and not get blocked.
31. On ADK accessory, run a UDP server on any valid port in the range 0-65535 except 9000 with the command `#iperf –s –p <Port Number> –i 1 -u`
32. On the Mac running HAT, open terminal and run the command `#iperf –c < Ip of the ADK client running iperf s as above> –p <Port Number as above> –i 1` -u. Make sure the UDP

traffic does not go through and gets blocked.

33. On Mac client, run a UDP server on any valid port in the range 0-65535 with the command `#iperf –s –p <Port Number> –i 1 -u`
34. On ADK accessory, open terminal and run the command iperf -c < Ip of the Mac client running iperf s as above> -p <Port Number as above> -i 1 -u. Make sure the UDP traffic does not go through and gets blocked.
35. In the left sidebar, select the name of the accessory that was added in step 3.
36. On the main screen under heading "Network Declarations" choose button "Select File" next to "Import JSON"
37. Browse and select the TCSR019.json file, downloaded from the MFi Portal.
38. Select "Apply rules" button in the "Network Declarations" section.
39. On the next screen, ensure "No Restrictions" is selected at the import rules line at top of page, and press "Apply" to apply those rules to the router for this accessory.
40. Verify the rules are correctly reflected on the router by selecting the "Show Configured" button.
41. On ADK terminal, try to access any web server by running the command `#curl –v <web address>`, Make sure the https server is connected.
42. Setup a Mac client on the WAN side of the router with any WAN IP. Open the terminal on the Mac client, run a TCP server on any valid port in the range 0-65535 with the command `#iperf –s –p <Port Number> –i 1 –u`
43. On the ADK terminal run `#iperf –c <Mac WAN IP as above> –p <Port Number as above> –i 1 –u`, verify the traffic is successfully sent without being blocked.
44. On ADK terminal, run the command `#ping <Mac WAN IP as above>`, make sure ping command is success with host reachable.
45. On ADK accessory, run a TCP server on any valid port in the range 0-65535 with the command `#iperf –s –p <Port Number> –i 1`
46. On the Mac running HAT, open terminal and run the command `#iperf –c < Ip of the ADK client running iperf s as above> –p <Port Number as above> –i 1` Make sure the TCP traffic goes through and not get blocked.
47. On the Mac, run a UDP server on any valid port in the range 0-65535 with the command `#iperf –s –p <Port Number> –i 1 -u`
48. On ADK client, open terminal and run the command iperf -c < Ip of the Mac client running iperf s as above> -p <Port Number as above> -i 1 -u Make sure the UDP traffic goes through and not get blocked.
49. On Mac running HAT client, open terminal and run the command `# ping <ip of the ADK accessory>`. Verify the ping success with host reachable.
50. Setup another ADK device as a HomeKit accessory, or use the same ADK accessory as above.
51. Connect the ADK device to the router's ethernet port (Make sure ADK accessory is only connected using the ethernet port and not connected wirelessly).

52. If the router supports ethernet Repeat the steps from step 8 to step 49 on the ADK accessory connected using Ethernet.

53. Setup another ADK device as a HomeKit accessory, or use the same ADK accessory as above.

54. Connect the ADK device to the network.

55. Ensure the ADK is connected to one of the satellites, and not the main gateway. (To achieve this, a shielding cloth can be used to cover the main gateway to cause satellite to have a stronger signal, or satellite and iOS device can be moved to a more isolated area where the signal strength from the satellite to the ADK device is strongest.)

56. Repeat the steps from step 8 to step 49 on the ADK accessory connected wirelessly to the mesh node.

TCSR020 **Verify the behavior of LAN UDP Inbound Static port rule with port ranges.**

Applies to HomeKit-enabled Wi-Fi router accessories. Perform this test case using HAT, the accessory app, a Raspberry Pi running an ADK IP accessory, a Mac client running iperf, and the test network declaration JSON files found on the MFi Portal.

1. Use the accessory's app to setup and create a new wireless network.
2. Setup an ADK device as a HomeKit accessory.
3. On the ADK terminal run `#sudo raspi-config` and follow Network Options to join the network created in step 1.
4. Join the newly created network on a Mac running HAT.
5. Pair and discover router accessory.
6. In the left sidebar, select "Managed Network Enable".
7. On the main screen, navigate to the "Prepare and Execute Timed Write [Unsigned Integer]" section.
8. Enter a value of "1" in the text field and click the "Timed Write" button.
9. In the left sidebar, select the name of the accessory that was added in step 3.
10. On the main accessory server view, navigate to the "Network Declarations" pane and select the "Select File" button next to "Import JSON".
11. Browse and select the TCSR020.json file, downloaded from the MFi Portal.
12. Select "Apply rules" button in the "Network Declarations" section.
13. On the next screen, ensure "No Restrictions" is selected at the import rules line at top of page, and press "Apply" to apply those rules to the router for this accessory.
14. Verify the rules are correctly reflected on the router by selecting the "Show Configured" button.
15. On ADK terminal, try to access any web server by running the command `#curl –v <web address>` , Make sure the https server is connected.
16. Setup a Mac client on the WAN side of the router with any WAN IP. Open the terminal on the Mac client, run a TCP server on any valid port in the range 0-65535 with the command `#iperf –s –p <Port Number> –i 1 –u`.

17. On the ADK terminal run `#iperf –c <Mac WAN IP as above> –p <Port Number as above> –i 1 –u`, verify the traffic is successfully sent without being blocked.
18. On ADK terminal, run the command `#ping <Mac WAN IP as above>`, make sure ping command is success with host reachable.
19. On ADK accessory, run a TCP server on any valid port in the range 0-65535 with the command `#iperf –s –p <Port Number> –i 1`.
20. On the Mac running HAT, open terminal and run the command `#iperf –c < Ip of the ADK client running iperf s as above> –p <Port Number as above> –i 1`. Make sure the TCP traffic goes through and not get blocked.
21. On the Mac, run a UDP server on any valid port in the range 0-65535 with the command `#iperf –s –p <Port Number> –i 1 –u`.
22. On ADK client, open terminal and run the command `#iperf –c < Ip of the Mac client running iperf s as above> –p <Port Number as above> –i 1 –u`. Make sure the UDP traffic goes through and not get blocked.
23. On Mac running HAT client, open terminal and run the command `#ping <ip of the ADK accessory>`.. Verify the ping success with host reachable.
24. In the left sidebar, select the name of the accessory that was added in step 3.
25. On the main screen under heading "Network Declarations" choose button "Select File" next to "Import JSON"
26. Browse and select the TCSR020.json file, downloaded from the MFi Portal.
27. Select "Apply rules" button in the "Network Declarations" section.
28. On the next screen, ensure "Auto" is selected at the import rules line at top of page, and press "Apply" to apply those rules to the router for this accessory.
29. Verify the rules are correctly reflected on the router by selecting the "Show Configured" button.
30. On ADK accessory, run a TCP server on a specific ports with command: `#iperf –s –p 9050 –i 1 –u`.
31. On a Mac running HAT, open terminal and run the command `iperf –c <ip of the Mac client running iperf s as above> –p 9050 –i 1 –u`. Make sure the UDP traffic goes through and not get blocked.
32. On ADK accessory Run a TCP server on a specific ports with command `iperf –s –p 9100 –i 1 –u`.
33. On a Mac running HAT, open terminal and run the command `iperf –c <ip of the Mac client running iperf s as above> –p 9100 –i 1 –u`. Make sure the UDP traffic goes through and not get blocked.
34. On ADK accessory, run a UDP server on any valid port in the range 0-65535 except 9000-9100 with the command `#iperf –s –p <Port Number> –i 1 –u`.
35. On the Mac running HAT, open terminal and run the command iperf -c < Ip of the ADK client running `#iperf s as above> –p <Port Number as above> –i 1 –u`. Make sure the UDP traffic does not go through and gets blocked.

36. On Mac client, run a UDP server on any valid port in the range 0-65535 with the command `#iperf –s –p <Port Number> –i 1 –u`.

37. On ADK accessory, open terminal and run the command `#iperf –c < Ip of the Mac client running iperf s as above> –p <Port Number as above> –i 1 –u`. Make sure the UDP traffic does not go through and gets blocked.

38. In the left sidebar, select the name of the accessory that was added in step 3.

39. On the main accessory server view, navigate to the "Network Declarations" pane and select the "Select File" button next to "Import JSON".

40. Browse and select the TCSR020.json file, downloaded from the MFi Portal.

41. Select "Apply rules" button in the "Network Declarations" section.

42. On the next screen, ensure "No Restrictions" is selected at the import rules line at top of page, and press "Apply" to apply those rules to the router for this accessory.

43. Verify the rules are correctly reflected on the router by selecting the "Show Configured" button.

44. On ADK terminal, try to access any web server by running the command `#curl –v <web address>` , Make sure the https server is connected.

45. Setup a Mac client on the WAN side of the router with any WAN IP. Open the terminal on the Mac client, run a TCP server on any valid port in the range 0-65535 with the command `#iperf –s –p <Port Number> –i 1 –u`.

46. On the ADK terminal run `#iperf –c <Mac WAN IP as above> –p <Port Number as above> –i 1 –u`, verify the traffic is successfully sent without being blocked.

47. On ADK terminal, run the command `#ping <Mac WAN IP as above>`, make sure ping command is success with host reachable.

48. On ADK accessory, run a TCP server on any valid port in the range 0-65535 with the command `#iperf –s –p <Port Number> –i 1`.

49. On the Mac running HAT, open terminal and run the command `#iperf –c < Ip of the ADK client running iperf s as above> –p <Port Number as above> –i 1`. Make sure the TCP traffic goes through and not get blocked.

50. On the Mac, run a UDP server on any valid port in the range 0-65535 with the command `#iperf –s –p <Port Number> –i 1 –u`.

51. On ADK client, open terminal and run the command `#iperf –c < Ip of the Mac client running iperf s as above> –p <Port Number as above> –i 1 –u`. Make sure the UDP traffic goes through and not get blocked.

52. On Mac running HAT client, open terminal and run the command `#ping <ip of the ADK accessory>`. Verify the ping success with host reachable.

53. Setup another ADK device as a HomeKit accessory, or use the same ADK accessory as above.

54. Connect the ADK device to the router's ethernet port (Make sure ADK accessory is only connected using the ethernet port and not connected wirelessly).

55. If the router supports ethernet Repeat the steps from step 9 to step 52 on the ADK accessory connected using Ethernet.
56. Setup another ADK device as a HomeKit accessory, or use the same ADK accessory as above.
57. Connect the ADK device to the network.
58. Ensure the ADK is connected to one of the satellites, and not the main gateway. (To achieve this, a shielding cloth can be used to cover the main gateway to cause satellite to have a stronger signal, or satellite and iOS device can be moved to a more isolated area where the signal strength from the satellite to the ADK device is strongest.)
59. Repeat the steps from step 9 to step 52 on the ADK accessory connected wirelessly to the mesh node.

TCSR021 **Verify the behavior of LAN ICMP Inbound rule.**

Applies to HomeKit-enabled Wi-Fi router accessories. Perform this test case using HAT, the accessory app, a Raspberry Pi running an ADK IP accessory, and the test network declaration JSON files found on the MFi Portal.

1. Use the accessory's app to setup and create a new wireless network.
2. Setup an ADK device as a HomeKit accessory.
3. On the ADK terminal, run `#sudo raspi-config` and follow Network Options to join the network created in step 1.
4. Join the newly created network on a Mac running HAT.
5. Pair and discover router accessory.
6. In the left sidebar, select "Managed Network Enable".
7. On the main screen, navigate to the "Prepare and Execute Timed Write [Unsigned Integer]" section.
8. Enter a value of "1" in the text field and click the "Timed Write" button.
9. In the left sidebar, select the name of the accessory that was added in step 3.
10. On the main screen under heading "Network Declarations" choose button "Select File" next to "Import JSON"
11. Browse and select the TCSR021.json file, downloaded from the MFi Portal.
12. Select "Apply rules" button in the "Network Declarations" section.
13. On the next screen, ensure "No Restrictions" is selected at the import rules line at top of page, and press "Apply" to apply those rules to the router for this accessory.
14. Verify the rules are correctly reflected on the router by selecting the "Show Configured" button.
15. On ADK terminal, try to access any web server by running the command `#curl -v <web address>` , Make sure the https server is connected.
16. Setup a Mac client on the WAN side of the router with any WAN IP. Open the terminal on the Mac client, run a TCP server on any valid port in the range 0-65535 with the command `#iperf -s -p <Port Number> -i 1 -u`.

17. On the ADK terminal run `#iperf –c <Mac WAN IP as above> –p <Port Number as above> –i 1 –u`, verify the traffic is successfully sent without being blocked.
18. On ADK terminal, run the command `#ping <Mac WAN IP as above>`, make sure ping command is success with host reachable.
19. On ADK accessory, run a TCP server on any valid port in the range 0-65535 with the command `#iperf –s –p <Port Number> –i 1`.
20. On the Mac running HAT, open terminal and run the command `#iperf –c < Ip of the ADK client running iperf s as above> –p <Port Number as above> –i 1`. Make sure the TCP traffic goes through and not get blocked.
21. On the Mac, run a UDP server on any valid port in the range 0-65535 with the command `#iperf –s –p <Port Number> –i 1 –u`.
22. On ADK client, open terminal and run the command iperf -c < Ip of the Mac client running `#iperf s as above> –p <Port Number as above> –i 1 –u`. Make sure the UDP traffic goes through and not get blocked.
23. On Mac running HAT client, open terminal and run the command `# ping <ip of the ADK accessory>`. Verify the ping success with host reachable.
24. In the left sidebar, select the name of the accessory that was added in step 3.
25. On the main screen under heading "Network Declarations" choose button "Select File" next to "Import JSON"
26. Browse and select the TCSR021.json file, downloaded from the MFi Portal.
27. Select "Apply rules" button in the "Network Declarations" section.
28. On the next screen, ensure "Auto" is selected at the import rules line at top of page, and press "Apply" to apply those rules to the router for this accessory.
29. Verify the rules are correctly reflected on the router by selecting the "Show Configured" button.
30. On Mac running HAT client, open terminal and run the command `# ping <ip of the ADK accessory>`. Verify the ping success with host reachable.
31. On ADK accessory client, open terminal and run the command `#ping <ip of the Mac running HAT tool >`. Verify the ping fails with host unreachable.
32. On ADK terminal start a TCP iperf server by running the following command on ADK terminal: `iperf –s –p 2234 –i 1`.
33. On the Mac which is running HAT, open the Terminal application.
34. Use iperf to connect to the server started in above step using the following command: `iperf –c <ip address of ADK accessory> –p 2234 –i 1`. Verify traffic is blocked.
35. In the left sidebar, select the name of the accessory that was added in step 3.
36. On the main screen under heading "Network Declarations" choose button "Select File" next to "Import JSON"
37. Browse and select the TCSR021.json file, downloaded from the MFi Portal.

38. Select “Apply rules” button in the “Network Declarations” section.
39. On the next screen, ensure “No Restrictions” is selected at the import rules line at top of page, and press “Apply” to apply those rules to the router for this accessory.
40. Verify the rules are correctly reflected on the router by selecting the “Show Configured” button.
41. On ADK terminal, try to access any web server by running the command `#curl –v <web address>` , Make sure the https server is connected.
42. Setup a Mac client on the WAN side of the router with any WAN IP. Open the terminal on the Mac client, run a TCP server on any valid port in the range 0-65535 with the command `#iperf –s –p <Port Number> –i 1 –u`
43. On the ADK terminal run `#iperf –c <Mac WAN IP as above> –p <Port Number as above> –i 1 –u`, verify the traffic is successfully sent without being blocked.
44. On ADK terminal, run the command `#ping <Mac WAN IP as above>`, make sure ping command is success with host reachable.
45. On ADK accessory, run a TCP server on any valid port in the range 0-65535 with the command `#iperf –s –p <Port Number> –i 1`.
46. On the Mac running HAT, open terminal and run the command `#iperf –c < Ip of the ADK client running iperf s as above> –p <Port Number as above> –i 1`. Make sure the TCP traffic goes through and not get blocked.
47. On the Mac, run a UDP server on any valid port in the range 0-65535 with the command `#iperf –s –p <Port Number> –i 1 –u`.
48. On ADK client, open terminal and run the command iperf -c < Ip of the Mac client running iperf s as above> -p <Port Number as above> -i 1 -u Make sure the UDP traffic goes through and not get blocked.
49. On Mac running HAT client, open terminal and run the command `# ping <ip of the ADK accessory>`. Verify the ping success with host reachable.
50. Setup another ADK device as a HomeKit accessory, or use the same ADK accessory as above..
51. Connect the ADK device to the router’s ethernet port (Make sure ADK accessory is only connected using the ethernet port and not conencted wirelessly).
52. If the router supports ethernet Repeat the steps from step 9 to step 49 on the ADK accessory connected using Ethernet.
53. Setup another ADK device as a HomeKit accessory, or use the same ADK accessory as above.
54. Connect the ADK device to the network.
55. Ensure the ADK is connected to one of the satellites, and not the main gateway. (To achieve this, a shielding cloth can be used to cover the main gateway to cause satellite to have a stronger signal, or satellite and iOS device can be moved to a more isolated area where the signal strength from the satellite to the ADK device is strongest.)
56. Repeat the steps from step 9 to step 49 on the ADK accessory connected wirelessly to the mesh node.

TCSR022 **Verify the behavior of LAN TCP Outbound Static port rule with individual port.**

Applies to HomeKit-enabled Wi-Fi router accessories. Perform this test case using HAT, the accessory app, a Raspberry Pi running an ADK IP accessory, and the test network declaration JSON files found on the MFi Portal.

1. Use the accessory's app to setup and create a new wireless network.
2. Setup an ADK device as a HomeKit accessory.
3. On the ADK terminal run `#sudo raspi-config` and follow Network Options to join the network created in step 1.
4. Join the newly created network on a Mac running HAT.
5. Pair and Discover router accessory
6. In the left sidebar, select "Managed Network Enable".
7. On the main screen, navigate to the "Prepare and Execute Timed Write [Unsigned Integer]" section.
8. Enter a value of "1" in the text field and click the "Timed Write" button.
9. In the left sidebar, select the name of the accessory that was added in step 3.
10. On the main screen under heading "Network Declarations" choose button "Select File" next to "Import JSON"
11. Browse and select the TCSR022.json file, downloaded from the MFi Portal.
12. Select "Apply rules" button in the "Network Declarations" section.
13. On the next screen, ensure "No Restrictions" is selected at the import rules line at top of page, and press "Apply" to apply those rules to the router for this accessory.
14. Verify the rules are correctly reflected on the router by selecting the "Show Configured" button.
15. On ADK terminal, try to access any web server by running the command `#curl -v <web address>` , Make sure the https server is connected.
16. Setup a Mac client on the WAN side of the router with any WAN IP. Open the terminal on the Mac client, run a TCP server on any valid port in the range 0-65535 with the command `#iperf -s -p <Port Number> -i 1 -u`
17. On the ADK terminal run `#iperf -c <Mac WAN IP as above> -p <Port Number as above> -i 1 -u`, verify the traffic is successfully sent without being blocked.
18. On ADK terminal, run the command `#ping <Mac WAN IP as above>`, make sure ping command is success with host reachable.
19. On ADK accessory, run a TCP server on any valid port in the range 0-65535 with the command `#iperf -s -p <Port Number> -i 1`.
20. On the Mac running HAT, open terminal and run the command `#iperf -c < Ip of the ADK client running iperf s as above> -p <Port Number as above> -i 1`. Make sure the TCP traffic goes through and not get blocked.

21. On the Mac, run a UDP server on any valid port in the range 0-65535 with the command `#iperf –s –p <Port Number> –i 1 –u`.
22. On ADK client, open terminal and run the command `#iperf –c < Ip of the Mac client running iperf s as above> –p <Port Number as above> –i 1 –u`. Make sure the UDP traffic goes through and not get blocked.
23. On Mac running HAT client, open terminal and run the command `#ping <ip of the ADK accessory>`. Verify the ping success with host reachable.
24. In the left sidebar, select the name of the accessory that was added in step 3.
25. On the main screen under heading "Network Declarations" choose button "Select File" next to "Import JSON"
26. Browse and select the TCSR022.json file, downloaded from the MFi Portal.
27. Select "Apply rules" button in the "Network Declarations" section.
28. On the next screen, ensure "Auto" is selected at the import rules line at top of page, and press "Apply" to apply those rules to the router for this accessory.
29. Verify the rules are correctly reflected on the router by selecting the "Show Configured" button.
30. On Mac client, run a TCP server on a specific ports with command: `iperf –s –p 9000 –i 1`
31. On ADK accessory, open terminal and run the command `iperf –c <ip of the Mac client running iperf s as above> –p 9000 –i 1`. Make sure the TCP traffic goes through and not get blocked.
32. On Mac client, run a TCP server on any valid port in the range 0-65535 except 9000 with the command `#iperf –s –p <Port Number> –i 1`
33. On ADK accessory, open terminal and run the command `#iperf –c < Ip of the Mac client running iperf s as above> –p <Port Number as above> –i 1`. Make sure the TCP traffic does not go through and gets blocked.
34. On ADK accessory, run a TCP server on any valid port in the range 0-65535 with the command `#iperf –s –p <Port Number> –i 1`.
35. On the Mac running HAT, open terminal and run the command `#iperf –c < Ip of the ADK client running iperf s as above> –p <Port Number as above> –i 1`. Make sure the TCP traffic does not go through and gets blocked.
36. In the left sidebar, select the name of the accessory that was added in step 3.
37. On the main screen under heading "Network Declarations" choose button "Select File" next to "Import JSON"
38. Browse and select the TCSR022.json file, downloaded from the MFi Portal.
39. Select "Apply rules" button in the "Network Declarations" section.
40. On the next screen, ensure "No Restrictions" is selected at the import rules line at top of page, and press "Apply" to apply those rules to the router for this accessory.
41. Verify the rules are correctly reflected on the router by selecting the "Show Configured" button.

42. On ADK terminal, try to access any web server by running the command `#curl –v <web address>` , Make sure the https server is connected.

43. Setup a Mac client on the WAN side of the router with any WAN IP. Open the terminal on the Mac client, run a TCP server on any valid port in the range 0-65535 with the command `#iperf –s –p <Port Number> –i 1 –u`

44. On the ADK terminal run `#iperf –c <Mac WAN IP as above> –p <Port Number as above> –i 1 –u`, verify the traffic is successfully sent without being blocked.

45. On ADK terminal, run the command `#ping <Mac WAN IP as above>`, make sure ping command is success with host reachable.

46. On ADK accessory, run a TCP server on any valid port in the range 0-65535 with the command `#iperf –s –p <Port Number> –i 1`

47. On the Mac running HAT, open terminal and run the command `#iperf –c < Ip of the ADK client running iperf s as above> –p <Port Number as above> –i 1`. Make sure the TCP traffic goes through and not get blocked.

48. On the Mac, run a UDP server on any valid port in the range 0-65535 with the command `#iperf –s –p <Port Number> –i 1 –u`.

49. On ADK client, open terminal and run the command `#iperf –c < Ip of the Mac client running iperf s as above> –p <Port Number as above> –i 1 –u`. Make sure the UDP traffic goes through and not get blocked.

50. On Mac running HAT client, open terminal and run the command `#ping <ip of the ADK accessory>`. Verify the ping success with host reachable.

51. Setup another ADK device as a HomeKit accessory, or use the same ADK accessory as above.

52. Connect the ADK device to the router's ethernet port (Make sure ADK accessory is only connected using the ethernet port and not connected wirelessly).

53. If the Router supports Ethernet, Repeat the steps from step 9 to step 50 on the ADK accessory connected using Ethernet.

54. Setup another ADK device as a HomeKit accessory, or use the same ADK accessory as above.

55. Connect the ADK device to the network.

56. Ensure the ADK is connected to one of the satellites, and not the main gateway. (To achieve this, a shielding cloth can be used to cover the main gateway to cause satellite to have a stronger signal, or satellite and iOS device can be moved to a more isolated area where the signal strength from the satellite to the ADK device is strongest.)

57. Repeat the steps from step 9 to step 50 on the ADK accessory connected wirelessly to the mesh node.

TCSR023 **Verify the behavior of LAN TCP Outbound Static port rule with port ranges.**

Applies to HomeKit-enabled Wi-Fi router accessories. Perform this test case using HAT, the accessory app, a Raspberry Pi running an ADK IP accessory, and the test network declaration JSON files found on the MFi Portal.

1. Use the accessory's app to setup and create a new wireless network.
2. Setup an ADK device as a HomeKit accessory.
3. On the ADK terminal run `#sudo raspi-config` and follow Network Options to join the network created in step 1.
4. Join the newly created network on a Mac running HAT.
5. Pair and Discover router accessory
6. In the left sidebar, select "Managed Network Enable".
7. On the main screen, navigate to the "Prepare and Execute Timed Write [Unsigned Integer]" section.
8. Enter a value of "1" in the text field and click the "Timed Write" button.
9. In the left sidebar, select the name of the accessory that was added in step 3.
10. On the main screen under heading "Network Declarations" choose button "Select File" next to "Import JSON"
11. Browse and select the TCSR023.json file, downloaded from the MFi Portal.
12. Select "Apply rules" button in the "Network Declarations" section.
13. On the next screen, ensure "No Restrictions" is selected at the import rules line at top of page, and press "Apply" to apply those rules to the router for this accessory.
14. Verify the rules are correctly reflected on the router by selecting the "Show Configured" button.
15. On ADK terminal, try to access any web server by running the command `#curl –v <web address>` , Make sure the https server is connected.
16. Setup a Mac client on the WAN side of the router with any WAN IP. Open the terminal on the Mac client, run a TCP server on any valid port in the range 0-65535 with the command `#iperf –s –p <Port Number> –i 1 –u`.
17. On the ADK terminal run `#iperf –c <Mac WAN IP as above> –p <Port Number as above> –i 1 –u`, verify the traffic is successfully sent without being blocked.
18. On ADK terminal, run the command `#ping <Mac WAN IP as above>`, make sure ping command is success with host reachable.
19. On ADK accessory, run a TCP server on any valid port in the range 0-65535 with the command `#iperf –s –p <Port Number> –i 1`.
20. On the Mac running HAT, open terminal and run the command `#iperf –c < Ip of the ADK client running iperf s as above> –p <Port Number as above> –i 1`. Make sure the TCP traffic goes through and not get blocked.
21. On the Mac, run a UDP server on any valid port in the range 0-65535 with the command `#iperf –s –p <Port Number> –i 1 –u`.
22. On ADK client, open terminal and run the command `#iperf –c < Ip of the Mac client running iperf s as above> –p <Port Number as above> –i 1 –u`. Make sure the UDP traffic goes through and not get blocked.

23. On Mac running HAT client, open terminal and run the command `#ping <ip of the ADK accessory>`. Verify the ping success with host reachable.
24. In the left sidebar, select the name of the accessory that was added in step 3.
25. On the main screen under heading "Network Declarations" choose button "Select File" next to "Import JSON"
26. Browse and select the TCSR023.json file, downloaded from the MFi Portal.
27. Select "Apply rules" button in the "Network Declarations" section.
28. On the next screen, ensure "Auto" is selected at the import rules line at top of page, and press "Apply" to apply those rules to the router for this accessory.
29. Verify the rules are correctly reflected on the router by selecting the "Show Configured" button.
30. On Mac accessory Run a TCP server on a specific ports with command: `iperf –s –p 9030 –i 1`.
31. On ADK accessory, open terminal and run the command `iperf –c <ip of the Mac client running iperf s as above> –p 9030 –i 1`. Make sure the TCP traffic goes through and not get blocked.
32. On Mac accessory Run a TCP server on a specific ports with command: `iperf –s –p 9100 –i 1`.
33. On ADK accessory, open terminal and run the command `iperf –c <ip of the Mac client running iperf s as above> –p 9100 –i 1`. Make sure the TCP traffic goes through and not get blocked.
34. On Mac client, run a TCP server on any valid port in the range 0-65535 except 9000-9100 with the command `#iperf –s –p <Port Number> –i 1`.
35. On ADK accessory, open terminal and run the command `#iperf –c < Ip of the Mac client running iperf s as above> –p <Port Number as above> –i 1`. Make sure the TCP traffic does not go through and gets blocked.
36. On ADK accessory, run a TCP server on any valid port in the range 0-65535 with the command `#iperf –s –p <Port Number> –i 1`.
37. On the Mac running HAT, open terminal and run the command `#iperf –c < Ip of the ADK client running iperf s as above> –p <Port Number as above> –i 1`. Make sure the TCP traffic does not go through and gets blocked.
38. In the left sidebar, select the name of the accessory that was added in step 3.
39. On the main screen under heading "Network Declarations" choose button "Select File" next to "Import JSON"
40. Browse and select the TCSR023.json file, downloaded from the MFi Portal.
41. Select "Apply rules" button in the "Network Declarations" section.
42. On the next screen, ensure "No Restrictions" is selected at the import rules line at top of page, and press "Apply" to apply those rules to the router for this accessory.
43. Verify the rules are correctly reflected on the router by selecting the "Show Configured" button.
44. On ADK terminal, try to access any web server by running the command `#curl –v <web address>`, Make sure the https server is connected.

45. Setup a Mac client on the WAN side of the router with any WAN IP. Open the terminal on the Mac client, run a TCP server on any valid port in the range 0-65535 with the command `#iperf –s –p <Port Number> –i 1 –u`
46. On the ADK terminal run `#iperf –c <Mac WAN IP as above> –p <Port Number as above> –i 1 –u`, verify the traffic is successfully sent without being blocked.
47. On ADK terminal, run the command `#ping <Mac WAN IP as above>`, make sure ping command is success with host reachable.
48. On ADK accessory, run a TCP server on any valid port in the range 0-65535 with the command `#iperf –s –p <Port Number> –i 1`.
49. On the Mac running HAT, open terminal and run the command `#iperf –c < Ip of the ADK client running iperf s as above> –p <Port Number as above> –i 1`. Make sure the TCP traffic goes through and not get blocked.
50. On the Mac, run a UDP server on any valid port in the range 0-65535 with the command `#iperf –s –p <Port Number> –i 1 –u`.
51. On ADK client, open terminal and run the command `#iperf –c < Ip of the Mac client running iperf s as above> –p <Port Number as above> –i 1 –u`. Make sure the UDP traffic goes through and not get blocked.
52. On Mac running HAT client, open terminal and run the command `#ping <ip of the ADK accessory>`. Verify the ping success with host reachable.
53. Setup another ADK device as a HomeKit accessory, or use the same ADK accessory as above.
54. Connect the ADK device to the router's ethernet port (Make sure ADK accessory is only connected using the ethernet port and not connected wirelessly).
55. If the router supports Ethernet, Repeat the steps from step 9 to step 52 on the ADK accessory connected using Ethernet.
56. Setup another ADK device as a HomeKit accessory, or use the same ADK accessory as above.
57. Connect the ADK device to the router mesh node (Make sure ADK accessory is connected to the mesh node and not the gateway node).
58. Repeat the steps from step 9 to step 52 on the ADK accessory connected wirelessly to the mesh node.

TCSR024 **Verify the behavior of a LAN UDP Outbound Static port rule with an individual port.**

Applies to HomeKit-enabled Wi-Fi router accessories. Perform this test case using HAT, the accessory app, a Raspberry Pi running an ADK IP accessory, and the test network declaration JSON files found on the MFi Portal.

1. Use the accessory's app to set up and create a new wireless network.
2. Setup an ADK device as a HomeKit accessory.
3. On the ADK terminal run `#sudo raspi–config` and follow Network Options to join the network created in step 1.

4. Join the newly created network on a Mac running HAT.
5. Pair and Discover router accessory
6. In the left sidebar, select “Managed Network Enable”.
7. On the main screen, navigate to the “Prepare and Execute Timed Write [Unsigned Integer]” section.
8. Enter a value of “1” in the text field and click the “Timed Write” button.
9. In the left sidebar, select the name of the accessory that was added in step 3.
10. On the main screen under heading “Network Declarations” choose button “Select File” next to “Import JSON”
11. Browse and select the TCSR024.json file, downloaded from the MFi Portal.
12. Select “Apply rules” button in the “Network Declarations” section.
13. On the next screen, ensure “No Restrictions” is selected at the import rules line at top of page, and press “Apply” to apply those rules to the router for this accessory.
14. Verify the rules are correctly reflected on the router by selecting the “Show Configured” button.
15. On ADK terminal, try to access any web server by running the command `#curl –v <web address>` , Make sure the https server is connected.
16. Setup a Mac client on the WAN side of the router with any WAN IP. Open the terminal on the Mac client, run a TCP server on any valid port in the range 0-65535 with the command `#iperf –s –p <Port Number> –i 1 –u`
17. On the ADK terminal run `#iperf –c <Mac WAN IP as above> –p <Port Number as above> –i 1 –u`, verify the traffic is successfully sent without being blocked.
18. On ADK terminal, run the command `#ping <Mac WAN IP as above>`, make sure ping command is success with host reachable.
19. On ADK accessory, run a TCP server on any valid port in the range 0-65535 with the command `#iperf –s –p <Port Number> –i 1`.
20. On the Mac running HAT, open terminal and run the command `#iperf –c < Ip of the ADK client running iperf s as above> –p <Port Number as above> –i 1`. Make sure the TCP traffic goes through and not get blocked.
21. On the Mac, run a UDP server on any valid port in the range 0-65535 with the command `#iperf –s –p <Port Number> –i 1 –u`.
22. On ADK client, open terminal and run the command `#iperf –c < Ip of the Mac client running iperf s as above> –p <Port Number as above> –i 1 –u`. Make sure the UDP traffic goes through and not get blocked.
23. On Mac running HAT client, open terminal and run the command `#ping <ip of the ADK accessory>`. Verify the ping success with host reachable.
24. In the left sidebar, select the name of the accessory that was added in step 3.
25. On the main screen under heading “Network Declarations” choose button “Select File” next to “Import JSON”

26. Browse and select the TCSR024.json file, downloaded from the MFi Portal.
27. Select “Apply rules” button in the “Network Declarations” section.
28. On the next screen, ensure “Auto” is selected at the import rules line at top of page, and press “Apply” to apply those rules to the router for this accessory.
29. Verify the rules are correctly reflected on the router by selecting the “Show Configured” button.
30. On the Mac, run a UDP server on a specific ports with command `iperf –s –p 9000 –i 1 –u`.
31. On ADK client, open terminal and run the command `iperf –c <ip of the Mac client running iperf s as above> –p 9000 –i 1 –u`. Make sure the UDP traffic goes through and not get blocked.
32. On Mac client, run a UDP server on any valid port in the range 0-65535 except 9000 with the command `#iperf –s –p <Port Number> –i 1 –u`.
33. On ADK accessory, open terminal and run the command `#iperf –c < Ip of the Mac client running iperf s as above> –p <Port Number as above> –i 1 –u`. Make sure the UDP traffic does not go through and gets blocked.
34. On ADK accessory, run a UDP server on any valid port in the range 0-65535 with the command `#iperf –s –p <Port Number> –i 1`.
35. On the Mac running HAT, open terminal and run the command `#iperf –c < Ip of the ADK client running iperf s as above> –p <Port Number as above> –i 1 –u`. Make sure the UDP traffic does not go through and gets blocked.
36. In the left sidebar, select the name of the accessory that was added in step 3.
37. On the main screen under heading “Network Declarations” choose button “Select File” next to “Import JSON”
38. Browse and select the TCSR024.json file, downloaded from the MFi Portal.
39. Select “Apply rules” button in the “Network Declarations” section.
40. On the next screen, ensure “No Restrictions” is selected at the import rules line at top of page, and press “Apply” to apply those rules to the router for this accessory.
41. Verify the rules are correctly reflected on the router by selecting the “Show Configured” button.
42. On ADK terminal, try to access any web server by running the command `#curl –v <web address>` , Make sure the https server is connected.
43. Setup a Mac client on the WAN side of the router with any WAN IP. Open the terminal on the Mac client, run a TCP server on any valid port in the range 0-65535 with the command `#iperf –s –p <Port Number> –i 1 –u`.
44. On the ADK terminal run `#iperf –c <Mac WAN IP as above> –p <Port Number as above> –i 1 –u`, verify the traffic is successfully sent without being blocked.
45. On ADK terminal, run the command `#ping <Mac WAN IP as above>`, make sure ping command is success with host reachable.
46. On ADK accessory, run a TCP server on any valid port in the range 0-65535 with the command `#iperf –s –p <Port Number> –i 1`.

47. On the Mac running HAT, open terminal and run the command `#iperf –c < Ip of the ADK client running iperf s as above> –p <Port Number as above> –i 1`. Make sure the TCP traffic goes through and not get blocked.
48. On the Mac, run a UDP server on any valid port in the range 0-65535 with the command `#iperf –s –p <Port Number> –i 1 –u`.
49. On ADK client, open terminal and run the command `#iperf –c < Ip of the Mac client running iperf s as above> –p <Port Number as above> –i 1 –u`. Make sure the UDP traffic goes through and not get blocked.
50. On Mac running HAT client, open terminal and run the command `#ping <ip of the ADK accessory>`. Verify the ping success with host reachable.
51. Setup another ADK device as a HomeKit accessory, or use the same ADK accessory as above.
52. Connect the ADK device to the router's ethernet port (Make sure ADK accessory is only connected using the ethernet port and not connected wirelessly).
53. If the router supports Ethernet, Repeat the steps from step 9 to step 50 on the ADK accessory connected using Ethernet.
54. Setup another ADK device as a HomeKit accessory, or use the same ADK accessory as above.
55. Connect the ADK device to the router mesh node (Make sure ADK accessory is connected to the mesh node and not the gateway node).
56. Repeat the steps from step 9 to step 50 on the ADK accessory connected wirelessly to the mesh node.

TCSR025 **Verify the behavior of a LAN UDP Outbound Static port rule with port ranges.**

Applies to HomeKit-enabled Wi-Fi router accessories. Perform this test case using HAT, the accessory app, a Raspberry Pi running an ADK IP accessory, and the test network declaration JSON files found on the MFi Portal.

1. Use the accessory's app to setup and create a new wireless network.
2. Setup an ADK device as a HomeKit accessory.
3. On the ADK terminal run `#sudo raspi–config` and follow Network Options to join the network created in step 1.
4. Join the newly created network on a Mac running HAT.
5. Pair and Discover router accessory
6. In the left sidebar, select "Managed Network Enable".
7. On the main screen, navigate to the "Prepare and Execute Timed Write [Unsigned Integer]" section.
8. Enter a value of "1" in the text field and click the "Timed Write" button.
9. In the left sidebar, select the name of the accessory that was added in step 3.
10. On the main screen under heading "Network Declarations" choose button "Select File" next to "Import JSON"

11. Browse and select the TCSR025 JSON file, downloaded from the MFi Portal.
12. Select “Apply rules” button in the “Network Declarations” section.
13. On the next screen, ensure “No Restrictions” is selected at the import rules line at top of page, and press “Apply” to apply those rules to the router for this accessory.
14. Verify the rules are correctly reflected on the router by selecting the “Show Configured” button.
15. On ADK terminal, try to access any web server by running the command `#curl –v <web address>` , Make sure the https server is connected.
16. Setup a Mac client on the WAN side of the router with any WAN IP. Open the terminal on the Mac client, run a TCP server on any valid port in the range 0-65535 with the command `#iperf –s –p <Port Number> –i 1 -u`
17. On the ADK terminal run `#iperf –c <Mac WAN IP as above> –p <Port Number as above> –i 1 –u`, verify the traffic is successfully sent without being blocked.
18. On ADK terminal, run the command `#ping <Mac WAN IP as above>`, make sure ping command is success with host reachable.
19. On ADK accessory, run a TCP server on any valid port in the range 0-65535 with the command `#iperf –s –p <Port Number> –i 1`.
20. On the Mac running HAT, open terminal and run the command `#iperf –c < Ip of the ADK client running iperf s as above> –p <Port Number as above> –i 1`. Make sure the TCP traffic goes through and not get blocked.
21. On the Mac, run a UDP server on any valid port in the range 0-65535 with the command `#iperf –s –p <Port Number> –i 1 –u`.
22. On ADK client, open terminal and run the command `#iperf –c < Ip of the Mac client running iperf s as above> –p <Port Number as above> –i 1 –u`. Make sure the UDP traffic goes through and not get blocked.
23. On Mac running HAT client, open terminal and run the command `#ping <ip of the ADK accessory>`. Verify the ping success with host reachable.
24. In the left sidebar, select the name of the accessory that was added in step 3.
25. On the main screen under heading “Network Declarations” choose button “Select File” next to “Import JSON”
26. Browse and select the TCSR025.json file, downloaded from the MFi Portal.
27. Select “Apply rules” button in the “Network Declarations” section.
28. On the next screen, ensure “Auto” is selected at the import rules line at top of page, and press “Apply” to apply those rules to the router for this accessory.
29. Verify the rules are correctly reflected on the router by selecting the ”Show Configured” button.
30. On the Mac, run a UDP server on a specific ports with the command `#iperf –s –p 9000 –i 1 –u`.

31. On ADK client, open terminal and run the command `iperf –c <ip of the Mac client running iperf s as above> –p 9000 –i 1 –u`. Make sure the UDP traffic goes through and not get blocked.
32. On the Mac, run a UDP server on a specific ports with the command `#iperf –s –p 9070 –i 1 –u`.
33. On ADK client, open terminal and run the command `iperf –c <ip of the Mac client running iperf s as above> –p 9070 –i 1 –u`. Make sure the UDP traffic goes through and not get blocked.
34. On Mac client, run a UDP server on any valid port in the range 0-65535 except 9000-9100 with the command `#iperf –s –p <Port Number> –i 1`.
35. On ADK accessory, open terminal and run the command `#iperf –c < Ip of the Mac client running iperf s as above> –p <Port Number as above> –i 1 –u`. Make sure the UDP traffic does not go through and gets blocked.
36. On ADK accessory, run a UDP server on any valid port in the range 0-65535 with the command `#iperf –s –p <Port Number> –i 1`.
37. On the Mac running HAT, open terminal and run the command `#iperf –c < Ip of the ADK client running iperf s as above> –p <Port Number as above> –i 1 –u`. Make sure the UDP traffic does not go through and gets blocked.
38. In the left sidebar, select the name of the accessory that was added in step 3.
39. On the main screen under heading "Network Declarations" choose button "Select File" next to "Import JSON"
40. Browse and select the TCSR025 JSON file, downloaded from the MFi Portal.
41. Select "Apply rules" button in the "Network Declarations" section.
42. On the next screen, ensure "No Restrictions" is selected at the import rules line at top of page, and press "Apply" to apply those rules to the router for this accessory.
43. Verify the rules are correctly reflected on the router by selecting the "Show Configured" button.
44. On ADK terminal, try to access any web server by running the command `#curl –v <web address>` , Make sure the https server is connected.
45. Setup a Mac client on the WAN side of the router with any WAN IP. Open the terminal on the Mac client, run a TCP server on any valid port in the range 0-65535 with the command `#iperf –s –p <Port Number> –i 1 –u`
46. On the ADK terminal run `#iperf –c <Mac WAN IP as above> –p <Port Number as above> –i 1 –u`, verify the traffic is successfully sent without being blocked.
47. On ADK terminal, run the command `#ping <Mac WAN IP as above>`, make sure ping command is success with host reachable.
48. On ADK accessory, run a TCP server on any valid port in the range 0-65535 with the command `#iperf –s –p <Port Number> –i 1`.
49. On the Mac running HAT, open terminal and run the command `#iperf –c < Ip of the ADK client`

`running iperf s as above> –p <Port Number as above> –i 1` Make sure the TCP traffic goes through and not get blocked.

50. On the Mac, run a UDP server on any valid port in the range 0-65535 with the command `#iperf –s –p <Port Number> –i 1 –u`.
51. On ADK client, open terminal and run the command `#iperf –c < Ip of the Mac client running iperf s as above> –p <Port Number as above> –i 1 –u`. Make sure the UDP traffic goes through and not get blocked.
52. On Mac running HAT client, open terminal and run the command `#ping <ip of the ADK accessory>`. Verify the ping success with host reachable.
53. Setup another ADK device as a HomeKit accessory, or use the same ADK accessory as above.
54. Connect the ADK device to the router's ethernet port (Make sure ADK accessory is only connected using the ethernet port and not connected wirelessly).
55. If the router supports Ethernet, Repeat the steps from step 9 to step 52 on the ADK accessory connected using Ethernet.
56. Setup another ADK device as a HomeKit accessory, or use the same ADK accessory as above.
57. Connect the ADK device to the router mesh node (Make sure ADK accessory is connected to the mesh node and not the gateway node).
58. Repeat the steps from step 9 to step 52 on the ADK accessory connected wirelessly to the mesh node.

TCSR026 **Verify the behavior of a LAN ICMP Outbound Static rule.**

Applies to HomeKit-enabled Wi-Fi router accessories. Perform this test case using HAT, the accessory app, a Raspberry Pi running an ADK IP accessory, and the test network declaration JSON files found on the MFi Portal.

1. Use the accessory's app to setup and create a new wireless network.
2. Setup an ADK device as a HomeKit accessory.
3. On the ADK terminal run `#sudo raspi-config` and follow Network Options to join the network created in step 1.
4. Join the newly created network on a Mac running HAT.
5. Pair and discover router accessory.
6. In the left sidebar, select "Managed Network Enable".
7. On the main screen, navigate to the "Prepare and Execute Timed Write [Unsigned Integer]" section.
8. Enter a value of "1" in the text field and click the "Timed Write" button.
9. In the left sidebar, select the name of the accessory that was added in step 3.
10. On the main screen under heading "Network Declarations" choose button "Select File" next to "Import JSON"

11. Browse and select the TCSR026.json file, downloaded from the MFi Portal.
12. Select “Apply rules” button in the “Network Declarations” section.
13. On the next screen, ensure “No Restrictions” is selected at the import rules line at top of page, and press “Apply” to apply those rules to the router for this accessory.
14. Verify the rules are correctly reflected on the router by selecting the “Show Configured” button.
15. On ADK terminal, try to access any web server by running the command `#curl –v <web address>` , Make sure the https server is connected.
16. Setup a Mac client on the WAN side of the router with any WAN IP. Open the terminal on the Mac client, run a TCP server on any valid port in the range 0-65535 with the command `#iperf –s –p <Port Number> –i 1 –u`
17. On the ADK terminal run `#iperf –c <Mac WAN IP as above> –p <Port Number as above> –i 1 –u`, verify the traffic is successfully sent without being blocked.
18. On ADK terminal, run the command `#ping <Mac WAN IP as above>`, make sure ping command is success with host reachable.
19. On ADK accessory, run a TCP server on any valid port in the range 0-65535 with the command `#iperf –s –p <Port Number> –i 1`.
20. On the Mac running HAT, open terminal and run the command `#iperf –c < Ip of the ADK client running iperf s as above> –p <Port Number as above> –i 1`. Make sure the TCP traffic goes through and not get blocked.
21. On the Mac, run a UDP server on any valid port in the range 0-65535 with the command `#iperf –s –p <Port Number> –i 1 –u`.
22. On ADK client, open terminal and run the command `#iperf –c < Ip of the Mac client running iperf s as above> –p <Port Number as above> –i 1 –u`. Make sure the UDP traffic goes through and not get blocked.
23. On Mac running HAT client, open terminal and run the command `#ping <ip of the ADK accessory>`. Verify the ping success with host reachable.
24. In the left sidebar, select the name of the accessory that was added in step 3.
25. On the main screen under heading “Network Declarations” choose button “Select File” next to “Import JSON”
26. Browse and select the TCSR026.json file, downloaded from the MFi Portal.
27. Select “Apply rules” button in the “Network Declarations” section.
28. On the next screen, ensure “Auto” is selected at the import rules line at top of page, and press “Apply” to apply those rules to the router for this accessory.
29. Verify the rules are correctly reflected on the router by selecting the “Show Configured” button.
30. On ADK accessory client, open terminal and run the command `#ping <ip of the Mac running HAT tool >`. Verify the ping success with host reachable.
31. On Mac running HAT client, open terminal and run the command `#ping <ip of the ADK accessory>`. Verify the ping fails with host unreachable.

32. On ADK terminal start a TCP iperf server by running the following command on ADK terminal: `#iperf –s –p 2234 –i 1`.
33. On the Mac which is running HAT, open the Terminal application.
34. Use iperf to connect to the server started in above step using the following command: `#iperf –c <ip address of ADK accessory> –p 2234 –i 1`. Verify traffic is blocked.
35. In the left sidebar, select the name of the accessory that was added in step 3.
36. On the main screen under heading "Network Declarations" choose button "Select File" next to "Import JSON"
37. Browse and select the TCSR026.json file, downloaded from the MFi Portal.
38. Select "Apply rules" button in the "Network Declarations" section.
39. On the next screen, ensure "No Restrictions" is selected at the import rules line at top of page, and press "Apply" to apply those rules to the router for this accessory.
40. Verify the rules are correctly reflected on the router by selecting the "Show Configured" button.
41. On ADK terminal, try to access any web server by running the command `#curl –v <web address>` , Make sure the https server is connected.
42. Setup a Mac client on the WAN side of the router with any WAN IP. Open the terminal on the Mac client, run a TCP server on any valid port in the range 0-65535 with the command `#iperf –s –p <Port Number> –i 1 –u`.
43. On the ADK terminal run `#iperf –c <Mac WAN IP as above> –p <Port Number as above> –i 1 –u`, verify the traffic is successfully sent without being blocked.
44. On ADK terminal, run the command `#ping <Mac WAN IP as above>`, make sure ping command is success with host reachable.
45. On ADK accessory, run a TCP server on any valid port in the range 0-65535 with the command `#iperf –s –p <Port Number> –i 1`.
46. On the Mac running HAT, open terminal and run the command `#iperf –c < Ip of the ADK client running iperf s as above> –p <Port Number as above> –i 1`. Make sure the TCP traffic goes through and not get blocked.
47. On the Mac, run a UDP server on any valid port in the range 0-65535 with the command `#iperf –s –p <Port Number> –i 1 -u`
48. On ADK client, open terminal and run the command `#iperf –c < Ip of the Mac client running iperf s as above> –p <Port Number as above> –i 1 –u`. Make sure the UDP traffic goes through and not get blocked.
49. On Mac running HAT client, open terminal and run the command `#ping <ip of the ADK accessory>`. Verify the ping success with host reachable.
50. Setup another ADK device as a HomeKit accessory, or use the same ADK accessory as above.
51. Connect the ADK device to the router's ethernet port (Make sure ADK accessory is only connected using the ethernet port and not connected wirelessly).

52. If Router supports Ethernet, Repeat the steps from step 9 to step 49 on the ADK accessory connected using Ethernet.

53. Setup another ADK device as a HomeKit accessory, or use the same ADK accessory as above.

54. Connect the ADK device to the router mesh node (Make sure ADK accessory is connected to the mesh node and not the gateway node).

55. Repeat the steps from step 9 to step 49 on the ADK accessory connected wirelessly to the mesh node.

TCSR027 **Verify the behavior of a LAN Inbound UDP DNS-SD Dynamic port rule.**

Applies to HomeKit-enabled Wi-Fi router accessories. Perform this test case using HAT, the accessory app, a Raspberry Pi running an ADK IP accessory, and the test network declaration JSON files found on the MFi Portal.

1. Use the accessory's app to setup and create a new wireless network.
2. Setup an ADK device as a HomeKit accessory.
3. On the ADK terminal run `#sudo raspi-config` and follow Network Options to join the network created in step 1.
4. Join the newly created network on a Mac running HAT.
5. Pair and Discover router accessory
6. In the left sidebar, select "Managed Network Enable".
7. On the main screen, navigate to the "Prepare and Execute Timed Write [Unsigned Integer]" section.
8. Enter a value of "1" in the text field and click the "Timed Write" button.
9. In the left sidebar, select the name of the accessory that was added in step 3.
10. On the main screen under heading "Network Declarations" choose button "Select File" next to "Import JSON"
11. Browse and select the TCSR027.json file, downloaded from the MFi Portal.
12. Select "Apply rules" button in the "Network Declarations" section.
13. On the next screen, ensure "No Restrictions" is selected at the import rules line at top of page, and press "Apply" to apply those rules to the router for this accessory.
14. Verify the rules are correctly reflected on the router by selecting the "Show Configured" button.
15. On ADK terminal, try to access any web server by running the command`#curl -v <web address>` , Make sure the https server is connected.
16. Setup a Mac client on the WAN side of the router with any WAN IP. Open the terminal on the Mac client, run a TCP server on any valid port in the range 0-65535 with the command `#iperf -s -p <Port Number> -i 1 -u`.
17. On the ADK terminal run `#iperf -c <Mac WAN IP as above> -p <Port Number as above> -i 1 -u`, verify the traffic is successfully sent without being blocked.

18. On ADK terminal, run the command `#ping <Mac WAN IP as above>`, make sure ping command is success with host reachable.

19. On ADK accessory, run a TCP server on any valid port in the range 0-65535 with the command `#iperf –s –p <Port Number> –i 1`.

20. On the Mac running HAT, open terminal and run the command `#iperf –c < Ip of the ADK client running iperf s as above> –p <Port Number as above> –i 1`. Make sure the TCP traffic goes through and not get blocked.

21. On the Mac, run a UDP server on any valid port in the range 0-65535 with the command `#iperf –s –p <Port Number> –i 1 –u`.

22. On ADK client, open terminal and run the command `#iperf –c < Ip of the Mac client running iperf s as above> –p <Port Number as above> –i 1 –u`. Make sure the UDP traffic goes through and not get blocked.

23. On Mac running HAT client, open terminal and run the command `#ping <ip of the ADK accessory>`. Verify the ping success with host reachable.

24. In the left sidebar, select the name of the accessory that was added in step 3.

25. On the main screen under heading "Network Declarations" choose button "Select File" next to "Import JSON"

26. Browse and select the TCSR027.json file, downloaded from the MFi Portal.

27. Select "Apply rules" button in the "Network Declarations" section.

28. On the next screen, ensure "Auto" is selected at the import rules line at top of page, and press "Apply" to apply those rules to the router for this accessory.

29. Verify the rules are correctly reflected on the router by selecting the "Show Configured" button.

30. On the ADK client terminal, open terminal and run `#dns–sd –R "My Test" _http._tcp . 80 path=/path–to–page.html`. This will start a mdns in server mode advertising.

31. On the Mac client running HAT tool, in the terminal window, run: `#dns–sd –B _http._tcp`. Make sure the above advertisement shows up in the list.

32. On ADK accessory Run a TCP server on a specific port with the command `#iperf –s –p 80 –i 1`.

33. On Mac running HAT client, open terminal and run the command `#iperf –c < Ip of the ADK client running iperf as above> –p 80 –i 1`. Make sure the TCP traffic goes through and not get blocked.

34. On the ADK client terminal, open terminal and run on any valid port in the range 0-65535 except used above (80) with the command `#dns–sd –R "Printer Test" _printer._tcp . <Port number> path=/path–to–page.html`. This will start a mdns in server mode advertising.

35. On the Mac client running HAT tool, in the terminal window, run: `#dns–sd –B _printer._tcp`. Make sure the above advertisement shows up in the list.

36. On ADK accessory Run a TCP server on a specific port with the command `#iperf –s –p <Port number as above> –i 1`.

37. On Mac running HAT client, open terminal and run the command `#iperf –c < Ip of the ADK client running iperf as above> –p <Port number as above> –i 1`. Make sure the TCP traffic goes through and not get blocked
38. In the left sidebar, select the name of the accessory that was added in step 3.
39. On the main screen under heading “Network Declarations” choose button “Select File” next to “Import JSON”
40. Browse and select the TCSR027.json file, downloaded from the MFi Portal.
41. Select “Apply rules” button in the “Network Declarations” section.
42. On the next screen, ensure “No Restrictions” is selected at the import rules line at top of page, and press “Apply” to apply those rules to the router for this accessory.
43. Verify the rules are correctly reflected on the router by selecting the “Show Configured” button.
44. On ADK terminal, try to access any web server by running the command `#curl –v <web address>` , Make sure the https server is connected.
45. Setup a Mac client on the WAN side of the router with any WAN IP. Open the terminal on the Mac client, run a TCP server on any valid port in the range 0-65535 with the command `#iperf –s –p <Port Number> –i 1 –u`
46. On the ADK terminal run `#iperf –c <Mac WAN IP as above> –p <Port Number as above> –i 1 –u`, verify the traffic is successfully sent without being blocked.
47. On ADK terminal, run the command `#ping <Mac WAN IP as above>`, make sure ping command is success with host reachable.
48. On ADK accessory, run a TCP server on any valid port in the range 0-65535 with the command `#iperf –s –p <Port Number> –i 1`.
49. On the Mac running HAT, open terminal and run the command `#iperf –c < Ip of the ADK client running iperf s as above> –p <Port Number as above> –i 1`. Make sure the TCP traffic goes through and not get blocked.
50. On the Mac, run a UDP server on any valid port in the range 0-65535 with the command `#iperf –s –p <Port Number> –i 1 –u`.
51. On ADK client, open terminal and run the command `#iperf –c < Ip of the Mac client running iperf s as above> –p <Port Number as above> –i 1 –u`. Make sure the UDP traffic goes through and not get blocked.
52. On Mac running HAT client, open terminal and run the command `#ping <ip of the ADK accessory>`. Verify the ping success with host reachable
53. Setup another ADK device as a HomeKit accessory, or use the same ADK accessory as above.
54. Connect the ADK device to the router’s ethernet port (Make sure ADK accessory is only connected using the ethernet port and not connected wirelessly).
55. If router supports Ethernet, Repeat the steps from step 9 to step 52 on the ADK accessory connected using Ethernet.
56. Setup another ADK device as a HomeKit accessory, or use the same ADK accessory as above.

57. Connect the ADK device to the router mesh node (Make sure ADK accessory is connected to the mesh node and not the gateway node).
58. Repeat the steps from step 9 to step 52 on the ADK accessory connected wirelessly to the mesh node.

TCSR028 **Verify the behavior of a LAN Inbound UDP SSDP Dynamic port rule.**

Applies to HomeKit-enabled Wi-Fi router accessories. Perform this test case using HAT, the accessory app, a Raspberry Pi running an ADK IP accessory, and the test network declaration JSON files found on the MFi Portal.

1. Use the accessory's app to setup and create a new wireless network.
2. Setup an ADK device as a HomeKit accessory.
3. On the ADK terminal run `#sudo raspi-config` and follow Network Options to join the network created in step 1.
4. Join the newly created network on a Mac running HAT.
5. Pair and discover router accessory.
6. In the left sidebar, select "Managed Network Enable".
7. On the main screen, navigate to the "Prepare and Execute Timed Write [Unsigned Integer]" section.
8. Enter a value of "1" in the text field and click the "Timed Write" button.
9. In the left sidebar, select the name of the accessory that was added in step 3.
10. On the main screen under heading "Network Declarations" choose button "Select File" next to "Import JSON"
11. Browse and select the TCSR028.json file, downloaded from the MFi Portal.
12. Select "Apply rules" button in the "Network Declarations" section.
13. On the next screen, ensure "Auto" is selected at the import rules line at top of page, and press "Apply" to apply those rules to the router for this accessory.
14. Verify the rules are correctly reflected on the router by selecting the "Show Configured" button.
15. On the ADK client terminal, open terminal and run the attached python ssdp script in server mode.
16. On the Mac client running HAT tool, in the terminal window, run: the attached python ssdp script in client mode, verify the ssdp traffic is not blocked.
17. Setup another ADK device as a HomeKit accessory, or use the same ADK accessory as above.
18. Connect the ADK device to the router's ethernet port (Make sure ADK accessory is only connected using the ethernet port and not connected wirelessly).
19. (If Applicable)Repeat the steps from step 9 to step 15 on the ADK accessory connected using Ethernet.
20. Setup another ADK device as a HomeKit accessory, or use the same ADK accessory as above.
21. Connect the ADK device to the router mesh node (Make sure ADK accessory is connected to the mesh node and not the gateway node).

22. Repeat the steps from step 9 to step 15 on the ADK accessory connected wirelessly to the mesh node.

TCSR029 **Verify the behavior of a a LAN Outbound UDP DNS-SD Dynamic port rule.**

Applies to HomeKit-enabled Wi-Fi router accessories. Perform this test case using HAT, the accessory app, a Raspberry Pi running an ADK IP accessory, and the test network declaration JSON files found on the MFi Portal.

1. Use the accessory's app to setup and create a new wireless network.
2. Setup an ADK device as a HomeKit accessory.
3. On the ADK terminal run `#sudo raspi-config` and follow Network Options to join the network created in step 1.
4. Join the newly created network on a Mac running HAT.
5. Pair and discover router accessory.
6. In the left sidebar, select "Managed Network Enable".
7. On the main screen, navigate to the "Prepare and Execute Timed Write [Unsigned Integer]" section.
8. Enter a value of "1" in the text field and click the "Timed Write" button.
9. In the left sidebar, select the name of the accessory that was added in step 3.
10. On the main screen under heading "Network Declarations" choose button "Select File" next to "Import JSON"
11. Browse and select the TCSR029.json file, downloaded from the MFi Portal.
12. Select "Apply rules" button in the "Network Declarations" section.
13. On the next screen, ensure "No Restrictions" is selected at the import rules line at top of page, and press "Apply" to apply those rules to the router for this accessory.
14. Verify the rules are correctly reflected on the router by selecting the "Show Configured" button.
15. On ADK terminal, try to access any web server by running the command `#curl –v <web address>` , Make sure the https server is connected.
16. Setup a Mac client on the WAN side of the router with any WAN IP. Open the terminal on the Mac client, run a TCP server on any valid port in the range 0-65535 with the command `#iperf –s –p <Port Number> –i 1 –u`
17. On the ADK terminal run `#iperf –c <Mac WAN IP as above> –p <Port Number as above> –i 1 –u`, verify the traffic is successfully sent without being blocked.
18. On ADK terminal, run the command `#ping <Mac WAN IP as above>`, make sure ping command is success with host reachable.
19. On ADK accessory, run a TCP server on any valid port in the range 0-65535 with the command `#iperf –s –p <Port Number> –i 1`.

20. On the Mac running HAT, open terminal and run the command `#iperf –c < Ip of the ADK client running iperf s as above> –p <Port Number as above> –i 1`. Make sure the TCP traffic goes through and not get blocked.

21. On the Mac, run a UDP server on any valid port in the range 0-65535 with the command `#iperf –s –p <Port Number> –i 1 –u`.

22. On ADK client, open terminal and run the command `#iperf –c < Ip of the Mac client running iperf s as above> –p <Port Number as above> –i 1 –u`. Make sure the UDP traffic goes through and not get blocked.

23. On Mac running HAT client, open terminal and run the command `#ping <ip of the ADK accessory>`. Verify the ping success with host reachable.

24. In the left sidebar, select the name of the accessory that was added in step 3.

25. On the main screen under heading "Network Declarations" choose button "Select File" next to "Import JSON"

26. Browse and select the TCSR029.json file, downloaded from the MFi Portal.

27. Select "Apply rules" button in the "Network Declarations" section.

28. On the next screen, ensure "Auto" is selected at the import rules line at top of page, and press "Apply" to apply those rules to the router for this accessory.

29. Verify the rules are correctly reflected on the router by selecting the "Show Configured" button.

30. On the Mac client terminal, open terminal and run `#dns-sd –R "My Test" _http._tcp . 80 path=/path-to-page.html`. This will start a mdns in server mode advertising.

31. On the ADK client in the terminal window, run: `#dns-sd –B _http._tcp`. Make sure the above advertisement shows up in the list.

32. On MAC accessory Run a TCP server on a specific port with the command `#iperf –s –p 80 –i 1`.

33. On ADK running HAT client, open terminal and run the command `#iperf –c < Ip of the MAC client running iperf as above> –p 80 –i 1`. Make sure the TCP traffic goes through and not get blocked.

34. On the Mac client terminal, open terminal and run on any valid port in the range 0-65535 except used above (80) with the command `#dns-sd –R "Printer Test" _printer._tcp . <Port number> path=/path-to-page.html`. This will start a mdns in server mode advertising.

35. On the ADK client in the terminal window, run: `#dns-sd –B _printer._tcp` . Make sure the above advertisement shows up in the list.

36. On MAC accessory Run a TCP server on a specific port with the command `#iperf –s –p <Port number as above> –i 1`.

37. On ADK running HAT client, open terminal and run the command `#iperf –c < Ip of the MAC client running iperf as above> –p <Port number as above> –i 1` .Make sure the TCP traffic goes through and not get blocked.

38. In the left sidebar, select the name of the accessory that was added in step 3.

39. On the main screen under heading “Network Declarations” choose button “Select File” next to “Import JSON”

40. Browse and select the TCSR029.json file, downloaded from the MFi Portal.

41. Select “Apply rules” button in the “Network Declarations” section.

42. On the next screen, ensure “No Restrictions” is selected at the import rules line at top of page, and press “Apply” to apply those rules to the router for this accessory.

43. Verify the rules are correctly reflected on the router by selecting the “Show Configured” button.

44. On ADK terminal, try to access any web server by running the command `#curl –v <web address>` , Make sure the https server is connected.

45. Setup a Mac client on the WAN side of the router with any WAN IP. Open the terminal on the Mac client, run a TCP server on any valid port in the range 0-65535 with the command `#iperf –s –p <Port Number> –i 1 –u`.

46. On the ADK terminal run `#iperf –c <Mac WAN IP as above> –p <Port Number as above> –i 1 –u`, verify the traffic is successfully sent without being blocked.

47. On ADK terminal, run the command `#ping <Mac WAN IP as above>`, make sure ping command is success with host reachable.

48. On ADK accessory, run a TCP server on any valid port in the range 0-65535 with the command `#iperf –s –p <Port Number> –i 1`.

49. On the Mac running HAT, open terminal and run the command iperf -c < Ip of the ADK client running `#iperf s as above> –p <Port Number as above> –i 1`. Make sure the TCP traffic goes through and not get blocked.

50. On the Mac, run a UDP server on any valid port in the range 0-65535 with the command `#iperf –s –p <Port Number> –i 1 –u`.

51. On ADK client, open terminal and run the command iperf -c < Ip of the Mac client running `#iperf s as above> –p <Port Number as above> –i 1 –u`. Make sure the UDP traffic goes through and not get blocked.

52. On Mac running HAT client, open terminal and run the command `#ping <ip of the ADK accessory>`. Verify the ping success with host reachable.

53. Setup another ADK device as a HomeKit accessory, or use the same ADK accessory as above.

54. Connect the ADK device to the router’s ethernet port (Make sure ADK accessory is only connected using the ethernet port and not connected wirelessly).

55. If Router supports Ethernet, Repeat the steps from step 9 to step 52 on the ADK accessory connected using Ethernet.

56. Setup another ADK device as a HomeKit accessory, or use the same ADK accessory as above.

57. Connect the ADK device to the router mesh node (Make sure ADK accessory is connected to the mesh node and not the gateway node).

58. Repeat the steps from step 9 to step 52 on the ADK accessory connected wirelessly to the mesh node.

TCSR030 **Verify the behavior of a LAN Outbound UDP SSDP Dynamic port rule.**

Applies to HomeKit-enabled Wi-Fi router accessories. Perform this test case using HAT, the accessory app, a Raspberry Pi running an ADK IP accessory, and the test network declaration JSON files found on the MFi Portal.

1. Use the accessory's app to setup and create a new wireless network.
2. Setup an ADK device as a HomeKit accessory.
3. On the ADK terminal run `#sudo raspi-config` and follow Network Options to join the network created in step 1.
4. Join the newly created network on a Mac running HAT.
5. Pair and discover router accessory.
6. In the left sidebar, select "Managed Network Enable".
7. On the main screen, navigate to the "Prepare and Execute Timed Write [Unsigned Integer]" section.
8. Enter a value of "1" in the text field and click the "Timed Write" button.
9. In the left sidebar, select the name of the accessory that was added in step 3.
10. On the main screen under heading "Network Declarations" choose button "Select File" next to "Import JSON"
11. Browse and select the TCSR030.json file, downloaded from the MFi Portal.
12. Select "Apply rules" button in the "Network Declarations" section.
13. On the next screen, ensure "Auto" is selected at the import rules line at top of page, and press "Apply" to apply those rules to the router for this accessory.
14. Verify the rules are correctly reflected on the router by selecting the "Show Configured" button.
15. On the Mac client terminal, open terminal and run the attached python ssdp script in server mode.
16. On the ADK client, in the terminal window, run: the attached python ssdp script in client mode, verify the ssdp traffic is not blocked.
17. Setup another ADK device as a HomeKit accessory, or use the same ADK accessory as above.
18. Connect the ADK device to the router's ethernet port (Make sure ADK accessory is only connected using the ethernet port and not connected wirelessly).
19. (If Applicable)Repeat the steps from step 9 to step 15 on the ADK accessory connected using Ethernet.
20. Setup another ADK device as a HomeKit accessory, or use the same ADK accessory as above.
21. Connect the ADK device to the router mesh node (Make sure ADK accessory is connected to the mesh node and not the gateway node).
22. Repeat the steps from step 9 to step 15 on the ADK accessory connected wirelessly to the mesh node.

TCSR031 **Verify that all DNS requests made by accessories with rules must go through the router's DNS server. Verify that the DNS query response TTL is clamped to 10sec.**

Applies to HomeKit-enabled Wi-Fi router accessories. Perform this test case using HAT, the accessory app, a Raspberry Pi running an ADK IP accessory, and the test network declaration JSON files found on the MFi Portal.

1. Use the accessory's app to setup and create a new wireless network.
2. Setup an ADK device as a HomeKit accessory.
3. On the ADK terminal run `#sudo raspi-config` and follow Network Options to join the network created in step 1.
4. Join the newly created network on a Mac running HAT.
5. Pair and Discover router accessory
6. In the left sidebar, select "Managed Network Enable".
7. On the main screen, navigate to the "Prepare and Execute Timed Write [Unsigned Integer]" section.
8. Enter a value of "1" in the text field and click the "Timed Write" button.
9. In the left sidebar, select the name of the accessory that was added in step 3.
10. On the main screen under heading "Network Declarations" choose button "Select File" next to "Import JSON"
11. Browse and select the TCSR031.json file, downloaded from the MFi Portal.
12. Select "Apply rules" button in the "Network Declarations" section.
13. On the next screen, ensure "Auto" is selected at the import rules line at top of page, and press "Apply" to apply those rules to the router for this accessory.
14. Verify the rules are correctly reflected on the router by selecting the "Show Configured" button.
15. On the ADK accessory. Go to terminal and run dig www.engadget.com
16. Output of the dig command should show the TTL to be 10sec, for example:
    - dig host.example.gov
    - <*SNIP*>
    - ;; ANSWER SECTION:
    - host.example.gov. 10 IN CNAME host1.example.gov.
    - host1.example.gov. 10 IN A 192.168.16.10
17. Setup another ADK device as a HomeKit accessory, or use the same ADK accessory as above.
18. Connect the ADK device to the router's ethernet port (Make sure ADK accessory is only connected using the ethernet port and not connected wirelessly).
19. If the accessory supports ethernet, repeat step 9 to 15 using the Ethernet connection.
20. Setup another ADK device as a HomeKit accessory, or use the same ADK accessory as above.

21. Connect the ADK device to the router mesh node (Make sure ADK accessory is connected to the mesh node and not the gateway node).
22. Repeat the steps from step 9 to step 15 on the ADK accessory connected wirelessly to the mesh node.

TCSR032 **Verify that recursive DNS requests are also handled by the router. (Router has to send the DNS query to the DNS server with resolve recursively bit enabled.)**

Applies to HomeKit-enabled Wi-Fi router accessories. Perform this test case using HAT and an iOS device running the accessory app.

1. Use the accessory's app to setup and create a new wireless network.
2. Setup an ADK device as a HomeKit accessory.
3. On the ADK terminal run `#sudo raspi-config` and follow Network Options to join the network created in step 1.
4. Join the newly created network on a Mac running HAT.
5. Pair and Discover router accessory
6. In the left sidebar, select "Managed Network Enable".
7. On the main screen, navigate to the "Prepare and Execute Timed Write [Unsigned Integer]" section.
8. Enter a value of "1" in the text field and click the "Timed Write" button.
9. In the left sidebar, select the name of the accessory that was added in step 3.
10. On the main screen under heading "Network Declarations" choose button "Select File" next to "Import JSON"
11. Browse and select the TCSR032.json file, downloaded from the MFi Portal.
12. Select "Apply rules" button in the "Network Declarations" section.
13. On the next screen, ensure "Auto" is selected at the import rules line at top of page, and press "Apply" to apply those rules to the router for this accessory.
14. Verify the rules are correctly reflected on the router by selecting the "Show Configured" button.
15. On the ADK accessory. Go to terminal and run dig api.xbcs.net (here example domain is the domain listed in the json file as allowed domains).
16. Output of the dig command should show the TTL to be 10sec, As this is recursive domain, the query response should contain AAAA/A record with CNAME records as well.
17. Setup another ADK device as a HomeKit accessory, or use the same ADK accessory as above.
18. Connect the ADK device to the router's ethernet port (Make sure ADK accessory is only connected using the ethernet port and not connected wirelessly).
19. (If Applicable)Repeat the steps from step 9 to step 15 on the ADK accessory connected using Ethernet.
20. Setup another ADK device as a HomeKit accessory, or use the same ADK accessory as above.

21. Connect the ADK device to the router mesh node (Make sure ADK accessory is connected to the mesh node and not the gateway node).
22. Repeat the steps from step 9 to step 15 on the ADK accessory connected wirelessly to the mesh node.

TCSR033 **Verify that all DNS requests made by client with full access WAN rules and LAN rules should not be clamped by the router's DNS server when using external DNS server.**

Applies to HomeKit-enabled Wi-Fi router accessories. Perform this test case using HAT and an iOS device running the accessory app.

1. Use the accessory's app to setup and create a new wireless network.
2. Setup an ADK device as a HomeKit accessory.
3. On the ADK terminal run `#sudo raspi-config` and follow Network Options to join the network created in step 1.
4. Join the newly created network on a Mac running HAT.
5. Pair and Discover router accessory
6. In the left sidebar, select "Managed Network Enable".
7. On the main screen, navigate to the "Prepare and Execute Timed Write [Unsigned Integer]" section.
8. Enter a value of "1" in the text field and click the "Timed Write" button.
9. In the left sidebar, select the name of the accessory that was added in step 3.
10. On the main screen under heading "Network Declarations" choose button "Select File" next to "Import JSON"
11. Browse and select the TCSR032.json file, downloaded from the MFi Portal.
12. Select "Apply rules" button in the "Network Declarations" section.
13. On the next screen, ensure "No Restriction" is selected at the import rules line at top of page, and press "Apply" to apply those rules to the router for this accessory.
14. Verify the rules are correctly reflected on the router by selecting the "Show Configured" button.
15. On the ADK accessory. Go to terminal and run `#dig @8.8.8.8 api.xbcs.net`.
16. Output of the dig command should show the TTL to be not 10sec, As DNS server should not clamp the tcl of this request.
17. Setup another ADK device as a HomeKit accessory, or use the same ADK accessory as above.
18. Connect the ADK device to the router's ethernet port (Make sure ADK accessory is only connected using the ethernet port and not connected wirelessly).
19. (If Applicable)Repeat the steps from step 9 to step 15 on the ADK accessory connected using Ethernet.
20. Setup another ADK device as a HomeKit accessory, or use the same ADK accessory as above.
21. Connect the ADK device to the router mesh node (Make sure ADK accessory is connected to the mesh node and not the gateway node).

22. Repeat the steps from step 9 to step 15 on the ADK accessory connected wirelessly to the mesh node.

TCSR034 **Verify that a client on the restricted group CANNOT utilize UPNP to open ports up on the firewall.**

Applies to HomeKit-enabled Wi-Fi router accessories. Perform this test case using HAT and an iOS device running the accessory app.

1. Use the accessory's app to setup and create a new wireless network.
2. Setup an ADK device as a HomeKit accessory.
3. On the ADK terminal run `#sudo raspi-config` and follow Network Options to join the network created in step 1.
4. Join the newly created network on a Mac running HAT.
5. Pair and discover router accessory.
6. In the left sidebar, select "Managed Network Enable".
7. On the main screen, navigate to the "Prepare and Execute Timed Write [Unsigned Integer]" section.
8. Enter a value of "1" in the text field and click the "Timed Write" button.
9. In the left sidebar, select the name of the accessory that was added in step 3.
10. On the main screen under heading "Network Declarations" choose button "Select File" next to "Import JSON"
11. Browse and select the TCSR034.json file, downloaded from the MFi Portal.
12. Select "Apply rules" button in the "Network Declarations" section.
13. On the next screen, ensure "Restricted" is selected at the import rules line at top of page, and press "Apply" to apply those rules to the router for this accessory.
14. Verify the rules are correctly reflected on the router by selecting the "Show Configured" button.
15. On the ADK accessory. Go to terminal and run `#upnpc –a <ip or ADK accessory> 22 3333 TCP`.
16. Output of the above command should fail and it should be not open the port on the WAN side of the router.
17. Setup another ADK device as a HomeKit accessory, or use the same ADK accessory as above.
18. Connect the ADK device to the router's ethernet port (Make sure ADK accessory is only connected using the ethernet port and not connected wirelessly).
19. If router supports Ethernet, Repeat the steps from step 9 to step 15 on the ADK accessory connected using Ethernet.
20. Setup another ADK device as a HomeKit accessory, or use the same ADK accessory as above.
21. Connect the ADK device to the router mesh node (Make sure ADK accessory is connected to the mesh node and not the gateway node).

22. Repeat the steps from step 9 to step 15 on the ADK accessory connected wirelessly to the mesh node.

TCSR035 **Verify that a clients part of Main group can utilize UPNP to open ports up on the firewall.**

Applies to HomeKit-enabled Wi-Fi router accessories. Perform this test case using HAT and an iOS device running the accessory app.

1. Use the accessory's app to setup and create a new wireless network.
2. Setup an ADK device as a HomeKit accessory.
3. On the ADK terminal run `#sudo raspi-config` and follow Network Options to join the network created in step 1.
4. Join the newly created network on a Mac running HAT.
5. Pair and discover router accessory.
6. In the left sidebar, select "Managed Network Enable".
7. On the main screen, navigate to the "Prepare and Execute Timed Write [Unsigned Integer]" section.
8. Enter a value of "1" in the text field and click the "Timed Write" button.
9. In the left sidebar, select the name of the accessory that was added in step 3.
10. On the main screen under heading "Network Declarations" choose button "Select File" next to "Import JSON"
11. Browse and select the TCSR035.json file, downloaded from the MFi Portal.
12. Select "Apply rules" button in the "Network Declarations" section.
13. On the next screen, ensure "No Restriction" is selected at the import rules line at top of page, and press "Apply" to apply those rules to the router for this accessory.
14. Verify the rules are correctly reflected on the router by selecting the "Show Configured" button.
15. On the ADK accessory. Go to terminal and run `#upnpc –a <Ip of ADK accessory> 22 3333 TCP`.
16. Output of the above command should Pass and it should be opening the port on the WAN side of the router.

TCSR036 **Verify the behavior of an outbound TCP rule to DNS name, verify that when DNS changes to new IP address rule follow name, not Dest IP.**

Applies to HomeKit-enabled Wi-Fi router accessories. Perform this test case using HAT and an iOS device running the accessory app.

1. Use the accessory's app to setup and create a new wireless network.
2. Setup an ADK device as a HomeKit accessory.

3. On the ADK terminal run `#sudo raspi-config` and follow Network Options to join the network created in step 1.
4. Join the newly created network on a Mac running HAT.
5. Pair and discover router accessory.
6. In the left sidebar, select "Managed Network Enable".
7. On the main screen, navigate to the "Prepare and Execute Timed Write [Unsigned Integer]" section.
8. Enter a value of "1" in the text field and click the "Timed Write" button.
9. In the left sidebar, select the name of the accessory that was added in step 3.
10. On the main screen under heading "Network Declarations" choose button "Select File" next to "Import JSON"
11. Browse and select the TCSR036.json file, downloaded from the MFi Portal.
12. Select "Apply rules" button in the "Network Declarations" section.
13. On the next screen, ensure "Auto" is selected at the import rules line at top of page, and press "Apply" to apply those rules to the router for this accessory.
14. Verify the rules are correctly reflected on the router by selecting the "Show Configured" button.
15. On a second ADK device which needs to be configured as a DNS server, Connect the ADK device to the router Wi-Fi network.
16. SSH into the second ADK accessory and install dnsmasq.
17. Edit /etc/dnsmasq.conf file to add entry for the domain name example.com.
18. Add this line to the above config file `address=/example.com/<ip of the Mac client on WAN>`.
19. Restart the dnsmasq service using `sudo /etc/init.d/dnsmasq restart`.
20. Go to the router app and apply the DNS server as the IP of the second ADK device. And apply the changes.
21. Connect a Mac client on the WAN side of the router and start iperf serve session using `#iperf -s -p 9000 -i 1`.
22. From the first ADK accessory, start iperf client session to access the server on the MAC client using `#iperf -c example.com -p 9000 -i 1`.
23. Verify the above session is allowed and traffic goes through.
24. Now ssh into the second ADK device and update /etc/dnsmasq.conf file and change the IP address next to the example.com domain. Restart dnsmasq using `sudo /etc/init.d/dnsmasq restart`.
25. Now on the Mac client connected on the WAN side, Change the IP of the MAC machine to match with the newly updated IP address next to the example.com domain in the dnsmasq.conf file on second ADK accessory.
26. From the first ADK accessory, start iperf client session to access the server on the Mac client using `#iperf -c example.com -p 9000 -i 1`.

27. Verify the above session is allowed and traffic goes through.

TCSR037 **Verify that TCP Inbound/Outbound rules do not get effected when accessory with network declaration changes the IP address.**

Applies to HomeKit-enabled Wi-Fi router accessories. Perform this test case using HAT and an iOS device running the accessory app.

1. Use the accessory's app to set up and create a new wireless network.
2. Set up an ADK device as a HomeKit accessory.
3. On the ADK terminal run `#sudo raspi-config` and follow Network Options to join the network created in step 1.
4. Join the newly created network on a Mac running HAT.
5. Pair and discover router accessory.
6. In the left sidebar, select "Managed Network Enable".
7. On the main screen, navigate to the "Prepare and Execute Timed Write [Unsigned Integer]" section.
8. Enter a value of "1" in the text field and click the "Timed Write" button.
9. In the left sidebar, select the name of the ADK accessory.
10. On the main screen under heading "Network Declarations" choose button "Select File" next to "Import JSON"
11. Browse and select the TCSR037.json file, downloaded from the MFi Portal.
12. Select "Apply rules" button in the "Network Declarations" section.
13. On the next screen, ensure "Auto" is selected at the import rules line at top of page, and press "Apply" to apply those rules to the router for this accessory.
14. Verify the rules are correctly reflected on the router by selecting the "Show Configured" button.
15. Start server session on Mac client using `iperf -s -p 9000 -i 1`.
16. From the ADK accessory, start iperf client session to access the server on the Mac client running HAT tool using `#iperf -c <ip of Mac client> -p 9000 -i 1`
17. Start server session on ADK client using `iperf -s -p 9005 -i 1`.
18. From the MAC, start iperf client session to access the server on the ADK client using `#iperf -c <ip of ADK client> -p 9005 -i 1`
19. Verify the above session is allowed and traffic goes through.
20. Now SSH into ADK accessory and change the IP address of the ADK accessory.
21. On the terminal of the ADK run `sudo nano /etc/dhcpcd.conf`.
22. Update the file with below details:
    - interface eth0

- static ip_address=192.168.0.4/24
- static routers=192.168.0.1
- static domain_name_servers=192.168.0.1

23. Run `sudo reboot` to reboot the adk accessory and have the new ip address applied.
24. After the ADK accessory is back online, From the ADK accessory, start iperf client session to access the server on the Mac client running HAT tool using `#iperf -c <ip of Mac client> -p 9000 -i`.
25. From the Mac, start iperf client session to access the server on the ADK client using `#iperf -c <ip of ADK client> -p 9005 -i 1`.

TCSR038 **Verify that UDP Inbound/Outbound rules do not get effected when accessory with network declaration changes the IP address.**

Applies to HomeKit-enabled Wi-Fi router accessories. Perform this test case using HAT and an iOS device running the accessory app.

1. Use the accessory's app to setup and create a new wireless network.
2. Setup an ADK device as a HomeKit accessory.
3. On the ADK terminal run `#sudo raspi-config` and follow Network Options to join the network created in step 1.
4. Join the newly created network on a Mac running HAT.
5. Pair and discover router accessory.
6. In the left sidebar, select "Managed Network Enable".
7. On the main screen, navigate to the "Prepare and Execute Timed Write [Unsigned Integer]" section.
8. Enter a value of "1" in the text field and click the "Timed Write" button.
9. In the left sidebar, select the name of the accessory that was added in step 3.
10. On the main screen under heading "Network Declarations" choose button "Select File" next to "Import JSON"
11. Browse and select the TCSR038.json file, downloaded from the MFi Portal.
12. Select "Apply rules" button in the "Network Declarations" section.
13. A popup screen should appear showing the rules th5at will be applied to that accessory. Ensure "Auto" is selected at the import rules line at top of page, and press "Apply" to apply those rules to the router for this accessory.
14. Verify the rules are correctly reflected on the router by selecting the "Show Configured" button.
15. Start server session on Mac client using `iperf -s -p 9000 -i 1 -u`.
16. From the ADK accessory, start iperf client session to access the server on the Mac client running HAT tool using `#iperf -c <ip of Mac client> -p 9000 -i 1 -u`.

17. Start server session on ADK client using `iperf –s –p 9005 –i 1 –u`.
18. From the Mac, start iperf client session to access the server on the ADK client using `#iperf –c <ip of ADK client> –p 9005 –i 1 –u`.
19. Verify the above session is allowed and traffic goes through.
20. Now SSH into ADK accessory and change the IP address of the ADK accessory.
21. On the terminal of the ADK run sudo nano /etc/dhcpcd.conf
22. Update the file with below details:
    - interface eth0
    - static ip_address=192.168.0.4/24
    - static routers=192.168.0.1
    - static domain_name_servers=192.168.0.1
23. Run `sudo reboot` to reboot the adk accessory and have the new ip address applied.
24. After the ADK accessory is back online, From the ADK accessory, start iperf client session to access the server on the Mac client running HAT tool using `#iperf –c <ip of MAC client> –p 9000 –i 1 –u`.
25. From the Mac, start iperf client session to access the server on the ADK client using `#iperf –c <ip of ADK client> –p 9005 –i 1 –u`.

TCSR039 **Verify that the HAT tool and Home app should discover both Gateway node and the Satellite node if the Satellite node is already setup in the mesh network.**

Applies to HomeKit-enabled Wi-Fi router accessories. Perform this test case using HAT and an iOS device running the accessory app.

1. Use the accessory's app to setup and create a new wireless network, and add a sattelite to create a mesh network.
2. Open HAT, and in the left sidebar under "IP Controllers", select "Controller 1".
3. With "Controller 1" selected on the left, click the "Start" button under the "Discovery" section on the main screen.
4. In the left sidebar, select the router.
5. On the main screen click the "Start Pairing" button under the "Pairing" section.
6. In the popup, enter the setup payload or setup code of the router.
7. In the main screen, click the "Discover" button in the "Summary" section.
8. Verify in the left sidebar, Both the Gateway node and the satellite node characteristics are shown.
9. Verify gateway and satellite nodes are correctly discovered after the pairing to the home using home app.

TCSR040 **Verify that the HAT tool and Home app should discover newly added satellite node to the mesh network.**

Applies to HomeKit-enabled Wi-Fi router accessories. Perform this test case using HAT and an iOS device running the accessory app.

1. Use the accessory's app to setup and create a new wireless network.
2. Open HAT, and in the left sidebar under "IP Controllers", select "Controller 1".
3. With "Controller 1" selected on the left, click the "Start" button under the "Discovery" section on the main screen.
4. In the left sidebar, select the router.
5. On the main screen click the "Start Pairing" button under the "Pairing" section.
6. In the popup, enter the setup payload or setup code of the router.
7. In the main screen, click the "Discover" button in the "Summary" section.
8. Verify in the left sidebar, Gateway node characteristics are shown.
9. From the third party app, Setup the satellite node to the mesh network
10. In the HAT tool, rediscover the services and characteristics of the router, This should list Both the Gateway node and the satellite node characteristics.
11. Verify newly added satellite nodes is discovered in the Home app.

TCSR041 **Verify that the HomeKit accessory is controllable when moved to the HK LAN.**

Applies to HomeKit-enabled Wi-Fi router accessories. Perform this test case using HAT and an iOS device running the accessory app.

1. Use the accessory's app to setup and create a new wireless network.
2. Setup an ADK device as a HomeKit accessory.
3. On the ADK terminal run `#sudo raspi-config` and follow Network Options to join the network created in step 1.
4. From the HAT tool, Pair the ADK accessory.
5. Verify you are able to read/write any charecterstics to the ADK accessory.
6. Join the newly created network on a Mac running HAT.
7. Pair and discover router accessory.
8. In the left sidebar, select "Managed Network Enable".
9. On the main screen, navigate to the "Prepare and Execute Timed Write [Unsigned Integer]" section.
10. Enter a value of "1" in the text field and click the "Timed Write" button.
11. In the left sidebar, select the name of the accessory that was added in step 3.

12. On the main screen under heading “Network Declarations” choose button “Select File” next to “Import JSON”
13. Browse and select the TCSR041.json file, downloaded from the MFi Portal.
14. Select “Apply rules” button in the “Network Declarations” section.
15. On the next screen, ensure “Restricted” is selected at the import rules line at top of page, and press “Apply” to apply those rules to the router for this accessory.
16. Verify the rules are correctly reflected on the router by selecting the “Show Configured” button.
17. Reverify from the HAT tool if the read/write to ADK accessory are still working.

TCSR042 **Verify that traffic is blocked between Restricted group and Main group.**

Applies to HomeKit-enabled Wi-Fi router accessories. Perform this test case using HAT and an iOS device running the accessory app.

1. Use the accessory’s app to setup and create a new wireless network.
2. Setup an ADK device as a HomeKit accessory.
3. On the ADK terminal run `#sudo raspi-config` and follow Network Options to join the network created in step 1.
4. Join the newly created network on a Mac running HAT.
5. Pair and discover router accessory.
6. In the left sidebar, select “Managed Network Enable”.
7. On the main screen, navigate to the “Prepare and Execute Timed Write [Unsigned Integer]” section.
8. Enter a value of “1” in the text field and click the “Timed Write” button.
9. In the left sidebar, select the name of the accessory that was added in step 3.
10. On the main screen under heading “Network Declarations” choose button “Select File” next to “Import JSON”
11. Browse and select the TCSR042.json file, downloaded from the MFi Portal.
12. Select “Apply rules” button in the “Network Declarations” section.
13. On the next screen, ensure “Restricted” is selected at the import rules line at top of page, and press “Apply” to apply those rules to the router for this accessory.
14. Verify the rules are correctly reflected on the router by selecting the “Show Configured” button.
15. On the ADK accessory (which is on the restricted group) start a TCP iperf server using the following command: `iperf –s –p 2234 –i 1`.
16. On the Mac (which is on the Main LAN) use iperf to connect to the server started in above step using the following command: `perf –c XXX.XXX.XXX.XXX –p 2234 –i 1` where XXX.XXX.XXX.XXX is the IP address of the ADK accessory.

17. Verify no connection is made in iperf.
18. Stop iperf on the Mac and ADK
19. On the ADK accessory (which is on the restricted group) start a UDP iperf server using the following command: `iperf –s –p 2234 –i 1 –u`
20. On the Mac (which is on the Main LAN) use iperf to connect to the server started in step 6 using the following command: `perf –c XXX.XXX.XXX.XXX –p 2234 –i 1 –u` where XXX.XXX.XXX.XXX is the IP address of the ADK accessory.
21. Verify no connection is made in iperf.
22. Stop iperf on the Mac and ADK.
23. On the Mac start a TCP iperf server using the following command: `iperf –s –p 2234 –i 1`
24. On the ADK use iperf to connect to the server started in the previous step using the following command: `iperf –c XXX.XXX.XXX.XXX –p 2234 –i 1` where XXX.XXX.XXX.XXX is the IP address of the Mac.
25. Verify no connection is made in iperf.
26. Stop iperf on the Mac and ADK.
27. On the Mac, start a UDP iperf server using the following command: `iperf –s –p 2234 –i 1 –u.`
28. On the ADK use iperf to connect to the server started in the above step using the following command: `iperf –c XXX.XXX.XXX.XXX –p 2234 –i 1 –u` where XXX.XXX.XXX.XXX is the IP address of the Mac.
29. Verify no connection is made in iperf.
30. Terminate the iperf session running on the Mac and the ADK accessory.
31. Repeat the above steps for ping tests between ADK accessory and Mac client running HAT tool. Pings should fail between ADK and Mac accessory.

TCSR043 **Any Wi-Fi Router services must include the required characteristics.**

Applies to HomeKit-enabled Wi-Fi router accessories. Perform this test case using HAT and an iOS device running the accessory app.

1. Use the accessory's app to set up and create a new wireless network.
2. Join the newly created network on a Mac running HAT.
3. Open HAT, and in the left sidebar under "IP Controllers", select "Controller 1".
4. With "Controller 1" selected on the left, click the "Start" button under the "Discovery" section on the main screen.
5. In the left sidebar, select the router.
6. On the main screen, click the "Start Pairing" button under the "Pairing" section.
7. In the popup, enter the setup payload or setup code of the router.

8. In the main screen, click the “Discover” button in the “Summary” section.
9. In the left sidebar, verify the following characteristics are present under the “Wi-Fi Router” service:
    - “Version”
    - “Supported Router Configuration”
    - “Configured Name”
    - “Router Status”
    - “WAN Configuration List”
    - “WAN Status List”
    - “Managed Network Enable”
    - “Network Client Profile Control”
    - “Network Client Status Control”
    - “Network Access Violation Control”

TCSR044 **Verify that the accessory can successfully perform a Network Client Profile Control “Add” operation.**

Applies to HomeKit-enabled Wi-Fi router accessories. Perform this test case using HAT and an iOS device running the accessory app.

1. Use the accessory’s app to setup and create a new wireless network.
2. Join the newly created network on a Mac running HAT.
3. Open HAT, and in the left sidebar under “IP Controllers”, select “Controller 1”.
4. With “Controller 1” selected on the left, click the “Start” button under the “Discovery” section on the main screen.
5. In the left sidebar, select the router.
6. On the main screen click the “Start Pairing” button under the “Pairing” section.
7. In the popup, enter the setup payload or setup code of the router.
8. In the main screen, click the “Discover” button in the “Summary” section.
9. In the left sidebar, select “Managed Network Enable”.
10. On the main screen, navigate to the “Prepare and Execute Timed Write [Unsigned Integer]” section.
11. Enter a value of “1” in the text field and click the “Timed Write” button.
12. In the left sidebar, select “Network Client Profile Control”.
13. Under the “Write Options” section check the box next to “Write With Response”
14. On the main screen, navigate to the “Prepare and Execute Timed Write [TLV8 Packet]” section.
15. Click on the “Build TLV” button.
16. In the “Network Client Profile Control” TLV builder, set the following parameters:
    - Set the “Operation” to “Add”.
    - Leave the “Client Profile Identifier” field blank.

- Enter “1” into the “Client Group Identifier” field.
- Credential data: Enter “31 31 31 31 31 31 31 31” (HEX converted value of “11111111”)
- Set the “Credential Type” to “PSK”.
- Enter “010100” into the “WAN Firewall Config” field.
- Enter “010100” into the “LAN Firewall Config” field.

17. Click the “Add” button to add rule to the TLV builder table.
18. Click the “Build TLV” button in the bottom right corner.
19. In the “Prepare and Execute Timed Write [TLV8 Packet]” section, click the “Timed Write” button to write this TLV to the router. Note: If the trace view shows a warning, this can be ignored.
20. Verify no errors created while adding the profile to the router.
21. Repeat steps 10 to 20 to add multiple client profiles with the same request, send the bulk request, and verify the accessory successfully performs the operation.
22. Repeat steps 10 to 20 to modify a client 1 configuration, also add a remove operation for the same client 1, and verify accessory rejects the operation. Note: If HAT shows an error that modify and remove cannot be done in the same bulk operation, continue anyway.

TCSR045 **Verify the Network client profile control List operation.**

Applies to HomeKit-enabled Wi-Fi router accessories. Perform this test case using HAT and an iOS device running the accessory app.

1. Use the accessory’s app to setup and create a new wireless network.
2. Setup an ADK accessory as a HomeKit accessory
3. Join the newly created network on a Mac running HAT.
4. Open HAT, and in the left sidebar under “IP Controllers”, select “Controller 1”.
5. With “Controller 1” selected on the left, click the “Start” button under the “Discovery” section on the main screen.
6. In the left sidebar, select the router.
7. On the main screen click the “Start Pairing” button under the “Pairing” section.
8. In the popup, enter the setup payload or setup code of the router.
9. In the main screen, click the “Discover” button in the “Summary” section.
10. In the left sidebar, select “Managed Network Enable”.
11. On the main screen, navigate to the “Prepare and Execute Timed Write [Unsigned Integer]” section.
12. Enter a value of “1” in the text field and click the “Timed Write” button.
13. In the left sidebar, select “Network Client Profile Control”.
14. Under the “Write Options” section check the box next to “Write With Response”

15. On the main screen, navigate to the “Prepare and Execute Timed Write [TLV8 Packet]” section.
16. Click on the “Build TLV” button.
17. In the “Network Client Profile Control” TLV builder, set the following parameters:
    - Set the “Operation” to “Add”.
    - Leave the “Client Profile Identifier” field blank.
    - Enter “1” into the “Client Group Identifier” field.
    - Credential data: Enter “31 31 31 31 31 31 31 31” (HEX converted value of “11111111”)
    - Set the “Credential Type” to “PSK”.
    - Enter “010100” into the “WAN Firewall Config” field.
    - Enter “010100” into the “LAN Firewall Config” field.
18. Click the “Add” button to add rule to the TLV builder table.
19. Click the “Build TLV” button in the bottom right corner.
20. In the “Prepare and Execute Timed Write [TLV8 Packet]” section, click the “Timed Write” button to write this TLV to the router. Note: If the trace view shows a warning, this can be ignored.
21. Use the ADK accessory and connect to the wireless network created above with “11111111” as the Wi-Fi password.
22. On the main screen, navigate to the “Prepare and Execute Timed Write [TLV8 Packet]” section.
23. Click on the “Build TLV” button.
24. In the “Network Client Profile Control” TLV builder, set the following parameters:
    - Set the “Operation” to “List”
    - Leave the “Client Profile Identifier” field blank.
    - Enter “1” into the “Client Group Identifier” field.
25. Click the “Add” button to add rule to the TLV builder table.
26. Click the “Build TLV” button in the bottom right corner.
27. In the “Prepare and Execute Timed Write [TLV8 Packet]” section, click the “Timed Write” button to write this TLV to the router. Note: If the trace view shows a warning, this can be ignored.
28. Verify that the router return list of all the Network client profile identifiers.

TCSR046 **Verify the Network client profile control Read operation.**

Applies to HomeKit-enabled Wi-Fi router accessories. Perform this test case using HAT and an iOS device running the accessory app.

1. Use the accessory’s app to setup and create a new wireless network.
2. Setup an ADK accessory as a HomeKit accessory
3. Join the newly created network on a Mac running HAT.
4. Open HAT, and in the left sidebar under “IP Controllers”, select “Controller 1”.

5. With “Controller 1” selected on the left, click the “Start” button under the “Discovery” section on the main screen.
6. In the left sidebar, select the router.
7. On the main screen click the “Start Pairing” button under the “Pairing” section.
8. In the popup, enter the setup payload or setup code of the router.
9. In the main screen, click the “Discover” button in the “Summary” section.
10. In the left sidebar, select “Managed Network Enable”.
11. On the main screen, navigate to the “Prepare and Execute Timed Write [Unsigned Integer]” section.
12. Enter a value of “1” in the text field and click the “Timed Write” button.
13. In the left sidebar, select “Network Client Profile Control”.
14. Under the “Write Options” section check the box next to “Write With Response”
15. On the main screen, navigate to the “Prepare and Execute Timed Write [TLV8 Packet]” section.
16. Click on the “Build TLV” button.
17. In the “Network Client Profile Control” TLV builder, set the following parameters:
    - Set the “Operation” to “Add”.
    - Leave the “Client Profile Identifier” field blank.
    - Enter “1” into the “Client Group Identifier” field.
    - Credential data: Enter “31 31 31 31 31 31 31 31” (HEX converted value of “11111111”)
    - Set the “Credential Type” to “PSK”.
    - Enter “010100” into the “WAN Firewall Config” field.
    - Enter “010100” into the “LAN Firewall Config” field.
18. Click the “Add” button to add rule to the TLV builder table.
19. Click the “Build TLV” button in the bottom right corner.
20. In the “Prepare and Execute Timed Write [TLV8 Packet]” section, click the “Timed Write” button to write this TLV to the router. Note: If the trace view shows a warning, this can be ignored.
21. Use the ADK accessory and connect to the wireless network created above with “11111111” as the Wi-Fi password.
22. On the main screen, navigate to the “Prepare and Execute Timed Write [TLV8 Packet]” section.
23. Click on the “Build TLV” button.
24. In the “Network Client Profile Control” TLV builder, set the following parameters:
    - Set the “Operation” to “Read”
    - Client profile identifier: 1
25. Click the “Add” button to add rule to the TLV builder table.
26. Click the “Build TLV” button in the bottom right corner.

27. In the “Prepare and Execute Timed Write [TLV8 Packet]” section, click the “Timed Write” button to write this TLV to the router. Note: If the trace view shows a warning, this can be ignored.
28. Verify that the router return the Network client profile configuration pertaining to the client identifier.

TCSR047 **Verify the Network client profile control Remove operation.**

Applies to HomeKit-enabled Wi-Fi router accessories. Perform this test case using HAT and an iOS device running the accessory app.

1. Use the accessory’s app to setup and create a new wireless network.
2. Setup an ADK accessory as a HomeKit accessory.
3. Join the newly created network on a Mac running HAT.
4. Open HAT, and in the left sidebar under “IP Controllers”, select “Controller 1”.
5. With “Controller 1” selected on the left, click the “Start” button under the “Discovery” section on the main screen.
6. In the left sidebar, select the router.
7. On the main screen click the “Start Pairing” button under the “Pairing” section.
8. In the popup, enter the setup payload or setup code of the router.
9. In the main screen, click the “Discover” button in the “Summary” section.
10. In the left sidebar, select “Managed Network Enable”.
11. On the main screen, navigate to the “Prepare and Execute Timed Write [Unsigned Integer]” section.
12. Enter a value of “1” in the text field and click the “Timed Write” button.
13. In the left sidebar, select “Network Client Profile Control”.
14. Under the “Write Options” section check the box next to “Write With Response”
15. On the main screen, navigate to the “Prepare and Execute Timed Write [TLV8 Packet]” section.
16. Click on the “Build TLV” button.
17. In the “Network Client Profile Control” TLV builder, set the following parameters:
    - Set the “Operation” to “Add”.
    - Leave the “Client Profile Identifier” field blank.
    - Enter “1” into the “Client Group Identifier” field.
    - Credential data: Enter “31 31 31 31 31 31 31 31” (HEX converted value of “11111111”)
    - Set the “Credential Type” to “PSK”.
    - Enter “010100” into the “WAN Firewall Config” field.
    - Enter “010100” into the “LAN Firewall Config” field.
18. Click the “Add” button to add rule to the TLV builder table.
19. Click the “Build TLV” button in the bottom right corner.

20. In the “Prepare and Execute Timed Write [TLV8 Packet]” section, click the “Timed Write” button to write this TLV to the router. Note: If the trace view shows a warning, this can be ignored.
21. Use the ADK accessory and connect to the wireless network created above with “11111111” as the Wi-Fi password.
22. On the main screen, navigate to the “Prepare and Execute Timed Write [TLV8 Packet]” section.
23. Click on the “Build TLV” button.
24. In the “Network Client Profile Control” TLV builder, set the following parameters:
    - Set the “Operation” to “Remove”.
    - Enter “1” into the Client Profile Identifier field.
25. Click the “Add” button to add rule to the TLV builder table.
26. Click the “Build TLV” button in the bottom right corner.
27. In the “Prepare and Execute Timed Write [TLV8 Packet]” section, click the “Timed Write” button to write this TLV to the router. Note: If the trace view shows a warning, this can be ignored.
28. Verify that the router returns success.
29. Try to read the Client Profile Identifier which is removed in the above step and make sure nothing is returned by the router.

TCSR048 **Verify the Network client profile control Update operation.**

Applies to HomeKit-enabled Wi-Fi router accessories. Perform this test case using HAT and an iOS device running the accessory app.

1. Use the accessory’s app to setup and create a new wireless network.
2. Setup an ADK accessory as a HomeKit accessory
3. Join the newly created network on a Mac running HAT.
4. Open HAT, and in the left sidebar under “IP Controllers”, select “Controller 1”.
5. With “Controller 1” selected on the left, click the “Start” button under the “Discovery” section on the main screen.
6. In the left sidebar, select the router.
7. On the main screen click the “Start Pairing” button under the “Pairing” section.
8. In the popup, enter the setup payload or setup code of the router.
9. In the main screen, click the “Discover” button in the “Summary” section.
10. In the left sidebar, select “Managed Network Enable”.
11. On the main screen, navigate to the “Prepare and Execute Timed Write [Unsigned Integer]” section.
12. Enter a value of “1” in the text field and click the “Timed Write” button.
13. In the left sidebar, select “Network Client Profile Control”.

14. Under the “Write Options” section check the box next to “Write With Response”
15. On the main screen, navigate to the “Prepare and Execute Timed Write [TLV8 Packet]” section.
16. Click on the “Build TLV” button.
17. In the “Network Client Profile Control” TLV builder, set the following parameters:
    - Set the “Operation” to “Add”.
    - Leave the “Client Profile Identifier” field blank.
    - Enter “1” into the “Client Group Identifier” field.
    - Credential data: Enter “31 31 31 31 31 31 31 31” (HEX converted value of “11111111”)
    - Set the “Credential Type” to “PSK”.
    - Enter “010100” into the “WAN Firewall Config” field.
    - Enter “010100” into the “LAN Firewall Config” field.
18. Click the “Add” button to add rule to the TLV builder table.
19. Click the “Build TLV” button in the bottom right corner.
20. In the “Prepare and Execute Timed Write [TLV8 Packet]” section, click the “Timed Write” button to write this TLV to the router. Note: If the trace view shows a warning, this can be ignored.
21. Use the ADK accessory and connect to the wireless network created above with “11111111” as the Wi-Fi password.
22. On the main screen, navigate to the “Prepare and Execute Timed Write [TLV8 Packet]” section.
23. Click on the “Build TLV” button.
24. In the “Network Client Profile Control” TLV builder, set the following parameters:
    - Set the “Operation” to “Update”
    - Enter “1” into the Client Profile Identifier field.
    - Enter “3” into the “Client Group Identifier” field.
    - Enter “0101010200” into the WAN Firewall Config field.
    - Enter “0101010200” into the LAN Firewalls Config field.
25. Click the “Add” button to add rule to the TLV builder table.
26. Click the “Build TLV” button in the bottom right corner.
27. In the “Prepare and Execute Timed Write [TLV8 Packet]” section, click the “Timed Write” button to write this TLV to the router. Note: If the trace view shows a warning, this can be ignored.
28. Verify the rules are update on the client identifier.

TCSR049 **Verify the Network client status control operation.**

Applies to HomeKit-enabled Wi-Fi router accessories. Perform this test case using HAT and an iOS device running the accessory app.

1. Use the accessory’s app to setup and create a new wireless network.

2. Setup an ADK accessory as a HomeKit accessory
3. Join the newly created network on a Mac running HAT.
4. Open HAT, and in the left sidebar under “IP Controllers”, select “Controller 1”.
5. With “Controller 1” selected on the left, click the “Start” button under the “Discovery” section on the main screen.
6. In the left sidebar, select the router.
7. On the main screen click the “Start Pairing” button under the “Pairing” section.
8. In the popup, enter the setup payload or setup code of the router.
9. In the main screen, click the “Discover” button in the “Summary” section.
10. In the left sidebar, select “Managed Network Enable”.
11. On the main screen, navigate to the “Prepare and Execute Timed Write [Unsigned Integer]” section.
12. Enter a value of “1” in the text field and click the “Timed Write” button.
13. In the left sidebar, select “Network Client Profile Control”.
14. Under the “Write Options” section check the box next to “Write With Response”.
15. On the main screen, navigate to the “Prepare and Execute Timed Write [TLV8 Packet]” section.
16. Click on the “Build TLV” button.
17. In the “Network Client Profile Control” TLV builder, set the following parameters:
    - Set the “Operation” to “Add”.
    - Leave the “Client Profile Identifier” field blank.
    - Enter “1” into the “Client Group Identifier” field.
    - Credential data: Enter “31 31 31 31 31 31 31 31” (HEX converted value of “11111111”)
    - Set the “Credential Type” to “PSK”.
    - Enter “010100” into the “WAN Firewall Config” field.
    - Enter “010100” into the “LAN Firewall Config” field.
18. Click the “Add” button to add rule to the TLV builder table.
19. Click the “Build TLV” button in the bottom right corner.
20. In the “Prepare and Execute Timed Write [TLV8 Packet]” section, click the “Timed Write” button to write this TLV to the router. Note: If the trace view shows a warning, this can be ignored.
21. Perform WAC on the ADK accessory, using “11111111” as the Wi-Fi password.
22. In the left sidebar, select the name of the accessory that was added in step 3.
23. Go to “Wi-Fi Router” service summary page.
24. There is a section for Network Client Status form.
25. Enter either the IP address, MAC address, or client ID for a given accessory to get the client status of the client.

26. Assuming the client ID created on the router for the above client profile is “1”, enter “1” into the Client ID field for network client status, and then click “Read”.
27. View the response in the trace, and verify the router returns all of the client information which matches client ID 1.
28. Get the IP address of the ADK accessory, and enter that IP address into the IP Address field for network client status, and then click “Read”.
29. View the response in the trace, and verify the router returns all of the client information which matches the IP address entered above.
30. Get the MAC address of the ADK accessory, and enter that MAC address into the MAC Address field for Network Client Status, and then click “Read”.
31. View the response in the trace, and verify the router returns all of the client information which matches the MAC address entered above.

TCSR050 **Verify the Network Access Violation list operation.**

Applies to HomeKit-enabled Wi-Fi router accessories. Perform this test case using HAT and an iOS device running the accessory app.

1. Use the accessory’s app to setup and create a new wireless network.
2. Setup an iOS device and join the network created in step 1.
3. Join the newly created network on a Mac running HAT.
4. Open HAT, and in the left sidebar under “IP Controllers”, select “Controller 1”.
5. With “Controller 1” selected on the left, click the “Start” button under the “Discovery” section on the main screen.
6. In the left sidebar, select the router.
7. On the main screen click the “Start Pairing” button under the “Pairing” section.
8. In the popup, enter the setup payload or setup code of the router.
9. In the main screen, click the “Discover” button in the “Summary” section.
10. In the left sidebar, select “Managed Network Enable”.
11. On the main screen, navigate to the “Prepare and Execute Timed Write [Unsigned Integer]” section.
12. Enter a value of “1” in the text field and click the “Timed Write” button.
13. In the left sidebar, select “Network Client Profile Control”.
14. Under the “Write Options” section check the box next to “Write With Response”
15. On the main screen, navigate to the “Prepare and Execute Timed Write [TLV8 Packet]” section.
16. Click on the “Build TLV” button.
17. In the “Network Client Profile Control” TLV builder, set the following parameters:

- Set the “Operation” to “Add”.
- Leave the “Client Profile Identifier” field blank.
- Enter “1” into the “Client Group Identifier” field.
- Credential data: Enter “31 31 31 31 31 31 31 31” (HEX converted value of “11111111”)
- Set the “Credential Type” to “PSK”.
- Enter “01010200” into the WAN Firewall Config field.
- Enter “0101010200” into the LAN Firewalls Config field.

18. Click the “Add” button to add rule to the TLV builder table.
19. Click the “Build TLV” button in the bottom right corner.
20. In the “Prepare and Execute Timed Write [TLV8 Packet]” section, click the “Timed Write” button to write this TLV to the router. Note: If the trace view shows a warning, this can be ignored.
21. Use the Raspi ADK accessory and pair to the iOS device using WAC, using “11111111” as the Wi-Fi password.
22. Once the ADK accessory is on the network, try to access external WAN domains which are not allowed.
23. In the side bar, select “Network Access Violation Control”
24. On the main screen, navigate to the “Prepare and Execute Timed Write [TLV8 Packet]” section.
25. Click on the “Build TLV” button.
26. In the operation section select “List”, click on the “Build TLV” button, and write the value.
27. Verify the accessory returns the last violation timestamp for each client.

TCSR051 **Verify the Network Access Violation reset operation.**

Applies to HomeKit-enabled Wi-Fi router accessories. Perform this test case using HAT and an iOS device running the accessory app.

1. Use the accessory’s app to setup and create a new wireless network.
2. Setup an ADK accessory as HomeKit accessory
3. Join the newly created network on a Mac running HAT.
4. Open HAT, and in the left sidebar under “IP Controllers”, select “Controller 1”.
5. With “Controller 1” selected on the left, click the “Start” button under the “Discovery” section on the main screen.
6. In the left sidebar, select the router.
7. On the main screen click the “Start Pairing” button under the “Pairing” section.
8. In the popup, enter the setup payload or setup code of the router.
9. In the main screen, click the “Discover” button in the “Summary” section.
10. In the left sidebar, select “Managed Network Enable”.

11. On the main screen, navigate to the “Prepare and Execute Timed Write [Unsigned Integer]” section.
12. Enter a value of “1” in the text field and click the “Timed Write” button.
13. In the left sidebar, select “Network Client Profile Control”.
14. Under the “Write Options” section check the box next to “Write With Response”
15. On the main screen, navigate to the “Prepare and Execute Timed Write [TLV8 Packet]” section.
16. Click on the “Build TLV” button.
17. In the “Network Client Profile Control” TLV builder, set the following parameters:
    - Set the “Operation” to “Add”.
    - Leave the “Client Profile Identifier” field blank.
    - Enter “1” into the Client Group Identifier field.
    - Enter “31 31 31 31 31 31 31 31” (HEX converted value of “11111111”) into the Credential Data field.
    - Set the Credential Type to “PSK”.
    - Enter “01010200” into the WAN Firewall Config field.
    - Enter “01010200” into the LAN Firewall Config field.
18. Click the “Add” button to add rule to the TLV builder table.
19. Click the “Build TLV” button in the bottom right corner.
20. In the “Prepare and Execute Timed Write [TLV8 Packet]” section, click the “Timed Write” button to write this TLV to the router. Note: If the trace view shows a warning, this can be ignored.
21. Perform WAC on the ADK accessory and use “11111111” as the Wi-Fi password.
22. Once the ADK accessory is on the network, try to access external WAN domains which are not allowed.
23. In the side bar, select the “Network Access Violation Control” characteristic.
24. On the main screen, navigate to the “Prepare and Execute Timed Write [TLV8 Packet]” section.
25. Click on the “Build TLV” button.
26. Set the Operation to “reset”.
27. In the Client Identifier field, enter “1” to reset the network access violation for the client identifier 1.
28. Verify the write response contains a TLV with a zero-length value.
29. In the side bar select “network Access Violation control”
30. On the main screen, navigate to the “Prepare and Execute Timed Write [TLV8 Packet]” section.
31. Click on the “Build TLV” button.
32. In the operation section select “List”
33. In the operation section select “List”, click on the “Build TLV” button and write the value.
34. Verify the accessory returns the last violation timestamp for each client.

TCSR052 **Verify that the router disconnects all the established connections for a client when the client configuration is modified (Move client from No Restriction to Auto).**

Applies to HomeKit-enabled Wi-Fi router accessories. Perform this test case using HAT, the accessory app, a Raspberry Pi running an ADK IP accessory, and the test network declaration JSON files found on the MFi Portal.

1. Use the accessory's app to setup and create a new wireless network.
2. Join the newly created network on a Mac running HAT.
3. Connect an ADK accessory to the network from step 1 via ethernet.
4. Open HAT, and in the left sidebar under "IP Controllers", select "Controller 1".
5. With "Controller 1" selected on the left, click the "Start" button under the "Discovery" section on the main screen.
6. In the left sidebar, select the router.
7. On the main screen, enter the Ownership Proof Token if needed, then click the "Start Pairing" button under the "Pairing" section to complete pair setup.
8. In the main screen, click the "Discover" button in the "Summary" section.
9. In the left sidebar, select "Managed Network Enable".
10. On the main screen, navigate to the "Prepare and Execute Timed Write [Unsigned Integer]" section.
11. Enter a value of "1" in the text field and click the "Timed Write" button.
12. In the left sidebar, select the name of the ADK accessory.
13. On the main accessory server view, navigate to the "Network Declarations" pane and select the "Select File" button next to "Import JSON".
14. Browse and select the TCSR052.json file, downloaded from the MFi Portal.
15. Select "Apply rules" button in the "Network Declarations" section.
16. On the next screen, ensure "No Restriction" is selected at the top of screen, and press "Apply" to apply those rules to the router for this accessory.
17. Verify the rules are correctly reflected on the router by selecting the "Show Configured" button.
18. Start an iperf TCP session from ADK accessory using `#iperf –s –p 9000 –i 1`.
19. On the Mac client running HAT tool `#iperf –c <ip of ADK accessory> –p 9000 –i 1`.
20. In the left sidebar, select the name of the accessory that was added in step 3.
21. On the main screen under heading "Network Declarations" choose button "Select File" next to "Import JSON"
22. Browse and select the TCSR052.json file, downloaded from the MFi Portal.
23. Select "Apply rules" button in the "Network Declarations" section.

24. On the next screen, ensure “Auto” is selected at the top of screen, and press “Apply” to apply those rules to the router for this accessory.
25. Verify the rules are correctly reflected on the router by selecting the “Show Configured” button.
26. Verify the iperf stream started above should not be broken and session is connected.
27. In the left sidebar, select the name of the ADK accessory.
28. On the main screen under heading “Network Declarations” choose button “Select File” next to “Import JSON”
29. Browse and select the TCSR052.json file, downloaded from the MFi Portal.
30. Select “Apply rules” button in the “Network Declarations” section.
31. On the next screen, ensure “No Restriction” is selected at the top of screen, and press “Apply” to apply those rules to the router for this accessory.
32. Verify the rules are correctly reflected on the router by selecting the “Show Configured” button.
33. Start an iperf TCP session from ADK accessory on any valid port in the range 0-65535 except 9000 as `#iperf –s –p <Port Number> –i 1`.
34. On the Mac client running HAT tool `#iperf –c <ip of ADK accessory> –p <Port Number as above> –i 1`.
35. In the left sidebar, select the name of the accessory that was added in step 3.
36. On the main screen under heading “Network Declarations” choose button “Select File” next to “Import JSON”
37. Browse and select the TCSR052.json file, downloaded from the MFi Portal.
38. Select “Apply rules” button in the “Network Declarations” section.
39. On the next screen, ensure “Auto” is selected at the top of screen, and press “Apply” to apply those rules to the router for this accessory.
40. Verify the rules are correctly reflected on the router by selecting the “Show Configured” button.
41. Verify the iperf stream started above should be broken and session is disconnected.
42. Start an iperf TCP session from ADK accessory as `#iperf –s –p 9000 –i 1`.
43. On the Mac client running HAT, `#iperf –c <ip of ADK accessory> –p 9000 –i 1`.
44. In the left sidebar, select the name of the ADK accessory.
45. On the main screen under heading “Network Declarations” choose button “Select File” next to “Import JSON”
46. Browse and select the TCSR052.json file, downloaded from the MFi Portal.
47. Select “Apply rules” button in the “Network Declarations” section.
48. On the next screen, ensure “restricted” is selected at the top of screen, and press “Apply” to apply those rules to the router for this accessory.
49. Verify the rules are correctly reflected on the router by selecting the “Show Configured” button.

50. Verify the iperf stream started above should be broken and session is disconnected.

TCSR053 **Verify the Traffic between clients within a restricted group.**

Applies to HomeKit-enabled Wi-Fi router accessories. Perform this test case using HAT, the accessory app, a Raspberry Pi running an ADK IP accessory, and the test network declaration JSON files found on the MFi Portal.

1. Use the accessory's app to setup and create a new wireless network.
2. Setup an ADK1 and ADK2 devices as HomeKit accessories.
3. On the ADK1 device terminal, run `#sudo raspi-config` and follow Network Options to join the network created in step 1. Repeats the steps for ADK2.
4. Join the newly created network on a Mac running HAT.
5. Pair and discover router accessory.
6. In the left sidebar, select "Managed Network Enable".
7. On the main screen, navigate to the "Prepare and Execute Timed Write [Unsigned Integer]" section.
8. Enter a value of "1" in the text field and click the "Timed Write" button.
9. In the left sidebar, select the ADK1 accessory that was added in step 3.
10. On the main screen under heading "Network Declarations" choose button "Select File" next to "Import JSON"
11. Browse and select the TCSR053.json file, downloaded from the MFi Portal.
12. Select "Apply rules" button in the "Network Declarations" section.
13. On the next screen, ensure "Auto" is selected at the import rules line at top of page, and press "Apply" to apply those rules to the router for this accessory.
14. Verify the rules are correctly reflected on the router by selecting the "Show Configured" button.
15. In the left sidebar, select the ADK2 accessory that was added in step 3.
16. On the main screen under heading "Network Declarations" choose button "Select File" next to "Import JSON"
17. Browse and select the TCSR053.json file, downloaded from the MFi Portal.
18. Select "Apply rules" button in the "Network Declarations" section.
19. On the next screen, ensure "Auto" is selected at the import rules line at top of page, and press "Apply" to apply those rules to the router for this accessory.
20. Verify the rules are correctly reflected on the router by selecting the "Show Configured" button.
21. On ADK1 accessory, run a TCP server on a specific ports with the command `#iperf -s -p 9000 -i 1`.

22. On ADK2 running HAT, open terminal and run the command `#iperf –c < Ip of the ADK1 client running iperf s as above> –p 9000 –i 1`. Make sure the TCP traffic goes through and not get blocked.
23. Verify traffic is blocked for all other ports.

TCSR054 **Verify the Network Access Violation Control Event Notifications.**

Applies to HomeKit-enabled Wi-Fi router accessories. Perform this test case using HAT and an iOS device running the accessory app.

1. Use the accessory's app to setup and create a new wireless network.
2. Setup an ADK accessory as HomeKit accessory
3. Join the newly created network on a Mac running HAT.
4. Open HAT, and in the left sidebar under "IP Controllers", select "Controller 1".
5. With "Controller 1" selected on the left, click the "Start" button under the "Discovery" section on the main screen.
6. In the left sidebar, select the router.
7. On the main screen click the "Start Pairing" button under the "Pairing" section.
8. In the popup, enter the setup payload or setup code of the router.
9. In the main screen, click the "Discover" button in the "Summary" section.
10. In the left sidebar, select "Managed Network Enable".
11. On the main screen, navigate to the "Prepare and Execute Timed Write [Unsigned Integer]" section.
12. Enter a value of "1" in the text field and click the "Timed Write" button.
13. In the left sidebar, select "Network Client Profile Control".
14. Under the "Write Options" section check the box next to "Write With Response"
15. On the main screen, navigate to the "Prepare and Execute Timed Write [TLV8 Packet]" section.
16. Click on the "Build TLV" button.
17. In the "Network Client Profile Control" TLV builder, set the following parameters:
18. Set the "Operation" to "Add".
19. Leave the "Client Profile Identifier" field blank.
20. Enter "3" into the Client Group Identifier field.
21. Enter "31 31 31 31 31 31 31 31" (HEX converted value of "11111111") into the Credential Data field.
22. Set the Credential Type to "PSK".
23. Enter "0101010200" into the WAN Firewall Config field.

24. Enter "0101010200" into the LAN Firewall Config field.
25. Click the "Add" button to add rule to the TLV builder table.
26. Click the "Build TLV" button in the bottom right corner.
27. In the "Prepare and Execute Timed Write [TLV8 Packet]" section, click the "Timed Write" button to write this TLV to the router. Note: If the trace view shows a warning, this can be ignored.
28. Use the ADK accessory and connect to the wireless network created above with "11111111" as the WiFi password.
29. In the side bar, select the "Network Access Violation Control" characteristic.
30. Navigate to the Event Notifications in the Summary section.
31. Click on the "Enable" button.
32. On the same screen, navigate to the "Prepare and Execute Timed Write [TLV8 Packet]" section.
33. Click on the "Build TLV" button.
34. Set the Operation to "reset".
35. In the Client Identifier field, enter "1" to reset the network access violation for the client identifier 1.
36. On the ADK accessory, try to access an external WAN domain which is not allowed.
37. Router should send the event notification for the Network Access Violation with the Client Profile Identifier of the ADK accessory.

1.3 HAP

TCH001: Accessory must have a single Accessory Information Service that includes the required characteristics.

TCH002: The Accessory Information service's Manufacturer, Model, Name, Serial Number, and Product Data characteristic values must persist through the lifetime of the accessory.

TCH004: Custom characteristics must not use the HAP base UUID.

TCH005: Identify characteristic must be included in Accessory Information Service and must not be included in any other accessory service.

TCH006: Accessory must contain an accessory category identifier value that advertises accessory's primary category.

TCH007: Any other Apple-defined characteristics added to the Accessory Information service must only contain the following properties: Paired Read or Notify*. Custom characteristics added to this service must only contain the following properties: Paired Read, Notify*, and Hidden. No other permissions are allowed. * Notify (ev) for BLE encompasses Indicate, Indicate (disconnected), and Broadcast. Please note, characteristics with TLV8 or String formats must not support Broadcast Events.

TCH008: Any Fan service must include the required characteristics.

TCH009: Accessories that contain a Fan service and support Rotation Direction characteristic must use characteristic value of "0" to represent clockwise and value of "1" to represent counterclockwise. Fan direction is based on the perspective looking into the front of the fan. (i.e. Looking up at ceiling-mounted fan.)

TCH010: Any garage door opener service must include the required characteristics.

TCH011: Any Light Bulb services must include the required characteristics.

TCH012: Accessories that contain the Light Bulb service and support Brightness, Hue, Saturation, and/or Color Temperature characteristics must report the correct characteristic value after accessory is manually power-cycled.

TCH014: Any lock mechanism service must include the required characteristics.

TCH015: Accessories that contain the Bluetooth LE Lock Mechanism Service must not return errors in HAT when writing and reading values to the lock mechanism target state characteristic.

TCH016: Any outlet service must include the required characteristics.

TCH017: Accessories that contain the outlet service must update the value of Outlet In Use characteristic regardless of the On characteristic state.

TCH018: Any switch service must include the required characteristics.

TCH019: Any thermostat service must include the required characteristics.

TCH020: Accessories that contain the thermostat service must update Current Heating/Cooling Mode characteristic when the Target Temperature characteristic is reached.

TCH021: Accessories that contain the Thermostat or Heater Cooler service must represent the maximum temperature that must be reached before cooling is turned on via the Cooling Threshold Temperature characteristic.

TCH022: Accessories that contain the Thermostat or Heater Cooler service must represent the minimum temperature that must be reached before heating is turned on via the Heating Threshold Temperature characteristic.

TCH023: Any air quality sensor service must include the required characteristics.

TCH024: Any security system service must include the required characteristics.

TCH025: Any carbon monoxide sensor service must include the required characteristics.

TCH026: Any contact sensor service must include the required characteristics.

TCH027: Any door service must include the required characteristics.

TCH028: Any humidity sensor service must include the required characteristics.

TCH029: Any leak sensor service must include the required characteristics.

TCH030: Any light sensor service must include the required characteristics.

TCH031: Any motion sensor service must include the required characteristics.

TCH032: Any occupancy sensor service must include the required characteristics.

TCH033: Any smoke sensor service must include the required characteristics.

TCH035: Any stateless programmable switch service must include the required characteristics.

TCH036: Any temperature sensor service must include the required characteristics.

TCH037: Any window service must include the required characteristics.

TCH038: Any window covering service must include the required characteristics.

TCH039: Any Battery service must include the required characteristics. This service must be included for all accessories and/or bridged endpoints that can operate via battery power.

TCH040: Any carbon dioxide sensor service must include the required characteristics.

TCH042: Accessories that contain custom characteristics must comply with requirements for paired Read and paired Write characteristics.

TCH043: Accessories that contain custom services must use Apple-defined characteristics to expose functionality of accessory rather than using custom characteristics for the same functionality, if available.

TCH044: Accessory must provide correct metadata for each characteristic. For Apple-defined characteristics, verify that provided metadata matches the latest HomeKit Accessory Protocol Specification. If characteristic properties includes a value for maxLen, the value must be not be > 256. The value of the characteristic (if it supports Paired Read) must be valid according to the specified format and metadata, as applicable. For example, if the minValue/maxValue metadata are 10, and 50, then the value should not be 60.

TCH045: A service must not contain duplicate Apple-defined characteristic types.

TCH046: If the accessory supports primary service, then the primary service must match the primary function of the accessory and must also match with the accessory category. An accessory must expose only one primary service from its list of available services. A custom service cannot be set as "primary".

TCH047: Accessories may expose services that could be used to configure the accessory or to update firmware on the accessory, these services should be marked as hidden. When all characteristics in a service are marked hidden then the service is also implicitly marked as hidden.

TCH048: All camera RTP stream management services must include the required characteristics.

TCH050: Any microphone service must include the required characteristics.

TCH051: Any speaker service must include the required characteristics.

TCH052: Any doorbell service must include the required characteristics.

TCH054: Any slat service must include the required characteristics.

TCH055: Any filter maintenance service must include the required characteristics.

TCH056: Any air purifier service must include the required characteristics.

TCH057: Accessories that contain the Service Label service must use characteristic:

TCH058: Status Flags in the accessory's advertisement must have bit0 set after the last admin pairing is removed.

TCH059: After a remove pairing is completed, accessory must tear down any existing connections with the removed controller within 5 seconds. After the last pairing has been removed, subsequent pairing attempts with accessory succeed.

TCH060: Accessory Information Service must have an instance ID of "1". Instance IDs must be >= 1. Instance IDs must be an integer. Instance IDs must not be reused across characteristics or services. Each instance ID must be unique for the lifetime of the pairing.

TCH061: When an accessory is factory reset, all cryptographic keys must be erased.

TCH062: If the last admin controller pairing is removed, all pairings on the accessory must be removed.

TCH064: Use of a fan service that is linked to an air purifier service.

TCH065: Linked service metadata.

TCH066: Any heater cooler service must include the required characteristics.

TCH067: Any humidifier dehumidifier service must include the required characteristics.

TCH068: The value of this Relative Humidity Dehumidifier Threshold characteristic represents the 'maximum relative humidity' that must be reached before dehumidifier is turned on.

TCH069: The value of the Relative Humidity Humidifier Threshold characteristic represents the 'minimum relative humidity' that must be reached before humidifier is turned on.

TCH070: Primary accessory must have a single Protocol Information Service that includes the required characteristics.

TCH071: If an accessory receives a write request with Additional Authorization Data on a characteristic that does not support aa (Additional Authorization) property, accessory should ignore the authorization data and accept the write request.

TCH072: If an accessory receives a write request without Additional Authorization Data on a characteristic that requires it, accessory should return the correct status code.

TCH073: Any Irrigation System service must include the required characteristics.

TCH074: All Valve services must include the required characteristics.

TCH075: Any Faucet service must include the required characteristics.

TCH076: Services which provide either user-visible or user-interactive functionality must include the Name characteristic. (i.e. ceiling fan - the fan and light would all include a Name characteristic.)

TCH077: Verify Security System returns the correct state when Arming/Disarming system.

TCH078: Any Target Control Management service must include the required characteristics.

TCH079: Any Target Control service must include the required characteristics.

TCH080: Any Audio Stream Management service must include the required characteristics.

TCH082: Any Siri service must include the required characteristics.

TCH083: Accessories must not allow a firmware image to be downgraded after a successful firmware update. Accessory must increment the configuration number (c#) after a firmware update. After a firmware update, services and characteristics that remain unchanged must retain their previous instance IDs. Newly added services and characteristics must not reuse instance IDs from services and characteristics that were removed in the firmware update. *HAP over BLE accessories - GSN must reset back to 1 when a firmware update occurs.

TCH084: Accessories must generate new public keys for each pair-setup.

TCH085: Characteristics must use appropriate delimiters in TLV value responses.

TCH086: If a pairing for `RemovedControllerPairingIdentifier` does not exist, the accessory must return success.

TCH088: Verify the accessory reports the correct state after an obstruction is detected.

TCH089: Any Camera Event Recording Management service must include the required characteristics.

TCH090: Any Camera Operating Mode service must include the required characteristics.

TCH091: Accessory must reject write requests if the value does not match the Valid Values or fall within the Valid Value Range.

TCH092: Accessories with uint/int format characteristics that use Step Values greather than 1 must reject write requests when the written value doesn't confirm to the Step Value.

TCH093: The instance ID for each accessory, service, and characteristic object must be unique for the lifetime of the server/client pairing.

TCH094: Accessory must specify new values as part of the characteristic metadata when modifying default properties. Percentages are excluded from properties that can be overridden.

TCH095: The accessory must implement an identify routine, a means of identifying the accessory so that it can be located by the user. Unpaired Identify must be allowed only if the accessory is unpaired (i.e., it has no paired controllers). The identify routine should run no longer than five seconds.

TCH096: The accessory must implement an identify routine, a means of identifying the accessory so that it can be located by the user. The identity routine should run no longer than five seconds.

TCH097: When the value of the Color Temperature characteristic changes, the accessory must update the values of Hue and Saturation characteristics accordingly and notify the controller.

TCH098: When the value of the Hue and Saturation characteristics change, the accessory must update the value of the Color Temperature characteristic accordingly.

TCH001 Accessory must have a single Accessory Information Service that includes the required characteristics.

Applies to all accessories. Perform this test case automatically with HCA. Below are steps that can be used to perform this test case manually in HAT.

Required characteristics:

- Identify

- Manufacturer
- Model
- Name
- Serial number
- Firmware revision
- Product Data

*Bridged endpoint must not contain a Product Data characteristic in its Accessory Information service.

Optional characteristics:

- Hardware revision
- Accessory Flags

1. Pair and discover accessory.
2. In left sidebar of Controllers window, select “Accessory Information Service.”
3. Verify required characteristics are included in Accessory Information Service.
4. Verify firmware revision and Hardware revision (if applicable) is not null and is in the following format: x[.y[.z]] (e.g. “100.1.1”): <x> is the major version number, required. <y> is the minor version number, required if non-zero or <z> is present. <z> is the revision version number, required if non-zero. The firmware revision must follow the below rules: - <x> is incremented when there is a significant change. e.g.., 1.0.0, 2.0.0, 3.0.0, etc. - <y> is incremented when minor features are introduced such as 1.1.0, 2.1.0, 3.1.0, etc. - <z> is incremented when bug-fixes are introduced such as 1.0.1, 2.0.1, 3.0.1, etc. - Subsequent firmware updates can have a lower <y> version if <x> is incremented than the previous revision - Subsequent firmware updates can have a lower <z> version only if <x> or <y> is incremented than the previous revision.
5. If applicable, verify custom characteristics do not contain the write permission.

TCH002 **The Accessory Information service’s Manufacturer, Model, Name, Serial Number, and Product Data characteristic values must persist through the lifetime of the accessory.**

Applies to all accessories. Perform this test case with HAT using the steps below.

1. Pair and discover accessory.
2. In left sidebar of the Controllers window, select “Accessory Information Service.”
3. Read the Manufacturer, Model, Name, Serial Number, and Product Data characteristic and note down the value.
4. Factory reset the accessory.
5. If applicable, WAC accessory to your network.
6. Repeat steps 1, 2 and 3.

7. Verify that accessory's Manufacturer, Model, Name, Serial Number, and Product Data characteristic values persist after the reset.

TCH004 **Custom characteristics must not use the HAP base UUID.**

Applies to accessories that implement custom characteristics. Perform this test case with HAT using the steps below.

1. Pair and discover accessory.
2. In left sidebar of the Controllers window, locate any custom UUIDs.
3. Verify that the custom UUIDs do not contain HAP base UUID: -0000-1000-8000-0026BB765291.

TCH005 **Identify characteristic must be included in Accessory Information Service and must not be included in any other accessory service.**

Applies to all accessories. Perform this test case with HAT using the steps below.

1. Pair and discover accessory.
2. In left sidebar of the Controllers window, locate accessory.
3. Verify identify characteristic is included in Accessory Information Service and is not located within any other service.

TCH006 **Accessory must contain an accessory category identifier value that advertises accessory's primary category.**

Applies to all accessories. Perform this test case with HAT using the steps below.

1. Pair and discover accessory.
2. Select accessory's name in left sidebar.
3. See Advertisement Information in Controllers window.
4. See value for category identifier.
5. Verify that accessory category identifier properly identifies accessory by its primary function and is not a reserved value (i.e., 16 & 24-25 & 27 & 31 & 35+).

TCH007 **Any other Apple-defined characteristics added to the Accessory Information service must only contain the following properties: Paired Read or Notify*. Custom characteristics added to this service must only contain the following properties: Paired Read, Notify*, and Hidden. No other permissions are allowed. * Notify (ev) for BLE encompasses Indicate, Indicate (disconnected), and Broadcast. Please note, characteristics with TLV8 or String formats must not support Broadcast Events.**

Applies to all accessories. Perform this test case with HAT using the steps below.

1. Pair and discover accessory.
2. In left sidebar of Controllers window, see accessory's Accessory Information Service.
3. If there are additional Apple-defined characteristics other than the required/optional characteristics, verify they only contain the following permissions: Paired Read or Notify.
4. Verify custom characteristics witin the Accessory Information Service only contain the following permissions: Paired Read, Notify, or Broadcast, or Hidden.

TCH008 **Any Fan service must include the required characteristics.**

Applies to accessories that implement the Fan service. For HAP over Wi-Fi or Ethernet accessories, perform this test case automatically with HCA. For HAP over BLE and Thread accessories, perform this test case manually with HAT using the steps below.

Required characteristics:

- Active (r/w/ev*)
- Current Fan State** (r/ev*)
- Target Fan State** (r/w/ev*)

* Notify (ev) for BLE encompasses Indicate, Indicate (disconnected), and Broadcast. Please note, characteristics with TLV8 or String formats must not support Broadcast Events.

** If the Fan service is included in air purifier accessories, Current Fan State and Target Fan State are required characteristics.

1. Pair and discover accessory.
2. In left sidebar of Controllers window, see each of accessory's services.
3. Verify characteristics are included for each supported service type.
4. Read each paired Read characteristic and perform a valid write operation to each paired Write characteristic.
5. Verify proper values are returned for all Read characteristics and all Writes characteristics properly update accessory's current state.

TCH009 **Accessories that contain a Fan service and support Rotation Direction characteristic must use characteristic value of "0" to represent clockwise and value of "1" to represent counterclockwise. Fan direction is based on the perspective looking into the front of the fan. (i.e. Looking up at ceiling-mounted fan.)**

Applies to accessories that implement the Fan service. Perform this test case with HAT using the steps below.

1. Pair and discover accessory.
2. Write "0" to the fan service's Rotation Direction characteristic.

3. Verify fan is rotating in the clockwise direction.
4. Write “1” to fan service Rotation Direction characteristic.
5. Verify fan is rotating in counterclockwise direction.

TCH010 **Any garage door opener service must include the required characteristics.**

Applies to accessories that implement the Garage Door Opener service. Perform this test case with HAT using the steps below.

Required characteristics:

- Current Door State (r/ev*)
- Target Door State (r/w/tw/ev*)
- Obstruction Detected (r/ev*)

* Notify (ev) for BLE encompasses Indicate, Indicate (disconnected), and Broadcast. Please note, characteristics with TLV8 or String formats must not support Broadcast Events.

Accessories with this service must not use non-programmable NFC tags for setup.

1. Pair and discover accessory.
2. In left sidebar of Controllers window, see each of accessory’s services.
3. Verify characteristics are included for each supported service type.
4. Read each paired Read characteristic and perform a valid write operation to each paired Write characteristic.
5. Verify proper values are returned for all Read characteristics and all Writes characteristics properly update accessory’s current state.

TCH011 **Any Light Bulb services must include the required characteristics.**

Applies to accessories that implement the Light Bulb service. For HAP over Wi-Fi or Ethernet accessories, perform this test case automatically with HCA. For HAP over BLE and Thread accessories, perform this test case manually with HAT using the steps below.

Required characteristics:

- On (r/w/ev*)

Optional characteristics:

- Brightness (r/w/ev*)
- Hue (r/w/ev*)
- Name (r)

- Saturation (r/w/ev*)
- Color Temperature (r/w/ev*) Note: If this characteristic is included in the Light Bulb service, Hue and Saturation must not be included as optional characteristics, unless the Light Bulb service also includes the Characteristic Value Transition Control and Supported Characteristic Value Transition Configuration characteristics.
- Characteristic Value Transition Control (r/w/wr/ev*)
- Supported Characteristic Value Transition Configuration (r/ev*)

* Notify (ev) for BLE encompasses Indicate, Indicate (disconnected), and Broadcast. Please note, characteristics with TLV8 or String formats must not support Broadcast Events.

1. Pair and discover accessory.
2. In left sidebar of Controllers window, see each of the accessory's services.
3. Verify characteristics are included for each supported service type.
4. Read each paired read characteristic and perform a valid write operation to each paired write characteristic.
5. Verify proper values are returned for all paired read characteristics and all paired writes characteristics properly update accessory's current state.

TCH012 **Accessories that contain the Light Bulb service and support Brightness, Hue, Saturation, and/or Color Temperature characteristics must report the correct characteristic value after accessory is manually power-cycled.**

Applies to accessories that implement the Light Bulb service. Perform this test case with HAT using the steps below.

1. Pair and discover accessory.
2. Change the accessory's Brightness, Hue, Saturation, and/or Color Temperature characteristics to new values.
3. In the Summary panel, select the "Disconnect" button.
4. Manually power-cycle the accessory (e.g., turn lamp off, wait 30 seconds, and then power back on).
5. Read the value for the Brightness, Hue, Saturation, and/or Color Temperature. Verify the values are equal to the values set in step 2.

TCH014 **Any lock mechanism service must include the required characteristics.**

Applies to accessories that implement the Lock Mechanism service. Perform this test case with HAT using the steps below.

Required characteristics:

- Lock Current State (r/ev*)

- Lock Target State (r/w/tw/ev*)

* Notify (ev) for BLE encompasses Indicate, Indicate (disconnected), and Broadcast. Please note, characteristics with TLV8 or String formats must not support Broadcast Events.

Writable characteristics on this service may be considered Security Class characteristics, which require the use of the Timed Write operation.

Accessories with this service must not use non-programmable NFC tags for setup.

No custom characteristics are allowed on this service.

1. Pair and discover accessory.
2. In left sidebar of Controllers window, see each of accessory's services.
3. Verify characteristics are included for each supported service type.
4. Read each paired Read characteristic and perform a valid write operation to each paired Write characteristic.
5. Verify proper values are returned for all Read characteristics and all Writes characteristics properly update accessory's current state.

TCH015 **Accessories that contain the Bluetooth LE Lock Mechanism Service must not return errors in HAT when writing and reading values to the lock mechanism target state characteristic.**

Applies to accessories that implement the Lock Mechanism service. Perform this test case with HAT using the steps below.

1. Pair and discover accessory.
2. Enable notifications on the Lock Target State and Lock Current State Characteristics.
3. In the Lock Mechanism Target State characteristic, under the Prepare and Execute Timed Write panel, Select "Timed Write 1".
4. Verify characteristic write completed without errors and Lock physically changed to the locked position.
5. Verify accessory delivers notifications for Lock Current State Characteristic for value "1".
6. In the Lock Mechanism Target State characteristic, under the Prepare and Execute Timed Write panel, Select "Timed Write 0".
7. Verify characteristic write completed without errors and Lock physically changed to the unlocked position.
8. Verify accessory delivers notifications for Lock Current State Characteristic for value "0".
9. Manually change the deadbolt to the Locked Position.
10. Verify accessory delivers notifications for Lock Target State and Lock Current State Characteristic for value "1".

11. Manually change the deadbolt to the Unlocked Position.
12. Verify accessory delivers notifications for Lock Target State and Lock Current State Characteristic for value “0”.

TCH016 **Any outlet service must include the required characteristics.**

Applies to accessories that implement the Outlet service. Perform this test case automatically with HCA. Below are steps that can be used to perform this test case manually in HAT.

Required characteristics:

- On (r/w/ev*)

* Notify (ev) for BLE encompasses Indicate, Indicate (disconnected), and Broadcast. Please note, characteristics with TLV8 or String formats must not support Broadcast Events.

1. Pair and discover accessory.
2. In left sidebar of Controllers window, see each of accessory’s services.
3. Verify characteristics are included for each supported service type.
4. Read each paired Read characteristic and perform a valid write operation to each paired Write characteristic.
5. Verify proper values are returned for all Read characteristics and all Writes characteristics properly update accessory’s current state.

TCH017 **Accessories that contain the outlet service must update the value of Outlet In Use characteristic regardless of the On characteristic state.**

Applies to accessories that implement the Outlet service. Perform this test case with HAT using the steps below.

1. Set On characteristic to “1.”
2. Plug appliance in to accessory.
3. Verify Outlet In Use characteristic equals “1.”
4. Unplug appliance from accessory.
5. Verify Outlet In Use characteristic equals “0.”
6. Set On characteristic to “0.”
7. Plug appliance in to accessory.
8. Verify Outlet In Use characteristic equals “1.”
9. Unplug appliance from accessory.
10. Verify Outlet In Use characteristic equals “0.”

TCH018 **Any switch service must include the required characteristics.**

Applies to accessories that implement the Switch service. Perform this test case with HAT using the steps below.

Required characteristics:

- On (r/w/ev*)

* Notify (ev) for BLE encompasses Indicate, Indicate (disconnected), and Broadcast. Please note, characteristics with TLV8 or String formats must not support Broadcast Events.

1. Pair and discover accessory.
2. In left sidebar of Controllers window, see each of accessory's services.
3. Verify characteristics are included for each supported service type.
4. Read each paired Read characteristic and perform a valid write operation to each paired Write characteristic.
5. Verify proper values are returned for all Read characteristics and all Writes characteristics properly update accessory's current state.

TCH019 **Any thermostat service must include the required characteristics.**

Applies to accessories that implement the Thermostat service. For HAP over Wi-Fi or Ethernet accessories, perform this test case automatically with HCA. For HAP over BLE and Thread accessories, perform this test case manually with HAT using the steps below.

Required characteristics:

- Current Heating/Cooling State (r/ev*)
- Target Heating/Cooling State (r/w/ev*)
- Current Temperature (r/ev*)
- Target Temperature (r/w/ev*)
- Temperature Display Units (r/w/ev*)

* Notify (ev) for BLE encompasses Indicate, Indicate (disconnected), and Broadcast. Please note, characteristics with TLV8 or String formats must not support Broadcast Events.

1. Pair and discover accessory.
2. In left sidebar of Controllers window, see each of accessory's services.
3. Verify characteristics are included for each supported service type.
4. Read each paired Read characteristic and perform a valid write operation to each paired Write characteristic.

5. Verify proper values are returned for all Read characteristics and all Writes characteristics properly update accessory's current state.

TCH020 **Accessories that contain the thermostat service must update Current Heating/Cooling Mode characteristic when the Target Temperature characteristic is reached.**

Applies to accessories that implement the Thermostat service. Perform this test case with HAT using the steps below.

1. Heat Mode.
2. Pair and discover accessory.
3. Read the Current Temperature characteristic.
4. Set the Target Heating/Cooling Mode characteristic to "Heat."
5. Set the Target Temperature characteristic above the current temperature.
6. Verify the Current Heating/Cooling Mode is Off once the target temperature is reached.
7. Cool Mode.
8. Pair and discover accessory.
9. Read the Current Temperature characteristic.
10. Set the Target Heating/Cooling Mode characteristic to "Cool."
11. Set the Target Temperature characteristic below the current temperature.
12. Verify the Current Heating/Cooling Mode is Off once the target temperature is reached.

TCH021 **Accessories that contain the Thermostat or Heater Cooler service must represent the maximum temperature that must be reached before cooling is turned on via the Cooling Threshold Temperature characteristic.**

Applies to accessories that implement the Thermostat service. Perform this test case with HAT using the steps below.

1. For Thermostat service, set the Target Heating Cooling State characteristic state to "3" (Auto). For Heater Cooler service, set Active Characteristic to "1" (Active) and set the Target Heater Cooler State characteristic state to "0" (Heat or Cool).
2. For Thermostat service, enable notifications on the Current Heating Cooling State characteristic. For Heater Cooler service, enable notifications on the Current Heater Cooler State characteristic.
3. Read the Current Temperature characteristic.
4. Set the Cooling Threshold characteristic 3 degrees above the current temperature read in step 3.
5. Heat ambient air around accessory until the Cooling Threshold temperature is reached.

6. For Thermostat service, verify notification received on the Current Heating Cooling State characteristic for value "2" (Cool). For Heater Cooler service, verify notification received on the Current Heater Cooler State characteristic for value "3" (Cooling).

TCH022 **Accessories that contain the Thermostat or Heater Cooler service must represent the minimum temperature that must be reached before heating is turned on via the Heating Threshold Temperature characteristic.**

Applies to accessories that implement the Thermostat service. Perform this test case with HAT using the steps below.

1. For Thermostat service, set the Target Heating Cooling State characteristic state to "3" (Auto). For Heater Cooler service, set Active Characteristic to "1" (Active) and set the Target Heater Cooler State characteristic state to "0" (Heat or Cool).
2. For Thermostat service, enable notifications on the Current Heating Cooling State characteristic. For Heater Cooler service, enable notifications on the Current Heater Cooler State characteristic.
3. Read the Current Temperature characteristic.
4. Set the Heating Threshold characteristic 3 degrees below the current temperature read in step 3.
5. Cool ambient air around accessory until the Heating Threshold temperature is reached.
6. For Thermostat service, verify notification received on the Current Heating Cooling State characteristic for value "1" (Heat). For Heater Cooler service, verify notification received on the Current Heater Cooler State characteristic for value "2" (Heating).

TCH023 **Any air quality sensor service must include the required characteristics.**

Applies to accessories that implement the Air Quality Sensor service. For HAP over Wi-Fi or Ethernet accessories, perform this test case automatically with HCA. For HAP over BLE and Thread accessories, perform this test case manually with HAT using the steps below.

Required characteristics:

- Air Quality (r/ev*)

* Notify (ev) for BLE encompasses Indicate, Indicate (disconnected), and Broadcast. Please note, characteristics with TLV8 or String formats must not support Broadcast Events.

1. Pair and discover accessory.
2. In left sidebar of Controllers window, see each of accessory's services.
3. Verify characteristics are included for each supported service type.
4. Read each paired Read characteristic and perform a valid write operation to each paired Write characteristic.
5. Verify proper values are returned for all Read characteristics and all Writes characteristics properly update accessory's current state.

TCH024 **Any security system service must include the required characteristics.**

Applies to accessories that implement the Security System service. Perform this test case with HAT using the steps below.

Required characteristics:

- Security System Current State (r/ev*)
- Security System Target State (r/w/ev*)

* Notify (ev) for BLE encompasses Indicate, Indicate (disconnected), and Broadcast. Please note, characteristics with TLV8 or String formats must not support Broadcast Events.

1. Pair and discover accessory.
2. In left sidebar of Controllers window, see each of accessory's services.
3. Verify characteristics are included for each supported service type.
4. Read each paired Read characteristic and perform a valid write operation to each paired Write characteristic.
5. Verify proper values are returned for all Read characteristics and all Writes characteristics properly update accessory's current state.

TCH025 **Any carbon monoxide sensor service must include the required characteristics.**

Applies to accessories that implement the Carbon Monoxide Sensor service. Perform this test case with HAT using the steps below.

Required characteristics:

- Carbon Monoxide Detected (r/ev*)

* Notify (ev) for BLE encompasses Indicate, Indicate (disconnected), and Broadcast. Please note, characteristics with TLV8 or String formats must not support Broadcast Events.

1. Pair and discover accessory.
2. In left sidebar of Controllers window, see each of accessory's services.
3. Verify characteristics are included for each supported service type.
4. Read each paired Read characteristic and perform a valid write operation to each paired Write characteristic.
5. Verify proper values are returned for all Read characteristics and all Writes characteristics properly update accessory's current state.

TCH026 **Any contact sensor service must include the required characteristics.**

Applies to accessories that implement the Contact Sensor service. Perform this test case with HAT using the steps below.

Required characteristics:

- Contact Sensor State (r/ev*)

* Notify (ev) for BLE encompasses Indicate, Indicate (disconnected), and Broadcast. Please note, characteristics with TLV8 or String formats must not support Broadcast Events.

1. Pair and discover accessory.
2. In left sidebar of Controllers window, see each of accessory's services.
3. Verify characteristics are included for each supported service type.
4. Read each paired Read characteristic and perform a valid write operation to each paired Write characteristic.
5. Verify proper values are returned for all Read characteristics and all Writes characteristics properly update accessory's current state.

TCH027 **Any door service must include the required characteristics.**

Applies to accessories that implement the Door service. Perform this test case with HAT using the steps below.

Required characteristics:

- Current Position(r/ev*)
- Target Position (r/w/ev*)
- Position State (r/ev*)

Writable characteristics on this service may be considered Security Class characteristics, which require the use of the Timed Write operation.

* Notify (ev) for BLE encompasses Indicate, Indicate (disconnected), and Broadcast. Please note, characteristics with TLV8 or String formats must not support Broadcast Events.

1. Pair and discover accessory.
2. In left sidebar of Controllers window, see each of accessory's services.
3. Verify characteristics are included for each supported service type.
4. Read each paired Read characteristic and perform a valid write operation to each paired Write characteristic.
5. Verify proper values are returned for all Read characteristics and all Writes characteristics properly update accessory's current state.

TCH028 **Any humidity sensor service must include the required characteristics.**

Applies to accessories that implement the Humidity Sensor service. Perform this test case with HAT using the steps below.

Required characteristics:

- Current Relative Humidity (r/ev*)

* Notify (ev) for BLE encompasses Indicate, Indicate (disconnected), and Broadcast. Please note, characteristics with TLV8 or String formats must not support Broadcast Events.

1. Pair and discover accessory.
2. In left sidebar of Controllers window, see each of accessory's services.
3. Verify characteristics are included for each supported service type.
4. Read each paired Read characteristic and perform a valid write operation to each paired Write characteristic.
5. Verify proper values are returned for all Read characteristics and all Writes characteristics properly update accessory's current state.

TCH029 **Any leak sensor service must include the required characteristics.**

Applies to accessories that implement the Leak Sensor service. Perform this test case with HAT using the steps below.

Required characteristics:

- Leak Detected (r/ev*)

* Notify (ev) for BLE encompasses Indicate, Indicate (disconnected), and Broadcast. Please note, characteristics with TLV8 or String formats must not support Broadcast Events.

1. Pair and discover accessory.
2. In left sidebar of Controllers window, see each of accessory's services.
3. Verify characteristics are included for each supported service type.
4. Read each paired Read characteristic and perform a valid write operation to each paired Write characteristic.
5. Verify proper values are returned for all Read characteristics and all Writes characteristics properly update accessory's current state.

TCH030 **Any light sensor service must include the required characteristics.**

Applies to accessories that implement the Light Sensor service. Perform this test case with HAT using the steps below.

Required characteristics:

- Current Ambient Light Level (r/ev*)

* Notify (ev) for BLE encompasses Indicate, Indicate (disconnected), and Broadcast. Please note, characteristics with TLV8 or String formats must not support Broadcast Events.

1. Pair and discover accessory.
2. In left sidebar of Controllers window, see each of accessory's services.
3. Verify characteristics are included for each supported service type.
4. Read each paired Read characteristic and perform a valid write operation to each paired Write characteristic.
5. Verify proper values are returned for all Read characteristics and all Writes characteristics properly update accessory's current state.

TCH031 **Any motion sensor service must include the required characteristics.**

Applies to accessories that implement the Motion Sensor service. For HAP over Wi-Fi or Ethernet accessories, perform this test case automatically with HCA. For HAP over BLE and Thread accessories, perform this test case manually with HAT using the steps below.

Required characteristics:

- Motion Detected (r/ev*)

* Notify (ev) for BLE encompasses Indicate, Indicate (disconnected), and Broadcast. Please note, characteristics with TLV8 or String formats must not support Broadcast Events.

1. Pair and discover accessory.
2. In left sidebar of Controllers window, see each of accessory's services.
3. Verify characteristics are included for each supported service type.
4. Read each paired Read characteristic and perform a valid write operation to each paired Write characteristic.
5. Verify proper values are returned for all Read characteristics and all Writes characteristics properly update accessory's current state.

TCH032 **Any occupancy sensor service must include the required characteristics.**

Applies to accessories that implement the Occupancy Sensor service. Perform this test case with HAT using the steps below.

Required characteristics:

- Occupancy Detected (r/ev*)

* Notify (ev) for BLE encompasses Indicate, Indicate (disconnected), and Broadcast. Please note, characteristics with TLV8 or String formats must not support Broadcast Events.

1. Pair and discover accessory.
2. In left sidebar of Controllers window, see each of accessory's services.
3. Verify characteristics are included for each supported service type.
4. Read each paired Read characteristic and perform a valid write operation to each paired Write characteristic.
5. Verify proper values are returned for all Read characteristics and all Writes characteristics properly update accessory's current state.

TCH033 **Any smoke sensor service must include the required characteristics.**

Applies to accessories that implement the Smoke Sensor service. Perform this test case with HAT using the steps below.

Required characteristics:

- Smoke Detected (r/ev*)

* Notify (ev) for BLE encompasses Indicate, Indicate (disconnected), and Broadcast. Please note, characteristics with TLV8 or String formats must not support Broadcast Events.

1. Pair and discover accessory.
2. In left sidebar of Controllers window, see each of accessory's services.
3. Verify characteristics are included for each supported service type.
4. Read each paired Read characteristic and perform a valid write operation to each paired Write characteristic.
5. Verify proper values are returned for all Read characteristics and all Writes characteristics properly update accessory's current state.

TCH035 **Any stateless programmable switch service must include the required characteristics.**

Applies to accessories that implement the Stateless Programmable Switch service. Perform this test case with HAT using the steps below.

Required characteristics:

- Programmable Switch Event (r/ev*)

* Notify (ev) for BLE encompasses Indicate, Indicate (disconnected), and Broadcast. Please note, characteristics with TLV8 or String formats must not support Broadcast Events.

Each physical switch on the accessory must be represented by a unique instance of this service.

If there is only one instance of this service on the accessory, Service Label is not required and consequently Service Label Index must not be present.

See the Stateless Programmable Switch section for testing.

1. Pair and discover accessory.
2. In left sidebar of Controllers window, see each of accessory's services.
3. Verify characteristics are included for each supported service type.
4. Read each paired Read characteristic and perform a valid write operation to each paired Write characteristic.
5. Verify proper values are returned for all Read characteristics and all Writes characteristics properly update accessory's current state.

TCH036 **Any temperature sensor service must include the required characteristics.**

Applies to accessories that implement the Temperature Sensor service. Perform this test case with HAT using the steps below.

Required characteristics:

- Current Temperature (r/ev*)

* Notify (ev) for BLE encompasses Indicate, Indicate (disconnected), and Broadcast. Please note, characteristics with TLV8 or String formats must not support Broadcast Events.

1. Pair and discover accessory.
2. In left sidebar of Controllers window, see each of accessory's services.
3. Verify characteristics are included for each supported service type.
4. Read each paired Read characteristic and perform a valid write operation to each paired Write characteristic.
5. Verify proper values are returned for all Read characteristics and all Writes characteristics properly update accessory's current state.

TCH037 **Any window service must include the required characteristics.**

Applies to accessories that implement the Window service. Perform this test case with HAT using the steps below.

Required characteristics:

- Current Position(r/ev*)
- Target Position (r/w/ev*)
- Position State (r/ev*)

Writable characteristics on this service may be considered Security Class characteristics, which require the use of the Timed Write operation.

* Notify (ev) for BLE encompasses Indicate, Indicate (disconnected), and Broadcast. Please note, characteristics with TLV8 or String formats must not support Broadcast Events.

1. Pair and discover accessory.
2. In left sidebar of Controllers window, see each of accessory's services.
3. Verify characteristics are included for each supported service type.
4. Read each paired Read characteristic and perform a valid write operation to each paired Write characteristic.
5. Verify proper values are returned for all Read characteristics and all Writes characteristics properly update accessory's current state.

TCH038 **Any window covering service must include the required characteristics.**

Applies to accessories that implement the Window Covering service. For HAP over Wi-Fi or Ethernet accessories, perform this test case automatically with HCA. For HAP over BLE and Thread accessories, perform this test case manually with HAT using the steps below.

Required characteristics:

- Current Position(r/ev*)
- Target Position (r/w/ev*)
- Position State (r/ev*)

* Notify (ev) for BLE encompasses Indicate, Indicate (disconnected), and Broadcast. Please note, characteristics with TLV8 or String formats must not support Broadcast Events.

1. Pair and discover accessory.
2. In left sidebar of Controllers window, see each of accessory's services.
3. Verify characteristics are included for each supported service type.
4. Read each paired Read characteristic and perform a valid write operation to each paired Write characteristic.
5. Verify proper values are returned for all Read characteristics and all Writes characteristics properly update accessory's current state.

TCH039 **Any Battery service must include the required characteristics. This service must be included for all accessories and/or bridged endpoints that can operate via battery power.**

Applies to accessories that implement the Battery service. Perform this test case with HAT using the steps below.

Required characteristics for accessories that support HomeKit Accessory Protocol specification R16 or later:

- Status Low Battery (r/ev*)

Optional characteristics for accessories that support HomeKit Accessory Protocol specification R16 or later:

- Name (r)
- Battery Level (r/ev*)
- Charging State (r/ev*)

Required characteristics for accessories that support HomeKit Accessory Protocol specification R15 or earlier:

- Status Low Battery (r/ev*)
- Battery Level (r/ev*)
- Charging State (r/ev*)

Optional characteristics for accessories that support HomeKit Accessory Protocol specification R15 or earlier:

- Name (r)

* Notify (ev) for BLE encompasses Indicate, Indicate (disconnected), and Broadcast. Please note, characteristics with TLV8 or String formats must not support Broadcast Events.

1. Pair and discover accessory.
2. In left sidebar of Controllers window, see each of accessory's services.
3. Verify characteristics are included for each supported service type.
4. Read each paired Read characteristic and perform a valid write operation to each paired Write characteristic.
5. Verify proper values are returned for all Read characteristics and all Writes characteristics properly update accessory's current state.

TCH040 **Any carbon dioxide sensor service must include the required characteristics.**

Applies to accessories that implement the Carbon Dioxide Sensor service. Perform this test case with HAT using the steps below.

Required characteristics:

- Carbon Dioxide Detected (r/ev*)

* Notify (ev) for BLE encompasses Indicate, Indicate (disconnected), and Broadcast. Please note, characteristics with TLV8 or String formats must not support Broadcast Events.

1. Pair and discover accessory.
2. In left sidebar of Controllers window, see each of accessory's services.
3. Verify characteristics are included for each supported service type.
4. Read each paired Read characteristic and perform a valid write operation to each paired Write characteristic.
5. Verify proper values are returned for all Read characteristics and all Writes characteristics properly update accessory's current state.

TCH042 **Accessories that contain custom characteristics must comply with requirements for paired Read and paired Write characteristics.**

Applies to accessories that implement custom characteristics.

Perform this test case as a user level test. Use the associated accessory application where necessary to expose custom service/characteristic functionality.

1. Verify that each characteristic performs its intended function and responds to Read and Write requests as supported.

TCH043 **Accessories that contain custom services must use Apple-defined characteristics to expose functionality of accessory rather than using custom characteristics for the same functionality, if available.**

Applies to all accessories. Perform this test case with HAT using the steps below.

1. Pair and discover accessory.
2. In left sidebar of Controllers window, see the accessory's services and characteristics.
3. Verify that there are no custom characteristics defining any Apple-defined characteristics.

TCH044 **Accessory must provide correct metadata for each characteristic. For Apple-defined characteristics, verify that provided metadata matches the latest HomeKit Accessory Protocol Specification. If characteristic properties includes a value for maxLen, the value must be not be > 256. The value of the characteristic (if it supports Paired Read) must be valid according to the specified format and metadata, as applicable. For example, if the minValue/maxValue metadata are 10, and 50, then the value should not be 60.**

Applies to all accessories. Perform this test case automatically with HCA. Below are steps that can be used to perform this test case manually in HAT.

The following properties of an Apple-defined characteristic may be modified in order to better fit the specific application: Minimum Value, Maximum Value, Step Value, Maximum Length, Maximum Data Length.

1. Pair and discover accessory.

2. Locate accessory's response to HTTP GET / accessories in the HTTP traffic view.
3. Select Details button to reveal Details sidebar.
4. See Attribute Database Object; use disclosure arrows to show and hide details.
5. Verify the metadata matches the spec's definition. Custom characteristics should provide applicable metadata for their intended use. Only the following properties of an Apple-defined characteristic are allowed to be modified - Minimum Value, Maximum Value, Step Value, Maximum Length, Maximum Data Length.

TCH045 **A service must not contain duplicate Apple-defined characteristic types.**

Applies to all accessories. Perform this test case with HAT using the steps below.

1. Pair and discover accessory.
2. Reveal accessory's traffic view.
3. Locate accessory's Discovered Accessories in Events traffic view.
4. Select the Details button.
5. See Attribute Database Object; use disclosure arrows to show and hide details.
6. Verify service does not have duplicate Apple-defined characteristics types.

TCH046 **If the accessory supports primary service, then the primary service must match the primary function of the accessory and must also match with the accessory category. An accessory must expose only one primary service from its list of available services. A custom service cannot be set as "primary".**

Applies to all accessories. Perform this test case with HAT using the steps below.

1. Pair and discover accessory.
2. Select accessory's name in left sidebar.
3. See Advertisement information panel in Controllers window.
4. See value for category identifier.
5. Verify that accessory category identifier properly identifies accessory by its primary function and is not a reserved value (i.e., 16 & 24-25 & 27 & 31 & 35+).
6. Locate Primary Service under the accessory's name in left sidebar. For HAP over Wi-Fi or Ethernet accessories, locate accessory's Discovered Accessories in Events traffic view. For HAP over BLE accessories, in the Controllers Window, select Accessory Services and view the Summary Panel for Primary Service.
7. Verify Primary Service is set on one of the available Apple-defined services (e.g., Primary: yes).
8. Verify the primary service matches the primary function of the accessory and accessory category (e.g., an outlet must use the Outlet (#7) Category identifier and must use Outlet Service as the primary service).

TCH047 **Accessories may expose services that could be used to configure the accessory or to update firmware on the accessory, these services should be marked as hidden. When all characteristics in a service are marked hidden then the service is also implicitly marked as hidden.**

Applies to all accessories. Perform this test case with HAT using the steps below.

1. Pair and discover accessory.
2. For HAP over Wi-Fi or Ethernet accessories, locate accessory's Discovered Accessories in Events traffic view. For HAP over BLE accessories, in the Controllers Window, select Accessory Services and view the Summary Panel for Hidden Service.
3. Select the Details button.
4. See Attribute Database Object; use disclosure arrows to show and hide details.
5. Verify services that could be used to configure the accessory or to update firmware on the accessory has the characteristic property hidden, "hidden :yes" set within the Service.

TCH048 **All camera RTP stream management services must include the required characteristics.**

Applies to accessories that implement the RTP Stream Management service. Applies to accessories that support Camera Event Recording. Perform this test case with HAT using the steps below.

Required characteristics:

- Streaming Status (r/ev)
- Supported Video Stream Configuration (r)
- Supported Audio Stream Configuration (r)
- Supported RTP Configuration (r)
- Setup Endpoints (r/w)
- Selected RTP Stream Configuration (r/w)

1. Pair and discover accessory.
2. In left sidebar of Controllers window, see each of accessory's services.
3. Verify characteristics are included for each supported service type.
4. Verify the correct characteristic format and permissions are set.
5. Read each paired Read characteristic and perform a valid write operation to each paired Write characteristic.
6. Verify proper values are returned for all Read characteristics and all Writes characteristics properly update accessory's current state.

TCH050 **Any microphone service must include the required characteristics.**

Applies to accessories that implement the Microphone service. For HAP over Wi-Fi or Ethernet accessories, perform this test case automatically with HCA. For HAP over BLE and Thread accessories, perform this test case manually with HAT using the steps below.

Required characteristics:

- Mute (r/w/ev)

1. Pair and discover accessory.
2. In left sidebar of Controllers window, see each of accessory's services.
3. Verify characteristics are included for each supported service type.
4. Read each paired Read characteristic and perform a valid write operation to each paired Write characteristic.
5. Verify proper values are returned for all Read characteristics and all Writes characteristics properly update accessory's current state.

TCH051 **Any speaker service must include the required characteristics.**

Applies to accessories that implement the Speaker service. Perform this test case with HAT using the steps below.

Required characteristics:

- Mute (r/w/ev)

1. Pair and discover accessory.
2. In left sidebar of Controllers window, see each of accessory's services.
3. Verify characteristics are included for each supported service type.
4. Read each paired Read characteristic and perform a valid write operation to each paired Write characteristic.
5. Verify proper values are returned for all Read characteristics and all Writes characteristics properly update accessory's current state.

TCH052 **Any doorbell service must include the required characteristics.**

Applies to accessories that implement the Doorbell service. Perform this test case with HAT using the steps below.

Required characteristics:

- Programmable Switch Event (r/ev*)

* Notify (ev) for BLE encompasses Indicate, Indicate (disconnected), and Broadcast. Please note, characteristics with TLV8 or String formats must not support Broadcast Events.

See the Stateless Programmable Switch section for testing instructions.

1. Pair and discover accessory.
2. In left sidebar of Controllers window, see each of accessory's services.
3. Verify characteristics are included for each supported service type.
4. Read each paired Read characteristic and perform a valid write operation to each paired Write characteristic.
5. Verify proper values are returned for all Read characteristics and all Writes characteristics properly update accessory's current state.

TCH054 **Any slat service must include the required characteristics.**

Applies to accessories that implement the Slat service. Perform this test case with HAT using the steps below.

Required characteristics:

- Current Slat State (r/ev*)
- Slat Type (r)

* Notify (ev) for BLE encompasses Indicate, Indicate (disconnected), and Broadcast. Please note, characteristics with TLV8 or String formats must not support Broadcast Events.

1. Pair and discover accessory.
2. In left sidebar of Controllers window, see each of accessory's services.
3. Verify characteristics are included for each supported service type.
4. Read each paired Read characteristic and perform a valid write operation to each paired Write characteristic.
5. Verify proper values are returned for all Read characteristics and all Writes characteristics properly update accessory's current state.

TCH055 **Any filter maintenance service must include the required characteristics.**

Applies to accessories that implement the Filter Maintenance service. Perform this test case with HAT using the steps below.

Required characteristics:

- Filter Change Indication (r/ev*)

* Notify (ev) for BLE encompasses Indicate, Indicate (disconnected), and Broadcast. Please note, characteristics with TLV8 or String formats must not support Broadcast Events.

1. Pair and discover accessory.
2. In left sidebar of Controllers window, see each of accessory's services.
3. Verify characteristics are included for each supported service type.
4. Read each paired Read characteristic and perform a valid write operation to each paired Write characteristic.
5. Verify proper values are returned for all Read characteristics and all Writes characteristics properly update accessory's current state.

TCH056 **Any air purifier service must include the required characteristics.**

Applies to accessories that implement the Air Purifier service. Perform this test case with HAT using the steps below.

Required characteristics:

- Active (r/w/ev*)
- Current Air Purifier State (r/ev*)
- Target Air Purifier State (r/w/ev*)

* Notify (ev) for BLE encompasses Indicate, Indicate (disconnected), and Broadcast. Please note, characteristics with TLV8 or String formats must not support Broadcast Events.

An Air Purifier accessory service may include Rotation Speed to control fan speed if the fan cannot be independently controlled.

1. Pair and discover accessory.
2. In left sidebar of Controllers window, see each of accessory's services.
3. Verify characteristics are included for each supported service type.
4. Read each paired Read characteristic and perform a valid write operation to each paired Write characteristic.
5. Verify proper values are returned for all Read characteristics and all Writes characteristics properly update accessory's current state.

TCH057 **Accessories that contain the Service Label service must use characteristic:**

Applies to accessories that implement the Service Label service. Perform this test case with HAT using the steps below.

Required characteristics:

- Service Label Namespace (r)

1. Pair and discover accessory.
2. In left sidebar of Controllers window, see each of accessory's services.
3. Verify characteristics are included for each supported service type.
4. Read each paired Read characteristic and perform a valid write operation to each paired Write characteristic.
5. Verify proper values are returned for all Read characteristics and all Writes characteristics properly update accessory's current state.

TCH058 **Status Flags in the accessory's advertisement must have bit0 set after the last admin pairing is removed.**

Applies to all accessories. Perform this test case automatically with HCA. Below are steps that can be used to perform this test case manually in HAT.

1. Pair and discover accessory.
2. In the Controllers window, select "+" to create Controller 2.
3. Select "+" again to create Controller 3.
4. Under Controller 1, select the accessory name, under the "Add Additional Controllers" panel, select "Controller 2" as Controller, then select the "Add Controller" button.
5. Under the "Add Additional Controllers" panel, select "Controller 3" as Controller, then select the "Add Controller" button.
6. For BLE accessories, select "Disconnect".
7. Find the most recent accessory advertisement and show the details. For BLE accessories, see the BLE Discovery view. For Ethernet or Wi-Fi accessories, see the IP Discovery view. For Thread accessories, see the Thread Discovery view.
8. Verify that the Status Flags does not have bit0 set.
9. For BLE accessories, select "Discover" using Controller 1.
10. Using Controller 1, select the "Remove Pairing" button.
11. After Remove Pairing completes, find the most recent accessory advertisement and show the details.
12. If applicable, perform user action such as button press, reboot etc., for accessory to begin advertising as unpaired.
13. Verify that the Status Flags has bit0 set on the most recent accessory advertisement.

TCH059 **After a remove pairing is completed, accessory must tear down any existing connections with the removed controller within 5 seconds. After the last pairing has been removed, subsequent pairing attempts with accessory succeed.**

Applies to accessories that use HAP over Ethernet or Wi-Fi. Applies to accessories that use HAP over BLE. Perform this test case automatically with HCA. Below are steps that can be used to perform this test case manually in HAT.

1. Pair and discover accessory.
2. Select the "Remove Pairing" button.
3. For HAP over Wi-Fi or Ethernet accessories, use the HTTP traffic view to verify the delta between the response to the remove pairing request and "HTTP Disconnect" message is within 5 seconds.
4. For HAP over BLE accessories, use the HAP Transactions traffic view to verify the delta between the response to the remove pairing request and "Disconnected" message is within 5 seconds.
5. If applicable, perform user action such as button press, reboot etc., for accessory to begin advertising as unpaired.
6. In Controllers window, select "Start Pairing".
7. If existing connections with the removed controller have not been closed, the accessory must refuse any request and return the appropriate HAP status response. IP: -70401 (Request denied due to insufficient privileges). BLE: 0x03 (Insufficient Authorization).
8. Complete Pair-Setup and then select "Discover".
9. Verify that Pair-Setup and Pair-Verify complete successfully.

TCH060 **Accessory Information Service must have an instance ID of "1". Instance IDs must be >= 1. Instance IDs must be an integer. Instance IDs must not be reused across characteristics or services. Each instance ID must be unique for the lifetime of the pairing.**

Applies to all accessories. Perform this test case with HAT using the steps below.

1. Pair and discover accessory.
2. Locate accessory's Discovered Accessories in Events traffic view.
3. Select the Details button.
4. See Attribute Database Object; use disclosure arrows to show and hide details. Note of the instance ID for each service and characteristic.
5. Verify that accessory's Accessory Information Service has an instance ID of "1.".
6. Verify the accessory object does not contain services with duplicate instance IDs.
7. Verify the accessory Service does not contain characteristics with duplicate instance IDs.
8. In the Summary panel, select the "Disconnect" button.
9. Power accessory off and on.
10. See Attribute Database, Verify that the instance ID values within each accessory object remain the same after power cycle.

TCH061 **When an accessory is factory reset, all cryptographic keys must be erased.**

Applies to all accessories. Perform this test case with HAT using the steps below.

1. Pair and discover accessory.
2. Factory reset the accessory.
3. Verify controller can no longer communicate with accessory.
4. Manually delete accessory key from controller key store.
5. Pair and discover accessory.
6. Verify accessory can successfully pair to the controller and that Pair Setup and Pair Verify complete successfully again.
7. For HAP over BLE accessories, verify Global State Number (GSN) resets back to "1" after a factory reset.

TCH062 **If the last admin controller pairing is removed, all pairings on the accessory must be removed.**

Applies to all accessories. Perform this test case automatically with HCA. Below are steps that can be used to perform this test case manually in HAT.

1. Pair and discover accessory (Controller 1).
2. In Controllers window, select "+" to create a new BLE/IP/Thread Controller.
3. Continue to add new virtual controllers until 7 additional controllers have been added.
4. Using Controller 1, select the accessory name. In the "Add Additional Controllers" pane, select Controller 2 as Controller, ensure the checkbox for Admin is disabled, and select the "Add Controller" button.
5. Repeat step 4 with each additional controller until 8 pairings have been established.
6. Select the "List Pairings" button in the Controllers window. See accessory's response to List Pairings completed. Verify accessory has 8 pairings.
7. On the left pane of the Controllers window, select the accessory name under each additional Controller, select the "Start" button, and select the "Discover" button.
8. Under Controller 1, select the accessory, and select the "Remove Pairing" button.
9. From each previously paired Controller, perform a Read operation on the Name characteristic in the Accessory Information service.
10. For IP and BLE accessories, verify that the accessory rejects establishing a secure session with TVL8 error: Type 0x07 (Error), Value: 2 (kTLVError_Authentication) <070102>. For Thread accessories, verify the Read request is rejected with CoAP status 4.04.
11. Repeat steps 9 and 10 for each non-admin Controller.

TCH064 **Use of a fan service that is linked to an air purifier service.**

Applies to accessories that implement a Fan v2 service that is linked to an Air Purifier service. Perform this test case with HAT using the steps below.

If Fan is included as a linked service in an air purifier accessory:

- Changing Active characteristic on the Air Purifier service must result in corresponding change to Active characteristic on the Fan service.
- Changing Active characteristic on the Fan service from Active to Inactive must result in the Active on the Air Purifier service to change to Inactive.
- Changing Active characteristic on the Fan service from Inactive to Active does not require the Active Characteristic on the Air Purifier service to change. This enables Fan Only mode on air purifier.
- An air purifier accessory service may include Rotation Speed characteristic to control fan speed if the fan cannot be independently controlled.

1. Pair and discover accessory.
2. Enable notifications on the Active characteristics in the Air Purifier and Fan V2 services.
3. Write "0" to the Air Purifier Service's Active characteristic.
4. Verify the Air Purifier Service's Current Air Purifier State Characteristic is "0" (Inactive). Verify Fan v2 Service Active characteristic is "0" (Inactive). If supported, Verify Fan v2 Service's Current Fan State is "0" (Inactive).
5. Write "1" to Air Purifier Service's Active characteristic.
6. Verify the Air Purifier Service's Current Air Purifier State Characteristic is "1" (Active). Verify Fan v2 Service Active characteristic is "1" (Active).
7. Write "0" to the Fan v2 Service's Active characteristic.
8. Verify the Air Purifier Service's Current Air Purifier State Characteristic is "0" (Inactive). If supported, Verify Fan v2 Service's Current Fan State is "0" (Inactive).

TCH065 **Linked service metadata.**

Applies to all accessories. Perform this test case with HAT using the steps below.

IP: the "linked" key is optional in the service object. If present, it must be an array of Numbers containing instance IDs of the services that this service links to.

BLE: Accessories must include a HAP-Param-Linked-Services parameter in any service signature read response as an array of two-byte linked services instance IDs. When the service does not link to other services it must return an empty list with length 0.

1. Pair and discover accessory.
2. For HAP over Wi-Fi or Ethernet accessories, locate the HTTP response to GET /accessories in the IP HTTP traffic view. Select Details and select Event. Locate the "linked" property. If present, verify it's a list of IDs the service links to.

3. For HAP over BLE accessories, locate the Service Signature Read Response in the BLE HAP Procedures traffic view. Select Details and select Event. Locate the HAP Linked Services. If the service does not link to other services, it must return an empty list with length = 0 (0 bytes). If used, verify list of IDs the service links to.

TCH066 **Any heater cooler service must include the required characteristics.**

Applies to accessories that implement the Heater Cooler service. Perform this test case with HAT using the steps below.

Required characteristics:

- Active (r/w/ev*)
- Current Temperature (r/ev*)
- Target Heater Cooler State (r/w/ev*)
- Current Heater Cooler State (r/ev*)

* Notify (ev) for BLE encompasses Indicate, Indicate (disconnected), and Broadcast. Please note, characteristics with TLV8 or String formats must not support Broadcast Events.

A heater must include the Heating Threshold Temperature characteristic. A cooler must include the Cooling Threshold Temperature characteristic. A heater cooler service may include the Rotation Speed characteristic to control fan speed if the fan cannot be independently controlled.

1. Pair and discover accessory.
2. In the left sidebar of Controllers window, see each of the accessory's services.
3. Verify that the characteristics are included for each supported service type.
4. Read each paired Read characteristic and perform a valid write operation to each paired Write characteristic.
5. Verify that proper values are returned for all Read characteristics and all Writes characteristics properly update accessory's current state.

TCH067 **Any humidifier dehumidifier service must include the required characteristics.**

Applies to accessories that implement the Humidifier/Dehumidifier service. Perform this test case with HAT using the steps below.

Required characteristics:

- Active (r/w/ev*)
- Current Relative Humidity (r/ev*)
- Target Humidifier Dehumidifier State (r/w/ev*)
- Current Humidifier Dehumidifier State (r/ev*)

* Notify (ev) for BLE encompasses Indicate, Indicate (disconnected), and Broadcast. Please note, characteristics with TLV8 or String formats must not support Broadcast Events.

- A dehumidifier must include Relative Humidity Dehumidifier Threshold characteristic.
- A humidifier must include Relative Humidity Humidifier Threshold characteristic.
- A humidifier/dehumidifier accessory service may include Rotation Speed characteristic to control fan speed if the fan cannot be independently controlled.

1. Pair and discover accessory.
2. In left sidebar of Controllers window, see each of accessory's services.
3. Verify characteristics are included for each supported service type.
4. Read each paired Read characteristic and perform a valid write operation to each paired Write characteristic.
5. Verify proper values are returned for all Read characteristics and all Writes characteristics properly update accessory's current state.

TCH068 **The value of this Relative Humidity Dehumidifier Threshold characteristic represents the 'maximum relative humidity' that must be reached before dehumidifier is turned on.**

Applies to accessories that implement the Humidifier/Dehumidifier service. Perform this test case with HAT using the steps below.

1. Verify that Dehumidifier turns on when the maximum relative humidity is reached.

TCH069 **The value of the Relative Humidity Humidifier Threshold characteristic represents the 'minimum relative humidity' that must be reached before humidifier is turned on.**

Applies to accessories that implement the Humidifier/Dehumidifier service. Perform this test case with HAT using the steps below.

1. Verify that Humidifier turns on when the minimum relative humidity is reached.

TCH070 **Primary accessory must have a single Protocol Information Service that includes the required characteristics.**

Applies to all accessories. Perform this test case automatically with HCA. Below are steps that can be used to perform this test case manually in HAT.

Required characteristics:

- Protocol Version (r)

For a bridge accessory, only the primary HAP accessory must contain this service.

The "pv" Key in Bonjour TXT record is truncated to "X.Y".

1. Pair and discover accessory.
2. In left sidebar of Controllers window, see accessory's Protocol Information Service.
3. Verify that the required characteristics are included in the Protocol Information Service.
4. Read the Protocol Version characteristic.
5. For HAP over BLE accessories, verify protocol version value is "2.2.0". For HAP over Wi-Fi or Ethernet accessories, verify protocol version value is "1.1.0". For HAP over Thread, verify protocol version value is "1.2.0".

TCH071 **If an accessory receives a write request with Additional Authorization Data on a characteristic that does not support aa (Additional Authorization) property, accessory should ignore the authorization data and accept the write request.**

Applies to accessories that support HomeKit Accessory Protocol specification R15 or earlier. Perform this test case automatically with HCA. Below are steps that can be used to perform this test case manually in HAT.

1. Pair and discover accessory.
2. Write with Additional Authorization Data to characteristic that does not support the aa (Additional Authorization) property.
3. Verify accessory ignores authorization data and accepts write request.

TCH072 **If an accessory receives a write request without Additional Authorization Data on a characteristic that requires it, accessory should return the correct status code.**

Applies to accessories that support HomeKit Accessory Protocol specification R15 or earlier. Perform this test case with HAT using the steps below.

1. IP Accessory
2. Pair and discover accessory.
3. Write to a characteristic that supports authorization data but do not include authorization data.
4. Verify HTTP status code -70411 (Insufficient Authorization) is returned.
5. Bluetooth LE Accessory
6. Pair and discover accessory.
7. Write to a characteristic that supports authorization data but do not include authorization data.
8. Verify status code 0x03 (Insufficient Authorization) is returned.

TCH073 **Any Irrigation System service must include the required characteristics.**

Applies to accessories that implement the Irrigation System service. Perform this test case with HAT using the steps below.

Required characteristics:

- Active (r/w/ev*)
- Program Mode (r/ev*)
- In Use (r/ev*)

* Notify (ev) for BLE encompasses Indicate, Indicate (disconnected), and Broadcast. Please note, characteristics with TLV8 or String formats must not support Broadcast Events.

- See Irrigation System and Valve Section for testing.
- This service must be present on an irrigation systems which supports on-device schedules or supports a top-level Active control across multiple valves.
- A sprinkler system accessory may be:
  - a combination of Irrigation System service on a bridge accessory with a collection of one or more Valve services (with Valve Type set to Irrigation) as bridged accessories (The bridge accessory is typically connected to the each valves using wires). OR
  - a combination of Irrigation System service with a collection of one or more linked Valve services (with Valve Type set to Irrigation) (The valves are colocated e.g. hose based system). OR
  - a combination Valve service(s) with Valve Type set to Irrigation (e.g., a system with one or more valves which does not support scheduling).

1. Pair and discover accessory.
2. In left sidebar of Controllers window, see each of accessory's services.
3. Verify characteristics are included for each supported service type.
4. Read each paired Read characteristic and perform a valid write operation to each paired Write characteristic.
5. Verify proper values are returned for all Read characteristics and all Writes characteristics properly update accessory's current state.

TCH074 **All Valve services must include the required characteristics.**

Applies to accessories that implement the Valve service. Perform this test case with HAT using the steps below.

Required characteristics:

- Active (r/w/ev*)
- In Use (r/ev*)
- Valve Type (r/ev*)

* Notify (ev) for BLE encompasses Indicate, Indicate (disconnected), and Broadcast. Please note, characteristics with TLV8 or String formats must not support Broadcast Events.

- See Valve Section for testing.

1. Pair and discover accessory.
2. In left sidebar of Controllers window, see each of accessory's services.
3. Verify characteristics are included for each supported service type.
4. Read each paired Read characteristic and perform a valid write operation to each paired Write characteristic.
5. Verify proper values are returned for all Read characteristics and all Writes characteristics properly update accessory's current state.

TCH075 **Any Faucet service must include the required characteristics.**

Applies to accessories that implement the Faucet service. Perform this test case with HAT using the steps below.

Required characteristics:

- Active (r/w/ev*)

* Notify (ev) for BLE encompasses Indicate, Indicate (disconnected), and Broadcast. Please note, characteristics with TLV8 or String formats must not support Broadcast Events.

- See Faucet and Valve Section for testing.
- This service must only be included when an accessory has either a linked "Heater Cooler" Service with single linked "Valve" Service or multiple linked "Valve" Services (with/without "Heater Cooler" service)to describe water outlets. This service serves as a top level service for such accessories.
- A faucet which supports heating of water must include "Heater Cooler" and "Valve" service as linked services. An accessory which supports one or multiple water outlets and heating of water through a common temperature control, must include "Heater Cooler" and "Valve" service(s) as linked services to the faucet service.

1. Pair and discover accessory.
2. In left sidebar of Controllers window, see each of accessory's services.
3. Verify characteristics are included for each supported service type.
4. Read each paired Read characteristic and perform a valid write operation to each paired Write characteristic.
5. Verify proper values are returned for all Read characteristics and all Writes characteristics properly update accessory's current state.

TCH076 **Services which provide either user-visible or user-interactive functionality must include the Name characteristic. (i.e. ceiling fan - the fan and light would all include a Name characteristic.)**

Applies to all accessories. Perform this test case with HAT using the steps below.

1. Pair and discover accessory.
2. In left sidebar of the Controllers window, inspect each of the accessory's services.
3. Verify the "Name" characteristic is included for each service type which provides user-visible or user-interactive functionality.
4. Verify that all other services do not include this characteristic (e.g., there may be an additional service providing firmware update capabilities that should not include the characteristic. The characteristics contained in this service are not meant to be user-visible or user-interactive, thus this service would not contain a "Name" characteristic).

TCH077 **Verify Security System returns the correct state when Arming/Disarming system.**

Applies to all accessories. Perform this test case with HAT using the steps below.

1. Pair and discover accessory.
2. Enable notifications on all Security System service characteristics and associated characteristic end-points.
3. Verify Security System is ready to be armed. i.e Verify all contact sensors show value "0" (Contact is detected) and that door/window is closed and/or no Motion is detected.
4. Write "0'' (Stay Arm) to the Security System Target State characteristic.
5. Verify a notification was received from the Security System Current State characteristic of value "0" (Stay Arm).
6. Write "3'' (Disarm) to the Security System Target State characteristic.
7. Verify a notification was received from the Security System Current State characteristic of value "3" (Disarmed).
8. Verify Security System is not ready to be armed. i.e Remove contact from one of the contact sensors by opening door/window and/or motion detected.
9. Write "1'' (Away Arm) to the Security System Target State characteristic and verify it completes without error.
10. Verify a notification was received from the Security System Target State of value "3" (Disarm).
11. Verify Security System Current State characteristic value remained value "3" (Disarmed).

TCH078 **Any Target Control Management service must include the required characteristics.**

Applies to accessories that implement the Target Control Management service. Perform this test case with HAT using the steps below.

Required characteristics:

- Target Control Supported Configuration (r)
- Target Control List (w,r,wr)

If an accessory can support control of multiple concurrent Apple TVs at the same time without requiring the user to select an Apple TV on the remote UI, it must expose multiple instances of this service.

1. Pair and discover accessory.
2. In left sidebar of Controllers window, see each of accessory's services.
3. Verify characteristics are included for each supported service type.
4. Read each paired Read characteristic and perform a valid write operation to each paired Write characteristic.
5. Verify proper values are returned for all Read characteristics and all Writes characteristics properly update accessory's current state.

TCH079 **Any Target Control service must include the required characteristics.**

Applies to accessories that implement the Target Control service. Perform this test case with HAT using the steps below.

Required characteristics:

- Active Identifier (r/ev)
- Active (w/r/ev)
- Button Event (r/ev)

1. Pair and discover accessory.
2. In left sidebar of Controllers window, see each of accessory's services.
3. Verify characteristics are included for each supported service type.
4. Read each paired Read characteristic and perform a valid write operation to each paired Write characteristic.
5. Verify proper values are returned for all Read characteristics and all Writes characteristics properly update accessory's current state.

TCH080 **Any Audio Stream Management service must include the required characteristics.**

Applies to accessories that implement the Audio Stream Management service. Perform this test case with HAT using the steps below.

Required characteristics:

- Supported Audio Stream Configuration (r)
- Selected Audio Stream Configuration (w/r)

1. Pair and discover accessory.
2. In left sidebar of Controllers window, see each of accessory's services.

3. Verify characteristics are included for each supported service type.
4. Read each paired Read characteristic and perform a valid write operation to each paired Write characteristic.
5. Verify proper values are returned for all Read characteristics and all Writes characteristics properly update accessory's current state.

TCH082 **Any Siri service must include the required characteristics.**

Applies to accessories that implement the Siri service. Perform this test case with HAT using the steps below.

Required characteristics:

- Siri Input Type (r)

1. Pair and discover accessory.
2. In left sidebar of Controllers window, see each of accessory's services.
3. Verify characteristics are included for each supported service type.
4. Read each paired Read characteristic and perform a valid write operation to each paired Write characteristic.
5. Verify proper values are returned for all Read characteristics and all Writes characteristics properly update accessory's current state.

TCH083 **Accessories must not allow a firmware image to be downgraded after a successful firmware update. Accessory must increment the configuration number (c#) after a firmware update. After a firmware update, services and characteristics that remain unchanged must retain their previous instance IDs. Newly added services and characteristics must not reuse instance IDs from services and characteristics that were removed in the firmware update. *HAP over BLE accessories - GSN must reset back to 1 when a firmware update occurs.**

Applies to all accessories. Perform this test case with HAT using the steps below.

1. Before performing the firmware update, pair and discover the accessory with HAT.
2. Inspect the accessory's most recent advertisement and note the Device ID and Config Number. For HAP over Wi-Fi or Ethernet accessories, use the Bonjour Discovery traffic view. For HAP over BLE accessories, use the BLE Discovery traffic view.
3. See Discovered Accessories in the Events traffic view.
4. Note the instance ID for each service and characteristic in the Attribute Database.
5. Note the Firmware Revision characteristic value.
6. Select the "Disconnect" button to close the connection.
7. If the accessory requires being paired with an iOS controller to perform the update, then remove pairing from HAT and pair accessory with the Home app. Using the Home app, ensure each of the accessory's user-interactive tiles are visible and do not report any errors.

8. Perform the firmware update on the accessory. For bridge accessories adding support for new endpoints, add the endpoints after the firmware update completes. Do not factory reset the accessory.
9. If pairing was removed in step 6 and an iOS device was used to perform the firmware update, use the Home app to ensure each of the accessory's user-interactive tiles remain visible and do not report any errors. Toggle any writable characteristic states and verify "No Response", "Accessory is unreachable", or other errors are not encountered. Remove the accessory from the Home App.
10. Wait for the accessory to begin advertising again. If applicable, perform user action such as button press, reboot etc., for accessory to begin advertising as unpaired.
11. Using HAT, inspect the accessory's most recent advertisement. Verify the DeviceID remains the same and Config Number has incremented the value found in step 2 by only 1. For HAP over BLE accessories, verify the Global State Number (GSN) has reset back to "1".
12. Pair, if applicable, and discover accessory using HAT.
13. See Discovered Accessories in the Events traffic view.
14. Note the instance ID for each service and characteristic in the Attribute Database.
15. If services or characteristics were removed or added by the current firmware update, verify accessory did not reuse or replace a previously used instance ID(s).
16. Verify Firmware Revision characteristic value has incremented.

TCH084 **Accessories must generate new public keys for each pair-setup.**

Applies to all accessories. Perform this test case with HAT using the steps below.

1. Pair and discover.
2. For HAP over Wi-Fi or Ethernet accessories, in HTTP traffic view, locate the response to pair setup and select the details button. Verify State is Value "2'' (M2) and note the Public Key.
3. For HAP over BLE accessories, in BLE HAP Procedures traffic view, locate the write response to pair setup and select the details. Verify State is Value "2'' (M2) and note the Public Key.
4. In the Pairing Panel, select the "Remove Pairing'' button
5. Pair and discover again.
6. For HAP over Wi-Fi or Ethernet accessories, in HTTP traffic view, locate the response to pair setup and select the details button. Verify State is Value "2'' (M2) and verify the Public Key differs from step 2.
7. For HAP over BLE accessories, in BLE HAP Procedures traffic view, locate the write response to pair setup and select the details button. Verify State is Value "2'' (M2) and verify the Public Key differs from step 3.

TCH085 **Characteristics must use appropriate delimiters in TLV value responses.**

Applies to all accessories. Perform this test case with HAT using the steps below.

- This applies to the following characteristics which may support delimiters:

- Logs
- Display Order
- Target Control List

1. Pair and discover.
2. If one or more of the above characteristics are supported on the accessory, read the value of each characteristic containing a delimited TLV.
3. Verify proper values are returned and no errors appear in the Events traffic view.

TCH086 **If a pairing for `RemovedControllerPairingIdentifier` does not exist, the accessory must return success.**

Applies to all accessories. Perform this test case with HAT using the steps below.

1. Pair and discover accessory with Controller 1.
2. In Controllers window, select "+'' to create a new IP or BLE Controller 2.
3. Under Controller 1, select the accessory name. Under "Remove Additional Controllers" panel, select "Controller 2" as Controller and select the "Remove Controller" button.
4. For HAP over Wi-Fi or Ethernet accessories, in the HTTP traffic view, verify the accessory responds with HTTP 200 OK.
5. For HAP over BLE accessories, in the HAP Procedures traffic view, verify the Write Response does not contain an error.

TCH088 **Verify the accessory reports the correct state after an obstruction is detected.**

Applies to accessories that implement the Garage Door Opener service. Perform this test case with HAT using the steps below.

1. Pair and discover accessory.
2. Write "0" (open) to the Target Door State characteristic, and wait for the door to stop in the fully open position.
3. Enable notifications on the Garage Door Opener Service's Current Door State, Target Door State, and Obstruction Detected Characteristic.
4. Write "1" (Closed) to the Target Door State Characteristic.
5. When the garage door is half way closed, trigger the obstruction sensors.
6. Verify a notification is received for "Obstruction Detected".
7. Verify visually that the garage door returns to the open position.
8. Verify notifications received for Target and Current State characteristic with a value of "0" (Open).

TCH089 **Any Camera Event Recording Management service must include the required characteristics.**

Applies to accessories that support Camera Event Recording. Perform this test case with HAT using the steps below.

Required characteristics:

- Active (r/w/tw/ev)
- Selected Camera Recording Configuration (r/w/ev)
- Supported Audio Recording Configuration (r/ev)
- Supported Video Recording Configuration (r/ev)
- Supported Camera Recording Configuration (r/ev)
- Recording Audio Active (r/w/tw/ev)

1. Pair and discover accessory.
2. In left sidebar of Controllers window, see each of accessory's services.
3. Verify characteristics are included for each supported service type.
4. Verify the correct characteristic format and permissions are set.
5. Read each paired Read characteristic and perform a valid write operation to each paired Write characteristic.
6. Verify proper values are returned for all Read characteristics and all Writes characteristics properly update accessory's current state.

TCH090 **Any Camera Operating Mode service must include the required characteristics.**

Applies to accessories that support Camera Event Recording. Perform this test case with HAT using the steps below.

Required characteristics for R15 or earlier specifications:

- HomeKit Camera Active (r/w/ev)
- Event Snapshots Active (r/w/ev)
- Periodic Snapshots Active (r/w/ev)

Optional characteristics for R15 or earlier specifications:

- Camera Operating Mode Indicator (r/w/ev)
- Diagonal Field of View (r/ev)
- Manually Disabled (r/ev)
- Night Vision (r/w/ev)
- Third Party Camera Active (r/ev)

Required characteristics for R16 or later specifications:

- HomeKit Camera Active (r/w/ev)
- Event Snapshots Active (r/w/ev)
- Periodic Snapshots Active (r/w/ev)
- Camera Operating Mode Indicator (r/w/ev)

Optional characteristics for R16 or later specifications:

- Diagonal Field of View (r/ev)
- Manually Disabled (r/ev)
- Night Vision (r/w/ev)
- Third Party Camera Active (r/ev)

1. Pair and discover accessory.
2. In left sidebar of Controllers window, see each of accessory's services.
3. Verify characteristics are included for each supported service type.
4. Verify the correct characteristic format and permissions are set.
5. Read each paired Read characteristic and perform a valid write operation to each paired Write characteristic.
6. Verify proper values are returned for all Read characteristics and all Writes characteristics properly update accessory's current state.

TCH091 **Accessory must reject write requests if the value does not match the Valid Values or fall within the Valid Value Range.**

Applies to all accessories. Perform this test case with HAT using the steps below.

1. Pair and discover accessory.
2. Navigate to a characteristic that has Valid Value or Valid Value Range metadata, either specified in the accessory's metadata or Apple-defined.
3. Write a value that is outside of the Valid Values or Valid Value Range (e.g., if Valid Values are 0,1,2,3, then write 4. If the valid values are 1,3,4, then write 2).
4. For HAP over Wi-Fi or Ethernet accessories, verify the accessory rejects the write request with the HAP status code -70410. For HAP over BLE accessories, verify in the BLE HAP Transactions traffic view, that the accessory responds with HAP status code (0x06).
5. Read characteristic and verify that accessory did not accept the invalid value.
6. Repeat steps 2-5 for each characteristic with Valid Values or Valid Value Range.

TCH092 **Accessories with uint/int format characteristics that use Step Values greather than 1 must reject write requests when the written value doesn't confirm to the Step Value.**

Applies to all accessories. Perform this test case with HAT using the steps below.

1. Pair and discover accessory.
2. Navigate to a characteristic that has Step Value metadata, either specified in the accessory's metadata or Apple-defined.
3. For uint/int formats with a Step Value greater than 1, write a value that does not conform to the Step Value (e.g., if Step Value is 2 and the Min Value is 0, then write 1).
4. For HAP over Wi-Fi or Ethernet accessories, verify the accessory rejects the write request with the HAP status code -70410. For HAP over BLE accessories, verify that the accessory does not update the characteristic to an out of bounds value. In the BLE HAP Transactions traffic view, verify accessory responds with HAP status code (0x06).
5. Repeat steps 2-4 for each characteristic with Step Value greater than 1.

TCH093 **The instance ID for each accessory, service, and characteristic object must be unique for the lifetime of the server/client pairing.**

Applies to all accessories. Perform this test case with HAT using the steps below.

1. Pair and discover accessory with HAT.
2. See "Discovered Accessories" message in the Events traffic view.
3. Select "Details" button and view attribute database. Notate the instance ID for each accessory, service, and characteristic.
4. In the Summary panel, select the "Disconnect" button.
5. Remove power from the accessory and power on again. i.e unplug power cord and plug in again, or remove and reinstall batteries.
6. After the accessory begins to advertise again, select "Discover" button in the Pairing panel.
7. Verify pair-verify completes successfully.
8. See "Discovered Accessories" message in the Events traffic view.
9. Select "Details" button and view attribute database.
10. Verify the attribute database remains the same from step 3, ensuring that the instance ID for each accessory, service, and characteristic remains unchanged.

TCH094 **Accessory must specify new values as part of the characteristic metadata when modifying default properties. Percentages are excluded from properties that can be overridden.**

Applies to all accessories. Perform this test case with HAT using the steps below.

The following properties of an Apple-defined characteristic may be modified in order to better fit the specific application:

- Minimum Value
- Maximum Value

- Step Value
- Maximum Length
- Maximum Data Length

1. Pair and discover accessory.
2. In the Events view of the trace, select the "Discovered Accessories" event and select Details button to reveal Details sidebar to verify the following steps.
3. Verify any modified metadata is present in the accessory attribute database (e.g. the modified min, max, max data length, max data, and/or valid values must be included in the characteristic metadata).
4. Verify any modified metadata consists of only Minimum Value, Maximum Value, Step Value, Maximum Length, and/or Maximum Data Legnth. Verify other metadata has not been modified and matches specification characteristic definition.
5. If a characteristic contains the unit "percentage" and metadata is present, verify metadata matches the specification characteristic definition.
6. Verify any accessory characteristic that supports only a sub-set of the Apple-defined enum values indicates these values as part of the characteristic's metadata (e.g., if the specification defines valid values "0,1,2,3,4", and the accessory only supports a subset of those valid values, for example "0,2,4", then the new valid values must be present in the characteristic metadata as "0,2,4").
7. If a modified characteristic includes min, max, and valid values, verify the valid values align with the new min and max or vice versa (e.g., if the specification defines a min value of "0", a max value of "3", and valid values of "0,1,2,3", and the max value has been modified to "1", then the valid values must also be updated to "0,1". If the valid values have been upated to "1,2,3", then the min value must present in the characteristic metadata as "1").

TCH095 **The accessory must implement an identify routine, a means of identifying the accessory so that it can be located by the user. Unpaired Identify must be allowed only if the accessory is unpaired (i.e., it has no paired controllers). The identify routine should run no longer than five seconds.**

Applies to all accessories. Perform this test case with HAT using the steps below.

1. With the accessory in an unpaired state, select the accessory server. Then in the Summary panel, select the "Identify'' button for Unpaired Identify.
2. Verify the accessory identifies itself to the user, and that the identify routine runs no longer than 5 seconds.
3. For BLE accessories, select the HAP Transactions view in the trace and verify the response to Identify contains 0x00 (Success).
4. For IP accessories, select the HTTP view in the trace and verify the response to POST /Identify is HTTP/1.1 204 No Content.
5. For Thread accessories, select the CoAP Traffic view in the trace, click on Details button, and verify the Details in the CoAP Response show Code: 2.04.
6. Pair and discover the accessory with controller 1.

7. For BLE accessories, select the “Disconnect’’ button to disconnect from the accessory.
8. Create a secondary controller.
9. Using the secondary controller 2, select the accessory server. Then in the Summary panel, select the “Identify’’ button for Unpaired Identify.
10. For BLE accessories, select the HAP Transactions view in the trace and verify the response to Identify contains Status: 0x05 (Requires Authentication).
11. For IP accessories, select the HTTP view in the trace and verify the response to POST /Identify is HTTP/1.1 400 Bad Request with HAP status code -70401.
12. For Thread accessories, select the CoAP Traffic view in the trace, click on Details button, and verify the Details in the CoAP Response show Code: 4.04.
13. Verify the accessory does not identify itself to the user.

TCH096 **The accessory must implement an identify routine, a means of identifying the accessory so that it can be located by the user. The identity routine should run no longer than five seconds.**

Applies to all accessories. Perform this test case automatically with HCA. Below are steps that can be used to perform this test case manually in HAT.

1. Pair and discover accessory.
2. Select the accessory server and then in the Summary panel, select the “Identify’’ button for Paired Identify.
3. Verify the accessory identifies itself to the user and that the identify routine runs no longer than 5 seconds.
4. For HAP over BLE accessories, select the HAP Transactions view in the trace and verify the response to Identify contains 0x00 (Success).
5. For HAP over Wi-Fi or Ethernet accessories, select the HTTP view in the trace and verify the response to POST /Identify is HTTP/1.1 204 No Content.
6. For HAP over Thread accessories, select the HAP Traffic view in the trace, click on Details button, and verify the Details in the Encrypted Thread Response show Status: Success (0x00).
7. For bridged accessories, navigate to the Identify characteristic for an endpoint accessory and write a value of “1”.
8. Verify the accessory identifies itself to the user and that the identify routine runs no longer than 5 seconds.
9. Repeat steps 5-6 for each bridged accessory endpoint.

TCH097 **When the value of the Color Temperature characteristic changes, the accessory must update the values of Hue and Saturation characteristics accordingly and notify the controller.**

Applies to Light Bulb accessories that support Hue and Saturation along with the Color Temperature characteristic. Perform this test case with HAT using the steps below.

1. Pair and discover accessory.
2. For HAP over BLE accessories, enable "Pair-Resume keep alive" with a 27 second interval.
3. In the sidebar of the Controllers window, select Color Temperature.
4. Select the "Enable" button to subscribe to event notifications.
5. In the sidebar of the Controllers window, select Hue.
6. Select the "Enable" button to subscribe to event notifications.
7. Select "Read" and notate the value.
8. In the sidebar of the Controllers window, select Saturation.
9. Select the "Enable" button to subscribe to event notifications.
10. Select "Read" and notate the value.
11. Navigate to the "Color Temperature" characteristic and perform a valid write operation to change the value.
12. Navigate to the "Hue" characteristic, select "Read", and verify the value has been updated accordingly.
13. Navigate to the "Saturation" characteristic, select "Read", and verify the value has been updated accordingly.
14. In the Events view of the trace, verify that notifications were received from "Hue" and "Saturation" with the updated values.
15. Navigate to the "Color Temperature" characteristic and perform a valid write operation to change the value again.
16. Navigate to the "Hue" characteristic, select "Read", and verify the value has been updated accordingly.
17. Navigate to the "Saturation" characteristic, select "Read", and verify the value has been updated accordingly.
18. In the Events view of the trace, verify that notifications were received from "Hue" and "Saturation" with the updated values.
19. Repeat step 2-18 for each Light Bulb service that supports Hue and Saturation along with the Color Temperature characteristic.

TCH098 **When the value of the Hue and Saturation characteristics change, the accessory must update the value of the Color Temperature characteristic accordingly.**

Applies to Light Bulb accessories that support Hue and Saturation along with the Color Temperature characteristic. Perform this test case with HAT using the steps below.

1. Pair and discover accessory. For HAP over BLE accessories, enable "Pair-Resume keep alive" with a 27 second interval.
2. In the sidebar of the Controllers window, select Color Temperature.
3. Select the "Enable" button to subscribe to event notifications.

4. Select “Read” and notate the value.
5. In the sidebar of the Controllers window, select “Hue”.
6. Perform a valid write operation to change “Hue” to a new value.
7. In the sidebar of the Controllers window, select “Saturation”.
8. Perform a valid write operation to change “Saturation” to a new value.
9. In the Events view of the trace, verify a notification was received that the “Color Temperature” has been updated.
10. In the sidebar of the Controllers window, select “Color Temperature”.
11. Select “Read” and verify the value has updated to match the notification value from step 9.
12. Perform valid write operations to the “Hue” and “Saturation” characteristics to values that do not map to a supported “Color Temperature” value.
13. In the Events view of the trace, verify a notification was received that the “Color Temperature” characteristic has been updated.
14. Read the value of “Color Temperature”, and verify that the value has updated to its minimum supported value.
15. Repeat step 3-14 for each Light Bulb service that supports “Hue” and “Saturation” along with the “Color Temperature” characteristic.

1.4 Stateless Programmable Switch

TCSPS001: Verify that the correct Stateless Programmable Switch button press event is sent with each button press type.

TCSPS002: Validate that the Service Label Namespace characteristic indicates one of the pre-defined namespaces (Dots and Arabic Numerals).

TCSPS003: If there are multiple Stateless Programmable Switch Services on the accessory, validate that there is a linked Service Label Service. If there are multiple Stateless Programmable Switch Services on the accessory, Service Label Index is a required characteristic.

TCSPS004: If there is a physical label on the product, validate that the label-index of a service matches the physical label on the product (i.e. If the product has a service label namespace characteristic of "Arabic Numerals", the product should be physically labeled with Arabic Numerals that correspond to the service label index number: Button 1, Button 2, etc.).

TCSPS005: Verify that event notifications are sent from the accessory with each supported button press type, and that the characteristic value is set to "null" in all read responses.

TCSPS006: If the accessory supports button press events outside of the ones defined in the HomeKit specification, verify the accessory does not send those event notifications for the "Programmable Switch Event" characteristic to the HomeKit controller(s). e.g. a triple press should not cause any notifications to be sent to the HomeKit controller(s).

TCSPS001 **Verify that the correct Stateless Programmable Switch button press event is sent with each button press type.**

Applies to accessories that implement the Stateless Programmable Switch service. Applies to accessories that use HAP over BLE. Perform this test case with HAT using the steps below.

1. Pair and discover accessory.
2. In the "Connetion" panel, enable "Pair Resume Keep Alive" with a 27 second interval.
3. Select the "Programmable Switch Event" characteristic on the "Stateless Programmable Switch" service and select "Enable" to subscribe to event notifications.
4. If the accessory supports the single press event, press the switch/button one time.
5. Verify a notifications was received in the GATT view of the trace window.
6. In the Controller's window, select "Read" on the characteristic, and verify the value in the read response is "0".
7. If the accessory supports the double press event, press the switch/button two times.
8. Verify a notifications was received in the GATT view of the trace window.
9. In the Controller's window, select "Read" on the characteristic, and verify the value in the read response is "1".
10. If the accessory supports the long press event, hold the the switch/button down for a few seconds, and then release.
11. Verify a notifications was received in the GATT view of the trace window.

12. In the Controller's window, select "Read" on the characteristic, and verify the value in the read response is "2".

TCSPS002 **Validate that the Service Label Namespace characteristic indicates one of the pre-defined namespaces (Dots and Arabic Numerals).**

Applies to accessories that implement the Stateless Programmable Switch service. Perform this test case with HAT using the steps below.

1. Pair and discover accessory.
2. Read the Service Label Namespace characteristic.
3. Verify Service Label Namespace characteristic value returns the appropriate Naming Label (0 - Dots, 1 - Arabic Numerals).

TCSPS003 **If there are multiple Stateless Programmable Switch Services on the accessory, validate that there is a linked Service Label Service. If there are multiple Stateless Programmable Switch Services on the accessory, Service Label Index is a required characteristic.**

Applies to accessories that implement the Stateless Programmable Switch service. Perform this test case with HAT using the steps below.

1. Pair and discover accessory.
2. Verify Service Label Service is supported. Verify Service Label Index characteristic is provided in each Programmable Switch Service. Verify appropriate Naming Label (Dots or Arabic Numerals) and that each programmable switch service is numbered sequentially (e.g. One Dot, Two Dots OR One, Two).

TCSPS004 **If there is a physical label on the product, validate that the label-index of a service matches the physical label on the product (i.e. If the product has a service label namespace characteristic of "Arabic Numerals", the product should be physically labeled with Arabic Numerals that correspond to the service label index number: Button 1, Button 2, etc.).**

Applies to accessories that implement the Stateless Programmable Switch service. Perform this test case with HAT using the steps below.

1. Pair and discover accessory.
2. Verify the appropriate label index (e.g. One Dot, Two Dots, etc.) corresponds to appropriate programmable switch accessory.

TCSPS005 **Verify that event notifications are sent from the accessory with each supported button press type, and that the characteristic value is set to "null" in all read responses.**

Applies to accessories that implement the Doorbell service. Applies to accessories that implement the Stateless Programmable Switch service. Applies to accessories that use HAP over Ethernet or Wi-Fi. Perform this test case with HAT using the steps below.

1. Pair and discover IP accessory.
2. Select the "Programmable Switch Event" characteristic on the "Stateless Programmable Switch" and/or "Doorbell" service and select "Enable" to subscribe to event notifications.
3. If the accessory supports the single press event, press the switch/button one time.
4. Verify an "/EVENT" was received in the HTTP view of the trace window, which includes the characteristic's IID, accessory AID, and a value of "0".
5. Select "Read" on the characteristic, and verify the value in the read response is "null".
6. If the accessory supports the double press event, press the switch/button two times.
7. Verify an "/EVENT" was received in the HTTP view of the trace window, which includes the characteristic's IID, accessory AID, and a value of "1".
8. Select "Read" on the characteristic, and verify the value in the read response is "null".
9. If the accessory supports the long press event, hold the the switch/button down for a few seconds, and then release.
10. Verify an "/EVENT" was received in the HTTP view of the trace window, which includes the characteristic's IID, accessory AID, and a value of "2".
11. Select "Read" on the characteristic, and verify the value in the read response is "null".

TCSPS006 **If the accessory supports button press events outside of the ones defined in the HomeKit specification, verify the accessory does not send those event notifications for the "Programmable Switch Event" characteristic to the HomeKit controller(s). e.g. a triple press should not cause any notifications to be sent to the HomeKit controller(s).**

Applies to accessories that implement the Stateless Programmable Switch service. Perform this test case with HAT using the steps below.

1. Pair and discover accessory.
2. Navigate to the Programmable Switch Event characteristic on the Stateless Programmable Switch service and select the "Enable" button to subscribe to event notifications.
3. Setup the programmable switch with different button presses that are not supported by HomeKit Specification (e.g. Triple Press).
4. Perform non-HomeKit supported button presses on the accessory. e.g. triple press.
5. In the "Events" traffic view of the Trace window, verify the accessory does not send event notifications for the non-supported HomeKit button presses. e.g. a triple press event should not send any notifications to the HomeKit controller(s).

1.5 Accessory Runtime Information Service

TCAR001: Verify that the accessory contains the "Accessory Runtime Information" service and includes the required characteristics.

TCAR002: The "Heart Beat" characteristic's interval is initialized to "1" on factory reset and power cycles. By default, the accessory must increment this characteristic after a random interval between 240 minutes and 300 minutes (4-5 hours).

TCAR003: Verify that the accessory responds to all pings from controller.

TCAR001 **Verify that the accessory contains the "Accessory Runtime Information" service and includes the required characteristics.**

Applies to accessories that support HomeKit Accessory Protocol specification R16 or later. Applies to accessories that implement the Accessory Runtime Information service. Applies to accessories using the HAP over Thread transport. Perform this test case with HAT using the steps below.

1. Pair and discover accessory.
2. In the sidebar of the Controllers window, verify a single instance of the "Accessory Runtime Information" service is present.
3. For HAP over Wi-Fi, Ethernet, or Thread accessories, verify the required "Ping" characteristic is present.
4. Verify the permissions for the "Ping" characteristic are "Paired Read" and the format is "data".
5. Read the "Ping" characteristic, and verify the read response is zero length value.
6. For HAP over BLE accessoires, verify the "Heart Beat" characteristic is present.
7. Verify the permissions for the "Heart Beat" characteristic are "Paired Read" and "Notify", and that the format is "uint32".
8. If the "Sleep Interval" characteristic is present, verify the permissions are "Paired Read", and the format is "uint32".

TCAR002 **The "Heart Beat" characteristic's interval is initialized to "1" on factory reset and power cycles. By default, the accessory must increment this characteristic after a random interval between 240 minutes and 300 minutes (4-5 hours).**

Applies to accessories that support HomeKit Accessory Protocol specification R16 or later. Applies to accessories that implement the Accessory Runtime Information service. Perform this test case with HAT using the steps below.

1. Factory reset accessory.
2. Pair and discover accessory.
3. Navigate to the "Heat Beat" characteristic and select "Read".
4. Verify the value in the read response is equal to "1".

5. Wait 3 hours and 50 minutes.
6. Navigate to the “Heat Beat” characteristic and select “Read”.
7. Verify the value in the read response is still equal to “1”.
8. Wait 1 hour and 10 minutes.
9. Navigate to the “Heat Beat” characteristic, and select “Read”.
10. Verify the value in the read response is equal to “2”.
11. From the main accessory server view Summary panel, select “Disconnect”.
12. Power cycle the accessory and wait for the accessory to begin advertising again.
13. Naviagate to the “Heat Beat” characteristic and select “Read”.
14. Verify the value in the read response is equal to “1”.

TCAR003 **Verify that the accessory responds to all pings from controller.**

Applies to accessories that implement the Accessory Runtime Information service. Applies to accessories using the HAP over Thread transport. Perform this test case with HAT using the steps below.

1. Pair and discover accessory.
2. In the sidebar of the Controllers window, select the accessory.
3. Under “Connection” panel, enter “10000” into the “Reachability Ping” text field and then select “Enable Ping.”
4. Using the Events view of the trace window, verify that the accessory responds to the Paired Read request to Ping characteristic.
5. Select the response to the read request and then select the “Details” button to show the details.
6. Verify the value in the response is “00”.
7. Wait 10 minutes.
8. Verify that the accessory responded to all of the Paired Read requests to the Ping characteristic successfully and without error.
9. Under the “Connection” panel, select “Disable Ping”.

1.6 Irrigation System

TCIS001: When the irrigation system is enabled, the Active characteristic on the Irrigation Service service must be set to "1".

TCIS002: When any valve service has an "In use" characteristic with value "1," the Irrigation System service's "In Use" must also be "1."

TCIS003: If an irrigation system does not auto detect the presence of valve(s), it must include the "Is Configured" characteristic on each Valve service.

TCIS004: All Valves used in irrigation systems must have their Valve Type characteristic set to "1" (Irrigation). An irrigation system must include the following characteristics within each Valve service:

TCIS005: If there are programs scheduled on the accessory and the accessory is later used for manual operation, the value of the Program Mode characteristic must be "2".

TCIS006: If an Irrigation System uses the "Remaining Duration" characteristic, notifications must not be sent during the accessory's usual count down. If a new "Remaining Duration" value is specified, e.g. to 95 from 92 (increase) or 85 from 92 (decrease which is not part of the usual duration countdown), it must send a notification.

TCIS001 **When the irrigation system is enabled, the Active characteristic on the Irrigation Service service must be set to "1".**

Applies to accessories that implement the Valve service. Perform this test case with HAT using the steps below.

1. Pair and discover accessory.
2. Enable notifications on the Irrigation System Service Active characteristic.
3. Physically enable the irrigation system, i.e manual override.
4. Verify notification received for Irrigation System Service Active characteristic value of "1".

TCIS002 **When any valve service has an "In use" characteristic with value "1," the Irrigation System service's "In Use" must also be "1."**

Applies to accessories that implement the Valve service. Perform this test case with HAT using the steps below.

1. Pair and discover accessory.
2. Enable notifications on the Irrigation System Service Active, Irrigation System "In Use" characteristic and Valve Service "In Use" characteristic .
3. Write "0" to the Irrigation System Active characteristic.
4. Verify Irrigation System "In Use" characteristic value is "0".
5. Write "1" to the Irrigation System Active characteristic.
6. Verify no valves are currently enabled and "In Use" characteristic returns value of "0".

7. Enable the irrigation system valve manually or through accessory UI.
8. Verify notification received for Irrigation System Service and Valve Service "In Use" characteristic value of "1".
9. Disable the valve from step 6.
10. Verify notification received for Irrigation System Service and Valve Service "In Use" characteristic value of "0".
11. Repeat steps 7-10 for all valves.

TCIS003 **If an irrigation system does not auto detect the presence of valve(s), it must include the "Is Configured" characteristic on each Valve service.**

Applies to accessories that implement the Valve service. Perform this test case with HAT using the steps below.

1. Pair and discover accessory.
2. In left sidebar of Controllers window, see each of accessory's services.
3. Verify characteristics are included for each supported service type.
4. Verify the "Is Configured" characteristic is included within each Valve Service in the Irrigation System.
5. Using the Read and Write buttons in Controllers window, read each paired Read characteristic and write to each paired Write characteristic.
6. Verify proper values are returned for all Read characteristics and all Writes characteristics properly update accessory's current state.

TCIS004 **All Valves used in irrigation systems must have their Valve Type characteristic set to "1" (Irrigation). An irrigation system must include the following characteristics within each Valve service:**

Applies to accessories that implement the Valve service. Perform this test case with HAT using the steps below.

Required characteristics:

- Set Duration (r/w/ev*)
- Remaining Duration (r/ev*)

* Event Notifications for BLE include connected, disconnected, and broadcast notifications.

1. Pair and discover accessory.
2. Read the Valve Service Valve Type Characteristic.
3. Verify Valve Type characteristic value is set to "1" (Irrigation).
4. Verify the "Set Duration" and "Remaining Duration" characteristic are included within each Valve Service.

TCIS005 **If there are programs scheduled on the accessory and the accessory is later used for manual operation, the value of the Program Mode characteristic must be "2".**

Applies to accessories that implement the Valve service. Perform this test case with HAT using the steps below.

1. Delete all scheduled programs.
2. Pair and discover IP accessory.
3. Read the Irrigation System Service Program Mode characteristic.
4. Verify Program Mode characteristic value returns "0" (No Programs Scheduled).
5. Set up a schedule program on the accessory.
6. Read the Irrigation System Service Program Mode characteristic.
7. Verify Program Mode characteristic value returns "1" (Program Scheduled).
8. On the accessory UI, enable a valve to run for at least 1 minute.
9. Read the Irrigation System Service Program Mode characteristic.
10. Verify Program Mode characteristic value returns "2" (Program Scheduled, currently overridden to manual mode).
11. Wait for the program in step 8 to end.
12. Read the Irrigation System Service Program Mode characteristic.
13. Verify Program Mode characteristic value returns "1" (Program Scheduled).

TCIS006 **If an Irrigation System uses the "Remaining Duration" characteristic, notifications must not be sent during the accessory's usual count down. If a new "Remaining Duration" value is specified, e.g. to 95 from 92 (increase) or 85 from 92 (decrease which is not part of the usual duration countdown), it must send a notification.**

Applies to accessories that implement the Valve service. Perform this test case with HAT using the steps below.

1. Pair and discover accessory.
2. Enable notifications on the "Remaining Duration" characteristic in the Valve service.
3. On the accessory, enable a valve to run for at least 5 minute.
4. Verify HAT does not receive notifications from the "Remaining Duration" characteristic during the countdown initiated in step 3.
5. While countdown is running, set a new duration, i.e. 10 minutes, for the previously selected valve.
6. Verify HAT receives a notification from the "Set Duration" characteristic for the duration set in step 5.
7. Verify the accessory continues to count down to the duration in step 3.

8. Verify the controller receives a notification for the “Remaining Duration” characteristic for the duration set in step 3 that it expired.
9. On the accessory, enable the valve from step 3.
10. Read the “Remaining Duration” characteristic and verify the time is count down from the duration set in step 5.
11. Verify the controller receives a notification for the “Remaining Duration” characteristic for the duration set in step 5 that it expired.
12. Repeat steps 3-11 for all valves.

1.7 Faucet

TCFT001: If an accessory has a Heater Cooler service linked to a single Valve service, or if an accessory has multiple linked Valve services, then the accessory must use the Faucet service.

TCFT002: If an accessory supports one or multiple water outlets and changing of water temperature through a common temperature control, the accessory must include Heater Cooler and Valve service(s) as linked services linked to the Faucet service.

TCFT003: Setting the Active characteristic to "0" on the Faucet Service must turn off the faucet.

TCFT004: The accessory must retain the Active state on all linked Valve services when the Active characteristic on the Faucet service is changed.

TCFT005: The accessory must retain the Active state on the Faucet service when the Active characteristic on any linked Valve service is changed.

TCFT006: When the value of a faucet's Active characteristic is set to 0 (Inactive), the linked Heater Cooler service Active characteristic value is set to 0 (Inactive).

TCFT001 **If an accessory has a Heater Cooler service linked to a single Valve service, or if an accessory has multiple linked Valve services, then the accessory must use the Faucet service.**

Applies to accessories that implement the Valve service. Applies to accessories that implement the Heater Cooler service. Perform this test case with HAT using the steps below.

1. Pair and discover accessory.
2. For HAP over Wi-Fi or Ethernet accessories, in the IP HTTP traffic view, locate the HTTP response to GET /accessories. Select Details to show the details. Locate the "linked" property. If used, verify the arrays list of IDs the service links to.
3. For HAP over BLE accessories, in the BLE HAP Procedures traffic view, locate the Service Signature Read Response. Select Details and then select Event. Locate the HAP Linked Services. If the service does not link to other services, it must return an empty list with length = 0 (0 bytes). If used, verify the list of IDs the service links to.
4. If accessory has a Heater Cooler Service linked to a single Valve Service or to multiple linked Valve Services, verify the accessory uses the Faucet Service.

TCFT002 **If an accessory supports one or multiple water outlets and changing of water temperature through a common temperature control, the accessory must include Heater Cooler and Valve service(s) as linked services linked to the Faucet service.**

Applies to accessories that implement the Valve service. Applies to accessories that implement the Heater Cooler service. Perform this test case with HAT using the steps below.

1. Pair and discover accessory.
2. For HAP over Wi-Fi or Ethernet accessories, in the IP HTTP traffic view, locate the HTTP response to GET /accessories. Select Details to show the details. Locate the "linked" property. If used, verify the arrays list of IDs the service links to.

3. For HAP over BLE accessories, in the BLE HAP Procedures traffic view, locate the Service Signature Read Response. Select Details and select Event. Locate the HAP Linked Services. If the service does not link to other services, it must return an empty list with length = 0 (0 bytes). If used, verify list of IDs the service links to.
4. Verify the Heater Cooler Service and Valve Service(s) are linked to the Faucet Service.

TCFT003 **Setting the Active characteristic to "0" on the Faucet Service must turn off the faucet.**

Applies to accessories that implement the Valve service. Applies to accessories that implement the Heater Cooler service. Perform this test case with HAT using the steps below.

1. Pair and discover accessory.
2. Enable notifications on the In Use Characteristic in the Valve Service.
3. Write "1" to the Valve Service Active Characteristic.
4. Write "1" to the Faucet Service Active Characteristic.
5. Manually turn on the faucet.
6. Verify Notification received for the In Use Characteristic of value "1".
7. Write "0" to the Faucet Service Active Characteristic.
8. Verify notifications received for the In Use Characteristic contain value "0" and verify the faucet turned off.

TCFT004 **The accessory must retain the Active state on all linked Valve services when the Active characteristic on the Faucet service is changed.**

Applies to accessories that implement the Valve service. Applies to accessories that implement the Heater Cooler service. Perform this test case with HAT using the steps below.

1. Pair and discover accessory.
2. Read each Valve Service's Active Characteristic.
3. Write "0" to the Faucet Service Active Characteristic.
4. Read each Valve Service's Active Characteristic. Verify the value did not change from step 2.
5. Write "1" to the Faucet Service Active Characteristic.
6. Read each Valve Service's Active Characteristic. Verify the value did not change from step 2.

TCFT005 **The accessory must retain the Active state on the Faucet service when the Active characteristic on any linked Valve service is changed.**

Applies to accessories that implement the Valve service. Applies to accessories that implement the Heater Cooler service. Perform this test case with HAT using the steps below.

1. Pair and discover accessory.
2. Read the Faucet Service Active Characteristic.
3. Write “0” to the Valve Service Active Characteristic.
4. Read the Faucet Service Active Characteristic. Verify the value did not change from step 2.
5. Write “1” to the Valve Service Active Characteristic.
6. Read the Faucet Service Active Characteristic. Verify the value did not change from step 2.

TCFT006 **When the value of a faucet’s Active characteristic is set to 0 (Inactive), the linked Heater Cooler service Active characteristic value is set to 0 (Inactive).**

Applies to accessories that implement the Valve service. Applies to accessories that implement the Heater Cooler service. Perform this test case with HAT using the steps below.

1. Pair and discover accessory.
2. Enable notifications on the Active Characteristic in the Heater Cooler Service.
3. Write “1” to the Faucet Service Active Characteristic.
4. Write “1” to the Heater Cooler Service Active Characteristic.
5. Write “0” to the Faucet Service Active Characteristic.
6. Verify Notification received for Heater Cooler Service Active Characteristic value of “0”.

1.8 Window Service, Window Covering Service, Door Service

TCBW001: Verify that the accessory correctly sends event notifications for the "Target Position", "Current Position" and "Position State" characteristics when partially opening and partially closing. When X > Y and X < Y. X and Y are values between the "Target Position" minimum and maximum values.

TCBW002: Verify that the accessory correctly sends Event Notifications for the "Target Position", "Current Position" and "Position State" characteristics when partially opening and partially closing. When X > Y and X < Y. X and Y are values between the "Target Position" minimum and maximum values.

TCBW003: If the "Hold Position" characteristic is supported, verify that the accessory correctly sends Event Notifications for the "Target Position", "Current Position", and "Position State" characteristics. X and Y can be any value.

TCBW004: If the accessory can be controlled outside of HomeKit with a non-HAP controller (e.g. a touchscreen or third-party accessory app) where a specific "Target Position" can be set, verify that the accessory correctly sends Event Notifications for the "Target Position", "Current Position", and "Position State" characteristics.

TCBW005: If the accessory can be controlled outside of HomeKit with a non-HAP controller (e.g. a remote with open and close buttons) where a specific "Target Position" cannot be set, verify that the accessory correctly sends Event Notifications for the "Target Position", "Current Position", and "Position State" characteristics.

TCBW001 **Verify that the accessory correctly sends event notifications for the "Target Position", "Current Position" and "Position State" characteristics when partially opening and partially closing. When X > Y and X < Y. X and Y are values between the "Target Position" minimum and maximum values.**

Applies to accessories with the Window service. Applies to accessories with the Window Covering service. Applies to accessories with the Door service. Perform this test case with HAT using the steps below.

1. Pair and discover accessory.
2. Navigate to the Target Position characteristic on the Window, Window Covering, and/or Door service(s), and write a value of "100" to set the initial position.
3. Enable event notifications for the Target Position, Current Position, and Position State characteristics on each applicable service.
4. Write a value of "0" to the Target Position characteristic.
5. Verify an event notification is received for the Position State characteristic with a value of 0 (Going to the min value specified in metadata).
6. Verify that the accessory physically reaches the Target Position written in step 4.
7. Verify that the accessory immediately sends an event notification that includes the "Position State" characteristic with a value of 2 (Stopped) and the "Current Position" characteristic with a value equal to the one set in step 4.
8. Write a value of "100" to the Target Position characteristic.
9. Verify that an event notification is received for the "Position State" characteristic with a value of 1 (Going to the max value specified in metadata).

10. Verify that the accessory physically reaches the Target Position written in step 8.
11. Verify that the accessory immediately sends an event notification that includes a "Position State" characteristic with a value of 2 (Stopped) and the "Current Position" characteristic with a value equal to the one set in step 8.
12. Repeat steps 4-11 for each applicable service.

TCBW002 **Verify that the accessory correctly sends Event Notifications for the "Target Position", "Current Position" and "Position State" characteristics when partially opening and partially closing. When X > Y and X < Y. X and Y are values between the "Target Position" minimum and maximum values.**

Applies to accessories with the Window service. Applies to accessories with the Window Covering service. Applies to accessories with the Door service. Perform this test case with HAT using the steps below.

1. Pair and discover accessory.
2. In the sidebar of the Controllers window, navigate to the "Target Position" characteristic on the Window, Window Covering, and/or Door service(s), and write a value of "X' to set the initial position, i.e. 75.
3. Enable Event Notifications for the "Target Position", "Current Position", and "Position State" characteristics on each applicable service.
4. Write a value of "Y" to the "Target Position" characteristic, i.e. 25.
5. In the Events view of Trace, verify that an Event Notification is received for the "Position State" characteristic with a value of "0" (Going to the min value specified in metadata).
6. Verify that the accessory physically reaches the "Target Position" written in step 4.
7. In the Events view of Trace, verify that the accessory immediately sends an Event Notification that includes the "Position State" characteristic with value of "2" (Stopped) and the "Current Position" characteristic with a value equal to the one set in step 4.
8. Write a value of "X" to the "Target Position" characteristic, i.e 50.
9. In the Events view of Trace, verify that an Event Notification is received for the "Position State" characteristic with a value of "1" (Going to the max value specified in metadata).
10. Verify that the accessory physically reaches the "Target Position" written in step 8.
11. In the Events view of Trace, verify that the accessory immediately sends an Event Notification that includes a "Position State" characteristic with value of "2" (Stopped) and the "Current Position" characteristic with a value equal to the one set in step 8.
12. Write a value of "Y" to the "Target Position" characteristic, i.e. 90.
13. In the Events view of Trace, verify that an Event Notification is received for the "Position State" characteristic with a value of "1", i.e. going to the max value specified in metadata.
14. Verify that the accessory physically reaches the "Target Position" written in step 12.
15. In the Events view of Trace, verify that the accessory immediately sends an Event Notification that includes a "Position State" characteristic with value of "2" (Stopped) and the "Current Position" characteristic with a value equal to the one set in step 12.

16. Repeat steps 4-15 for each applicable service.

TCBW003 **If the "Hold Position" characteristic is supported, verify that the accessory correctly sends Event Notifications for the "Target Position", "Current Position", and "Position State" characteristics. X and Y can be any value.**

Applies to accessories with the Window service. Applies to accessories with the Window Covering service. Applies to accessories with the Door service. Perform this test case with HAT using the steps below.

1. Pair and discover accessory.
2. In the sidebar of the Controllers window, navigate to the "Target Position" characteristic on the Window, Window Covering, and/or Door service(s), and write a value of "X' to set the initial position i.e. 5.
3. Enable Event Notifications for the "Target Position", "Current Position", and "Position State" characteristics in the currently tested service.
4. Write a value of "Y" to the "Target Position" characteristic, i.e. 85.
5. In the Events view of Trace, verify that an Event Notification is received for the "Position State" characteristic with a value of "1" (Going to the max value specified in metadata).
6. Before the accessory reaches its final state, write a value of "1" to the "Hold Position" characteristic.
7. In the Events view of Trace, verify that the accessory immediately sends an Event Notification that includes a "Position State" with a value of "2" (Stopped) and the current position at which the accessory was stopped.
8. Write a value of "Y" to the "Target Position" characteristic again.
9. In the Events view of Trace, verify that an Event Notification is received for the "Position State" characteristic with a value of "1" (Going to the max value specified in metadata).
10. Verify that the accessory physically reaches the "Target Position" written in step 7.
11. In the Events view of Trace, verify that the accessory immediately sends an Event Notification that includes a "Position State" characteristic with value of "2" (Stopped) and the "Current Position" characteristic with a value equal to the one set in step 7.

TCBW004 **If the accessory can be controlled outside of HomeKit with a non-HAP controller (e.g. a touchscreen or third-party accessory app) where a specific "Target Position" can be set, verify that the accessory correctly sends Event Notifications for the "Target Position", "Current Position", and "Position State" characteristics.**

Applies to accessories with the Window service. Applies to accessories with the Window Covering service. Applies to accessories with the Door service. Perform this test case with HAT using the steps below.

1. Pair and discover accessory.
2. Enable Event Notifications for the "Target Position", "Current Position", and "Position State" characteristics on each applicable service.

3. Using the non-HAP controller, trigger a change on the accessory.
4. In the Events view of Trace, verify that an Event Notification is received for the "Target Position" characteristic with the value set in step 3.
5. In the Events view of Trace, verify an Event Notification is received for the "Position State" characteristic with a value corresponding to the direction that the accessory is moving, i.e. going to max or min values.
6. When the accessory physically reaches the desired state, in the Events view of Trace, verify that an Event Notification is received for "Position State" with a value of "2" (Stopped) and "Current Position" with the value equal to the position from step 3. If the accessory was stopped using the non-HAP controller, verify that an Event Notification is received for "Position State" with a value of "2" (Stopped), "Current Position" and "Target Position" with the updated position values.

TCBW005 **If the accessory can be controlled outside of HomeKit with a non-HAP controller (e.g. a remote with open and close buttons) where a specific "Target Position" cannot be set, verify that the accessory correctly sends Event Notifications for the "Target Position", "Current Position", and "Position State" characteristics.**

Applies to accessories with the Window service. Applies to accessories with the Window Covering service. Applies to accessories with the Door service. Perform this test case with HAT using the steps below.

1. Pair and discover accessory.
2. Using the sidebar of the Controllers window, enable Event Notifications for the "Target Position", "Current Position", and "Position State" characteristics on each applicable service.
3. Using the non-HAP controller, trigger a change on the accessory.
4. If the non-HAP controller triggers a position change from max to min, verify that the accessory sends an Event Notification for the "Target Position" with a value of "0" in the Events view of the Trace. If the non-HAP controller triggers a position change from min to max, verify that the accessory sends an Event Notification for "Target Position" with a value of "100".
5. In the Events view of Trace, verify that an Event Notification is received for the "Position State" characteristic with a value corresponding to the direction that the accessory is moving, i.e. max or min.
6. When the accessory physically reaches the desired state, in the Events view of the Trace, verify an Event Notification is received for "Position State" with value of "2" (Stopped) and for the "Current Position" with the value equal to the position it stopped at. If the accessory was stopped using the non-HAP controller, verify an Event Notification is received for "Position State" with a value of "2" (Stopped), and for the "Current Position" and "Target Position" with the updated position values.

1.9 Valve

TCV001: A bridge consisting of multiple Valve endpoints must include the following characteristic within each Valve service:

TCV002: If the accessory uses multiple linked Valve services, the valve services must include the following characteristic:

TCV003: If a valve has the "Service Label Index" characteristic and if the user has not assigned a name to the valve before HomeKit pairing (i.e. through the accessory app), the "Name" characteristic on the valves must be an empty string.

TCV004: If a user sets the valve name before HomeKit pairing, the valves should include the user-defined name as the default value for the "Name" characteristic.

TCV005: The Is Configured characteristic must be included in each Valve service, if the Valve service is used in an irrigation system or shower where all valves may not be setup for use.

TCV001 **A bridge consisting of multiple Valve endpoints must include the following characteristic within each Valve service:**

Applies to accessories that implement the Valve service. Applies to accessories that implement the Irrigation System service. Applies to accessories that implement the Faucet service. Perform this test case with HAT using the steps below.

Required characteristics:

- Sevice Label Index (r)

1. Pair and discover accessory.
2. Verify the Service Label Index characteristic is included within each Valve Service.

TCV002 **If the accessory uses multiple linked Valve services, the valve services must include the following characteristic:**

Applies to accessories that implement the Valve service. Applies to accessories that implement the Irrigation System service. Applies to accessories that implement the Faucet service. Perform this test case with HAT using the steps below.

Required characteristics:

- Sevice Label Index (r)

1. Pair and discover accessory.
2. For HAP over Wi-Fi or Ethernet accessories, in the IP HTTP traffic view, locate the HTTP response to GET /accessories. Select Details and select Event. Locate the "linked" property. If used, verify arrays list of IDs the service links to.
3. For HAP over BLE accessories, in the BLE HAP Procedures traffic view, locate the Service Signature Read Response. Select Details and select Event. Locate the HAP Linked Services. If the service does not link to other services it must return an empty list with length = 0 (0 bytes). If used, verify list of IDs the service links to.

4. Verify all instances of the Valve service include the Service Label Index characteristic.

TCV003 **If a valve has the "Service Label Index" characteristic and if the user has not assigned a name to the valve before HomeKit pairing (i.e. through the accessory app), the "Name" characteristic on the valves must be an empty string.**

Applies to accessories that implement the Valve service. Applies to accessories that implement the Irrigation System service. Applies to accessories that implement the Faucet service. Perform this test case with HAT using the steps below.

1. Factory reset the accessory.
2. Pair and discover accessory.
3. Verify the Service Label Index characteristic is included within each Valve Service.
4. Read the Valve Service Name characteristic.
5. Verify the read to the Name characteristic returns an empty string.
6. Repeat for all instances of the Valve service.

TCV004 **If a user sets the valve name before HomeKit pairing, the valves should include the user-defined name as the default value for the "Name" characteristic.**

Applies to accessories that implement the Valve service. Applies to accessories that implement the Irrigation System service. Applies to accessories that implement the Faucet service. Perform this test case with HAT using the steps below.

1. Using the accessory app, name all valves.
2. Pair and discover accessory.
3. Verify the Service Label Index characteristic is included within each Valve Service.
4. Read the Valve Service Name characteristic.
5. Verify the value returned by the Name characteristic is the same value as entered in step 1.
6. Repeat for all instances of the Valve service.

TCV005 **The Is Configured characteristic must be included in each Valve service, if the Valve service is used in an irrigation system or shower where all valves may not be setup for use.**

Applies to accessories that implement the Valve service. Applies to accessories that implement the Irrigation System service. Applies to accessories that implement the Faucet service. Perform this test case with HAT using the steps below.

1. Pair and discover accessory.
2. Verify all instances of the Valve Service includes the Is Configured characteristic.

3. Read each instance of the Is Configured characteristic and verify the proper value is returned.

1.10 Software Token-Based Authentication

TCSTA001: Verify that the accessory successfully updates to HomeKit-compatible firmware and can be paired (and subsequently paired again after factory reset) using the Home app.

TCSTA002: Controller can successfully set and retrieve a maximum size authentication token.

TCSTA003: Provisioned software token must persist through factory reset.

TCSTA004: Accessories must save the setup code or SRP verifier used during pair-setup with both split and transient flags and use that same setup code or SRP verifier for the next pair-setup if the split flag is provided.

TCSTA008: If the connection is closed after Transient + Split pair-setup M2, a subsequent pair-setup must succeed.

TCSTA009: If the connection is closed after Split pair-setup M2, a subsequent pair-setup must succeed.

TCSTA010: If the connection is closed after Split pair-setup M4, a subsequent pair-setup must succeed.

TCSTA011: If the accessory received pair-setup M1 with the kTLVType_Flags set as kPairingFlag_Split (without the Transient Flag ever being sent), it must respond with the following TLV items: kTLVType_State <M2> and kTLVType_Error <kTLVError_-Authentication>.

TCSTA012: Verify that the accessory can perform Software Token Authentication procedures during Pair-Setup, and that the token persists through factory reset.

TCSTA001 **Verify that the accessory successfully updates to HomeKit-compatible firmware and can be paired (and subsequently paired again after factory reset) using the Home app.**

Applies to accessories that support in-field provisioning for software token authentication. Perform this test case using the Home app on iOS.

1. Verify accessory does not integrate an Apple Authentication Coprocessor and has not yet been updated to support HomeKit.
2. Add accessory with vendor app and note down current firmware version.
3. Use vendor app to upgrade to a firmware version that supports HomeKit.
4. Verify firmware update completed successfully. Verify firmware version after update is greater than firmware version in step 2.
5. Using Home app, add the accessory to the Home.
6. Verify read/write functionality.
7. Perform factory reset on accessory.
8. Remove the accessory from the Home using the Home app.
9. Re-add the accessory to the Home.
10. Verify read/write functionality.
11. Remove the accessory from the Home using the Home app.

TCSTA002 **Controller can successfully set and retrieve a maximum size authentication token.**

Applies to accessories that use software token authentication. Perform this test case automatically with HCA. Below are steps that can be used to perform this test case manually in HAT.

1. Pair and discover accessory.
2. In the Controllers window, under the Pairing Panel, select the "Get Authentication Token" button.
3. In the Events traffic view, find the "Get Authentication Token Completed" response, select Details, and copy the Token.
4. In the Controllers window, under the Pairing Panel, enter a 2000-character string into the Authentication Token field and select the "Set Authentication Token" button.
5. In the Controllers window, under the Pairing Panel, select the "Get Authentication Token" button.
6. In the Events traffic view, find the "Get Authentication Token Completed" response, select Details, verify the token matches from step 4.
7. In the Controllers window, under the Pairing Panel, input the original token from step 3 into the Authentication Token field, and select the "Set Authentication Token" button.
8. In the Controllers window, under the Pairing Panel, select the "Get Authentication Token" button.
9. In the Events traffic view, find the "Get Authentication Token Completed" response, select Details, and verify token matches original token from step 3.

TCSTA003 **Provisioned software token must persist through factory reset.**

Applies to accessories that use software token authentication. Perform this test case with HAT using the steps below.

1. Pair and discover accessory.
2. In the Controllers window, under the Pairing Panel, select the "Get Authentication Token" button.
3. In the Events traffic view, find the "Get Authentication Token Completed" response, select Details, and copy the Token.
4. Perform factory reset.
5. Launch HAT and then select the accessory on the left side of the Controllers window.
6. For HAP over Wi-Fi or Ethernet accessories that support WAC, proceed to to step 7. For all other accessories, pair and discover and then proceed to step 14.
7. In the Wi-Fi Accessory Configuration panel, select the "Join access point" button.
8. In the Pairing Panel, select the "Start Pairing" and enter setup code when prompted.
9. In the Wi-Fi Accessory Configuration panel, enter the Wi-fi SSID and Wi-Fi PSK and select the "Send WAC Configuration" button.

10. Ensure your Mac is on the network you expect the accessory to join, otherwise re-join the expected network.
11. Once the accessory begins advertising via Bonjour on the newly joined network, select the "Confirm WAC Configuration" button.
12. Verify the accessory responds to /Configured with HTTP Response: HTTP/1.1 200 OK
13. Select the "Discover" button.
14. In the Controllers window, under the Pairing Panel, select the "Get Authentication Token" button.
15. In the Events traffic view, find the "Get Authentication Token Completed" response, select Details, verify token matches token from step 3.

TCSTA004 **Accessories must save the setup code or SRP verifier used during pair-setup with both split and transient flags and use that same setup code or SRP verifier for the next pair-setup if the split flag is provided.**

Applies to accessories that use software token authentication. Perform this test case automatically with HCA. Below are steps that can be used to perform this test case manually in HAT.

1. Select the accessory server from the left sidebar of the Controllers window.
2. In the Controllers window, under the Pairing panel, check the box "on" for both "Transient pair-setup" and "Split pair-setup".
3. Select the "Start Pairing" button.
4. For accessories that use dynamic setup codes, note the setup code on the accessory's display.
5. Verify the details of the M2 response to pair-setup, Pairing Type Flags value is set to 16777232 (bit 4). For HAP over Wi-Fi or Ethernet accessories, check in the HTTP traffic view. For HAP over BLE accessories, check the HAP procedures traffic view.
6. Select the "Disconnect" button.
7. Verify that the Status Flag is still "0x01" in the most recent accessory advertisement.
8. Select the "Start Pairing" button.
9. For accessories that use dynamic setup codes, note the setup code on the accessory's display. Verify the accessory generates and displays a new setup code that differs from step 4.
10. Select the "Disconnect" button.
11. Verify that the Status Flag is still "0x01" in the most recent accessory advertisement.
12. Uncheck "Transient pair-setup" and ensure "Split pair-setup" is still selected.
13. Select the "Start Pairing" button.
14. Pair to the accessory. For accessories that use dynamic setup codes, verify the new setup code from step 9 can be used to successfully pair to the accessory.
15. Verify the details of the M2 response to pair-setup, Pairing Type Flags value is set to 16777216 (bit 24). For HAP over Wi-Fi or Ethernet accessories, check in the HTTP traffic view. For HAP over BLE accessories, check the HAP procedures traffic view.

16. Verify pair-setup completes successfully.
17. In the summary panel, select the “Discover” button.
18. Verify pair-verify completes successfully.
19. Select the “Disconnect” button.
20. Verify that the Status Flag is “0x00” in the most recent accessory advertisement.
21. In the summary panel, select the “Discover” button.
22. In the pairing panel, select the “Remove Pairing” button.
23. Uncheck “Split pair-setup”.

TCSTA008 **If the connection is closed after Transient + Split pair-setup M2, a subsequent pair-setup must succeed.**

Applies to accessories that use software token authentication. Does not apply to accessories while using the HAP over Thread transport. Perform this test case automatically with HCA. Below are steps that can be used to perform this test case manually in HAT.

1. Select the accessory server from the left sidebar of the Controllers window.
2. In the Controllers window, under the Pairing panel, check the box “on” for both “Transient pair-setup” and “Split pair-setup”.
3. In Pairing panel, select “Start Pairing” button.
4. When the setup code pairing window appears, select the “Stop” button.
5. In the Summary panel, select the “Disconnect” button.
6. Verify that the Status Flag is still “0x01” in the most recent accessory advertisement.
7. In Pairing panel, select the “Start Pairing” button.
8. Enter the setup code and select send button or pair using the Companion app.
9. Select the “Disconnect” button.
10. Verify that the Status Flag is still “0x01” in the most recent accessory advertisement.
11. In the Pairing panel, uncheck the “Transient pair-setup” box.
12. In Pairing panel, select the “Start Pairing” button.
13. Enter the setup code and select send button or pair using the Companion app.
14. Verify pair-setup completes successfully.
15. In the summary panel, select the “Discover” button.
16. Verify pair-verify completes successfully.
17. Select the “Disconnect” button.
18. Verify that the Status Flag is “0x00” in the most recent accessory advertisement.

19. In the summary panel, select the “Discover” button.
20. In the pairing panel, select the “Remove Pairing” button.
21. Uncheck “Split pair-setup”.

TCSTA009 **If the connection is closed after Split pair-setup M2, a subsequent pair-setup must succeed.**

Applies to accessories that use software token authentication. Does not apply to accessories while using the HAP over Thread transport. Perform this test case automatically with HCA. Below are steps that can be used to perform this test case manually in HAT.

1. Select the accessory server from the left sidebar of the Controllers window.
2. In the Controllers window, under Pairing panel, check the box “on” for both “Transient pair-setup” and “Split pair-setup”.
3. Select the “Start Pairing” button.
4. Enter the setup code and select send button or pair using the Companion app.
5. Select the “Disconnect” button.
6. After the accessory disconnects, verify that the Status Flag is “0x01” in the next accessory advertisement.
7. In the Pairing panel, uncheck the “Transient pair-setup” box.
8. Select the “Start Pairing” button.
9. When the setup code pairing window appears, select the “Stop” button.
10. Select the “Disconnect” button.
11. After the accessory disconnects, verify that the Status Flag is “0x01” in the next accessory advertisement.
12. In the Controllers window, under Pairing panel, check the box “on” for both “Transient pair-setup” and “Split pair-setup”.
13. Select the “Start Pairing” button
14. Enter the setup code and select send button or pair using the Companion app.
15. Select the “Disconnect” button.
16. After the accessory disconnects, verify that the Status Flag is “0x01” in the next accessory advertisement.
17. In the Pairing panel, uncheck the “Transient pair-setup” box.
18. In Pairing panel, select the “Start Pairing” button.
19. Enter the setup code and select send button or pair using the Companion app.
20. In the summary panel, select the “Discover” button.
21. Verify pair-verify completes successfully.

22. In Pairing panel, select the “Remove Pairing” button and uncheck the “Split pair-setup” box.

TCSTA010 **If the connection is closed after Split pair-setup M4, a subsequent pair-setup must succeed.**

Applies to accessories that use software token authentication. Does not apply to accessories while using the HAP over Thread transport. Perform this test case using HCA.

TCSTA011 **If the accessory received pair-setup M1 with the kTLVType_Flags set as kPairingFlag_Split (without the Transient Flag ever being sent), it must respond with the following TLV items: kTLVType_State <M2> and kTLVType_Error <kTLVError_Authentication>.**

Applies to accessories that use software token authentication. Perform this test case with HAT using the steps below.

1. Factory reset the accessory.
2. Launch HAT and then select the accessory on the left side of the Controllers window.
3. For HAP over Wi-Fi or Ethernet accessories that support WAC2, proceed to step 4. For all other accessories, proceed to step 6.
4. In the Wi-Fi Accessory Configuration panel, select the “Join access point” button.
5. Wait for the accessory to begin advertising via Bonjour.
6. In the Controllers window, in the Pairing panel, check the box ‘on’ for “Split pair-setup”.
7. Select the “Start Pairing” button.
8. Verify accessory rejects the pair-setup request with State value 2 (M2) and Error value 2 (Authentication error).

TCSTA012 **Verify that the accessory can perform Software Token Authentication procedures during Pair-Setup, and that the token persists through factory reset.**

Applies to accessories that use software token authentication. Perform this test case with HAT using the steps below.

1. Select the accessory in HAT sidebar.
2. In the Pairing panel, check the box “on” for “Transient pair-setup” and “Split pair-setup”.
3. Select “Start Pairing” button.
4. Provide the setup code for the accessory and select “Send”, or select “HomeKit Companion” to scan NFC tag.
5. After the controller receives the Pair-Setup M4 response, select the “Get Server Information” button.
6. Using the Events view in the trace, verify the response to “Get Server Info” contains valid information for: State Number, Config Number, Device ID, Pairing Feature Flags, Model Name, Protocol Version, Status Flags, Category Identifier, and Setup Hash.

7. For HAP over Wi-Fi or Ethernet accessories, find the most recent advertisement in the Bonjour Discovery view and verify the Configuration Number, Device ID, Pairing Feature Flags, Model Name, Protocol Version, Status Flags, Category Identifier, and Setup Hash from the "Get Server Info" response matches the information in the most recent accessory advertisement. For HAP over BLE accessories, find the most recent advertisement in the BLE Discovery view and verify the Configuration Number, Device ID (Advertising Identifier), Compatible Version = 2, Status Flags, Category Identifier, and Setup Hash from the "Get Server Info" response matches the information in the most recent accessory advertisement. The BLE Advertisement data can contain either the Shortened Local Name or the Complete Local Name.
8. Select "Get Authentication Token" button in the Pairing panel.
9. Using the Events view in the trace, verify the response to "Get Authentication Token" contains valid information for UUID and Token. Notate the UUID and Token values.
10. Enter a new 2000-character string in the "Software Authentication Token" text field and select the "Set Authentication Token" button.
11. Using the Events view in the trace, verify the response to "Set Authentication Token" does not contain an error.
12. Select "Get Authentication Token" button in the Pairing panel.
13. Using the Events view in the trace, verify the response to "Get Authentication Token" contains the same UUID from step 9 and the new Token set in step 10.
14. Enter the first token value from step 9 in the "Software Authentication Token" text field and select the "Set Authentication Token" button.
15. Using the Events view in the trace, verify the response to "Set Authentication Token" does not contain an error.
16. From the Summary panel, select "Disconnect" button.
17. Disable the "Transient pair-setup" checkbox and ensure the "Split pair-setup" checkbox is still enabled.
18. Select "Start Pairing" button.
19. Provide the same setup code from step 4 and select "Send", or select "HomeKit Companion" to scan NFC tag.
20. Verify pair setup completes successfully.
21. Factory reset accessory.
22. Select the accessory on the left side of the Controllers window.
23. For accessories that do not support WAC2, proceed to step 30. For accessories that support WAC2, complete the WAC2 procedure below:
24. Select the accessory. Then in the Wi-Fi Accessory Configuration panel, select the "Join access point" button.
25. Pair with the accessory.

26. In the Wi-Fi Accessory Configuration panel, enter the Wi-fi SSID and Wi-Fi PSK and select the “Send WAC Configuration” button.
27. Ensure your Mac is on the network you expect the accessory to join, otherwise join the expected network.
28. Once the accessory begins advertising via Bonjour on the newly joined network, select the “Confirm WAC Configuration” button.
29. After the accessory successfully responds to the /Configured request, select the “Discover” button and then select “Remove Pairing”.
30. Repeat steps 1-9, and verify that the UUID and Token matches the UUID and Token used in step 14.

1.11 Remotes for Apple TV

TCRC001: Verify that the Siri service is linked to the Audio Stream Management and Data Stream Management services.

TCRC002: Verify that the Button Event for each supported button type.

TCRC003: Verify that the Button Events are received for each Up down button state.

TCRC004: If button name is present, the accessory must use this name instead of the name derived from the button type.

TCRC005: Verify when releasing the Siri button, the target processes the full Siri request.

TCRC006: Verify that remotes use Opus codec 16kHz.

TCRC007: Verify the correct value of Siri input type.

TCRC012: Verify that the accessory supports Add, Update, Remove, Reset and List operations on the Target Control List characteristic, and the correct information is returned in write responses when applicable.

TCRC014: Target Control List configurations persist over an accessory power cycle.

TCRC015: Target Control List configurations must delete after last admin pairing removed.

TCRC016: Target Control List configurations must delete after factory reset.

TCRC017: Any requests to enable notification on the Button Event characteristic by non admin controllers must result in error -70401 (Insufficient Privileges).

TCRC018: Only admin controllers are allowed to perform any operations on the Target Control List characteristic. Any read/writes to this characteristic by non admin controllers must result in error -70401 (Insufficient Privileges).

TCRC019: If no Target is currently selected (i.e. not configured or non ATV selected), the Active Identifier characteristic value must be 0.

TCRC020: When Active Identifier is changed the Active characteristic must reset to inactive.

TCRC022: Accessory must not allow new targets to be added once the advertised Maximum Targets limit has been met.

TCRC023: Verify that the accessory reports the correct "Type" of remote in its Target Control Supported Configuration characteristic. Verify software based remotes do not include the Siri service.

TCRC001 **Verify that the Siri service is linked to the Audio Stream Management and Data Stream Management services.**

Applies to remote control accessories. Perform this test case with HAT using the steps below.

1. Pair and discover accessory.
2. For HAP over Wi-Fi or Ethernet accessories, in the IP HTTP traffic view, locate the HTTP response to GET /accessories. Select Details and select Event. Locate the "linked" property in the Siri service. Verify it lists the ids of the Audio Stream Management and Data Stream Management services.

TCRC002 **Verify that the Button Event for each supported button type.**

Applies to remote control accessories. Perform this test case with HAT using the steps below.

Press one of each button type where supported: 1 - Menu 2 - Play/Pause 3 - TV/Home 4 - Select 5 - Arrow Up 6 - Arrow Right 7 - Arrow Down 8 - Arrow Left 9 - Volume Up 10 - Volume Down 11 - Siri 12 - Power 13 - Generic (optional)

1. Pair and discover accessory.
2. In the left side of the controller's window select the Target Control Management service.
3. In the Target Control List Operations panel, set the "Target Identifier'' to any non-zero value, set the "Target Name'', and set the "Target Category'' value to 24.
4. Select the "Read Supported Configuration'' button.
5. Select the "Add'' operation button.
6. Read the Target Control service Active Identifier characteristic.
7. Verify the Identifier value matches the Target Identifier.
8. Write 1 to the Target Control service Active characteristic.
9. Enable notifications on the Button Event characteristic in the Target Control service.
10. On the accessory, press each supported button once.
11. Verify that a notification for a button down (Button State 0x02, value 1 (down)) and notification for a button up (Button State 0x02, value 0 (up)) event are received for that button type.
12. On the accessory press each supported button 5 times in rapid succession.
13. Verify that 5 notifications for a button down (Button State 0x02, value 1 (down)) and 5 notifications for button up (Button State 0x02, value 0 (up)) events are seen for that button type.
14. On the accessory hold down each supported button for 3 seconds and then release.
15. Verify that a notification for a button down event (Button State 0x02, value 1 (down)) is received followed by a notification for a button up (Button State 0x02, value 0 (up)) event is received for that button type 3 seconds later.
16. Repeat for all buttons (where applicable).

TCRC003 **Verify that the Button Events are received for each Up down button state.**

Applies to remote control accessories. Perform this test case with HAT using the steps below.

1. Pair and discover accessory.
2. In the left side of the controller's window select the Target Control Management service.
3. In the Target Control List Operations panel, set the "Target Identifier'' to any non-zero value, set the "Target Name'', and set the "Target Category'' value to 24.
4. Select the "Read Supported Configuration'' button.

5. Select the “Add’’ operation button.
6. Read the Target Control service Active Identifier characteristic.
7. Verify the Identifier value matches the Target Identifier.
8. Write 1 to the Target Control service Active characteristic.
9. Enable notifications on the Button Event characteristic in the Target Control service.
10. On the accessory press each supported button 30 times.
11. Verify that 30 notifications for a button down (Button State 0x02, value 1 (down)) and 30 notifications for button up (Button State 0x02, value 0 (up)) events are seen for that button type.

TCRC004 **If button name is present, the accessory must use this name instead of the name derived from the button type.**

Applies to remote control accessories. Perform this test case with HAT using the steps below.

1. Pair and discover accessory.
2. Read the Target Control List characteristic.
3. In the events traffic view, locate the characteristic read completed. Select Details, expand the Parsed TLV8 disclosure arrow and locate the buttonConfiguration.
4. On the remote, verify any text labels for buttons reflects the button name and not the button type.

TCRC005 **Verify when releasing the Siri button, the target processes the full Siri request.**

Applies to remote control accessories. Perform this test case with HAT using the steps below.

1. Pair and discover accessory.
2. In the left side of the controller’s window select the Target Control Management service.
3. In the Target Control List Operations panel, set the “Target Identifier’’ to any non-zero value, set the “Target Name’’, and set the “Target Category’’ value to 24.
4. Select the “Read Supported Configuration’’ button.
5. Select the “Add’’ operation button.
6. Read the Target Control service Active Identifier characteristic.
7. Verify the Identifier value matches the Target Identifier.
8. Write 1 to the Target Control service Active characteristic.
9. Enable notifications on the Button Event characteristic in the Target Control service.
10. Select the Audio Stream Management service.
11. In the Audio Stream Configuration panel, select the “Read from Supported’’ button.

12. Verify Audio Codec Type returns value 3 (Opus) and audio channels is 1.
13. Set the RTP time (in ms) to 20 and select the "Write to Selected'' button.
14. Select the Data Stream Management service.
15. In the HomeKit Data Stream panel, select the "Send Start Command" button.
16. Select the "Connect'' button.
17. Press and hold the Siri button and issue a voice command. i.e "Show me new movies on iTunes".
18. Release button and wait 250ms.
19. Command Siri to do something.
20. Verify the target does not show a Siri response to step 19.

TCRC006 **Verify that remotes use Opus codec 16kHz.**

Applies to remote control accessories. Perform this test case with HAT using the steps below.

1. Pair and discover accessory.
2. Read the Supported Audio Stream Configuration characteristic.
3. In the Events traffic view, select details, and expand the Parsed TLV8 disclosure arrow.
4. Verify the audioCodecType (0x01) value is 3 for Opus.
5. Verify the audioCodecParameters sampleRate (0x03) value is 1 for 16 kHz.

TCRC007 **Verify the correct value of Siri input type.**

Applies to remote control accessories. Perform this test case with HAT using the steps below.

1. Pair and discover accessory.
2. Read the value of the Siri Input Type characteristic.
3. Verify this value is "0" for Push button triggered Apple TV.

TCRC012 **Verify that the accessory supports Add, Update, Remove, Reset and List operations on the Target Control List characteristic, and the correct information is returned in write responses when applicable.**

Applies to remote control accessories. Perform this test case with HAT using the steps below.

1. Pair and discover accessory.
2. In the left side of the controller's window, select the "Target Control Management" service.
3. In the "Target Control List Operations" panel, set the "Target Identifier" to any non-zero value, set the "Target Name", and set the "Target Category" value to 24.

4. Select the “Read Supported Configuration’’ button.
5. Select the “Add’’ operation button.
6. In the events traffic view, locate the “Encrypted Characteristic Write Completed” event, select “Details”, and verify the response value contains the added Target Configuration information.
7. In the Target Control List Operations panel, enter the same identifier from step 3, change the “Target Name’’, and select the “Update’’ operation button.
8. In the events traffic view, locate the “Encrypted Characteristic Write Completed” event, select “Details”, and then verify the response value contains the updated Target Configuration information.
9. In the Target Control List Operations panel, set the “Target Identifier’’ to a different value, and set a different “Target Name’’.
10. Select the “Read Supported Configuration’’ button.
11. Select the “Add’’ operation button.
12. Select the “List’’ operation button.
13. In the events traffic view, locate the “Encrypted Characteristic Write Completed” event, select “Details”, and verify the response value contains the Target Configuration information of all of the added targets.
14. In the “Target Control List Operations” panel, set the “Target Identifier” to the one set in step 9.
15. Select the “Remove’’ operation button.
16. In the events traffic view, locate the “Encrypted Characteristic Write Completed” event, select “Details”, and then verify the response value contains the new Target Configuration information, which excludes the target removed in step 15.
17. In the “Target Control List Operations” panel, Select the “Reset’’ operation button.
18. In the events traffic view, locate the “Encrypted Characteristic Write Completed” event, select “Details”, and then verify the response contains no value.

TCRC014 **Target Control List configurations persist over an accessory power cycle.**

Applies to remote control accessories. Perform this test case with HAT using the steps below.

1. Pair and discover accessory.
2. In the left side of the controller’s window select the Target Control Management service.
3. In the Target Control List Operations panel, set the “Target Identifier” to any non-zero value, set the “Target Name”, and set the “Target Category” value to 24.
4. Select the “Read Supported Configuration” button.
5. Select the “Add” operation button.
6. Read the Target Control List characteristic.

7. In the events traffic view, locate the characteristic read completed. Select the details view and locate the buttonConfiguration Parse TLV8.
8. Note down the Target Control List configuration.
9. In the Summary panel, select the "Disconnect" button.
10. Power cycle the accessory.
11. Select discover button.
12. Read the Target Control List characteristic.
13. In the events traffic view, locate the characteristic read completed. Select the details view and locate the buttonConfiguration Parse TLV8.
14. Verify the accessory has retained the previous Target Control List configuration.

TCRC015 **Target Control List configurations must delete after last admin pairing removed.**

Applies to remote control accessories. Perform this test case with HAT using the steps below.

1. Pair and discover accessory.
2. In the left side of the controller's window select the Target Control Management service.
3. In the Target Control List Operations panel, set the "Target Identifier" to any non-zero value, set the "Target Name", and set the "Target Category" value to 24.
4. Select the "Read Supported Configuration" button.
5. Select the "Add" operation button.
6. Read the Target Control List characteristic.
7. In the events traffic view, locate the characteristic read completed. Select the details view and locate the buttonConfiguration Parse TLV8.
8. In the Pairing panel, select the Remove Pairing Button.
9. Pair and discover accessory.
10. Read the Target Control List characteristic.
11. In the events traffic view, locate the characteristic read completed. Select the details view and verify TLV8 value is not listed for the previous Target Control List.

TCRC016 **Target Control List configurations must delete after factory reset.**

Applies to remote control accessories. Perform this test case with HAT using the steps below.

1. Pair and discover accessory.
2. In the left side of the controller's window select the Target Control Management service.

3. In the Target Control List Operations panel, set the “Target Identifier” to any non-zero value, set the “Target Name”, and set the “Target Category” value to 24.
4. Select the “Read Supported Configuration” button.
5. Select the “Add” operation button.
6. Read the Target Control List characteristic.
7. In the events traffic view, locate the characteristic read completed. Select the details view and locate the buttonConfiguration Parse TLV8.
8. Perform a factory reset on the accessory.
9. Pair and discover accessory.
10. Read the Target Control List characteristic.
11. In the events traffic view, locate the characteristic read completed. Select the details view and verify TLV8 value is not listed for the previous Target Control List.

TCRC017 **Any requests to enable notification on the Button Event characteristic by non admin controllers must result in error -70401 (Insufficient Privileges).**

Applies to remote control accessories. Perform this test case with HAT using the steps below.

1. Pair and discover accessory.
2. In Controllers window, select “+” to create a new IP Controller 2.
3. Under Controller 1, select the accessory name, under “Add Additional Controllers” panel, select Controller 2 as Controller and select the “Add Controller” button.
4. Under Controller 2, select accessory, and select the “Discover” button.
5. From Controller 2, Enable notifications on the Button Event characteristic.
6. In the HTTP traffic view, verify the accessory responds with HTTP 207 with HAP status code -70401 (Insufficient Privileges).

TCRC018 **Only admin controllers are allowed to perform any operations on the Target Control List characteristic. Any read/writes to this characteristic by non admin controllers must result in error -70401 (Insufficient Privileges).**

Applies to remote control accessories. Perform this test case with HAT using the steps below.

1. Pair and discover accessory.
2. In Controllers window, select “+” to create a new BLE or IP Controller 2.
3. Under Controller 1, select the accessory name, under “Add Additional Controllers” panel, select Controller 2 as Controller and select the “Add Controller” button.
4. Under Controller 2, select accessory, and select the “Discover” button.

5. In the left side of the controller's window select the Target Control Management service.
6. In the Target Control List Operations panel, set the "Target Identifier'' to any non-zero value, set the "Target Name'', and set the "Target Category'' value to 24.
7. Select the "Read Supported Configuration'' button.
8. Select the "Add'' operation button.
9. In the HTTP traffic view, verify the accessory responds with HTTP 207 with HAP status code -70401 (Insufficient Privileges).
10. Read the Target Control List characteristic.
11. In the HTTP traffic view, verify the accessory responds with HTTP 207 with HAP status code -70401 (Insufficient Privileges).

TCRC019 **If no Target is currently selected (i.e. not configured or non ATV selected), the Active Identifier characteristic value must be 0.**

Applies to remote control accessories. Perform this test case with HAT using the steps below.

1. Pair and discover accessory.
2. Read the Active Identifier characteristic.
3. Verify the value is "0".
4. In the left side of the controller's window select the Target Control Management service.
5. In the Target Control List Operations panel, set the "Target Identifier'' to any non-zero value, set the "Target Name'', and set the "Target Category'' value to 24.
6. Select the "Read Supported Configuration'' button.
7. Select the "Add'' operation button.
8. Read the Target Control service Active Identifier characteristic.
9. Verify the Identifier value matches the Target Identifier.

TCRC020 **When Active Identifier is changed the Active characteristic must reset to inactive.**

Applies to remote control accessories. Perform this test case with HAT using the steps below.

1. Pair and discover accessory.
2. Read the Active characteristic and verify value is 0.
3. In the left side of the controller's window select the Target Control Management service.
4. In the Target Control List Operations panel, set the "Target Identifier'' to "111", set the "Target Name'' to "one", and set the "Target Category'' value to 24.
5. Select the "Read Supported Configuration'' button.

6. Select the “Add’’ operation button.
7. Read the Active Identifier characteristic and verify the value is 111.
8. Enable notifications on the Active characteristic.
9. Write 1 to the Active characteristic.
10. In the left side of the controller’s window select the Target Control Management service.
11. In the Target Control List Operations panel, set the “Target Identifier’’ to “222”, set the “Target Name’’ to “two”, and set the “Target Category’’ value to 24.
12. Select the “Read Supported Configuration’’ button.
13. Select the “Add’’ operation button.
14. From the remote, select the “222” Target Identifier.
15. Verify a notification was received from the Active characteristic for value 0.

TCRC022 **Accessory must not allow new targets to be added once the advertised Maximum Targets limit has been met.**

Applies to remote control accessories. Perform this test case with HAT using the steps below.

1. Pair and discover accessory.
2. Read the Target Control Supported Configuration characteristic in the Target Control Management Service.
3. In the Events view, notate the Maximum Targets value.
4. In the left side of the controller’s window, select the Target Control Management service.
5. In the Target Control List Operations panel, set the “Target Identifier’’ to 1, set the “Target Name’’, and set the “Target Category’’ value to 24.
6. Select the “Read Supported Configuration’’ button.
7. Select the “Add’’ operation button.
8. Repeat steps 5 thru 7 until the Maximum Targets limit is met, incrementing the Target Identifier value accordingly.
9. Add another target to exceed the Maximum Target limit, and verify the accessory returns an error response.

TCRC023 **Verify that the accessory reports the correct “Type” of remote in its Target Control Supported Configuration characteristic. Verify software based remotes do not include the Siri service.**

Applies to remote control accessories. Perform this test case with HAT using the steps below.

1. Pair and discover accessory.

2. Read the Target Control Supported Configuration characteristic in the Target Control Management Service.
3. In the Events view, locate the Characteristic Read Completed response, and select the Details button.
4. Verify the TLV "Type" (0x04) value is either "0" for software based remotes, or "1" for hardware based remotes.
5. If the value is "0" for software based, verify the Siri service is not included.

1.12 IP

TCI005: The minimum number of pairing relationships that an accessory must support is 16.

TCI006: Accessory must be able to support 8 simultaneous connections.

TCI008: Accessory must close the connection if authTag verification fails.

TCI009: If the HAP accessory server uses Wi-Fi, it must either be listed as a Wi-Fi CERTIFIED product with the Wi-Fi Alliance or has passed the MFi Wi-Fi Verification.

TCI010: If the HAP accessory server uses Wi-Fi, it must support Wi-Fi Accessory Configuration (WAC) as one method of getting Wi-Fi credentials.

TCI011: Accessory must be able to join a Wi-Fi network that uses a 32-bit unicode character in its network name.

TCI012: Accessory in WAC mode must have the correct bits set in the Apple IE.

TCI013: Bonjour Conformance Test

TCI014: The accessory must support event notifications for multiple connections.

TCI015: Accessory must respond to every HTTP request with a HTTP response that includes the appropriate status code (200, 204).

TCI018: The HAP accessory object with the instance ID of "1" must be the primary HAP accessory object. For bridges, the primary HAP accessory object must be the bridge itself.

TCI019: Bridge accessories must expose all user-addressable functionality supported by any connected devices as HAP accessory objects.

TCI020: Bridge accessories containing more than one accessory object must ensure that object instance IDs persist across reboots and power cycles. A bridge must not contain multiple accessories with the same instance ID.

TCI021: Services contained within the HAP accessory must be colocated. For example, a fan with a light on it would expose a single HAP accessory with 3 services: the required accessory information service, a fan service and a Light Bulb service. Conversely, a bridge that bridges 2 independent lights that may be in different physical locations must expose a HAP accessory object for each independent light.

TCI022: Accessories must supply required properties in each service object: Type, Instance ID, Characteristics, Hidden Services (Conditional), Primary Services (Optional), Linked Services (Optional). When Apple-defined UUIDs are encoded as JSON strings, a short form must be used by including only the first 8 characters with leading zeros removed.

TCI023: The accessory must include the following properties for each characteristic that supports paired read: Type, Instance ID, Permissions, Value, Format. When Apple-defined UUIDs are encoded as JSON strings, a short form must be used by including only the first 8 characters with leading zeros removed.

TCI025: The accessory must include the following properties for each characteristic that does not support paired read: Type, Instance ID, Permissions, and Format. When Apple-defined UUIDs are encoded as JSON strings, a short form must be used by including only the first 8 characters with leading zeros removed.

TCI026: For characteristics that do not support notifications, verify correct error response when trying to enable notifications.

TCI027: The notification event must be generated on the characteristic that notifications are enabled on and not generated on other characteristic.

TCI028: The name of the Bonjour service (i.e., the user-visible name of accessory) must match the accessory name.

TCI029: The required Bonjour TXT keys must be supplied in the accessory's Bonjour advertisement.

TCI030: The Device ID must persist across a reboot and is randomly generated when accessory is factory reset.

TCI032: The following device information must persist across reboots and power cycles: Device ID (id), Configuration number (c#), Accessory category identifier (ci).

TCI033: If supported, accessory's configuration number must increment when a service or characteristic is added to or removed from accessory server.

TCI034: For characteristics that don't support paired write, write attempts should fail with the correct error.

TCI035: For characteristics that don't support paired read, read attempts should fail with the correct error.

TCI036: When a bridged accessory is powered off, the bridge must return the correct error.

TCI037: If an accessory has characteristics that have minimum value and maximum value metadata, writing values below the minimum value and above the maximum value must not be accepted by accessory.

TCI042: Accessories must supply required properties inside each accessory: Accessory Instance ID and Services.

TCI044: The accessory must support writing values to one or more characteristics via a single HTTP request.

TCI045: Sending an unencrypted HTTP request after opening a connection must fail with the HTTP Status Code 470 Connection Authorization Required. A paired or unpaired accessory must reject HTTP requests to the following paths on connections where a security session has not been established: GET /accessories, GET /characteristics, PUT /characteristics, POST /pairings.

TCI047: Accessory must always successfully deliver event notifications for every characteristic that supports them when a single client has subscribed multiple times.

TCI048: State number (s#) must have an initial value of 1. State number (s#) must have a range of 1-65535 and wrap to 1 when it overflows. State number (s#) increments each time a paired accessory's state changes while not connected to a paired HomeKit controller.

TCI050: Accessories must respond to all requests within 10 seconds.

TCI005 **The minimum number of pairing relationships that an accessory must support is 16.**

Applies to accessories that use HAP over Ethernet or Wi-Fi. Perform this test case automatically with HCA. Below are steps that can be used to perform this test case manually in HAT.

1. Pair and discover accessory.
2. Select the "+" at the bottom of left sidebar.
3. Select "Create IP Controller" to create 15 additonal controllers, for a total of 16 controllers.
4. Using admin Controller 1, add pairings to each of the 15 secondary controllers.
5. Select "List Pairings" button in the Controllers window.
6. In the Event view of the Trace window, verify the response to "List Pairings Completed" event shows all 16 pairings.

7. On the left pane of the Controllers window, under Controller 2, select the accessory name, select the "Start" button, and select the "Discover" button.
8. In the Event view of the Trace window, verify Pair-Verify completes successfully.
9. Repeat steps 7-8 for each additional controller.

TCI006 **Accessory must be able to support 8 simultaneous connections.**

Applies to accessories that use HAP over Ethernet or Wi-Fi. Perform this test case automatically with HCA. Below are steps that can be used to perform this test case manually in HAT.

1. Pair and discover accessory (Controller 1).
2. Select the "+" at the bottom of left sidebar.
3. Select "Create IP Controller" to make new virtual controllers.
4. Continue to add new virtual controllers until 7 additional controllers have been added.
5. In left sidebar, under the Admin controller, select the paired accessory server.
6. In Controllers window, under "Add Additional Controllers" panel, select Controller # for each IP virtual controller and select "Add Controller" button.
7. Continue until 8 pairings have been established.
8. In left sidebar, select each newly added Controller and select the "Start" button.
9. In left sidebar, under the newly added controller, select the paired accessory server and select the "Discover" (which performs a pair-verify) button.
10. Repeat steps 8 and 9 until the 8th controller has been Pair-Verified.
11. Verify all 8 controllers simultaneously maintain connection with accessory without any disconnection.

TCI008 **Accessory must close the connection if authTag verification fails.**

Applies to accessories that use HAP over Ethernet or Wi-Fi.

No test steps beyond test case description at this time.

TCI009 **If the HAP accessory server uses Wi-Fi, it must either be listed as a Wi-Fi CERTIFIED product with the Wi-Fi Alliance or has passed the MFi Wi-Fi Verification.**

Applies to accessories that use HAP over Wi-Fi.

1. Register accessory as a Wi-Fi CERTIFIED product with the Wi-Fi Alliance (http://www.wi-fi.org/certification). For accessories that completed MFi Wi-Fi Verification, proceed to step 2.
2. Enter the Wi-Fi certificate ID or MFi W-Fi Verification ID on the Accessory Compliance Questionnaire.
3. The Accessory Compliance Questionnaire is submitted through the MFi Portal after Product Plan approval.

TCI010 **If the HAP accessory server uses Wi-Fi, it must support Wi-Fi Accessory Configuration (WAC) as one method of getting Wi-Fi credentials.**

Applies to accessories that use HAP over Wi-Fi.

- This applies to accessories that support legacy Wi-Fi Accessory Configuration.
- If the accessory supports Wi-Fi Accessory Configuration 2, see Wi-Fi Accessory Configuration 2 section for testing.
- Complete the Wi-Fi Accessory Association Verification Tests as defined in the Accessory Interface Specification - Wi-Fi Accessory Configuration Addendum, including all tests therein:

1. Wi-Fi Accessory Configuration Mode Automatic Shutoff (15 min timeout)
2. 802.11 b/g Association Verification
3. 802.11 Non-broadcast SSID Association Verification
4. 2.4 GHz vs 5 GHz Beaconing Tests
5. Security Mode Verification Tests for WPA2 Personal
6. IPv4 DHCP
7. IPv4 Link Local
8. Bonjour TXT Records Tests for ADD and RMV

TCI011 **Accessory must be able to join a Wi-Fi network that uses a 32-bit unicode character in its network name.**

Applies to accessories that use HAP over Wi-Fi. Perform this test case using the Home app on iOS.

1. Put Wi-Fi accessory in WAC mode.
2. Add unicode character to your Wi-Fi Network Name.
3. Select the accessory to join your network with an iOS device.
4. Verify accessory has successfully joined the network.

TCI012 **Accessory in WAC mode must have the correct bits set in the Apple IE.**

Applies to accessories that use HAP over Wi-Fi.

For accessories that support legacy WAC, the following bits must be set:

- MFi Accessory Configuration (bit 2)
- HomeKit Accessory Protocol (bit 17)

For accessories that support WAC2, the following bits must be set:

- HomeKit Accessory Configuration (bit 17)
- WAC with pair-setup (bit 20)

For accessories that support WAC2, one of the following bits must also be set:

- MFi Authentication Chip (bit 21)
- HomeKit Software Token-Based Authentication (bit 22)

1. Perform a factory reset on the accessory and put into WAC mode.
2. Launch HAT.
3. On the left hand side of the Controllers Window, under IP Controllers, select Controller, in the Discovery panel, select Start button for Status.
4. In the WAC Discovery traffic view, select the accessory and select the "Details" button.
5. Expand the Flags view and verify the correct Apple IE bits have been set.

TCI013 **Bonjour Conformance Test**

Applies to accessories that use HAP over Ethernet or Wi-Fi.

Use the Bonjour Conformance Test from https://developer.apple.com/softwarelicensing/agreements/bonjour.php

For additional information regarding the Bonjour Conformance Test (BCT), refer to the Bonjour Conformance Test Guideline at https://developer.apple.com/bonjour/index.html

Accessory must pass the following tests and all of the associated sub-tests on all supported IP interfaces:

- Link-local Address Allocation
- IPv4 Multicast-DNS
- IPv6 Multicast-DNS
- Mixed-network Interoperability

* If the accessory is left unpaired, accessory must disable HAP server within 10 minutes, unless pairing is completed. This may prevent the accessory from being available beyond 10 minutes for BCT testing. To overcome this, pair to the accessory using HAT, then close the HAT app so that the accessory is still paired, and then start BCT testing.

TCI014 **The accessory must support event notifications for multiple connections.**

Applies to accessories that use HAP over Ethernet or Wi-Fi. Perform this test case using HCA.

1. Pair and discover accessory (Controller 1).
2. Select the "+" at the bottom of left the sidebar, then select "Create IP Controller" to make a new controller.
3. In the left sidebar, under Controller 1, select the accessory name.

4. Under the "Add Additional Controllers" panel, select "Controller 2" as Controller and "on" for Admin, then select the "Add Controller" button.
5. In the left sidebar, under Controller 2, select the accessory name, then select the "Discover" button.
6. Select the "+" at the bottom of the left sidebar, then select "Create IP Controller" to make a new controller.
7. In the left sidebar, under Controller 1, select the accessory name.
8. In the Controllers window, under the "Add Additional Controllers" panel, select "Controller 3" as Controller and "on" for Admin, then select the "Add Controller" button.
9. In the left sidebar, under Controller 3, select the accessory name, then select the "Discover" button.
10. Using Controller 1, select the first characteristic that supports the "Notify"permission, and select the "Enable" button to enable event notifications.
11. Repeat step 10 for controllers 2 and 3.
12. If the characteristic supports the "Paired Write" permission, using Controller 1 write a valid value other than the current value to the characteristic.
13. Using the HTTP view of the trace window, verify that "EVENT/" notifications were only sent to Controller 2 and Controller 3.
14. Verify that the notifications contain the correct value, characteristic IID, and AID.
15. If the characteristic supports the "Paired Read" permission, and the characteristic's state can be changed outside of HAP (e.g., by physical means of interaction, remote control, or app), change the state of the accessory once (e.g., turn light on).
16. Verify that "EVENT/"notifications were sent to all controllers: Controller 1, Controller 2, and Controller 3.
17. Verify that the notifications contain the correct value, IID, and AID.
18. Using Controller 1, select the characteristic that supports the "Notify"permission from step 10, and select the "Disable" button to disable event notifications.
19. Repeat step 18 for Controller 2 and 3.
20. Repeat steps 10-19 for the next characteristic that supports the "Notify" permission.

TCI015 **Accessory must respond to every HTTP request with a HTTP response that includes the appropriate status code (200, 204).**

Applies to accessories that use HAP over Ethernet or Wi-Fi. Perform this test case automatically with HCA. Below are steps that can be used to perform this test case manually in HAT.

1. Pair and discover accessory.
2. In the HTTP traffic view, verify response to GET /accessories request contains HTTP header "200 OK."
3. Write to a paired write characteristic.

4. In the HTTP traffic view, verify response HTTP header contains “204 No Content”.
5. Write to a read-only characteristic.
6. In the HTTP traffic view, verify response HTTP header contains “207 Multi-Status”.

TCI018 **The HAP accessory object with the instance ID of “1” must be the primary HAP accessory object. For bridges, the primary HAP accessory object must be the bridge itself.**

Applies to accessories that use HAP over Ethernet or Wi-Fi. Perform this test case automatically with HCA. Below are steps that can be used to perform this test case manually in HAT.

1. Pair and discover accessory.
2. In the Events traffic view, select Discovered Accessories packet.
3. Select the Details button.
4. See Accessories; scroll down to locate “Primary: Yes.”
5. Verify that the primary accessory object has instance ID “1.”
6. If accessory is a bridge, verify that the bridge itself is the primary accessory object.

TCI019 **Bridge accessories must expose all user-addressable functionality supported by any connected devices as HAP accessory objects.**

Applies to bridge accessories. Perform this test case with HAT using the steps below.

1. Connect non-HAP accessory sample to the bridge.
2. In HAT, pair and discover the bridge and its accessory objects.
3. Verify that one HAP accessory is seen for the bridge itself.
4. Verify that each accessory sample connected to the bridge is identified as a HAP object (e.g., a Light Bulb accessory identifies itself as a light accessory object and includes the required characteristics for a Light Bulb service).

TCI020 **Bridge accessories containing more than one accessory object must ensure that object instance IDs persist across reboots and power cycles. A bridge must not contain multiple accessories with the same instance ID.**

Applies to bridge accessories. Perform this test case with HAT using the steps below.

1. Connect at least one accessory to the bridge.
2. In HAT, pair and discover the bridge and its accessories.
3. Verify the instance ID values of the bridge and its accessory objects.
4. In the Summary panel, select the “Disconnect” button.

5. Power the bridge off and on.
6. Select Discover in Controllers window.
7. Verify the instance ID values of the bridge and its accessories remain the same after power cycle. Verify accessories do not contain duplicate instance IDs.

TCI021 **Services contained within the HAP accessory must be colocated. For example, a fan with a light on it would expose a single HAP accessory with 3 services: the required accessory information service, a fan service and a Light Bulb service. Conversely, a bridge that bridges 2 independent lights that may be in different physical locations must expose a HAP accessory object for each independent light.**

Applies to accessories that use HAP over Ethernet or Wi-Fi. Perform this test case with HAT using the steps below.

1. Pair and discover accessory.
2. Locate accessory's response to HTTP GET /accessories in the HTTP traffic view.
3. Select the Details button.
4. See Attribute Database Object; use disclosure arrows to show and hide details.
5. If accessory server contains 1 physical accessory with more than 1 service:
6. Verify that its services are located within 1 accessory object.
7. If accessory server contains more than 1 physical accessory:
8. Verify there is 1 accessory object for each physical accessory.

TCI022 **Accessories must supply required properties in each service object: Type, Instance ID, Characteristics, Hidden Services (Conditional), Primary Services (Optional), Linked Services (Optional). When Apple-defined UUIDs are encoded as JSON strings, a short form must be used by including only the first 8 characters with leading zeros removed.**

Applies to accessories that use HAP over Ethernet or Wi-Fi. Perform this test case with HAT using the steps below.

1. Pair and discover accessory.
2. Locate accessory's response to HTTP GET / accessories in the HTTP traffic view.
3. Select Details button to reveal Details sidebar.
4. Within each service object, verify that the following properties are provided: type, instance ID, characteristics, Hidden Services (Conditional), Primary Services (Optional), and Linked Services (Optional).
5. Under the type property verify short format Apple-defined UUIDs are used. (e.g. Accessory Information Service full UUID "0000003E-0000-1000-8000-0026BB765291" and short UUID "3E")

TCI023 **The accessory must include the following properties for each characteristic that supports paired read: Type, Instance ID, Permissions, Value, Format. When Apple-defined UUIDs are encoded as JSON strings, a short form must be used by including only the first 8 characters with leading zeros removed.**

Applies to accessories that use HAP over Ethernet or Wi-Fi. Perform this test case automatically with HCA. Below are steps that can be used to perform this test case manually in HAT.

1. Pair and discover accessory.
2. Locate accessory's response to HTTP GET /accessories in the HTTP traffic view.
3. Select the Details button.
4. Within each characteristic object, verify that the following properties are provided: type, instance ID, permissions, value and format.
5. Under the type property, verify short format Apple-defined UUIDs are used. (e.g. Name Characteristic full UUID "00000023-0000-1000-8000-0026BB765291" and short UUID "23")

TCI025 **The accessory must include the following properties for each characteristic that does not support paired read: Type, Instance ID, Permissions, and Format. When Apple-defined UUIDs are encoded as JSON strings, a short form must be used by including only the first 8 characters with leading zeros removed.**

Applies to accessories that use HAP over Ethernet or Wi-Fi. Perform this test case automatically with HCA. Below are steps that can be used to perform this test case manually in HAT.

1. Pair and discover accessory.
2. Locate accessory's response to HTTP GET /accessories in the HTTP traffic view.
3. Select the Details button.
4. Within each characteristic object, verify that the following properties are provided: type, instance ID, permissions and format.
5. For each characteristic object that does not contain the permission "paired read", the "value" key-value pair must not be supplied.
6. Under the type property, verify short format Apple-defined UUIDs are used. (e.g. Name Characteristic full UUID "00000023-0000-1000-8000-0026BB765291" and short UUID "23")

TCI026 **For characteristics that do not support notifications, verify correct error response when trying to enable notifications.**

Applies to accessories that use HAP over Ethernet or Wi-Fi. Perform this test case automatically with HCA. Below are steps that can be used to perform this test case manually in HAT.

1. Pair and discover accessory.
2. From the Controller's Window, locate each characteristic that does not support "ev" (event notifications) in the perms key, select the enable notifications button.

3. In the HTTP traffic view, verify accessory responds with HTTP 207 with HAP status code -70406.

TCI027 **The notification event must be generated on the characteristic that notifications are enabled on and not generated on other characteristic.**

Applies to accessories that use HAP over Ethernet or Wi-Fi. Perform this test case with HAT using the steps below.

1. Pair and discover accessory.
2. In the Controllers window, locate characteristics that support Event Notifications and select enable.
3. Make a change from accessory to each of the characteristics that has notifications enabled.
4. Verify that the notification event is generated on each characteristic that the notifications are enabled on.
5. Verify that notifications are not generated on other characteristics.

TCI028 **The name of the Bonjour service (i.e., the user-visible name of accessory) must match the accessory name.**

Applies to accessories that use HAP over Ethernet or Wi-Fi. Perform this test case automatically with HCA. Below are steps that can be used to perform this test case manually in HAT.

1. Pair and discover accessory.
2. Select accessory's name in left sidebar of Controllers window.
3. To locate name of the HAP accessory object, see Summary > Accessory Name.
4. To locate Bonjour name, see Advertisement Information > Bonjour Service Name.
5. Verify that these 2 values match.

TCI029 **The required Bonjour TXT keys must be supplied in the accessory's Bonjour advertisement.**

Applies to accessories that use HAP over Ethernet or Wi-Fi. Perform this test case automatically with HCA. Below are steps that can be used to perform this test case manually in HAT.

1. Pair and discover accessory.
2. Go to the Bonjour Discovery traffic view, select the Bonjour record for the accessory, and open the details sidebar.
3. Verify the presence of the Device ID (id) key.
4. Verify the presense of the Model (md) key.
5. Verify the presense of the Protocol version (pv) key.
6. Verify the presense of the Configuration number (c#) key.

7. Verify the presense of the Status flags (sf) key, required if non-zero value.
8. Verify the presense of the Accessory category identifier (ci) key.
9. Verify the presense of the State number (s#) key.
10. Verify the presense of the Setup Hash (sh) key, required if the the accessory supports enhanced setup payload information.
11. Verify the presense of the Pairing Feature flags (ff) key, required if non-zero.

TCI030 **The Device ID must persist across a reboot and is randomly generated when accessory is factory reset.**

Applies to accessories that use HAP over Ethernet or Wi-Fi. Perform this test case with HAT using the steps below.

1. For accessories that support Legacy WAC, WAC accessory to the network and perform pair-setup, then continue to step 10. For HAP over Ethernet, complete pairing and then continue to step 10.
2. For accessories that support WAC2, proceed to step 3.
3. Launch HAT and select the accessory on the left side of the controllers window.
4. In the Wi-Fi Accessory Configuration panel, select the “Join access point” button.
5. In the Pairing Panel, select the “Start Pairing” and complete pair-setup.
6. In the Wi-Fi Accessory Configuration panel, enter the Wi-fi SSID and Wi-Fi PSK, then select the “Send WAC Configuration” button.
7. Ensure your Mac is on the network you expect the accessory to join, otherwise re-join the expected network.
8. Once the accessory begins advertising via Bonjour on the newly joined network via _hap._tcp, select the “Confirm WAC Configuration” button.
9. Verify the accessory responds to /Configured with HTTP Response: HTTP/1.1 200 OK
10. In Controllers window, note the Device ID in the Advertisement Information Panel.
11. In the Summary panel, select the “Disconnect” button.
12. Power cycle accessory.
13. Select the Discover button in the Pairing Panel.
14. Verify the Device ID did not change.
15. Perform factory reset on accessory.
16. Wait for the accessory to begin advertising over Bonjour. For HAP over Ethernet, proceed to step 18.
17. In the WAC Discovery traffic view, look for the most recent accessory advertisement. Select details button and verify Device ID is randomly generated and differs from Device ID from step 10.
18. Perform WAC or WAC2 (if applicable), then perform pair-setup and select Discover.

19. Wait for the accessory to begin advertising over Bonjour. In Controllers window, note the Device ID in the Advertisement Information Panel. Verify the Device ID not change from step 16.
20. Power cycle the accessory.
21. Wait for the accessory to begin advertising over Bonjour. In Controllers window, note the Device ID in the Advertisement Information Panel. Verify the Device ID not change from step 16.

TCI032 **The following device information must persist across reboots and power cycles: Device ID (id), Configuration number (c#), Accessory category identifier (ci).**

Applies to accessories that use HAP over Ethernet or Wi-Fi. Perform this test case with HAT using the steps below.

1. Pair and discover accessory.
2. Select accessory's name in left sidebar of Controllers window.
3. See Advertisement Information in Controllers window.
4. Note the current Device ID, configuration number and accessory category identifier values.
5. In the Summary panel, select the "Disconnect" button.
6. Power accessory off and on.
7. Select Discover in Controllers window.
8. Repeat steps 2-4.
9. Verify that the device information values did not change.

TCI033 **If supported, accessory's configuration number must increment when a service or characteristic is added to or removed from accessory server.**

Applies to accessories that use HAP over Ethernet or Wi-Fi. Perform this test case with HAT using the steps below.

1. Pair and discover accessory.
2. Select accessory's name in left sidebar of Controllers window.
3. See the Advertisement Information panel in the Controllers window.
4. Note the current configuration number.
5. Add a service or characteristic to accessory or remove a service or characteristic from accessory.
6. Note the configuration number.
7. Verify that the configuration number has incremented.

TCI034 **For characteristics that don't support paired write, write attempts should fail with the correct error.**

Applies to accessories that use HAP over Ethernet or Wi-Fi. Perform this test case automatically with HCA. Below are steps that can be used to perform this test case manually in HAT.

1. Pair and discover accessory.
2. In Controllers window, select a characteristic that does not support paired write.
3. Write a value to this characteristic.
4. See the HTTP traffic view.
5. Verify that the accessory responds with "HTTP 207" with HAP status code "-70404."

TCI035 **For characteristics that don't support paired read, read attempts should fail with the correct error.**

Applies to accessories that use HAP over Ethernet or Wi-Fi. Perform this test case automatically with HCA. Below are steps that can be used to perform this test case manually in HAT.

1. Pair and discover accessory.
2. In Controllers window, select a characteristic that does not support a paired read.
3. Read this characteristic.
4. See the HTTP traffic view.
5. Verify that the accessory responds with "HTTP 207" with HAP status code "-70405."

TCI036 **When a bridged accessory is powered off, the bridge must return the correct error.**

Applies to bridge accessories. Perform this test case with HAT using the steps below.

1. Pair and discover accessory.
2. Remove power from an end point (e.g., a light).
3. Read and/or write to the end point's Characteristics.
4. See the HTTP traffic view.
5. Verify that the accessory responds with "HTTP 207" with HAP status code "-70402."

TCI037 **If an accessory has characteristics that have minimum value and maximum value metadata, writing values below the minimum value and above the maximum value must not be accepted by accessory.**

Applies to accessories that use HAP over Ethernet or Wi-Fi. Perform this test case automatically with HCA. Below are steps that can be used to perform this test case manually in HAT.

1. Pair and discover accessory.
2. In left sidebar of Controllers window, see each of the accessory's services.

3. Using the Write button in Controllers window, write to each paired Read/Write characteristic with a value that is below the minValue if the value to be written is valid for the specified format (e.g., for a format of uint8 and a minValue of 0, this step can be skipped).
4. Verify that Accessory responds to out of range values with the HTTP status code "207" and HAP status code "-70410."
5. Using the Write button in Controllers window, write to each paired Read/Write characteristic with a value that is above the maxValue if the value to be written is valid for the specified format (e.g., for a format of uint8 and a maxValue of 255, this step can be skipped).
6. Verify that Accessory responds to out of range values with the HTTP status code "207 "and HAP status code "-70410."

TCI042 **Accessories must supply required properties inside each accessory: Accessory Instance ID and Services.**

Applies to accessories that use HAP over Ethernet or Wi-Fi. Perform this test case with HAT using the steps below.

1. Pair and discover accessory.
2. Locate accessory's response to HTTP GET /accessories in the HTTP traffic view.
3. Select Details button to reveal Details sidebar.
4. Within each accessory, verify that the following properties are provided: Accessory instance ID and Services.

TCI044 **The accessory must support writing values to one or more characteristics via a single HTTP request.**

Applies to accessories that use HAP over Ethernet or Wi-Fi. Perform this test case automatically with HCA. Below are steps that can be used to perform this test case manually in HAT.

1. Pair and discover accessory.
2. In Controllers window, select any characteristic.
3. In the Queued Writing Panel, Enable Queue State.
4. Write value in the Queue Characteristic Panel and select Queue button.
5. Select another characteristic.
6. Write value in the Queue Characteristic Panel to a read-only characteristic and select Queue button.
7. In the pop-up window, select the Write button.
8. View in accessory response in the HTTP traffic view, select Details button.
9. Verify the appropriate characteristic response with HTTP 207 HAP and for a successful write with a status: 0 and a unsuccessful write with a status: -70404.

TCI045 **Sending an unencrypted HTTP request after opening a connection must fail with the HTTP Status Code 470 Connection Authorization Required. A paired or unpaired accessory must reject HTTP requests to the following paths on connections where a security session has not been established: GET /accessories, GET /characteristics, PUT /characteristics, POST /pairings.**

Applies to accessories that use HAP over Ethernet or Wi-Fi. Perform this test case using HCA.

TCI047 **Accessory must always successfully deliver event notifications for every characteristic that supports them when a single client has subscribed multiple times.**

Applies to accessories that use HAP over Ethernet or Wi-Fi. Perform this test case with HAT using the steps below.

1. Pair and discover accessory with Controller 1.
2. In Controllers window, select "+" to create a new IP Controller 2.
3. Under Controller 1, select the accessory name, under "Add Additional Controllers" panel, select "Controller 2" as Controller select the "Add Controller" button.
4. On the left pane of the Controllers window, select the accessory name under Controller 2, select "Start" button and select "Discover" button.
5. Repeat steps 3-5 to add a 3rd controller.
6. From Controller 1, navigate to each characteristic that supports Event Notifications, and subscribe multiple times to Event Notifications by selecting "Enable" button 3 times.
7. Repeat step 6 with each additional controller.
8. For characteristics that provide physical means of interaction, physically toggle each applicable characteristic on the accessory.
9. Verify that each of the controllers receive only a single notification for each state change.
10. For characteristics that supports paired write and event notifications, use Controller 1 to write a new value to each of the characteristics.
11. Verify Controller 2 and Controller 3 each receive only a single notification for each state change.

TCI048 **State number (s#) must have an initial value of 1. State number (s#) must have a range of 1-65535 and wrap to 1 when it overflows. State number (s#) increments each time a paired accessory's state changes while not connected to a paired HomeKit controller.**

Applies to accessories that use HAP over Ethernet or Wi-Fi. Perform this test case with HAT using the steps below.

Accessory state change for HAP over Wi-Fi or Ethernet accessories are defined for the following cases:

- Accessory reboot or restart due to any reason. (i.e. after every reboot or restart, the s# value must increment by one from the last value prior to the restart).
- Accessory re-publishes HomeKit service while in disconnected state (i.e. changes in HomeKit-specific Bonjour text records while it is not connected to any controller).

- A characteristic that supports notification changes while a paired accessory is in disconnected state (i.e. when not connected with a secure session to at least one HomeKit paired controller). The state number should increment only once for multiple characteristic value changes while in disconnected state until the accessory state changes from disconnected state to connected state.

1. Factory reset the accessory.
2. For accessories that do not support WAC2, proceed to step 4.
3. For accessories that support WAC2, select the accessory server, select "Join access point", and then wait for the accessory to begin advertising via Bonjour.
4. Select the most recent Bonjour advertisement in the Bonjour Discovery view of the trace window.
5. Open the details sidebar and verify the s# value is less than 5.
6. Pair and discover accessory. For accessories that support WAC2, select "Start Pairing" and complete pairing process.
7. In the Bonjour Discovery view, select the most recent Bonjour advertisement and open the details sidebar. Verify the s# value is less than 5.
8. For accessories that do not support WAC2 (i.e. Ethernet), proceed to step 15.
9. For accessories that support WAC2, complete the WAC2 procedure below:
10. In the Wi-Fi Accessory Configuration panel, enter the Wi-fi SSID and Wi-Fi PSK and select the "Send WAC Configuration" button.
11. Ensure your Mac is on the network you expect the accessory to join, otherwise re-join the expected network.
12. Once the accessory begins advertising via Bonjour on the newly joined network, select the "Confirm WAC Configuration" button.
13. After the accessory successfully responds to the /Configured request, select the "Discover" button.
14. Select the most recent Bonjour advertisement and then open the details sidebar. Verify the s# value does not increment more than 2 from the value in step 7 for WAC2.
15. Select the "Disconnect" button.
16. Power cycle the accessory and wait for accessory to begin advertising via Bonjour.
17. In the Bonjour Discovery view, select the most recent Bonjour advertisement and open the details sidebar. Verify the s# has incremented. For WAC2 the s# value has not incremented more than 2. For Ethernet the s# value has not incremented more than 2 from the value in step 7.
18. Select "Discover" button in the Controllers window.
19. For each characteristic that supports notifications, change the characteristic value(s) multiple times from controller with valid write operation.
20. In the Bonjour Discovery view, select the most recent Bonjour advertisement and open the details sidebar. Verify the s# has not incremented. For WAC2 the s# value is the same as the value from step 17. For Ethernet the s# value does not change from the value in step 17.

21. Select the “Disconnect” button.
22. For each characteristic that provides physical means of interaction and supports Event Notifications, change the state of the characteristic several times (e.g., toggle switch on/off).
23. In the Bonjour traffic view, select the most recent Bonjour advertisement and open the details sidebar. Verify the s# has incremented by at least one and is greater than the value in step 20.
24. Remove pairings from the accessory using HAT.

TCI050 **Accessories must respond to all requests within 10 seconds.**

Applies to accessories that use HAP over Ethernet or Wi-Fi. Perform this test case with HAT using the steps below.

1. Pair and discover accessory.
2. In the HTTP traffic view, locate the Pair Setup Requests/Responses, and select details.
3. Verify that the Pair Setup procedure, from state value 3 (M3) through state value 6 (M6), completes within 10 seconds.
4. In the HTTP traffic view, locate the Pair Verify Requests/Responses, and select details.
5. Verify that the Pair Verify procedure, from state value 1 (M1) through state value 4 (M4), completes within 10 seconds.
6. In the Summary panel, select the Disconnect button.
7. Discover the accessory.
8. In the HTTP traffic view, locate the Pair Verify Requests/Responses, and select details.
9. Verify that the Pair Verify procedure, from state value 1 (M1) through state value 4 (M4), completes within 10 seconds.

1.13 IP Timed Write

TCITW001: An accessory must support timed writes to all characteristics even if the characteristic does not require it.

TCITW002: Prepare-write with PID 111. Prepare-write with PID 222. Execute-write with PID 222.

TCITW003: Prepare-write with PID 111. Prepare-write with PID 222. Execute-write with PID 111.

TCITW004: Prepare-write with PID 111. Wait for TTL to expire. Prepare-write with PID 111. Execute-write with PID 111.

TCITW005: Execute-write with PID 111 without Prepare-write.

TCITW006: Prepare-write with PID 222. Execute-write with PID 111.

TCITW007: Prepare-write with PID 111. Execute-write with PID 111. Execute-write with PID 111.

TCITW008: If the accessory receives an Execute Write Request after the TTL has expired it must respond with HAP status error code -70410 (HAPIPStatusErrorCodeInvalidWrite).

TCITW009: If the accessory receives a standard write request on a characteristic which requires timed write, the accessory must respond with HAP status error code -70410 (HAPIPStatusErrorCodeInvalidWrite).

TCITW010: The accessory receives consecutive Prepare Write Requests in the same session.

TCITW011: From Controller 1, Prepare-write with PID 111. From Controller 2, Execute-write with PID 111.

TCITW012: Prepare-write on 8 controllers. Execute-write on each controller.

TCITW013: Prepare-write with PID 111. Disconnect HTTP connection. Pair verify and Execute-write with PID 111.

TCITW001 **An accessory must support timed writes to all characteristics even if the characteristic does not require it.**

Applies to HAP over Ethernet or Wi-Fi accessories that support protocol version 1.1 or later. Protocol version 1.1 is required by HomeKit Accessory Specification R10 and later. Perform this test case automatically with HCA. Below are steps that can be used to perform this test case manually in HAT.

1. Pair and discover accessory.
2. In Controllers window, select a characteristic that supports paired write but does not require timed write.
3. In the Prepare Write Panel, Set PID to 111. Set TTL to 60000. Select Prepare Write.
4. In the Execute Write Panel, Set PID to 111. Enter a value and select Execute write.
5. Verify accessory completes the write without error.

TCITW002 **Prepare-write with PID 111. Prepare-write with PID 222. Execute-write with PID 222.**

Applies to HAP over Ethernet or Wi-Fi accessories that support protocol version 1.1 or later. Protocol version 1.1 is required by HomeKit Accessory Specification R10 and later. Perform this test case automatically with HCA. Below are steps that can be used to perform this test case manually in HAT.

1. Pair and discover accessory.
2. In Controllers window, select a characteristic that supports paired write.
3. In the Prepare Write Panel, Set PID to 111. Set TTL to 60000 (ms). Select Prepare Write.
4. In the Prepare Write Panel, Set PID to 222. Set TTL to 60000 (ms). Select Prepare Write.
5. In the Execute Write Panel, Set PID to 222. Enter a value and select Execute write. (i.e turn on light)
6. Verify accessory completes the write without error.

TCITW003 **Prepare-write with PID 111. Prepare-write with PID 222. Execute-write with PID 111.**

Applies to HAP over Ethernet or Wi-Fi accessories that support protocol version 1.1 or later. Protocol version 1.1 is required by HomeKit Accessory Specification R10 and later. Perform this test case automatically with HCA. Below are steps that can be used to perform this test case manually in HAT.

1. Pair and discover accessory.
2. In Controllers window, select a characteristic that supports paired write.
3. In the Prepare Write Panel, Set PID to 111. Set TTL to 120000 (ms). Select Prepare Write.
4. In the Prepare Write Panel, Set PID to 222. Set TTL to 120000 (ms). Select Prepare Write.
5. In the Execute Write Panel, Set PID to 111. Enter a value and select Execute write. (i.e turn on light)
6. See the HTTP traffic view.
7. Verify that the accessory responds with the HAP status code "-70410."

TCITW004 **Prepare-write with PID 111. Wait for TTL to expire. Prepare-write with PID 111. Execute-write with PID 111.**

Applies to HAP over Ethernet or Wi-Fi accessories that support protocol version 1.1 or later. Protocol version 1.1 is required by HomeKit Accessory Specification R10 and later. Perform this test case automatically with HCA. Below are steps that can be used to perform this test case manually in HAT.

1. Pair and discover accessory.
2. In Controllers window, select a characteristic that supports paired write.
3. In the Prepare Write Panel, Set PID to 111. Set TTL to 20 (ms). Select Prepare Write.
4. In the Prepare Write Panel, Set PID to 111. Set TTL to 180000 (ms). Select Prepare Write.
5. In the Execute Write Panel, Set PID to 111. Enter a value and select Execute write. (i.e turn on light)
6. Verify accessory completes the write without error.

TCITW005 **Execute-write with PID 111 without Prepare-write.**

Applies to HAP over Ethernet or Wi-Fi accessories that support protocol version 1.1 or later. Protocol version 1.1 is required by HomeKit Accessory Specification R10 and later. Perform this test case automatically with HCA. Below are steps that can be used to perform this test case manually in HAT.

1. Pair and discover accessory.
2. In Controllers window, select a characteristic that supports paired write.
3. In the Execute Write Panel, Set PID to 111. Enter a value and select Execute write. (i.e turn on light)
4. See the HTTP traffic view.
5. Verify that the accessory responds with the HAP status code "-70410."

TCITW006 **Prepare-write with PID 222. Execute-write with PID 111.**

Applies to HAP over Ethernet or Wi-Fi accessories that support protocol version 1.1 or later. Protocol version 1.1 is required by HomeKit Accessory Specification R10 and later. Perform this test case automatically with HCA. Below are steps that can be used to perform this test case manually in HAT.

1. Pair and discover accessory.
2. In Controllers window, select a characteristic that supports paired write.
3. In the Prepare Write Panel, Set PID to 222. Set TTL to 60000 (ms). Select Prepare Write.
4. In the Execute Write Panel, Set PID to 111. Enter a value and select Execute write. (i.e turn on light)
5. See the HTTP traffic view.
6. Verify that the accessory responds with the HAP status code "-70410."

TCITW007 **Prepare-write with PID 111. Execute-write with PID 111. Execute-write with PID 111.**

Applies to HAP over Ethernet or Wi-Fi accessories that support protocol version 1.1 or later. Protocol version 1.1 is required by HomeKit Accessory Specification R10 and later. Perform this test case automatically with HCA. Below are steps that can be used to perform this test case manually in HAT.

1. Pair and discover accessory.
2. In Controllers window, select a characteristic that supports paired write.
3. In the Prepare Write Panel, Set PID to 111. Set TTL to 60000 (ms). Select Prepare Write.
4. In the Execute Write Panel, Set PID to 111. Enter a value and select Execute write. (i.e turn on light)
5. Verify write completes without errors.
6. In the Execute Write Panel, Set PID to 111. Enter a value and select Execute write. (i.e turn off light)
7. See the HTTP traffic view.
8. Verify that the accessory responds with the HAP status code "-70410."

TCITW008 **If the accessory receives an Execute Write Request after the TTL has expired it must respond with HAP status error code -70410 (HAPIPStatusErrorCodeInvalidWrite).**

Applies to HAP over Ethernet or Wi-Fi accessories that support protocol version 1.1 or later. Protocol version 1.1 is required by HomeKit Accessory Specification R10 and later. Perform this test case automatically with HCA. Below are steps that can be used to perform this test case manually in HAT.

1. Pair and discover accessory.
2. In Controllers window, select a characteristic that support paired write.
3. In the Prepare Write Panel, Set PID to 111. Set TTL to 200. Select Prepare Write.
4. In the Execute Write Panel, Set PID to 111. Enter a value and select Execute write.
5. Verify the accessory returns HAP status error code -70410.

TCITW009 **If the accessory receives a standard write request on a characteristic which requires timed write, the accessory must respond with HAP status error code -70410 (HAPIPStatusErrorCodeInvalidWrite).**

Applies to HAP over Ethernet or Wi-Fi accessories that support protocol version 1.1 or later. Protocol version 1.1 is required by HomeKit Accessory Specification R10 and later. Perform this test case automatically with HCA. Below are steps that can be used to perform this test case manually in HAT.

1. Pair and discover accessory.
2. In Controllers window, select a characteristic that requires timed-write.
3. Write a value to this characteristic.
4. See the HTTP traffic view.
5. Verify that the accessory responds with the HAP status code "-70410."

TCITW010 **The accessory receives consecutive Prepare Write Requests in the same session.**

Applies to HAP over Ethernet or Wi-Fi accessories that support protocol version 1.1 or later. Protocol version 1.1 is required by HomeKit Accessory Specification R10 and later. Perform this test case automatically with HCA. Below are steps that can be used to perform this test case manually in HAT.

1. Pair and discover accessory.
2. In Controllers window, select a characteristic that supports paired write.
3. In the Prepare Write Panel, Set PID to 111. Set TTL to 20000 (ms). Select Prepare Write.
4. In the Prepare Write Panel, Set PID to 111. Set TTL to 90000 (ms). Select Prepare Write.
5. Wait 20 seconds.
6. In the Execute Write Panel, Set PID to 111. Enter a value and select Execute write. (i.e turn on light)

7. Verify accessory completes the writes without error.

TCITW011 **From Controller 1, Prepare-write with PID 111. From Controller 2, Execute-write with PID 111.**

Applies to HAP over Ethernet or Wi-Fi accessories that support protocol version 1.1 or later. Protocol version 1.1 is required by HomeKit Accessory Specification R10 and later. Perform this test case automatically with HCA. Below are steps that can be used to perform this test case manually in HAT.

1. Pair and discover accessory (Controller 1).
2. Select the "+" at the bottom of left sidebar.
3. Select "Create IP Controller" to make new virtual controllers.
4. In left sidebar, select the Controller 1, and select the accessory name.
5. In Controllers window, under "Add Additional Controllers" panel, select "Controller 2" as Controller and select the "Add Controller" button.
6. In left sidebar, select the Controller 2, select the accessory name, select the "Discover" button.
7. From Controller 1, In Controllers window, select a characteristic that supports paired write.
8. In the Prepare Write Panel, Set PID to 111. Set TTL to 20000 (ms). Select Prepare Write.
9. From Controller 2, In Controllers window, select the same characteristic. In the Execute Write Panel, Set PID to 111. Enter a value and select Execute write. (i.e turn on light)
10. See the HTTP traffic view.
11. Verify that the accessory responds with the HAP status code "-70410."

TCITW012 **Prepare-write on 8 controllers. Execute-write on each controller.**

Applies to HAP over Ethernet or Wi-Fi accessories that support protocol version 1.1 or later. Protocol version 1.1 is required by HomeKit Accessory Specification R10 and later. Perform this test case automatically with HCA. Below are steps that can be used to perform this test case manually in HAT.

1. Pair and discover accessory (Controller 1).
2. Select the "+" at the bottom of left sidebar.
3. Select "Create IP Controller" to make 7 additional controllers.
4. In left sidebar, select the Controller 1, and select the accessory name.
5. In Controllers window, under the "Add Additional Controllers" panel, select "Controller 2" as Controller, check the box 'on' for Admin and select the "Add Controller" button.
6. Continue step #5 to add 8 controllers in total.
7. In left sidebar, select the Controller 2, select the accessory name, select the "Discover" button.
8. Continue step #7 to discover controllers 3 through 8.

9. From Controller 1, In Controllers window, select a characteristic that supports paired write.
10. In the Prepare Write Panel, Set PID to 111. Set TTL to 600000 (ms). Select Prepare Write.
11. From Controller 2, In Controllers window, select a characteristic that supports paired write.
12. In the Prepare Write Panel, Set PID to 111. Set TTL to 600000 (ms). Select Prepare Write.
13. From Controller 3, In Controllers window, select a characteristic that supports paired write.
14. In the Prepare Write Panel, Set PID to 111. Set TTL to 600000 (ms). Select Prepare Write.
15. From Controller 4, In Controllers window, select a characteristic that supports paired write.
16. In the Prepare Write Panel, Set PID to 111. Set TTL to 600000 (ms). Select Prepare Write.
17. From Controller 5, In Controllers window, select a characteristic that supports paired write.
18. In the Prepare Write Panel, Set PID to 111. Set TTL to 600000 (ms). Select Prepare Write.
19. From Controller 6, In Controllers window, select a characteristic that supports paired write.
20. In the Prepare Write Panel, Set PID to 111. Set TTL to 600000 (ms). Select Prepare Write.
21. From Controller 7, In Controllers window, select a characteristic that supports paired write.
22. In the Prepare Write Panel, Set PID to 111. Set TTL to 600000 (ms). Select Prepare Write.
23. From Controller 8, In Controllers window, select a characteristic that supports paired write.
24. In the Prepare Write Panel, Set PID to 111. Set TTL to 600000 (ms). Select Prepare Write.
25. From Controller 1, In Controllers window, select the same characteristic. In the Execute Write Panel, Set PID to 111. Enter a value and select Execute write. (i.e turn on light)
26. Verify accessory completes the writes without error.
27. From Controller 2, In Controllers window, select the same characteristic. In the Execute Write Panel, Set PID to 111. Enter a value and select Execute write. (i.e turn off light)
28. Verify accessory completes the writes without error.
29. From Controller 3, In Controllers window, select the same characteristic. In the Execute Write Panel, Set PID to 111. Enter a value and select Execute write. (i.e turn on light)
30. Verify accessory completes the writes without error.
31. From Controller 4, In Controllers window, select the same characteristic. In the Execute Write Panel, Set PID to 111. Enter a value and select Execute write. (i.e turn off light)
32. Verify accessory completes the writes without error.
33. From Controller 5, In Controllers window, select the same characteristic. In the Execute Write Panel, Set PID to 111. Enter a value and select Execute write. (i.e turn on light)
34. Verify accessory completes the writes without error.
35. From Controller 6, In Controllers window, select the same characteristic. In the Execute Write Panel, Set PID to 111. Enter a value and select Execute write. (i.e turn off light)

36. Verify accessory completes the writes without error.
37. From Controller 7, In Controllers window, select the same characteristic. In the Execute Write Panel, Set PID to 111. Enter a value and select Execute write. (i.e turn on light)
38. Verify accessory completes the writes without error.
39. From Controller 8, In Controllers window, select the same characteristic. In the Execute Write Panel, Set PID to 111. Enter a value and select Execute write. (i.e turn off light)
40. Verify accessory completes the writes without error.

TCITW013 **Prepare-write with PID 111. Disconnect HTTP connection. Pair verify and Execute-write with PID 111.**

Applies to HAP over Ethernet or Wi-Fi accessories that support protocol version 1.1 or later. Protocol version 1.1 is required by HomeKit Accessory Specification R10 and later. Perform this test case automatically with HCA. Below are steps that can be used to perform this test case manually in HAT.

1. Pair and discover accessory.
2. In Controllers window, select a characteristic that support paired write.
3. In the Prepare Write Panel, Set PID to 111. Set TTL to 60000. Select Prepare Write.
4. From the Controllers window select the Disconnect button.
5. From the Controllers window select the Discover button to pair-verify.
6. In the Execute Write Panel, Set PID to 111. Enter a value and select Execute write.
7. See the HTTP traffic view.
8. Verify that the accessory responds with the HAP status code "-70410."

1.14 Wi-Fi Accessory Configuration 2

TCW001: Verify that the accessory can join a dual-band 2.4 GHz and 5 GHz WPA2 Access Point via WAC2.

TCW002: Verify that the accessory can join a 2.4 GHz WPA2 Access Point via WAC2.

TCW003: If supported, verify that the accessory can join a 5 GHz WPA2 Access Point via WAC2.

TCW004: Verify that the accessory can join a dual-band 2.4 GHz and 5 GHz unsecured Access Point via WAC2.

TCW005: Verify that the accessory can join a 2.4 GHz unsecured Access Point via WAC2.

TCW006: If supported, verify that the accessory can join a 5 GHz unsecured Access Point via WAC2.

TCW007: Verify that the accessory successfully joins an Access Point via WAC2 using Home app with incorrect setup code followed by correct setup code.

TCW008: Accessories that support WAC2 must not support legacy WAC.

TCW015: Verify accessory no longer advertises via WAC after the Wi-Fi Accessory Configuration Mode Automatic Shutoff timer expires.

TCW016: If the accessory can be put into WAC mode without removing the HomeKit pairing, then verify the WAC advertisement has bit 9 set during the Wi-Fi reconfiguration procedure.

TCW001 **Verify that the accessory can join a dual-band 2.4 GHz and 5 GHz WPA2 Access Point via WAC2.**

Applies to accessories that use HAP over Wi-Fi.

1. Set up a dual-band 2.4 GHz and 5 GHz Access Point with WPA2 encryption.
2. Using Home app, add the accessory to the Home.
3. Verify read/write functionality.
4. Perform factory reset on accessory.
5. Delete accessory from the Home using the Home app.
6. Re-add the accessory to the Home.
7. Verify read/write functionality.
8. Delete the accessory from the Home.

TCW002 **Verify that the accessory can join a 2.4 GHz WPA2 Access Point via WAC2.**

Applies to accessories that use HAP over Wi-Fi.

1. Set up an Access Point to 2.4 GHz only with WPA2 encryption.
2. Using Home app, add the accessory to the Home.
3. Verify read/write functionality.

4. Perform factory reset on accessory.
5. Delete accessory from the Home using the Home app.

TCW003 **If supported, verify that the accessory can join a 5 GHz WPA2 Access Point via WAC2.**

Applies to accessories that use HAP over Wi-Fi.

1. Set up an Access Point to 5 GHz only with WPA2 encryption.
2. Using Home app, add the accessory to the Home.
3. Verify read/write functionality.
4. Perform factory reset on accessory.
5. Delete accessory from the Home using the Home app.

TCW004 **Verify that the accessory can join a dual-band 2.4 GHz and 5 GHz unsecured Access Point via WAC2.**

Applies to accessories that use HAP over Wi-Fi.

1. Set up a dual-band 2.4 GHz and 5 GHz unsecured Access Point.
2. Using Home app, add the accessory to the Home.
3. Verify read/write functionality.
4. Perform factory reset on accessory.
5. Delete accessory from the Home using the Home app.

TCW005 **Verify that the accessory can join a 2.4 GHz unsecured Access Point via WAC2.**

Applies to accessories that use HAP over Wi-Fi.

1. Set up a 2.4 GHz unsecured Access Point.
2. Using Home app, add the accessory to the Home.
3. Verify read/write functionality.
4. Perform factory reset on accessory.
5. Delete accessory from the Home using the Home app.

TCW006 **If supported, verify that the accessory can join a 5 GHz unsecured Access Point via WAC2.**

Applies to accessories that use HAP over Wi-Fi.

1. Set up a 5 GHz unsecured Access Point.

2. Using Home app, add the accessory to the Home.
3. Verify read/write functionality.
4. Perform factory reset on accessory.
5. Delete accessory from the Home using the Home app.

TCW007 **Verify that the accessory successfully joins an Access Point via WAC2 using Home app with incorrect setup code followed by correct setup code.**

Applies to accessories that use software token authentication.

1. Set up an Access Point with the frequency band supported by accessory.
2. In Home app tap "Add Accessory".
3. Tap "Don't Have a Code or Can't Scan?".
4. Using Home app, select the accessory to be added.
5. Enter an incorrect setup code when prompted.
6. Enter correct setup code when prompted.
7. Verify read/write functionality.
8. Perform factory reset on accessory.
9. Delete the accessory from the Home.

TCW008 **Accessories that support WAC2 must not support legacy WAC.**

Applies to accessories that use HAP over Wi-Fi.

1. Perform a factory reset on the accessory and put into WAC mode.
2. Launch HAT.
3. On the left hand side of the Controllers Window, under IP Controllers, select Controller, in the Discovery panel, select Start button for Status.
4. In the WAC Discovery traffic view, select the accessory and select the "Details" button.
5. Expand the Flags view.
6. Verify WAC2 flag (bit 20) is set and Supports MFi Configuration V1 (bit 2) is not set.

TCW015 **Verify accessory no longer advertises via WAC after the Wi-Fi Accessory Configuration Mode Automatic Shutoff timer expires.**

Applies to accessories that use HAP over Wi-Fi. Applies to accessories that support HomeKit Accessory Protocol specification R15 or earlier. Perform this test case with HAT using the steps below.

1. Factory reset the accessory.
2. In the left sidebar of the Controllers window, select the controller, and then select “Start” to begin discovering advertisements.
3. Using the WAC Discovery view in the trace window, verify accessory begins to advertise.
4. Do not interact with the accessory for at least 15 minutes.
5. In the left sidebar of the Controllers window, select the controller, and then select “Stop” to stop discovering advertisements.
6. Select “Start” to begin discovering advertisements again.
7. Using the WAC Discovery view in the trace window, verify accessory is no longer advertising.
8. Power cycle the accessory.
9. Using the WAC Discovery view in the trace window, verify accessory begins to advertise again.

TCW016 **If the accessory can be put into WAC mode without removing the HomeKit pairing, then verify the WAC advertisement has bit 9 set during the Wi-Fi reconfiguration procedure.**

Applies to accessories that use HAP over Wi-Fi.

1. Verify Mac is on network you want the accessory to join.
2. Factory reset the accessory.
3. Launch HAT and select the accessory on the left side of the controllers window.
4. In the Wi-Fi Accessory Configuration panel, select the “Join access point” button.
5. In the Pairing Panel, select the “Start Pairing” and enter setup code when prompted.
6. In the Wi-Fi Accessory Configuration panel, enter the Wi-fi SSID and Wi-Fi PSK and select the “Send WAC Configuration” button.
7. Ensure your Mac is on the network you expect the accessory to join, otherwise re-join the expected network.
8. Once the accessory begins advertising via Bonjour on the newly joined network, select the “Confirm WAC Configuration” button.
9. Verify the accessory responds to /Configured with HTTP Response: HTTP/1.1 200 OK
10. Put accessory into WAC mode without removing the HomeKit pairing.
11. In the WAC Discovery traffic view, select the accessory’s most recent WAC advertisement, select the “Details” button, and expand the Flags view.
12. Verify “Unconfigured” flag (Bit 1) is listed.
13. Verify “Device is paired to a HomeKit Controller flag (Bit 9)” is listed.

1.15 Product Plan

TCPP001: Verify Accessory Network Declarations. HAP over Wi-Fi or Ethernet accessories must have their network services information declared via the MFi Portal in order to establish WAN and LAN firewall rules in a HomeKit-enabled Wi-Fi router.

TCPP002: Accessory must expose the Product Data characteristic on the Accessory Information service, and the characteristic's value must match the Product Plan's assigned 8 byte Product Data value from the MFi Portal. The Product Data must also be present in the accessory's Pair-Setup M4 encrypted sub-tlv. Bridged accessories must not include the Product Data characteristics as part of their Accessory Information service(s).

TCPP003: Manufacturer, Model, and Firmware Revision on the Accessory Information Service of the accessory must match the Product Plan's assigned Manufacturer, Model, and Firmware Revision from the MFi Portal.

TCPP001 **Verify Accessory Network Declarations. HAP over Wi-Fi or Ethernet accessories must have their network services information declared via the MFi Portal in order to establish WAN and LAN firewall rules in a HomeKit-enabled Wi-Fi router.**

Applies to accessories that use HAP over Ethernet or Wi-Fi. Perform this test case with HAT using the steps below.

1. Please contact your MFi representative for more information regarding the Network Declarations submission and testing process.

TCPP002 **Accessory must expose the Product Data characteristic on the Accessory Information service, and the characteristic's value must match the Product Plan's assigned 8 byte Product Data value from the MFi Portal. The Product Data must also be present in the accessory's Pair-Setup M4 encrypted sub-tlv. Bridged accessories must not include the Product Data characteristics as part of their Accessory Information service(s).**

Applies to all accessories. Perform this test case automatically with HCA. Below are steps that can be used to perform this test case manually in HAT.

1. Pair and Discover accessory.
2. In the Events traffic view, select the "Pair-Setup - M4" event, and verify the Encrypted Data contains the Product Data TLV type (0x1c).
3. Verify the TLV value is exactly 8 bytes. Notate the value.
4. Navigate to the Accessory Information service and verify it contains the "Product Data" Characteristic.
5. Read the "Product Data" Characteristic.
6. Verify the read value matches the value found in step 3.
7. Verify that the Product Data value matches the value assigned to this accessory's Product Plan on the MFi Portal.
8. Navigate to each bridged accessory and verify the "Accessory Information" service(s) do not contain the "Product Data" Characteristic.

TCPP003 **Manufacturer, Model, and Firmware Revision on the Accessory Information Service of the accessory must match the Product Plan's assigned Manufacturer, Model, and Firmware Revision from the MFi Portal.**

Applies to all accessories. Perform this test case automatically with HCA. Below are steps that can be used to perform this test case manually in HAT.

1. Pair and Discover accessory.
2. Navigate to the Accessory Information service and verify it contains the "Manufacturer" Characteristic.
3. Read the "Manufacturer" Characteristic.
4. Verify that the Manufacturer value matches the value assigned to this accessory's Product Plan on the MFi Portal.
5. Navigate to the Accessory Information service and verify it contains the "Model" Characteristic.
6. Read the "Model" Characteristic.
7. Verify that the Model value matches the value assigned to this accessory's Product Plan on the MFi Portal.
8. Navigate to the Accessory Information service and verify it contains the "Firmware Revision" Characteristic.
9. Read the "Firmware Revision" Characteristic.
10. Verify that the Firmware Revision value matches the value assigned to this accessory's Product Plan on the MFi Portal.

1.16 Bluetooth

TCB001: Accessory must expose a single instance of the Pairing Service with the following required characteristics: Pair Setup, Pair Verify, Features, Pairings.

TCB002: Accessory must close the Bluetooth connection after 30 seconds of inactivity (i.e without any HAP Transactions).

TCB007: If an accessory receives a HAP PDU with an opcode that it does not support, it shall reject the PDU and respond with a status code "Unsupported PDU (0x01)" in its HAP response.

TCB008: Accessory must support a timed write on any characteristic that supports paired write.

TCB009: Accessory must reject HAP-Characteristic-Execute-Write request after TTL and discard the queued HAP-Timed-Write request.

TCB010: Accessory must reject HAP-Characteristic-Write-Request on characteristics that require timed writes.

TCB011: Accessories must support restoring sessions for at least 8 sessions.

TCB012: If authTag verification of encrypted data fails during Pair-Verify, the accessory must respond with "M4" and "kTLVError_Authentication 0x02".

TCB015: 48-bit Device ID must persist across a reboot and is randomly generated when accessory is factory reset.

TCB016: Global state number (GSN) must increment correctly based on connection state and characteristic configuration.

TCB024: Changes on characteristics that do not support disconnected events must not change the Global State Number (GSN).

TCB026: Accessories must support multiple iterations of Pair Verify on a single Bluetooth LE connection.

TCB027: The authTag must be appended to the encrypted value and must be sent as part of the same GATT message.

TCB030: Accessory must handle fragmented PDUs.

TCB035: Accessory must indicate that Security Class characteristics require HAP-Characteristic-Timed-Write using the HAP Characteristic Properties Descriptor.

TCB036: Accessory must reject any HAP operation to a GATT characteristic with instance ID that does not match the instance ID in the HAP-BLE PDU.

TCB037: Accessory must tear down the security session when the Bluetooth LE link disconnects.

TCB041: The minimum number of pairing relationships that an accessory must support is 16.

TCB043: Bluetooth LE accessories must pass Bluetooth Qualification and Declaration from the Bluetooth SIG (https://www.bluetooth.org/en-us/test-qualification/qualification-overview).

TCB044: Characteristic properties must include only the allowed permissions.

TCB049: Accessory must reject GATT Read Requests on a HAP characteristic if it was not preceded by an GATT Write Request with the same transaction ID at most 10 seconds prior.

TCB050: Accessory must reject characteristic writes where the written value is above the maximum value or below the minimum value defined by the characteristic metadata.

TCB052: Characteristics using fixed-length formats that support Disconnected Events must also support Connected Events and Broadcasted Events. For Broadcasted Events, when the value associated with a characteristic that is configured for broadcast notification changes while in a disconnected state the Global State Number (GSN) must be incremented for every such

change and reflected in the Encrypted advertisement payload. Default value Advertising ID should be equal to device ID. Broadcast key shouldn't be set after initial pairing. After last admin paring is removed, the Advertisement ID should be reset to Device ID. Characteristics with TLV or String formats must not support Broadcast Events.

TCB053: If authTag verification of encrypted data fails during pair-resume, the accessory must respond with "M2" and "kTLVError_Authentication 0x02".

TCB054: Accessories must implement a 10 second HAP procedure timeout. All HAP procedures including Pair-Verify and Pair-Resume must complete within 10 seconds.

TCB055: After a Bluetooth link is established, the first HAP procedure must begin within 10 seconds. Accessories must drop the Bluetooth Link if the controller fails to start a HAP procedure within 10 seconds of establishing the Bluetooth link.

TCB056: If the session ID from a pair-resume <M1> request is not found or has expired, the accessory must treat this as <M1> of pair-verify.

TCB057: The accessory must support connected events.

TCB058: Accessory must always successfully deliver event notifications for every characteristic that supports them when a single client has subscribed multiple times.

TCB059: Accessory shall not advertise while it is connected to a HomeKit controller.

TCB060: If the accessory receives another HAP procedure to the same characteristics in the middle of the timed write procedure, it shall drop the security session and disconnect the Bluetooth link.

TCB001 **Accessory must expose a single instance of the Pairing Service with the following required characteristics: Pair Setup, Pair Verify, Features, Pairings.**

Applies to accessories that use HAP over BLE. Perform this test case with HAT using the steps below.

1. Pair and discover accessory.
2. In left sidebar of Controllers window, see accessory's Pairing Service.
3. Verify that the required characteristics are included in the Pairing Service.

TCB002 **Accessory must close the Bluetooth connection after 30 seconds of inactivity (i.e without any HAP Transactions).**

Applies to accessories that use HAP over BLE. Perform this test case with HAT using the steps below.

1. Pair and discover accessory.
2. Do not read/write to the accessory.
3. Use the BLE HAP Transactions traffic view to check the delta between the last HAP transaction and the "Disconnected" message.
4. Verify accessory closes the Bluetooth connection 30 seconds after the last HAP transaction completed.
5. Select "Discover" in Controllers window to reconnect and establish a security session.

6. Read or Write to the accessory.
7. Use the BLE HAP Transactions traffic view to check the delta between the last HAP transaction and the "Disconnected" message.
8. Verify accessory closes the Bluetooth connection 30 seconds after the last HAP transaction completed.

TCB007 **If an accessory receives a HAP PDU with an opcode that it does not support, it shall reject the PDU and respond with a status code "Unsupported PDU (0x01)" in its HAP response.**

Applies to accessories that use HAP over BLE. Perform this test case automatically with HCA. Below are steps that can be used to perform this test case manually in HAT.

1. Pair and discover accessory.
2. Select any characteristic.
3. In the Controllers window, under Miscellaneous, select "Send Unsupported Opcode".
4. In the BLE HAP Transactions traffic view, verify accessory responds with "HAP status code (0x01)."

TCB008 **Accessory must support a timed write on any characteristic that supports paired write.**

Applies to accessories that use HAP over BLE. Perform this test case automatically with HCA. Below are steps that can be used to perform this test case manually in HAT.

1. Pair and discover accessory.
2. Use Timed Write feature to write to characteristics that do not require it. (e.g., timed write to turn on/off light).
3. Verify accessory completes the write without error within 10 seconds.

TCB009 **Accessory must reject HAP-Characteristic-Execute-Write request after TTL and discard the queued HAP-Timed-Write request.**

Applies to accessories that use HAP over BLE. Perform this test case automatically with HCA. Below are steps that can be used to perform this test case manually in HAT.

1. Pair and discover accessory.
2. Select any characteristic that supports paired write.
3. Select the checkbox on for "Timed Write After Prepared Write Timer Expired."
4. Write value in the Timed Write field.
5. Select Timed Write.
6. Verify in Events Traffic View, Accessory response to Encrypted Characteristic Timed Write Completed Error: Failed to execute write to characteristic.

TCB010 **Accessory must reject HAP-Characteristic-Write-Request on characteristics that require timed writes.**

Applies to accessories that use HAP over BLE. Perform this test case automatically with HCA. Below are steps that can be used to perform this test case manually in HAT.

1. Pair and discover accessory.
2. Write to a characteristic that supports timed writes.
3. Verify status code "Invalid Request" (0x06) in its HAP response.

TCB011 **Accessories must support restoring sessions for at least 8 sessions.**

Applies to accessories that use HAP over BLE. Perform this test case automatically with HCA. Below are steps that can be used to perform this test case manually in HAT.

1. Pair and discover accessory with Controller 1.
2. In Controllers window, select "+" to create a new Bluetooth LE Controller 2.
3. Repeat step 3 until 8 Controllers are created.
4. Under Controller 1, select the accessory name, under "Add Additional Controllers" panel, select "Controller 2" as Controller and select the "Add Controller" button.
5. Repeat step 4 until the 8th controller has been added.
6. Click "Disconnect" button in the Controllers window.
7. On the left pane of the Controllers window, select Controller 2, select the "Pair Verify" button under the Connection panel.
8. After Pair-Verify (M1-M4) completes successfully, click the "Disconnect" button in the Controllers window.
9. Repeat step 7 and 8 until the 8th Controller session has been established.
10. Using controller 1, select the "Pair Resume" button from the connection panel to initiate a pair-resume.
11. Using the HAP Procedures traffic view, verify accessory successfully completes pair-resume and responds with kTLVType_State <M2>, kTLVType_Method <kTLVMethod_Resume>, kTLVType_SessionID <New Session ID>, kTLVType_EncryptedData <16 bytes of auth tag>.
12. Verify the response from the accessory contains a new SessionID that differs from the SessionID used in the pair-resume request from the controller.
13. Click "Disconnect" button in the Controllers window.
14. Repeat steps 11-14 for each additional controller.

TCB012 **If authTag verification of encrypted data fails during Pair-Verify, the accessory must respond with "M4" and "kTLVError_Authentication 0x02".**

Applies to accessories that use HAP over BLE. Perform this test case automatically with HCA. Below are steps that can be used to perform this test case manually in HAT.

1. Pair and discover accessory.
2. Click "Disconnect" in the Controllers window or wait for the accessory to disconnect.
3. In the "Options" section of the accessory server view, enable the "Use Bad Auth Tag during Pair-Setup, Pair-Verify, and Pair-Resume" checkbox.
4. Click "Pair-Verify" in the Connection section of the Controllers window.
5. See the BLE HAP Transactions Traffic view.
6. Verify that the accessory response in the Pair-Verify value is "M4" and "`kTLVError_Authentication` 0x02" (TLV error response 070102 in HAT).
7. In the "Options" section of the accessory server view, disable the "Use Bad Auth Tag during Pair-Setup, Pair-Verify and Pair-Resume" checkbox.

TCB015 **48-bit Device ID must persist across a reboot and is randomly generated when accessory is factory reset.**

Applies to accessories that use HAP over BLE. Perform this test case with HAT using the steps below.

1. Pair and discover accessory.
2. In Controllers window, under Advertisement Information, note the device ID.
3. Power cycle accessory.
4. Verify the device id did not change.
5. Perform factory reset on accessory.
6. Pair and discover accessory.
7. Verify device identifier is randomly generated and differs from device id from step 2.

TCB016 **Global state number (GSN) must increment correctly based on connection state and characteristic configuration.**

Applies to accessories that use HAP over BLE. Perform this test case automatically with HCA. Below are steps that can be used to perform this test case manually in HAT.

- While a controller is connected, the GSN must increment only once for multiple value changes.
- While no controller is connected, the GSN must increment only once for multiple value changes.

1. Pair and discover accessory.

2. Select the “Disconnect” button to close the connection.
3. Using the Advertisement Information panel on the main accessory view, notate the Global State Number (GSN).
4. For each characteristic that supports Disconnect Events and has physical means of interaction, toggle each characteristic multiple times.
5. After toggling characteristics, check the GSN from the Advertisement Information panel to ensure it has incremented the value found in step 2 by only 1.
6. Select the “Discover” button in the summary panel.
7. Enable “Pair-Resume keep alive” with a 27 second interval.
8. For each characteristic that supports Disconnected Events and has write permissions, write a new value multiple times.
9. Select the “Disconnect” button to close the connection.
10. After the accessory begins to advertise again, check the GSN from the Advertisement Information panel to ensure it has increment the value found in step 5 by only 1.
11. Power cycle the accessory.
12. After the accessory begins to advertise again, check the GSN from the Advertisement Information panel to ensure it has not incremented the value found during step 10.

TCB024 **Changes on characteristics that do not support disconnected events must not change the Global State Number (GSN).**

Applies to accessories that use HAP over BLE. Perform this test case with HAT using the steps below.

1. Pair and discover accessory.
2. Wait 30 seconds for Bluetooth LE to disconnect.
3. Change the state of a characteristic that does NOT support HAP events multiple times. E.g. by means of accessory’s app or physical interaction with the accessory, if applicable.
4. Verify GSN does not change.
5. Discover the accessory again, and navigate to a characteristic that does not support Event Notifications and supports paired write.
6. Perform a valid write operation to change the characteristic value.
7. Select the “Disconnect” button and wait for the accessory to begin advertising again.
8. Verify in the most recent BLE advertisement that the GSN did not change.
9. Repeat steps 5-8 for each characteristic that does not support Event Notifications and supports paired write.

TCB026 **Accessories must support multiple iterations of Pair Verify on a single Bluetooth LE connection.**

Applies to accessories that use HAP over BLE. Perform this test case automatically with HCA. Below are steps that can be used to perform this test case manually in HAT.

1. Pair and discover accessory.
2. Verify that Pair Verify payloads are seen in the BLE HAP Transactions traffic view.
3. Read or write any characteristic.
4. Under the Connection panel select “Pair Verify” button.
5. In the Events traffic view, Verify the message “Pair-Verify Completed” was received.
6. Under the Connection panel select “Pair Verify” button again.
7. In the Events traffic view, Verify the message “Pair-Verify Completed” was received.

TCB027 **The authTag must be appended to the encrypted value and must be sent as part of the same GATT message.**

Applies to accessories that use HAP over BLE. Perform this test case automatically with HCA. Below are steps that can be used to perform this test case manually in HAT.

1. Pair and discover accessory.
2. In the Controllers window under the Pairing section, check the box on for “Read/Write Without Auth Tag.”
3. Select any characteristic.
4. Read or write to the characteristic.
5. See HAP Transactions traffic view.
6. Verify disconnected message is received.
7. In the Controllers window under the Pairing section, check the box off for “Read/Write Without Auth Tag.”

TCB030 **Accessory must handle fragmented PDUs.**

Applies to accessories that use HAP over BLE. Perform this test case with HAT using the steps below.

1. Pair and discover accessory.
2. In the Controllers window, under Connection, set “Maximum HAP packet size 512” to 10.
3. Read and write to any characteristic.
4. Verify the accessory completes read or write procedure successfully.

TCB035 **Accessory must indicate that Security Class characteristics require HAP-Characteristic-Timed-Write using the HAP Characteristic Properties Descriptor.**

Applies to accessories that use HAP over BLE. Perform this test case with HAT using the steps below.

1. Pair and discover accessory.
2. See Discovered Accessories in the Events traffic view.
3. Select "Details" and view Attribute Database.
4. Verify Security Class characteristics use the Timed-Write permission.
5. Write a value to the characteristics that support Timed-Write. i.e lock or unlock
6. Verify physical action completed. i.e deadbolt locked or unlocked.

TCB036 **Accessory must reject any HAP operation to a GATT characteristic with instance ID that does not match the instance ID in the HAP-BLE PDU.**

Applies to accessories that use HAP over BLE. Perform this test case using HCA.

If the accessory receives a HAP-Service-Signature-Read-Request with service instance ID "0", it must respond with a valid HAP-Service-Signature-Read-Response with HAP Service Properties set to "0x0000" and no linked services.

TCB037 **Accessory must tear down the security session when the Bluetooth LE link disconnects.**

Applies to accessories that use HAP over BLE. Perform this test case with HAT using the steps below.

1. Pair and discover accessory.
2. In Controllers window, select accessory's name.
3. Select the Remove Pairing button.
4. See the Events traffic view.
5. Verify that the Disconnected message is seen.

TCB041 **The minimum number of pairing relationships that an accessory must support is 16.**

Applies to accessories that use HAP over BLE. Perform this test case automatically with HCA. Below are steps that can be used to perform this test case manually in HAT.

1. Pair and discover accessory.
2. Select the "+" at the bottom of left sidebar.
3. Select "Create BLE Controller" to create 15 additonal controllers, for a total of 16 controllers.
4. Using admin Controller 1, add pairings to each of the 15 secondary controllers.
5. Select "List Pairings" button in the Controllers window.

6. In the Event view of the Trace window, verify the response to “List Pairings Completed” event shows all 16 pairings.
7. On the left pane of the Controllers window, under Controller 2, select the accessory name, select the “Start” button, and select the “Discover” button.
8. In the Event view of the Trace window, verify Pair-Verify completes successfully.
9. Repeat steps 7-8 for each additional controller.

TCB043 **Bluetooth LE accessories must pass Bluetooth Qualification and Declaration from the Bluetooth SIG (https://www.bluetooth.com/test-qualification/qualification-overview).**

Applies to accessories that use HAP over BLE.

1. Enter the accessory’s Declaration ID or Quality Design ID on the Accessory Compliance Questionnaire.
2. The Accessory Compliance Questionnaire is submitted through the MFi Portal after Product Plan approval.

TCB044 **Characteristic properties must include only the allowed permissions.**

Applies to accessories that use HAP over BLE. Perform this test case with HAT using the steps below.

Required permissions for R15 or earlier specifications:

- Paired Read
- Paired Write
- Additional Authorization
- Timed Write
- Event Notifications (Disconnected, Connected, Broadcasted)

Required characteristics for R16 or later specifications:

- Paired Read
- Paired Write
- Timed Write
- Event Notifications (Disconnected, Connected, Broadcasted)

Characteristics that support disconnected notifications must also support connected notifications and broadcast notifications.

1. Pair and discover accessory
2. In the Events traffic view, select the Discovered Accessories packet.
3. Select the Details button.

4. See Accessory Info; use disclosure arrows to show and hide details.
5. Verify that the supplied properties are only a combination of: Paired read, Paired write, indicate, indicate (disconnected), timed write, Additional Authorization.

TCB049 **Accessory must reject GATT Read Requests on a HAP characteristic if it was not preceded by an GATT Write Request with the same transaction ID at most 10 seconds prior.**

Applies to accessories that use HAP over BLE. Perform this test case with HAT using the steps below.

1. Pair and discover accessory with controller 1.
2. In the Controllers window, under Miscellaneous window, check the box 'on' for "Delay GATT Read Request from Controller".
3. Read a characteristic.
4. In the GATT traffic view, verify accessory does not respond to the Read request and drops BLE connection within 10 seconds.
5. In the Controllers window, under Miscellaneous window, check the box 'off' for "Delay GATT Read Request from Controller".

TCB050 **Accessory must reject characteristic writes where the written value is above the maximum value or below the minimum value defined by the characteristic metadata.**

Applies to HAP over BLE accessories with characteristics that use minimum value and/or maximum value characteristic metadata. Perform this test case automatically with HCA. Below are steps that can be used to perform this test case manually in HAT.

1. Pair and discover accessory.
2. In left sidebar of the Controllers window, see each of accessory's services.
3. Using the Write button in Controllers window, write to each paired Read/Write characteristic with a value that is below the minValue if the value to be written is valid for the specified format (e.g., for a format of uint8 and a minValue of 0, this step can be skipped).
4. Verify that Accessory does not update the characteristic to an out of bounds value.
5. Using the Write button in Controllers window, write to each paired Read/Write characteristic with a value that is above the maxValue if the value to be written is valid for the specified format (e.g., for a format of uint8 and a maxValue of 255, this step can be skipped).
6. Verify that Accessory does not update the characteristic to an out of bounds value. In the BLE HAP Transactions traffic view, verify accessory responds with "HAP status code (0x06)."

TCB052 **Characteristics using fixed-length formats that support Disconnected Events must also support Connected Events and Broadcasted Events. For Broadcasted Events, when the value associated with a characteristic that is configured for broadcast notification changes while in a disconnected state the Global**

State Number (GSN) must be incremented for every such change and reflected in the Encrypted advertisement payload. Default value Advertising ID should be equal to device ID. Broadcast key shouldn't be set after initial pairing. After last admin paring is removed, the Advertisement ID should be reset to Device ID. Characteristics with TLV or String formats must not support Broadcast Events.

Applies to accessories that use HAP over BLE. Perform this test case automatically with HCA. Below are steps that can be used to perform this test case manually in HAT.

1. Pair and discover accessory.
2. In the HAP Protocol Information Service, select the Service Signature characteristic, and then in the Protocol Configuration panel, enable the "Get all params" checkbox, and select send.
3. In the HAP Procedures traffic view, see Protocol Configuration Response.
4. Verify advertising identifier matches Device Identifier in the Controller Window Advertisement Information Panel.
5. Change the state of the characteristic that supports broadcast notifications.
6. In the BLE Discovery traffic view, Verify no Encrypted Broadcast advertisement.
7. In the Controllers window, select Disconnect.
8. In the BLE Discovery traffic view, verify accessory is sending Regular Homekit Advertisement.
9. In the HAP Protocol Information Service, select the Service Signature characteristic, and then in the Protocol Configuration panel, enable the "Set Advertising Identifier" and "Get all params" checkboxes.
10. Under "Desired Advertising Identifier", fill in: "AABBCCDDEEFF", and then select send.
11. In the HAP Procedures traffic view, see Protocol Configuration Response.
12. Verify advertising identifier matches "AABBCCDDEEFF".
13. In the controller window, select Disconnect.
14. In the Verify Accessory is sending Regular HomeKit Advertisement.
15. Select Discover.
16. In the HAP Protocol Information Service, select the Service Signature characteristic. In the Protocol Configuration panel, enable the "Get all params" and "generate broadcast key" checkboxes, and then select the send button.
17. Find a characteristic that supports Broadcast permissions, and in the Characteristic Configuration panel, enter "1" for broadcast interval and select "Enable Broadcast Notifications".
18. In the Controllers window, in the Summary Panel, select "Disconnect".
19. Change the state of the characteristic (from step 18).
20. Verify a Broadcast Notification was received in the BLE Discovery traffic view. Verify the GSN incremented one time.
21. Change the state of the characteristic (from step 18) again.
22. Verify a Broadcast Notification was received in the BLE Discovery traffic view. Verify the GSN incremented one time.

23. From the Controllers window, Select “Remove pairing”.
24. If applicable, perform user action such as button press, reboot etc., for accessory to begin advertising as unpaired.
25. Pair and discover accessory.
26. Select the Service Signature characteristic, and in the Protocol Configuration panel, enable the “Get all params” checkbox, and then select send.
27. In the HAP Procedures traffic view, see Protocol Configuration Response.
28. Verify advertising identifier matches Device Device Identifier in the Controller Window Advertisement Information Panel.
29. Change the state of the characteristic that supports broadcast notifications.
30. In the BLE Discovery traffic view, verify no new Encrypted Broadcast advertisements are seen.
31. In the controller window, select “Disconnect”.
32. In the BLE Discovery traffic view, verify accessory is sending Regular Homekit Advertisements.
33. Navigate to each characteristic that uses a TLV/data or String format, and in the Summary pane, verify “Permissions” does not include the “Broadcast” permission.

TCB053 **If authTag verification of encrypted data fails during pair-resume, the accessory must respond with “M2” and “kTLVError_Authentication 0x02”.**

Applies to accessories that use HAP over BLE. Perform this test case automatically with HCA. Below are steps that can be used to perform this test case manually in HAT.

1. Pair and discover accessory.
2. Click “Disconnect” in the Controllers window or wait for the accessory to disconnect.
3. In the “Options” section of the accessory server view, enable the “Use Bad Auth Tag during Pair-Setup, Pair-Verify and Pair-Resume” check box.
4. In the Connection panel, select the “Pair Resume” button.
5. See BLE HAP Transactions Traffic view.
6. Verify accessory response in Pair Verify value is “M2” and “kTLVError_Authentication 0x02” (TLV error response 070102 in HAT).
7. In the “Options” section of the accessory server view, disable the “Use Bad Auth Tag during Pair-Setup, Pair-Verify and Pair-Resume” check box.

TCB054 **Accessories must implement a 10 second HAP procedure timeout. All HAP procedures including Pair-Verify and Pair-Resume must complete within 10 seconds.**

Applies to accessories that use HAP over BLE. Perform this test case with HAT using the steps below.

1. Pair and discover accessory.

2. See BLE HAP Transactions traffic view, wait until a Disconnected message is displayed.
3. In the Connection Panel, select the Pair Verify button.
4. In BLE HAP Procedures traffic view, locate Pair Verify Write Requests/Responses, and select details.
5. Verify the Pair Verify procedure, from state value 1 (M1) through state value 4 (M4), completes within 10 seconds.
6. See BLE HAP Transactions traffic view, wait until a Disconnected message is displayed.
7. In the Connection Panel, select the Pair Resume button.
8. In BLE HAP Procedures traffic view, locate Pair Verify Write Requests/Responses, and select details.
9. Verify the Pair Resume procedure, from state value 1 (M1) through state value 2 (M2), completes within 10 seconds.

TCB055 **After a Bluetooth link is established, the first HAP procedure must begin within 10 seconds. Accessories must drop the Bluetooth Link if the controller fails to start a HAP procedure within 10 seconds of establishing the Bluetooth link.**

Applies to accessories that use HAP over BLE. Perform this test case with HAT using the steps below.

1. Pair and discover accessory.
2. In the Summary panel, select Disconnect button or wait for the accessory to timeout.
3. After the Disconnect message is seen in the HAP procedure view, select the Connect button.
4. Verify the “Disconnected” message is seen 10 seconds after the last “Connected” message is seen.

TCB056 **If the session ID from a pair-resume <M1> request is not found or has expired, the accessory must treat this as <M1> of pair-verify.**

Applies to accessories that use HAP over BLE. Perform this test case with HAT using the steps below.

1. Pair and discover accessory.
2. View the key store for the controller.
3. Modify the session ID so that it is a different, 8-byte integer.
4. Select the Pair-Resume button in the Connection panel.
5. In the HAP procedures traffic view, verify the accessory successfully completes pair-verify (M1-M4).

TCB057 **The accessory must support connected events.**

Applies to accessories that use HAP over BLE. Perform this test case using HCA.

Does not apply to the Programmable Switch Event characteristic.

1. Pair and discover accessory.

2. In the connection panel, select “Pair Resume Keep Alive Enabled”, then set the interval to 27 seconds.
3. Using the controller, select the first characteristic that supports the “Notify“permission, then select the “Enable” button to enable event notifications.
4. Using a value different than the current value, write to a characteristic that contains both the paired write and notify permissions.
5. Verify that the controller does not receive notifications for the characteristic that was written to.
6. While connected to the accessory, physically change the state of a characteristic that contains the notify permission. Note: If the accessory disconnects, Event notifications must be re-enabled for each characteristic.
7. Using the GATT traffic view in the trace, verify that the accessory sends a notification for the correct service and characteristic that was changed.
8. Using the HAP Procedures traffic view, verify that the value in the read response after the notification arrives contains the correct value.
9. Using the controller, select the characteristic that supports the “Notify“permission from step 3, then select the “Disable” button to disable event notifications.
10. Repeat steps 3-9 for the next characteristic that contains the notify permissions.
11. In the connection panel, deselect “Pair Resume Keep Alive Enabled”.

TCB058 **Accessory must always successfully deliver event notifications for every characteristic that supports them when a single client has subscribed multiple times.**

Applies to accessories that use HAP over BLE. Perform this test case with HAT using the steps below.

1. Pair and discover accessory.
2. Enable Pair-Resume keep alive with an interval of 27 seconds.
3. Navigate to each characteristic that supports connected Event Notifications, and subscribe multiple times to Event Notifications by selecting “Enable” button 3 times.
4. For characteristics that provide physical means of interaction, physically toggle each applicable characteristic on the accessory.
5. Verify that the controller receives only a single notification for each state change.

TCB059 **Accessory shall not advertise while it is connected to a HomeKit controller.**

Applies to accessories that use HAP over BLE. Perform this test case with HAT using the steps below.

1. Pair and discover accessory.
2. Once Discover completes, select the “Disconnect” button.
3. Navigate to the “BLE Discovery” view in the Trace window and verify the accessory begins advertising.

4. In the Controllers window, select “Discover”.
5. Once Discover completes, enable the Pair-Resume keep alive checkbox in the Connections pane, using an interval of 27 seconds.
6. Navigate to the “BLE Discovery” view in the Trace window and verify the accessory is not advertising.
7. Wait one minute and then verify the accessory is still not advertising.
8. In the Controllers window, select the “Disconnect” button.
9. Navigate to the “BLE Discovery” view in the Trace window, and verify the accessory begins advertising again.

TCB060 **If the accessory receives another HAP procedure to the same characteristics in the middle of the timed write procedure, it shall drop the security session and disconnect the Bluetooth link.**

Applies to accessories that use HAP over BLE. Perform this test case with HAT using the steps below.

1. Pair and discover accessory.
2. Navigate to a characteristic that supports Paired Write.
3. Send a Prepare Write request with TTL set to “100”. (For a TTL of 10 seconds.)
4. Before the above TTL expires, send a non-timed write request to the same characteristic.
5. In the HAP Procedures view, verify accessory does not respond to the write request and a Disconnect message is seen immediately.
6. Select “Discover” to discover accessory.
7. Navigate to a characteristic that supports Paired Write.
8. Send a Prepare Write request with TTL set to “100”. (For a TLL of 10 seconds.)
9. Before the above TTL expires, select “Read” to send a Read request on the same characteristic.
10. In the HAP Procedures view, verify accessory does not respond to the read request and that a Disconnect message is seen immediately.

1.17 IP Cameras

TCICF004: Ensure the IP Camera stream starts successfully and that video is received after stop and re-starting streaming.

TCICF006: Streaming to 2x controllers. Both controllers streaming, stop stream on controller 2 and restart controller 2 stream, verify no issues seen from stream 1. Stop stream on controller 1 and restart controller 1 stream, verify no issues seen from stream 2.

TCICF007: Ensure a snapshot is retrieved successfully and streaming is uninterrupted.

TCICF008: 2x Controllers should receive snapshots successfully. Streams should be uninterrupted as the snapshot is processed.

TCICF010: 2x Controllers should receive snapshots successfully. As close to simultaneously as possible.

TCICF011: Speaker Volume - if applicable.

TCICF012: Speaker Mute - if applicable.

TCICF013: Microphone Volume - if applicable.

TCICF014: Microphone Mute - if applicable.

TCICF015: Night Vision - if applicable.

TCICF016: Optical Zoom - if applicable.

TCICF017: Digital Zoom - if applicable.

TCICF018: Rotating - if applicable.

TCICF019: Mirroring - if applicable.

TCICF023: Verify that the time it takes to spin up the stream is less than 5 seconds.

TCICF027: Ensure camera responds with error and the next (proper) stream request succeeds.

TCICF028: Streams should stop upon Pair-Verify sessions closing. Camera should stop sending packets to Controller and reset the stream configuration service back to available.

TCICF029: Starting streaming on a busy service must fail. Ensure the accessory returns error of "accessory is busy".

TCICF030: Whenever a pairing is removed, any associated data stream (e.g RTP, HDS) must also be stopped and removed immediately.

TCICF031: The RTP port number must be >= 1024.

TCICF032: IP Cameras must be able to support 8 simultaneous snapshot requests.

TCICF033: Verify that multiple controllers receive snapshots for different resolutions successfully and as close to simultaneously as possible.

TCICF034: Any IP Camera accessory must include the required services. If Camera Event Recording is supported, the optional services are also required.

TCICMT001: Gather IP Camera RTP streaming information using the RTPParser.

TCICAV001: Verify IP Camera Audio and Video streaming requirements.

TCICAV069: Verify that the accessory supports the minimum video stream requirements.

TCICR001: Verify that the camera can successfully reconfigure live stream parameters during an active live stream.

TCICF004 **Ensure the IP Camera stream starts successfully and that video is received after stop and re-starting streaming.**

Applies to IP camera accessories.

1. Pair and discover accessory.
2. In Controller window, select the accessory name and locate the Camera RTP Stream Management Service.
3. Select the "Select Stream Parameters" button, select supported parameters, and then select "Configure".
4. Select "Negotiate" and wait for "Successfully negotiated".
5. Select "Start Streaming" and verify the accessory begins to stream successfully.
6. Select "Stop Streaming".
7. Select "Start Streaming" and verify the accessory begins to stream successfully.

TCICF006 **Streaming to 2x controllers. Both controllers streaming, stop stream on controller 2 and restart controller 2 stream, verify no issues seen from stream 1. Stop stream on controller 1 and restart controller 1 stream, verify no issues seen from stream 2.**

Applies to IP camera accessories.

1. Perform this test case using HAT.
2. Pair and discover accessory (Controller 1).
3. Select the "+" at the bottom of left sidebar.
4. Select "Create IP Controller" to make a new virtual controller.
5. In left sidebar, select the Controller 1, and select the accessory name.
6. In Controllers window, under the "Add Additional Controllers" panel, select Controller 2 as Controller and select the "Add Controller" button.
7. In left sidebar, select Controller 2, select the accessory name, and select discover.
8. In left sidebar, select Controller 1, select the accessory name, and locate the Camera RTP Stream Management Service.
9. Select the "Select Stream Parameters" button, select supported parameters, and then select "Configure".
10. Select Negotiate and wait for "Successfully negotiated".

11. Select Start Streaming.
12. In left sidebar, select Controller 2, select the accessory name, and locate the Camera RTP Stream Management Service.
13. Select the "Select Stream Parameters" button, select supported parameters, and then select "Configure".
14. Select Negotiate and wait for "Successfully negotiated".
15. Select Start Streaming.
16. Verify that both controllers are streaming without error.
17. In left sidebar, select Controller 2, select the accessory name, and locate the Camera RTP Stream Management Service.
18. In the IP Camera RTP Stream Management panel, Select Stop Streaming.
19. Verify that stream on Controller 1 continues streaming.
20. In left sidebar, select Controller 2, select the accessory name, and locate the Camera RTP Stream Management Service.
21. Select the "Select Stream Parameters" button, select supported parameters, and then select "Configure".
22. Select Negotiate and wait for "Successfully negotiated".
23. Select Start Streaming.
24. Verify that both controllers are streaming without error.
25. In left sidebar, select Controller 1, select the accessory name, and locate the Camera RTP Stream Management Service.
26. In the IP Camera RTP Stream Management panel, Select Stop Streaming.
27. Verify that stream on Controller 2 continues streaming.
28. In left sidebar, select Controller 1, select the accessory name, and locate the Camera RTP Stream Management Service.
29. Select the "Select Stream Parameters" button, select supported parameters, and then select "Configure".
30. Select Negotiate and wait for "Successfully negotiated".
31. Select Start Streaming.
32. Verify that both controllers are streaming without error.

TCICF007 **Ensure a snapshot is retrieved successfully and streaming is uninterrupted.**

Applies to IP camera accessories.

1. Perform this test case using HAT.

2. Pair and discover accessory (Controller 1).
3. Select the "+" at the bottom of left sidebar.
4. Select "Create IP Controller" to make a new virtual controller.
5. In left sidebar, select the Controller 1, and select the accessory name.
6. In Controllers window, under the "Add Additional Controllers" panel, select Controller 2 as Controller and select the "Add Controller" button.
7. In left sidebar, select Controller 2, select the accessory name, and select discover.
8. In left sidebar, select Controller 1, select the accessory name, and locate the Camera RTP Stream Management Service.
9. Select the "Select Stream Parameters" button, select supported parameters, and then select "Configure".
10. Select Negotiate and wait for "Successfully negotiated".
11. Select Start Streaming.
12. In left sidebar, select Controller 2, select the accessory name, and locate the Camera RTP Stream Management Service.
13. Select the "Supported Video Stream Configuration" characteristic and select "Read" to perform Paired Read.
14. In the Events view of the trace, select the read response, show the details, and notate the resolutions supported by the accessory.
15. In the Camera Snapshot Management panel on the main accessory server view, enter the "Image Width" and "Image Height" for a supported resolution, and then select "Take Snapshot".
16. Verify a snapshot is received in the resolution specified in step 15, and the stream is uninterrupted for Controller 1.
17. Repeat steps 15 and 16 for all of the supported resolutions found in step 14.

TCICF008 **2x Controllers should receive snapshots successfully. Streams should be uninterrupted as the snapshot is processed.**

Applies to IP camera accessories.

1. Perform this test case using HAT.
2. Pair and discover accessory (Controller 1).
3. Select the "+" at the bottom of left sidebar.
4. Select "Create IP Controller" to make a new virtual controller.
5. In left sidebar, select the Controller 1, and select the accessory name.
6. In Controllers window, under the "Add Additional Controllers" panel, select Controller 2 as Controller and select the "Add Controller" button.

7. In left sidebar, select Controller 2, select the accessory name, and select discover.
8. In left sidebar, select Controller 1, select the accessory name, and locate the Camera RTP Stream Management Service.
9. Select the "Select Stream Parameters" button, select supported parameters, and then select "Configure".
10. Select Negotiate and wait for "Successfully negotiated".
11. Select Start Streaming.
12. In left sidebar, select Controller 2, select the accessory name, and locate the Camera RTP Stream Management Service.
13. Select the "Select Stream Parameters" button, select supported parameters, and then select "Configure".
14. Select Negotiate and wait for "Successfully negotiated".
15. Select Start Streaming.
16. Verify that streams run on both controllers.
17. In left sidebar, select Controller 1, select the accessory name, locate the Camera RTP Stream Management Service.
18. In the Snapshot Management panel, select "Take Snapshot".
19. Verify picture is taken for Controller 1 without HAT errors and stream is uninterrupted for Controller 2.
20. In left sidebar, select Controller 2, select the accessory name, locate the Camera RTP Stream Management Service.
21. In the Snapshot Management panel, select "Take Snapshot".
22. Verify picture is taken for Controller 2 without HAT errors and stream is uninterrupted for Controller 1.

TCICF010 **2x Controllers should receive snapshots successfully. As close to simultaneously as possible.**

Applies to IP camera accessories.

1. Sign in to iCloud account on a resident device (HomePod, AppleTV, or iPad).
2. Pair and discover accessory using Home app with iOS Controller on the same iCloud account.
3. Start Streaming.
4. Share home with 2nd iOS Controller.
5. Start Streaming with 2nd iOS Controller.
6. From the accessory, ring doorbell or trigger the motion sensor.
7. Verify that both iOS Controllers receive a banner notification with snapshot.

TCICF011 **Speaker Volume - if applicable.**

Applies to IP camera accessories.

1. Perform this test case using HAT.
2. Pair and discover accessory.
3. In Controller window, select the accessory name and locate the Camera RTP Stream Management Service.
4. Select the "Select Stream Parameters" button, select supported parameters, and then select "Configure".
5. Select Negotiate and wait for "Successfully negotiated".
6. Select Start Streaming.
7. Verify "Speaker Volume" characteristic reads/writes without error.
8. Verify stream reacts correctly.

TCICF012 **Speaker Mute - if applicable.**

Applies to IP camera accessories.

1. Perform this test case using HAT.
2. Pair and discover accessory.
3. In Controller window, select the accessory name, and locate the Camera RTP Stream Management Service.
4. Select the "Select Stream Parameters" button, select supported parameters, and then select "Configure".
5. Select Negotiate and wait for "Successfully negotiated".
6. Select Start Streaming.
7. Verify "Speaker Mute" characteristic reads/writes without error.
8. Verify stream reacts correctly.

TCICF013 **Microphone Volume - if applicable.**

Applies to IP camera accessories.

1. Perform this test case using HAT.
2. Pair and discover accessory.
3. In Controller window, select the accessory name and locate the Camera RTP Stream Management Service.

4. Select the “Select Stream Parameters” button, select supported parameters, and then select “Configure”.
5. Select Negotiate and wait for “Successfully negotiated”.
6. Select Start Streaming.
7. Verify “Mic Volume” characteristic reads/writes without error.
8. Verify stream reacts correctly.

TCICF014 **Microphone Mute - if applicable.**

Applies to IP camera accessories.

1. Perform this test case using HAT.
2. Pair and discover accessory.
3. In Controller window, select the accessory name and locate the Camera RTP Stream Management Service.
4. Select the “Select Stream Parameters” button, select supported parameters, and then select “Configure”.
5. Select Negotiate and wait for “Successfully negotiated”.
6. Select Start Streaming.
7. Verify “Microphone Mute” characteristic reads/writes without error.
8. Verify stream reacts correctly.

TCICF015 **Night Vision - if applicable.**

Applies to IP camera accessories.

1. Perform this test case using HAT.
2. Pair and discover accessory.
3. In Controller window, select the accessory name and locate the Camera RTP Stream Management Service.
4. Select the “Select Stream Parameters” button, select supported parameters, and then select “Configure”.
5. Select Negotiate and wait for “Successfully negotiated”.
6. Select “Start Streaming”.
7. Place the accessory in a dark location to test the Night Vision capability of the accessory.
8. Select the “Night Vision” characteristic and perform a Timed Write with the value “1”.
9. Verify the Night Vision Light turns On.

10. Verify live stream reacts correctly.
11. Select the "Night Vision" characteristic and perform a Timed Write with the value "0".
12. Verify the Night Vision Light turns Off.
13. Verify live stream reacts correctly.

TCICF016 **Optical Zoom - if applicable.**

Applies to IP camera accessories.

1. Perform this test case using HAT.
2. Pair and discover accessory.
3. In Controller window, select the accessory name and locate the Camera RTP Stream Management Service.
4. Select the "Select Stream Parameters" button, select supported parameters, and then select "Configure".
5. Select Negotiate and wait for "Successfully negotiated".
6. Select Start Streaming.
7. Verify "Optical Zoom" characteristic reads/writes without error.
8. Verify stream reacts correctly.

TCICF017 **Digital Zoom - if applicable.**

Applies to IP camera accessories.

1. Perform this test case using HAT.
2. Pair and discover accessory.
3. In Controller window, select the accessory name and locate the Camera RTP Stream Management Service.
4. Select the "Select Stream Parameters" button, select supported parameters, and then select "Configure".
5. Select Negotiate and wait for "Successfully negotiated".
6. Select Start Streaming.
7. Verify "Digital Zoom" characteristic reads/writes without error.
8. Verify stream reacts correctly.

TCICF018 **Rotating - if applicable.**

Applies to IP camera accessories.

1. Perform this test case using HAT.
2. Pair and discover accessory.
3. In Controller window, select the accessory name and locate the Camera RTP Stream Management Service.
4. Select the “Select Stream Parameters” button, select supported parameters, and then select “Configure”.
5. Select Negotiate and wait for “Successfully negotiated”.
6. Select Start Streaming.
7. Verify “Rotating” characteristic reads/writes without error.
8. Verify stream reacts correctly.

TCICF019 **Mirroring - if applicable.**

Applies to IP camera accessories.

1. Perform this test case using HAT.
2. Pair and discover accessory.
3. In Controller window, select the accessory name and locate the Camera RTP Stream Management Service.
4. Select the “Select Stream Parameters” button, select supported parameters, and then select “Configure”.
5. Select Negotiate and wait for “Successfully negotiated”.
6. Select Start Streaming.
7. Verify “Mirroring” characteristic reads/writes without error.
8. Verify stream reacts correctly.

TCICF023 **Verify that the time it takes to spin up the stream is less than 5 seconds.**

Applies to IP camera accessories.

1. Perform this test case using HAT.
2. Pair and discover accessory.
3. In Controller window, select the accessory name and locate the Camera RTP Stream Management Service.
4. Select the “Select Stream Parameters” button, select supported parameters, and then select “Configure”.
5. Select Negotiate and wait for “Successfully negotiated”.

6. Select Start Streaming.
7. Verify that stream did not take more than 5 seconds to begin.

TCICF027 **Ensure camera responds with error and the next (proper) stream request succeeds.**

Applies to IP camera accessories.

1. Perform this test case using HAT.
2. Pair and discover accessory.
3. In Controller window, select the accessory name and locate the Camera RTP Stream Management Service.
4. Select the "Select Stream Parameters" button, select non-supported parameters, and then select "Configure".
5. Select Negotiate and wait for "Successfully negotiated".
6. Select "Start Streaming" button.
7. In the HTTP traffic view, verify that that accessory responds with "HTTP 207" with HAP status code "-70410".
8. Select the "Select Stream Parameters" button, select supported parameters, and then select "Configure".
9. Select Negotiate and wait for "Successfully negotiated".
10. Select "Start Streaming" button.
11. Verify controller is streaming without error.

TCICF028 **Streams should stop upon Pair-Verify sessions closing. Camera should stop sending packets to Controller and reset the stream configuration service back to available.**

Applies to IP camera accessories.

1. Perform this test case using HAT.
2. Pair and discover accessory.
3. Navigate to the Camera RTP Stream Management Service.
4. Select the "Select Stream Parameters" button, select supported parameters, and then select "Configure".
5. Select Negotiate and wait for "Successfully negotiated".
6. Select Start Streaming to show the stream window.
7. Select the accessory's name in the left sidebar of Controllers window, then select the "Disconnect" button to close the security session.

8. Once the disconnect message is seen, verify the video in the stream window stops immediately.

TCICF029 Starting streaming on a busy service must fail. Ensure the accessory returns error of "accessory is busy".

Applies to IP camera accessories.

1. Perform this test case using HAT.
2. Pair and discover accessory (Controller1).
3. Select the "+" at the bottom of left sidebar.
4. Select "Create IP Controller" to make a new virtual controller.
5. In left sidebar, select the Controller 1, and select the accessory name.
6. In Controllers window, under the "Add Additional Controllers" panel, select Controller 2 as Controller and select the "Add controller" button.
7. In left sidebar, select Controller 2, select the accessory name, and select discover.
8. Repeat steps 3-7 to create and add pairing to Controller 3.
9. Using Controller 1, select the accessory name and locate the first Camera RTP Stream Management Service.
10. Select the "Select Stream Parameters" button, select supported parameters, and then select "Configure".
11. Select Negotiate and wait for "Successfully negotiated".
12. Select Start Streaming.
13. From Controller 2, select the accessory name and locate the second Camera RTP Stream Management Service.
14. Select the "Select Stream Parameters" button, select supported parameters, and then select "Configure".
15. Select Negotiate and wait for "Successfully negotiated".
16. Select Start Streaming.
17. Using Controller 3, select the accessory name and read the first "Camera RTP Stream Management" service's "Streaming Status" characteristic.
18. In the Events traffic view, locate the Characteristic Read Completed, select details, verify the Status (0x01) has the value "1" (In Use).
19. Read the second Camera RTP Stream Management Service Streaming Status Characteristic.
20. In the Events traffic view, locate the Characteristic Read Completed, select details, verify the Status (0x01) has the value "1" (In Use).

TCICF030 **Whenever a pairing is removed, any associated data stream (e.g RTP, HDS) must also be stopped and removed immediately.**

Applies to IP camera accessories.

1. Perform this test case using HAT.
2. Pair and discover accessory.
3. Navigate to the Camera RTP Stream Management Service.
4. Select the "Select Stream Parameters" button, select supported parameters, and then select "Configure".
5. Select Negotiate and wait for "Successfully negotiated".
6. Select Start Streaming to show the stream window.
7. Select accessory's name in the left sidebar of Controllers window and select Remove Pairing button.
8. Once the pairing is removed, verify the video in the stream window stops immediately.

TCICF031 **The RTP port number must be >= 1024.**

Applies to IP camera accessories. Perform this test case automatically with HCA. Below are steps that can be used to perform this test case manually in HAT.

1. Pair and discover accessory.
2. In Controller window, select the accessory name and locate the Camera RTP Stream Management Service.
3. Select the "Select Stream Parameters" button, select supported parameters, and then select "Configure".
4. Select Negotiate and wait for "Successfully negotiated".
5. In the Events traffic view, select last "IP Camera Session Negotiation Completed" event and open details.
6. Verify that "Video Port" and "Audio Port" under Accessory Config are both above or equal to 1024.
7. Repeat Step 3-6 100 times.

TCICF032 **IP Cameras must be able to support 8 simultaneous snapshot requests.**

Applies to IP camera accessories.

1. Pair and discover accessory with Home app on iOS Controller.
2. Login to the same iCloud account on 7 additional iOS Controllers.
3. Enable Notifications for motion on the motion detector settings in the Home App for the camera, on all 8 phones.

4. Sleep all iPhones.
5. Cover the camera's lens to prevent it from detecting motion.
6. Wait for 30s.
7. Uncover the camera's lens.
8. Verify that each of the iOS Controllers receive a notification of motion and a Snapshot with it.

TCICF033 **Verify that multiple controllers receive snapshots for different resolutions successfully and as close to simultaneously as possible.**

Applies to IP camera accessories. Perform this test case with HAT using the steps below.

1. Pair and discover accessory with Controller 1.
2. Select the "+" at the bottom of left sidebar.
3. Select "Create IP Controller" to make a new virtual controller.
4. In left sidebar, select the Controller 1, and select the accessory name.
5. In Controllers window, under the "Add Additional Controllers" panel, select Controller 2, and select the "Add Controller" button.
6. In left sidebar, select Controller 2, select the accessory name, then select "Discover".
7. In left sidebar, select Controller 2, select the accessory name, and then select the "Supported Video Stream Configuration" under the "Camera RTP Stream Management" service.
8. Select "Read" to perform a Paired Read.
9. In the Events view of the trace, select the read response, show the details, and notate the resolutions supported by the accessory.
10. In left sidebar, select Controller 1, select the accessory name, and locate the "Camera RTP Stream Management" service.
11. Select the "Select Stream Parameters" button, select supported parameters, and then select "Configure".
12. Select "Negotiate".
13. Select "Start Streaming".
14. Complete steps 17 and 20 as simultaneously as possible.
15. Verify the accessory begins to stream without issue.
16. In left sidebar, select Controller 1.
17. In the "Camera Snapshot Management" panel on the main accessory server view, enter the "Image Width" and "Image Height" for the lowest supported resolution, and then select "Take Snapshot".
18. Verify a snapshot is received in the resolution specified in step 17.
19. In left sidebar, select Controller 2.

20. In the Camera Snapshot Management panel on the main accessory server view, enter the "Image Width" and "Image Height" for the highest supported resolution, and then select "Take Snapshot".
21. Verify a snapshot is received in the resolution specified in step 20.
22. Verify there are no abrupt changes in the live stream, such as brightness or contrast, when simultaneous snapshots are requested.

TCICF034 **Any IP Camera accessory must include the required services. If Camera Event Recording is supported, the optional services are also required.**

Applies to IP camera accessories. Perform this test case with HAT using the steps below.

Required IP Camera Services:

- Camera RTP Stream Management
- Microphone

Optional IP Camera Services:

- Camera Event Recording Management
- Camera Operating Mode
- Data Stream Transport Management

1. Pair and discover accessory.
2. In left sidebar of Controllers window, see each of the accessory's services.
3. Verify the required services are included.
4. If the accessory supports Camera Event Recording, verify the optional services are included.

TCICMT001 **Gather IP Camera RTP streaming information using the RTPParser.**

Applies to IP camera accessories. Perform this test case with HAT using the steps below.

1. Follow provided instructions in the "RTPParser Script - User Guide" to install all necessary dependencies and prepare the script for testing.
2. Streams should be captured in a clean room where WiFi should use a clean channel, and there should be no packet loss.
3. Pair and discover accessory.
4. Verify that corresponding Microphone and Speaker services have the Mute characteristic set to "0'' (Unmuted).
5. In Controller window, select the accessory name and locate the Camera RTP Stream Management Service.

6. Select the “Select Stream Parameters” button, select supported parameters, and then select “Configure”.
7. Select“Negotiate” and then select “Start Streaming’’.
8. Allow the stream to run for approximately 2 minutes, ensuring that audio can be heard and that the camera is pointed to a target in steady motion.
9. Save the trace and close HAT.
10. Repeat steps 4-7 for all supported resolutions.
11. Copy the StreamConfigruations.plist and hat-traffic.pcap to the same folder as the CamParser script.
12. Run RTPParser.py to validate the stream performance.
13. Repeat steps 9-10 for each captured configuration.
14. Submit all log files as part of your MFi portal submission.

TCICAV001 **Verify IP Camera Audio and Video streaming requirements.**

Applies to IP camera accessories. Perform this test case with HAT using the steps below.

1. Pair and discover accessory.
2. Select the “Camera RTP Stream Management” service.
3. Select the “Select Stream Parameters” button, select supported parameters, and then select “Configure”.
4. Select“Negotiate” and then select “Start Streaming”.
5. Verify the following for each combination of resolution, framerate, and bitrate:
    - Video downlink initiates successfully in HAT.
    - There are no visual artifacts in the video downlink.
    - There are no audio artifacts in the HAT audio downlink.
    - There are no audio artifacts in the Camera audio downlink.
    - Video stream is smooth, is not jittery, and does not lag.
    - Camera is encoding at the correct resolution.
    - Framerate does not exceed target framerate.
    - Operating Bitrate is within the Target Bitrate.
    - Audio and Video are synced.
    - Verify Mandatory resolutions for corresponding aspect ratio are supported.
    - Attempt to start a stream with unsupported resolution and verify that the accessory responds with “HTTP 207” with HAP status code “70410”.

Verify for each of the following combinations:

Table 1.1: Audio and Video Configurations using AAC-ELD Audio Codec

Test Case	Resolution	HAP Specification	Video Frame Rate	Video Bit Rate	Audio Sample Rate	Aspect Ratio
TCICAV001	320 x 240	All	Highest Supported	228k	16kHz and/or 24kHz	4:3
TCICAV002	480 x 360	All	Highest Supported	228k	16kHz and/or 24kHz	4:3
TCICAV003	640 x 480	All	Highest Supported	502k	16kHz and/or 24kHz	4:3
TCICAV004	1024 x 768	All	Highest Supported	930k	16kHz and/or 24kHz	4:3
TCICAV005	1280 x 960	All	Highest Supported	1.25M	16kHz and/or 24kHz	4:3
TCICAV006	640 x 360	All	Highest Supported	422k	16kHz and/or 24kHz	16:9
TCICAV007	1280 x 720	All	Highest Supported	1.078M	16kHz and/or 24kHz	16:9
TCICAV008	1920 x 1080	All	Highest Supported	1.45M	16kHz and/or 24kHz	16:9
TCICAV033	480 x 270	R16 or later	Highest Supported	164k	16kHz and/or 24kHz	16:9
TCICAV034	1080 x 1920	R16 or later	Highest Supported	1.450M	16kHz and/or 24kHz	9:16
TCICAV035	720 x 1280	R16 or later	Highest Supported	930k	16kHz and/or 24kHz	9:16
TCICAV036	360 x 640	R16 or later	Highest Supported	422k	16kHz and/or 24kHz	9:16
TCICAV037	270 x 480	R16 or later	Highest Supported	164k	16kHz and/or 24kHz	9:16
TCICAV038	1600 x 1200	R16 or later	Highest Supported	1.940M	16kHz and/or 24kHz	4:3
TCICAV039	1440 x 1080	R16 or later	Highest Supported	1.680M	16kHz and/or 24kHz	4:3
TCICAV040	1200 x 1600	R16 or later	Highest Supported	1.940M	16kHz and/or 24kHz	3:4
TCICAV041	1080 x 1440	R16 or later	Highest Supported	1.680M	16kHz and/or 24kHz	3:4
TCICAV042	960 x 1280	R16 or later	Highest Supported	1.250M	16kHz and/or 24kHz	3:4
TCICAV043	768 x 1024	R16 or later	Highest Supported	802k	16kHz and/or 24kHz	3:4
TCICAV044	480 x 640	R16 or later	Highest Supported	502k	16kHz and/or 24kHz	3:4
TCICAV045	360 x 480	R16 or later	Highest Supported	228k	16kHz and/or 24kHz	3:4
TCICAV046	240 x 320	R16 or later	Highest Supported	228k	16kHz and/or 24kHz	3:4
TCICAV047	1536 x 1536	R16 or later	Highest Supported	2.200M	16kHz and/or 24kHz	1:1
TCICAV048	1080 x 1080	R16 or later	Highest Supported	1.180M	16kHz and/or 24kHz	1:1
TCICAV049	720 x 720	R16 or later	Highest Supported	580M	16kHz and/or 24kHz	1:1
TCICAV050	320 x 320	R16 or later	Highest Supported	218k	16kHz and/or 24kHz	1:1

Table 1.2: Mandatory Audio and Video Configurations using AAC-ELD Audio Codec

Test Case	Requirement
TCICAV001	Mandatory for R16 or later for landscape cameras
TCICAV007	Mandatory for landscape cameras
TCICAV035	Mandatory for portrait cameras
TCICAV046	Mandatory for R16 or later for portrait cameras
TCICAV050	Mandatory for R16 or later for square cameras

Table 1.3: Audio and Video Configurations using OPUS Audio Codec

Test Case	Resolution	HAP Specification	Video Frame Rate	Video Bit Rate	Audio Sample Rate	Aspect Ratio
TCICAV017	320 x 240	All	Highest Supported	228k	16kHz and/or 24kHz	4:3
TCICAV018	480 x 360	All	Highest Supported	228k	16kHz and/or 24kHz	4:3
TCICAV019	640 x 480	All	Highest Supported	502k	16kHz and/or 24kHz	4:3
TCICAV020	1024 x 768	All	Highest Supported	930k	16kHz and/or 24kHz	4:3
TCICAV021	1280 x 960	All	Highest Supported	1.25M	16kHz and/or 24kHz	4:3
TCICAV022	640 x 360	All	Highest Supported	422k	16kHz and/or 24kHz	16:9
TCICAV023	1280 x 720	All	Highest Supported	1.078M	16kHz and/or 24kHz	16:9
TCICAV024	1920 x 1080	All	Highest Supported	1.45M	16kHz and/or 24kHz	16:9
TCICAV051	480 x 270	R16 or later	Highest Supported	164k	16kHz and/or 24kHz	16:9
TCICAV052	1080 x 1920	R16 or later	Highest Supported	1.450M	16kHz and/or 24kHz	9:16
TCICAV053	720 x 1280	R16 or later	Highest Supported	930k	16kHz and/or 24kHz	9:16
TCICAV054	360 x 640	R16 or later	Highest Supported	422k	16kHz and/or 24kHz	9:16
TCICAV055	270 x 480	R16 or later	Highest Supported	164k	16kHz and/or 24kHz	9:16
TCICAV056	1600 x 1200	R16 or later	Highest Supported	1.940M	16kHz and/or 24kHz	4:3
TCICAV057	1440 x 1080	R16 or later	Highest Supported	1.680M	16kHz and/or 24kHz	4:3
TCICAV058	1200 x 1600	R16 or later	Highest Supported	1.940M	16kHz and/or 24kHz	3:4
TCICAV059	1080 x 1440	R16 or later	Highest Supported	1.680M	16kHz and/or 24kHz	3:4
TCICAV060	960 x 1280	R16 or later	Highest Supported	1.250M	16kHz and/or 24kHz	3:4
TCICAV061	768 x 1024	R16 or later	Highest Supported	802k	16kHz and/or 24kHz	3:4
TCICAV062	480 x 640	R16 or later	Highest Supported	502k	16kHz and/or 24kHz	3:4
TCICAV063	360 x 480	R16 or later	Highest Supported	228k	16kHz and/or 24kHz	3:4
TCICAV064	240 x 320	R16 or later	Highest Supported	228k	16kHz and/or 24kHz	3:4
TCICAV065	1536 x 1536	R16 or later	Highest Supported	2.200M	16kHz and/or 24kHz	1:1
TCICAV066	1080 x 1080	R16 or later	Highest Supported	1.180M	16kHz and/or 24kHz	1:1
TCICAV067	720 x 720	R16 or later	Highest Supported	580M	16kHz and/or 24kHz	1:1
TCICAV068	320 x 320	R16 or later	Highest Supported	218k	16kHz and/or 24kHz	1:1

Table 1.4: Mandatory Audio and Video Configurations using OPUS Audio Codec

Test Case	Requirement
TCICAV017	Mandatory for R16 or later for landscape cameras
TCICAV023	Mandatory for landscape cameras
TCICAV053	Mandatory for portrait cameras
TCICAV064	Mandatory for R16 or later for portrait cameras
TCICAV068	Mandatory for R16 or later for square cameras

TCICAV069 **Verify that the accessory supports the minimum video stream requirements.**

Applies to IP camera accessories. Applies to accessories that support HomeKit Accessory Protocol specification R16 or later. Perform this test case with HAT using the steps below.

1. Pair and discover accessory.
2. Navigate to the first instance of the "Camera RTP Stream Management" service.
3. Select the "Supported Video Stream Configuration" characteristic and select "Read" to perform a Paired Read.
4. In the Events view of the trace, select the read response, select "Details" to show the details, and notate all of the resolutions the accessory claims to support.
5. Verify the accessory supports a minimum resolution of either 1600x1200 or 1200x1600 (with 3:4 or 4:3 aspect ratio) or 1920x1080 or 1080x1920 (with 9:16 or 16:9 aspect ratio) as part of the supported resolutions.
6. Select the"Camera RTP Stream Management" service.
7. Select the "Select Stream Parameters" button, select supported parameters, enter a supported resolution of either 1600x1200 or 1200x1600 (with 3:4 or 4:3 aspect ratio) or 1920x1080 or 1080x1920 (with 9:16 or 16:9 aspect ratio) and then select "Configure".
8. Select "Negotiate" and then select "Start Stream".
9. Verify video stream is smooth, is not jittery, and does not lag.
10. Navigate to the next instance of the "Camera RTP Stream Management" service.
11. Select the "Supported Video Stream Configuration" characteristic and select "Read" to perform a Paired Read.
12. In the Events view of the trace, select the read response, select "Details" to show the details, and notate all of the resolutions the accessory claims to support.
13. Verify the accessory supports a minimum resolution of either 1280x960 or 960x1280 (with 3:4 or 4:3 aspect ratio) or 1280x720 or 720x1280 (with 9:16 or 16:9 aspect ratio) as part of the supported resolutions.

14. Select the “Camera RTP Stream Management” service.
15. Select the “Select Stream Parameters” button, select supported parameters, enter a supported resolution of either 960x1280 (with 3:4 or 4:3 aspect ratio) or 720x1280 (with 9:16 or 16:9 aspect ratio) and then select “Configure”.
16. Select “Negotiate” and then select “Start Stream”.
17. Verify both video streams are smooth, not jittery, and does not lag.
18. Select Stop Streaming to close the stream windows.

TCICR001 **Verify that the camera can successfully reconfigure live stream parameters during an active live stream.**

Applies to IP camera accessories. Perform this test case with HAT using the steps below.

- Video downlink initiates successfully in HAT.
- There are no visual artifacts in the video downlink.
- There are no audio artifacts in the HAT audio downlink.
- There are no audio artifacts in the Camera audio downlink.
- Verify the camera stream starts with the initial parameters, and reconfigures with the target parameters.

1. Pair and discover accessory.
2. In Controller window, select the accessory name, locate the Camera RTP Stream Management Service.
3. Select the “Select Stream Parameters” button, set the initial resolution and other supported parameters, and then select “Configure”.
4. Select“Negotiate” and then select “Start Streaming’’.
5. Verify that camera is streaming at the initial resolution.
6. Select the “Reconfigure Video” button, set the new resolution and other supported parameters, and then select “Reconfigure”.
7. Verify that the camera immediately starts streaming at the new resolution.
8. If the camera does not support the initial resolution, verify that the accessory responds with “HTTP 207” with HAP status code “-70410” and camera does not attempt renegotiation. If the camera does not support the renegotiated resolution, validate that the renegotiation fails and that the accessory responds with “HTTP 207” with HAP status code “-70410”.

Verify for each of the following combinations:

Table 1.5: 16 x 9 Video Renegotiation

Test Case	Initial Resolution	Target Resolution	HAP Specification
TCICR001	640 x 360	1280 x 720	All
TCICR002	640 x 360	1920 x 1080	All
TCICR003	1280 x 720	640 x 360	All
TCICR004	1280 x 720	1920 x 1080	All
TCICR005	1920 x 1080	640 x 360	All
TCICR006	1920 x 1080	1280 x 720	All

Table 1.6: 4 x 3 Video Renegotiation

Test Case	Initial Resolution	Target Resolution	HAP Specification
TCICR007	480 x 360	640 x 480	All
TCICR008	480 x 360	1024 x 768	All
TCICR009	480 x 360	1280 x 960	All
TCICR010	640 x 480	480 x 360	All
TCICR011	640 x 480	1024 x 768	All
TCICR012	640 x 480	1280 x 960	All
TCICR013	1024 x 768	480 x 360	All
TCICR014	1024 x 768	640 x 480	All
TCICR015	1024 x 768	1280 x 960	All
TCICR016	1280 x 960	480 x 360	All
TCICR017	1280 x 960	640 x 480	All
TCICR018	1280 x 960	1024 x 768	All
TCICR019	1280 x 960	1440 x 1080	R16 or later
TCICR020	1280 x 960	1600 x 1200	R16 or later
TCICR021	480 x 360	1440 x 1080	R16 or later
TCICR022	480 x 360	1600 x 1200	R16 or later
TCICR023	640 x 480	1440 x 1080	R16 or later
TCICR024	640 x 480	1600 x 1200	R16 or later
TCICR025	1024 x 768	1440 x 1080	R16 or later
TCICR026	1024 x 768	1600 x 1200	R16 or later
TCICR027	1440 x 1080	480 x 360	R16 or later
TCICR028	1440 x 1080	640 x 480	R16 or later
TCICR029	1440 x 1080	1024 x 768	R16 or later
TCICR030	1440 x 1080	1280 x 960	R16 or later
TCICR031	1440 x 1080	1600 x 1200	R16 or later
TCICR032	1600 x 1200	480 x 360	R16 or later
TCICR033	1600 x 1200	640 x 480	R16 or later
TCICR034	1600 x 1200	1024 x 768	R16 or later
TCICR035	1600 x 1200	1280 x 960	R16 or later
TCICR036	1600 x 1200	1440 x 1080	R16 or later

Table 1.7: 3 x 4 Video Renegotiation

Test Case	Initial Resolution	Target Resolution	HAP Specification
TCICR037	360 x 480	480 x 640	R16 or later
TCICR038	360 x 480	768 x 1024	R16 or later
TCICR039	360 x 480	960 x 1280	R16 or later
TCICR040	360 x 480	1080 x 1440	R16 or later
TCICR041	360 x 480	1200 x 1600	R16 or later
TCICR042	480 x 640	360 x 480	R16 or later
TCICR043	480 x 640	768 x 1024	R16 or later
TCICR044	480 x 640	960 x 1280	R16 or later
TCICR045	480 x 640	1080 x 1440	R16 or later
TCICR046	480 x 640	1200 x 1600	R16 or later
TCICR047	768 x 1024	360 x 480	R16 or later
TCICR048	768 x 1024	480 x 640	R16 or later
TCICR049	768 x 1024	960 x 1280	R16 or later
TCICR050	768 x 1024	1080 x 1440	R16 or later
TCICR051	768 x 1024	1200 x 1600	R16 or later
TCICR052	960 x 1280	360 x 480	R16 or later
TCICR053	960 x 1280	480 x 640	R16 or later
TCICR054	960 x 1280	768 x 1024	R16 or later
TCICR055	960 x 1280	1080 x 1440	R16 or later
TCICR056	960 x 1280	1200 x 1600	R16 or later
TCICR057	1080 x 1440	360 x 480	R16 or later
TCICR058	1080 x 1440	480 x 640	R16 or later
TCICR059	1080 x 1440	768 x 1024	R16 or later
TCICR060	1080 x 1440	960 x 1280	R16 or later
TCICR061	1080 x 1440	1200 x 1600	R16 or later
TCICR062	1200 x 1600	360 x 480	R16 or later
TCICR063	1200 x 1600	480 x 640	R16 or later
TCICR064	1200 x 1600	768 x 1024	R16 or later
TCICR065	1200 x 1600	960 x 1280	R16 or later
TCICR067	1200 x 1600	1080 x 1440	R16 or later

Table 1.8: 9 x 16 Video Renegotiation

Test Case	Initial Resolution	Target Resolution	HAP Specification
TCICR068	360 x 640	720 x 1280	R16 or later
TCICR069	360 x 640	1080 x 1920	R16 or later
TCICR070	720 x 1280	360 x 640	R16 or later
TCICR071	720 x 1280	1080 x 1920	R16 or later
TCICR072	1080 x 1920	360 x 640	R16 or later
TCICR073	1080 x 1920	720 x 1280	R16 or later

Table 1.9: 1 x 1 Video Renegotiation

Test Case	Initial Resolution	Target Resolution	HAP Specification
TCICR074	720 x 720	1080 x 1080	R16 or later
TCICR075	720 x 720	1536 x 1536	R16 or later
TCICR076	1080 x 1080	720 x 720	R16 or later
TCICR077	1080 x 1080	1536 x 1536	R16 or later
TCICR078	1536 x 1536	720 x 720	R16 or later
TCICR079	1536 x 1536	1080 x 1080	R16 or later

1.18 Video Doorbell

TCVD001: Any Doorbell service must include the required characteristics.

TCVD002: Verify that the controller can Mute and Unmute the doorbell's in-house chime.

TCVD003: Verify that the accessory rejects writes from non-admin controllers to the Mute characteristic under the Doorbell service with HAP status code -70401 (Insufficient Privileges).

TCVD004: Verify that the accessory functions as expected when the ambient temperature is within normal operating range.

TCVD005: If the accessory supports the "Limited Functionality" state, verify that the accessory is in a "Limited Functionality" state when ambient temperature nears abnormal temperature.

TCVD006: Verify that the accessory enters "Shutdown" mode when the ambient temperature nears shutdown temperature.

TCVD007: Any Video Doorbell accessory must include the required services. If Camera Event Recording is supported, it must also include the optional services.

TCVD001 **Any Doorbell service must include the required characteristics.**

Applies to Video Doorbell accessories. Perform this test case with HAT using the steps below.

Required characteristics:

- Programmable Switch Event** (r/ev*)
- Mute (r/w/ev*)
- Operation State Response (r/ev*)

* Event Notifications for BLE include Indicate (connected), Indicate (disconnected), and Broadcast.

** See the Stateless Programmable Switch chapter for additional "Programmable Switch Event" characteristic tests.

1. Pair and discover accessory.
2. In left sidebar of Controllers window, see each of the accessory's services.
3. Verify characteristics are included for each supported service type.
4. Read each paired Read characteristic and perform a valid write operation to each paired Write characteristic.
5. Verify proper values are returned for all Read characteristics and all Write characteristics properly update the accessory's current state.

TCVD002 **Verify that the controller can Mute and Unmute the doorbell's in-house chime.**

Applies to Video Doorbell accessories. Perform this test case with HAT using the steps below.

1. Pair and discover accessory.

2. Ensure accessory is connected to the in-home chime module.
3. In the sidebar of the Controllers window, select “Mute’’ characteristic under the “Doorbell” service.
4. Enter “1” (Mute) into the text field in the Prepare and Execute Timed write pane, and then select the “Timed write” button.
5. Perform a Paired Read on the Mute characteristic.
6. In the HTTP view of the trace, verify the value in the read response is “1”.
7. Press the doorbell button and verify the in-house chime does not ring.
8. Enter “0” (Unmute) into the text field in the Prepare and Execute Timed write pane, and then select the “Timed write” button.
9. Perform Paired Read on the Mute characteristic.
10. In the HTTP view of the trace, verify the value in the read response is “0”.
11. Press the doorbell button and verify in-house chime does ring.

TCVD003 **Verify that the accessory rejects writes from non-admin controllers to the Mute characteristic under the Doorbell service with HAP status code -70401 (Insufficient Privileges).**

Applies to Video Doorbell accessories. Perform this test case with HAT using the steps below.

1. Pair and discover accessory.
2. Create a new IP Controller 2 and add the additional controller as non-admin.
3. Discover the accessory with Controller 2.
4. Using Controller 2, in the sidebar of the Controllers window , select the “Mute” characteristic under the “Doorbell” service.
5. Enter “1” into the text field in the Write pane and then select the “Write” button.
6. Using the HTTP view of the trace, verify the response to the write request is HTTP 207 Multi-Status with HAP status code -70401.
7. Using Controller 2, in the sidebar of the Controllers window , select the “Mute” characteristic under the “Doorbell” service.
8. Enter “0” into the text field in the Write pane and then select the “Write” button.
9. Using the HTTP view of the trace, verify the response to the write request is HTTP 207 Multi-Status with HAP status code -70401.

TCVD004 **Verify that the accessory functions as expected when the ambient temperature is within normal operating range.**

Applies to Video Doorbell accessories. Perform this test case with HAT using the steps below.

1. Pair and discover accessory.

2. Place the accessory in room temperature with the ambient temperature within the accessory's normal operating temperature for at least 30 minutes.
3. In the sidebar of the Controllers window, select the "Operating State Response" characteristic
4. Read the "Operating State Response" characteristic.
5. Using the Events view of the trace, select the read response, and show the details.
6. Verify the value includes TLV8 type "01" (State) with a value of "0" (Normal).
7. In left sidebar, select "Camera RTP Stream Management" service and select "Select Stream Parameters".
8. In the Selected RTP Stream Configuration, enter the default parameters, set the image resolution to 960x1280 or 720x1280 for portrait camera or 1280x960 or 1280x720 for landscape camera, set the frame rate to highest supported, select "Configure", select "Negotiate" and then select "Start Streaming".
9. Verify the accessory successfully displays a live stream.
10. In left sidebar, select "Camera Event Recording Management" service and enable the "Active" characteristic by performing a timed write with a value of "1".
11. In left sidebar, select "Camera Event Recording Management" service and enable the "Recording Audio Active" characteristic by performing a timed write with a value of "1".
12. Select the "Selected Camera Recording Configuration" characteristic and select the "Build TLV" button from the Write TLV panel.
13. Select 4000msec pre-buffer duration, 4000msec fragment duration, 2000kbps video bit rate, 4000 i-Frame Rate Interval, 64kbps audio bit rate, highest Video Configuration Level, other default parameters, and enter image resolution as 1200x1600 or 1080x1920 for portrait camera or 1600x1200 or 1920x1080 for landscape camera, set the frame rate to highest and then select "Build TLV".
14. Select the "Write TLV" button.
15. In left sidebar, select the "Data Stream Transport Management" service.
16. Under the "HomeKit Data Stream pane select "Send Start Command" and then select "Connect".
17. Verify the status shows as "Connected".
18. Under Camera Recording pane, enter same stream ID on Start Message and Close Message fields and select "Send Start Request".
19. Using the HDS frames view in the trace, verify the accessory begins to send Binary Data Events.
20. Wait for 30 minutes and verify that the doorbell continues to stream during the 30 minutes.
21. Using the HDS Frames view of the trace, verify the controller continues to recieve Binary Data Events during the 30 minutes.
22. While streaming and recording is still ongoing, select the accessory name in the left sidebar. At the bottom of the main controllers window in the Camera Snapshop Management pane, enter supported resolutions for width and height, and then select "Take Secured Snapshot".
23. Verify the snapshot appears.

24. Wait 10 seconds and within the Camera Snapshop Management pane, enter supported resolutions for width and height, and then select "Take Secured Snapshot" again.
25. Verify the snapshot appears.
26. Verify the snapshots can be saved and reopened without issue.
27. After 30 minutes has elapsed, stop the HDS recording transfer by selecting the "Data Stream Transport Management" service and select the "Send Close Event" button.
28. Save and reopen the video recording, and verify it can be played back successfully.
29. Select the "Operating State Response" characteristic, select "Read".
30. Using the Events view of the trace, select the read response, and show the details.
31. Verify the value includes TLV8 type "01" (State) with a value of "0" (Normal).
32. In left sidebar, select "Camera RTP Stream Management" service and select "Stop Streaming".

TCVD005 **If the accessory supports the "Limited Functionality" state, verify that the accessory is in a "Limited Functionality" state when ambient temperature nears abnormal temperature.**

Applies to Video Doorbell accessories. Perform this test case with HAT using the steps below.

1. Pair and discover accessory.
2. In the sidebar of the Controllers window, select the "Operating State Response" characteristic under the "Doorbell" service.
3. Select "Enable" to subscribe to event notifications.
4. Set the doorbell in a temperature chamber with ambient temperature within the abnormal operating temperature for at least 30 minutes.
5. In the HTTP view of the trace, verify an "/EVENT" notification for "Operating State Response" is sent to the controller with a TLV8 value that contains type "01" (State) with value "1" (Limited Functionality), and type "02" (Abonormal Reasons) with value "4"(High Temperature).
6. In left sidebar, select "Camera RTP Stream Management" service and select "Select Stream Parameters".
7. In the Selected RTP Stream Configuration, enter the default parameters, set the image resolution to "480x640" or "360x640" for portrait camera or "640x480" or "640x360" for landscape camera, frame rate as 15fps, select "Configure", select "Negotiate" and then select "Start Streaming".
8. Verify accessory successfully begins to stream.
9. In left sidebar, select the "Active" characteristic on the "Camera Event Recording Management" service and perform a timed write with value "1".
10. In left sidebar, select "Camera Event Recording Management" service and enable the "Recording Audio Active" characteristic by performing a timed write with a value of "1".
11. Select the "Selected Camera Recording Configuration" characteristic and select the "Build TLV" button from the Write TLV panel.

12. Select 4000msec pre-buffer duration, 4000msec fragment duration, 2000kbps video bit rate, 4000 i-Frame Rate Interval, 64kbps audio bit rate, highest Video Configuration Level, other default parameters, and enter image resolution as 1200x1600 or 1080x1920 for portrait camera or 1600x1200 or 1920x1080 for landscape camera, set the frame rate as 15fps and then select "Build TLV".
13. Select the "Write TLV" button.
14. In the left sidebar, select the "Data Stream Transport Management" service.
15. Under HomeKit Data Stream panel, select "Send Start Command" and then select "Connect".
16. Under the Camera Recording panel, enter same stream ID on Start Message and Close Message fields and then select "Send Start Request".
17. Using the HDS Frames view of the trace, verify the controller recieves Binary Data Events for recording.
18. While HDS transfer and live stream is ongoing, select the accessory name in the left sidebar.
19. At the bottom of the main controllers window, in the Camera Snapshop Management panel, enter supported resolutions for width and height, and then select "Take Secured Snapshot".
20. Verify the snapshot appears, and can be saved and reopened without issue.
21. Stop the HDS recording transfer by selecting the "Data Stream Transport Management" service and select the "Send Close Event" button.
22. Save and reopen the video recording, and verify it can be played back successfully.
23. In left sidebar, select "Camera RTP Stream Management" service and select "Stop Streaming".
24. In the sidebar of the Controllers window , select "Mute'' characteristic under the "Doorbell" service.
25. Enter "0" (Unmute) into the text field in the Prepare and Execute Timed write pane and then select the "Timed write" button.
26. Perform a Paired Read on the "Mute" characteristic.
27. In the HTTP view of the trace, verify the value in the read response is "0".
28. Press the doorbell button and verify in-house chime does ring while the accessory is in the "Limited Functionality" state.
29. Bring the temperature back to normal operating range in the temperature chamber, and allow the accessory to sit for at least 30 minutes.
30. In the HTTP view of the trace, verify the accessory sends "/EVENT" characteristic notifications for the "Operating State Response" characteristic with TLV8 Type 1 (State) with value "0" (Normal).
31. In left sidebar, select "Camera RTP Stream Management" service and select "Select Stream Parameters".
32. In the Selected RTP Stream Configuration, enter the default parameters, set the image resolution to 960x1280 or 720x1280 for portrait camera or 1280x960 or 1280x720 for landscape camera, set the frame rate to highest supported, select "Configure", select "Negotiate" and then select "Start Streaming".
33. Verify the accessory successfully displays a live stream.

34. In left sidebar, select "Camera Event Recording Management" service and enable the "Active" characteristic by performing a timed write with a value of "1".
35. In left sidebar, select "Camera Event Recording Management" service and enable the "Recording Audio Active" characteristic by performing a timed write with a value of "1".
36. Select the "Selected Camera Recording Configuration" characteristic and select the "Build TLV" button from the Write TLV panel.
37. Select 4000msec pre-buffer duration, 4000msec fragment duration, 2000kbps video bit rate, 4000 i-Frame Rate Interval, 64kbps audio bit rate, highest Video Configuration Level, other default parameters, and enter image resolution as 1200x1600 or 1080x1920 for portrait camera or 1600x1200 or 1920x1080 for landscape camera, set the frame rate to highest and then select "Build TLV".
38. Select the "Write TLV" button.
39. In left sidebar, select the "Data Stream Transport Management" service.
40. Under the "HomeKit Data Stream pane select "Send Start Command" and then select "Connect".
41. Verify the status shows as "Connected".
42. Under Camera Recording pane, enter same stream ID on Start Message and Close Message fields and select "Send Start Request".
43. Using the HDS frames view in the trace, verify the accessory begins to send Binary Data Events.
44. While streaming and recording is still ongoing, select the accessory name in the left sidebar. At the bottom of the main controllers window in the Camera Snapshop Management pane, enter supported resolutions for width and height, and then select "Take Secured Snapshot".
45. Verify the snapshot appears, and can be saved and reopened without issue.
46. Stop the HDS recording transfer by selecting the "Data Stream Transport Management" service and select the "Send Close Event" button.
47. Save and reopen the video recording, and verify it can be played back successfully.
48. In left sidebar, select "Camera RTP Stream Management" service and select "Stop Streaming".
49. Verify the LED state on doorbell is Red.

TCVD006 **Verify that the accessory enters "Shutdown" mode when the ambient temperature nears shutdown temperature.**

Applies to Video Doorbell accessories. Perform this test case with HAT using the steps below.

1. Pair and discover accessory.
2. In the sidebar of the Controllers window, select the "Operating State Response" characteristic under the "Doorbell" service.
3. Select "Enable" to subscribe to event notifications.
4. Set the accessory in a temperature chamber, with ambient temperature nearing the shutdown temperature, until the accessory shuts down.

5. In the HTTP view of the trace, verify an "/EVENT" notification for the "Operating State Response" is sent to the controller before it shutdown with a TLV8 value that contains type "01" (State) with value "2" (Shutdown Imminent), and type "02" (Abnormal Reason) with value "4" (High Temperature).
6. Bring the ambient temperature of temperature chamber back to normal operating range for the accessory and let it sit for 30 minutes, or until the doorbell recovers.
7. In the "IP Discovery" view of the trace window, wait for the accessory to begin advertising again.
8. In left sidebar, select "Camera RTP Stream Management" service and select "Select Stream Parameters".
9. In the Selected RTP Stream Configuration, enter the default parameters, set the image resolution to 960x1280 or 720x1280 for portrait camera or 1280x960 or 1280x720 for landscape camera, set the frame rate to highest supported, select "Configure", select "Negotiate" and then select "Start Streaming".
10. Verify the accessory successfully displays a live stream.
11. In left sidebar, select "Camera Event Recording Management" service and enable the "Active" characteristic by performing a timed write with a value of "1".
12. In left sidebar, select "Camera Event Recording Management" service and enable the "Recording Audio Active" characteristic by performing a timed write with a value of "1".
13. Select the "Selected Camera Recording Configuration" characteristic and select the "Build TLV" button from the Write TLV panel.
14. Select 4000msec pre-buffer duration, 4000msec fragment duration, 2000kbps video bit rate, 4000 i-Frame Rate Interval, 64kbps audio bit rate, highest Video Configuration Level, other default parameters, and enter image resolution as 1200x1600 or 1080x1920 for portrait camera or 1600x1200 or 1920x1080 for landscape camera, set the frame rate to highest and then select "Build TLV".
15. Select the "Write TLV" button.
16. In left sidebar, select the "Data Stream Transport Management" service.
17. Under the "HomeKit Data Stream pane select "Send Start Command" and then select "Connect".
18. Under the Camera Recording panel, enter same stream ID on Start Message and Close Message fields and select "Send Start Request".
19. Using the HDS frames view in the trace, verify the accessory begins to send Binary Data Events.
20. While streaming and recording is still ongoing, select the accessory name in the left sidebar. At the bottom of the main controllers window in the Camera Snapshop Management pane, enter supported resolutions for width and height, and then select "Take Secured Snapshot".
21. Verify the snapshop appears, and can be saved and reopened without issue.
22. Stop the HDS transfer by selecting the "Data Stream Transport Management" service and select the "Send Close Event" button.
23. Save and reopen the video recording, and verify it can be played back successfully.
24. In left sidebar, select "Camera RTP Stream Management" service and select "Stop Streaming".

25. Verify the LED state on the accessory is red.

TCVD007 **Any Video Doorbell accessory must include the required services. If Camera Event Recording is supported, it must also include the optional services.**

Applies to Video Doorbell accessories. Perform this test case with HAT using the steps below.

Required Video Doorbell Services:

- Camera RTP Stream Management
- Speaker
- Microphone
- Accessory Runtime Information

Optional Video Doorbell Services:

- Camera Event Recording Management
- Camera Operating Mode
- Data Stream Transport Management

1. Pair and discover accessory.
2. In left sidebar of Controllers window, see each of accessory's services.
3. Verify the required services are included.
4. If the accessory supports Camera Event Recording, verify the optional services are included.

1.19 Camera Event Recording

Before performing each test within this section, it is important to ensure the accessory's characteristics are set to the proper values.

The following characteristic values should be set automatically after performing initial pair-setup, or can be set manually before performing each test case:

- "Active" characteristic(s) on the "Camera Stream Management" service(s) set to "1" (On).
- "Active" characteristic on the "Camera Event Recording Management" service set to "0" (Off).
- "Recording Audio Active" characteristic on the "Camera Event Recording Management" service set to "0" (Off).
- "Status Active" characteristic on the "Motion Sensor" service set to "1" (Active).
- "Event Snapshots Active", "Periodic Snapshots Active", and "HomeKit Camera Active" characteristics on the "Camera Operating Mode" service set to "1" (On).

The following characteristic values may not be set by default, and can be set manually before performing each test case:

- "Manually Disabled" characteristic on the "Camera Operating Mode" service set to "0" (Off).
- "Camera Operating Mode Indicator" characteristic on the "Camera Operating Mode" service set to "1" (On).
- "Night Vision" characteristic on the "Camera Operating Mode" service set to "1" (On).
- "Mute" characteristic on the "Speaker" and "Microphone" services set to "0" (Off).

TCR001: Verify that the accessory supports event recording and transferring of recorded video data via HomeKit Data Stream.

TCR002: Verify that the accessory is able to record for long durations without interruption.

TCR003: Verify that the ongoing live stream is not interrupted when a recording has started.

TCR004: Verify that the ongoing recording is not interrupted when a live stream is started.

TCR005: Verify that the accessory can successfully stream at either 1280x960 (with 4:3 or 3:4 aspect ratio) or 1280x720 (with 16:9 or 9:16 aspect ratio) while actively recording at either 1600x1200 (with 4:3 or 3:4 aspect ratio) or 1920x1080 (with 16:9 or 9:16 aspect ratio).

TCR008: The accessory must reject any Start Stream request over HDS when the "Active" characteristic on the "Camera Event Recording Management" service is set to "0" (Inactive).

TCR009: The accessory must reject any request to start a stream when Active is set to False and respond to any read/write to the Setup Endpoints characteristic and the Selected RTP Stream Configuration characteristic with HTTP 207 Multi-Status response including HAP Status Code -70412 (Not allowed in the current state).

TCR010: Verify that the accessory responds with HTTP Status Code 207 Multi-Status indicating HAP Status Code -70412 when Event Snapshots Active is set to False and a controller issues a snapshot request.

TCR013: Verify that snapshots are enabled when the accessory is set to stream and record.

TCR014: The accessory must reject any write to the "Active" characteristic associated with the "Camera Event Recording" service from a non-admin controller with HTTP Status Code 207 Multi-Status indicating HAP Status Code -70401 (Request denied due to insufficient privileges).

TCR015: The accessory must reject any write to the Active characteristics associated with any Camera RTP Management services from non-admin controllers with HTTP Status Code 207 Multi-Status indicating HAP Status Code -70401 (Request denied due to insufficient privileges).

TCR016: The accessory must be able to handle consecutive HDS data transfer operations.

TCR017: Verify accessory supports the required corresponding service for each Event Trigger Type specified in the accessory's Supported Camera Recording Configuration.

TCR019: The accessory must respond to the start data stream request within 5 seconds.

TCR020: The accessory must include the requested prebuffer duration's worth of video at the beginning of the video data sent over HDS.

TCR021: If a previously advertised audio or video recording configuration becomes unsupported (e.g., after a user-triggered configuration change via the accessory app), the accessory shall update the respective Supported Video Recording Configuration and/or Audio Recording Configuration, send a notification to all controllers, and locally discard the Selected Recording Configuration if it is no longer compatible for the next recording session. Do not discard the Selected Recording Configuration if a session is currently active (i.e., the accessory is still sending fragments with the previous configuration for the current session). In this case, the Selected Recording Configuration reflects the parameters for the next recording session, but the recording currently in progress shall continue using the previous parameters. If the controller tries to read the Selected Recording Configuration prior to writing to it, the accessory should return HAP error -70402 in case it is no longer compatible and was locally discarded.

TCR022: Reconfiguring recording parameters while recording is ongoing must not cause the recording to be interrupted. Changes to the configuration must only be applied to the next recording session.

TCR023: Verify Camera Event Recording functionality in the Home app continues after enabling 3rd party recording services (if applicable) via the accessory app.

TCR024: After first enabling 3rd party recording services via the accessory app, enable Camera Event Recording functionality in the Home app and verify Home app receives recorded video clips.

TCR025: The accessory must reject any write to the Selected Camera Recording Configuration characteristic on the Camera Event Recording Management service from non-admin controllers with HTTP Status Code 207 Multi-Status indicating HAP Status Code -70401 (Request denied due to insufficient privileges).

TCR026: The accessory shall support multiple HDS connections, but must only send video content to one connection at a time. If the accessory is already sending video content, it should respond to the new Start request with a status of Busy.

TCR027: An accessory supporting HomeKit Camera Event Recording must include the Active characteristic as part of the Camera RTP Stream Management service.

TCR032: Verify when the HomeKit Camera Active characteristic and/or Active characteristics for Camera RTP Stream Management Services is set to False while a live stream is on-going, that the accessory stops the live stream.

TCR034: Verify that the LED Indicator reflects the accessory's current state and configuration.

TCR035: Verify that the accessory rejects any HDS Start Request when the Camera Event Recording Management service's Active characteristic is set to False with a response that contains Status 1 = Not Allowed.

TCR037: Verify the presence and functionality of the Third Party Camera Active characteristic if the accessory supports streaming and/or recording outside of HomeKit.

TCR038: When Supported Video Recording Configuration is updated via accessory app, verify a notification is sent to applicable controllers, and accessory discards the previously configured configuration. If a recording is ongoing, the recording remains uninterrupted, and the subsequent recordings will use the newly defined configuration.

TCR039: If a previously advertised audio or video recording configuration becomes unsupported (e.g., after a user-triggered configuration change via the accessory app), the accessory shall locally discard the Selected Recording Configuration if it is no longer compatible for the next recording session. Do not discard the Selected Recording Configuration if a session is currently active (i.e., the accessory is still sending fragments with the previous configuration for the current session). In this case, the recording currently in progress shall continue using the previous parameters.

TCR040: Verify the values for the Event Snapshot Active, HomeKit Camera Active, and Third Party Camera Active (if applicable) characteristics on the Camera Operating Mode service are correctly set by default, after reboot after factory reset, and after pairings are removed.

TCR041: Verify the values of the Active characteristic for the Camera Event Recording Management service are correctly set by default, after reboot, after factory reset, and after pairings are removed.

TCR042: Verify that when the accessory is already sending video content, it responds to the new Start Request message with a Busy status in the Start Response message.

TCR045: Verify that the accessory returns HAP error -70402 when the controller attempts to read the value of Selected Recording Configuration before anything has been written to the Selected Recording Configuration.

TCR046: Verify that the prebuffer length is less than or equal to the value reported by the accessory in the Supported Camera Recording Configuration characteristic.

TCR047: Verify that the accessory responds with HAP Status Code -70410 when the controller attempts to write an incomplete value (tlv) to the Selected Camera Recording Configuration.

TCR048: Verify that the value written to the recording configuration persists across reboots of the accessory.

TCR055: Verify that the accessory responds with HTTP Status Code 207 Multi-Status indicating HAP Status Code -70401 when a controller attempts to write to the Event Snapshot Active characteristic if the Admin bit is not set to 1.

TCR056: Verify that the accessory responds with HTTP Status Code 207 Multi-Status indicating HAP Status Code -70401 when a controller attempts to write to the HomeKit Camera Active characteristic if the Admin bit is not set to 1.

TCR057: Verify that the accessory responds with HTTP Status Code 207 Multi-Status indicating HAP Status Code -70401 when a controller attempts to write to the Camera Operating Mode Indicator characteristic if the Admin bit is not set to 1.

TCR058: Verify that the selected recording configuration persists when changing the state of the characteristic to inactive/active.

TCR059: Verify that the Active characteristics associated to the Camera RTP Stream Management service, and Camera Event Recording Management service retain their value when HomeKit Camera Active is set to Off.

TCR060: Verify that the Active characteristic associated to the Camera RTP Stream Management, and Camera Event Recording Management services retains their values when Third Party Camera Active is set to Off.

TCR061: Verify that the accessory rejects any start request over HDS if the HomeKit Camera Active characteristic or Active characteristic for Camera Event Recording Management service is set to “0” (Off) by responding to the dataSend.open request with the Protocol Specific Error, “Not Allowed”. If the accessory is actively sending HDS data to the controller, and the HomeKit

Camera Active characteristic or Active characteristic for Camera Event Recording Management service is set to “0” (Off), the accessory shall send a Close Event with the reason key set to 1.

TCR063: Verify that if the accessory exposes a Sensor service, such as a Motion Sensor or Occupancy sensor service, that it disables the sensor when HomeKit Camera Active is set to Off by setting the Status Active characteristic on the corresponding Sensor Service to False.

TCR066: Verify that the accessory responds with HTTP Status Code 207 Multi-Status indicating HAP Status Code -70412 when HomeKit Camera Active is set to Off and a controller issues a snapshot request.

TCR067: Verify that the accessory responds to any read/write to the Setup Endpoints request with HTTP 207 Multi-Status response including HAP Status Code -70412 when the Active characteristic for the Camera RTP Stream Management service is set to false.

TCR068: Verify that when HomeKit Camera Active is set to Inactive, that the accessory responds to any read/write to the Setup Endpoints characteristic with HTTP 207 Multi-Status response including HAP Status Code -70412.

TCR076: Request an unencrypted snapshot while recording and streaming are ongoing.

TCR079: Verify the required Audio Bit Rate configurations are supported.

TCR080: Verify that the Periodic Snapshots Active characteristic controls the accessory’s ability to capture and deliver snapshots for motion and doorbell notifications.

TCR082: Verify that the accessory responds with HTTP Status Code 207 Multi-Status indicating HAP Status Code -70412 when Periodic Snapshots Active is set to “False” and a controller issues a snapshot request.

TCR083: Verify that the accessory responds with HTTP Status Code 207 Multi-Status indicating HAP Status Code -70401 when a controller attempts to write to the Periodic Snapshot Active characteristic if the Admin bit is not set to 1.

TCR085: Verify the accessory advertises the Timed Write permission for the Active characteristics on the Camera RTP Stream Management and Camera Event Recording Management services.

TCR086: Verify that the accessory accepts reasons of 0 or 1 for Event Snapshot Active and the accessory returns the a valid snapshot for each respective reason.

TCR087: Verify that the accessory responds with HTTP Status Code 207 Multi-Status indicating HAP Status Code -70412 when Periodic Snapshots Active is set to False and a controller issues a snapshot request.

TCR089: Verify the correct format, permissions, and valid values, for the Manually Disabled characteristic.

TCR090: Verify that if accessory is turned off manually (e.g via a physical button on the accessory), that it overrides both HomeKit and Third Party Camera active.

TCR091: Verify that if the accessory can be re-enabled via software, the accessory must advertise Paired Write for the Manually Disabled characteristic.

TCR093: The accessory must not include audio in the mp4 fragments sent over HDS when the Recording Audio Active characteristic is set to 0 (Disable audio in recordings).

TCR094: The accessory must reject any write to the Recording Audio Active characteristic if the Admin bit is not set to 1 in the request with HTTP Status Code 207 Multi-Status indicating HAP Status Code -70401 (Request denied due to insufficient privileges).

TCR095: Verify that the accessory rejects any periodic snapshot request or snapshot request without a (valid) reason field when the Periodic Snapshots Active characteristic is set to Disabled with HTTP Status Code 207 Multi-Status indicating HAP Status Code -70412 (Not allowed in the current state).

TCR096: Check the default values of audio recording active for all four instances (default, pairing reset, factory reset, and reboot).

TCR097: Verify that Audio Recording Active is set to 0 (Inactive) by default, after factory reset, after pairings are removed, and whenever the characteristic is set to 0 the video recording does not contain audio.

TCR098: The accessory must reject the Start Data Stream request when the Manually Disabled characteristic is set to "1" (Manually Disabled) with Protocol Specific Error, Status: 1 (Now Allowed).

TCR099: Verify that the accessory accepts read/writes to characteristics, excluding Manually Disabled, for the Camera RTP Stream Management, Data Stream Transport Management, Camera Event Recording Management, and Camera Operating Mode services when Manually Disabled characteristic is set to 1 (Manually Disabled) unless otherwise specified, and reflect the changes once Manually Disabled characteristic is set to 0 (Manually Enabled).

TCR100: Verify that the accessory supports concurrent live streams from each of its Camera RTP Stream Management services.

TCR101: Verify Audio and Video recording requirements.

TCR001 **Verify that the accessory supports event recording and transferring of recorded video data via HomeKit Data Stream.**

Applies to accessories that support Camera Event Recording. Perform this test case with HAT using the steps below.

1. Perform Pair Setup adding the accessory to HAT and discover the accessory's services and characteristics.
2. Set the HomeKit Camera Active characteristic in the Camera Operating Mode service to "1" (On) with a timed write.
3. Set the Active characteristic associated to the Camera Event Recording Management service to "1" (Active) with a timed write.
4. Set the value of the Recording Audio Active characteristic to "1" (Enabled) with a timed write.
5. Read the Supported Audio Recording Configuration characteristic. In the Details view of the Characteristic Read Completed, note the bitRate under the audioCodecRecordingParameters.
6. Configure the accessory to record by using the Build TLV button associated with the Selected Camera Recording Configuration characteristic, and using a 4000msec prebuffer duration, 4000msec fragment duration, 800kbps video bit rate, 4000 i-Frame Rate Interval, and the bitRate value from the read audioCodecRecordingParameters.
7. Select the "Write" button in the Write TLV panel.
8. Navigate to Data Stream Transport Management service and select Send Start Command, then select Connect.
9. Enter in the same integer in both Stream ID fields under Camera Recording.
10. Select Send Start Request.
11. Wait 5 seconds.

12. Select Send Close Event with reason “0”.
13. Select Save Recording.
14. Open and view the recording and verify that it is playable.
15. Remove all pairings from the accessory via HAT.

TCR002 **Verify that the accessory is able to record for long durations without interruption.**

Applies to accessories that support Camera Event Recording. Perform this test case with HAT using the steps below.

1. Perform Pair Setup adding the accessory to HAT.
2. Discover the accessory’s services and characteristics.
3. Set the HomeKit Camera Active characteristic in the Camera Operating Mode service to “1” (On) with a timed write.
4. Set the Active characteristic associated to the Camera Event Recording Management service to “1” (Active) with a timed write.
5. Set the value of the Recording Audio Active characteristic to “1” (Enabled) with a timed write.
6. Read the Supported Audio Recording Configuration characteristic. In the Details view of the Characteristic Read Completed, note the bitRate under the audioCodecRecordingParameters.
7. Configure the accessory to record by using the Build TLV button associated with the Selected Camera Recording Configuration characteristic, and using a 4000msec prebuffer duration, 4000msec fragment duration, 800kbps video bit rate, 4000 i-Frame Rate Interval, and the bitRate value from the read audioCodecRecordingParameters.
8. Select the “Write” button in the Write TLV panel.
9. Navigate to Data Stream Transport Management service and select Send Start Command, then choose Connect.
10. Enter the same integer in both Stream ID fields under the Camera Recording pane for the Start and Close messages.
11. Select Send Start Request.
12. Wait 60 minutes.
13. Select Send Close Event.
14. Select Save Recording.
15. Open and view the recording and verify that it is playable.
16. Remove all pairings from the accessory via HAT.

TCR003 **Verify that the ongoing live stream is not interrupted when a recording has started.**

Applies to accessories that support Camera Event Recording. Perform this test case with HAT using the steps below.

1. Perform Pair Setup adding the accessory to HAT.
2. Discover the accessory's services and characteristics.
3. Navigate to the Camera RTP Stream Management Service.
4. Select the "Select Stream Parameters" button, select supported parameters, and then select "Configure".
5. Select"Negotiate" and then select "Start Streaming".
6. Navigate to the Camera Event Recording Management Service.
7. Set the Active characteristic associated to the Camera Event Recording Management Service service to "1" (Active) with a timed write.
8. Set the value of the Recording Audio Active characteristic to "1" (Enabled) with a timed write.
9. Read the Supported Audio Recording Configuration characteristic. In the Details view of the Characteristic Read Completed, note the bitRate under the audioCodecRecordingParameters.
10. Configure the accessory to record by using the Build TLV button associated with the Selected Camera Recording Configuration characteristic, and using a 4000msec prebuffer duration, 4000msec fragment duration, 800kbps video bit rate, 4000 i-Frame Rate Interval, and the bitRate value from the read audioCodecRecordingParameters.
11. Select the "Write" button in the Write TLV panel.
12. Navigate to the Data Stream Transport Management service and select Send Start Command, and then choose Connect.
13. Enter the same integer in both Stream ID fields under Camera Recording.
14. Select Send Start Request.
15. Wait 5 seconds.
16. Select Send Close Event with reason "0".
17. Select Save Recording.
18. Open and view the recording and verify that it is playable.
19. Verify that the stream started in step 4 continues without interruption.
20. Remove all pairings from the accessory via HAT.

TCR004 **Verify that the ongoing recording is not interrupted when a live stream is started.**

Applies to accessories that support Camera Event Recording. Perform this test case with HAT using the steps below.

1. Perform Pair Setup adding the accessory to HAT.
2. Discover the accessory's services and characteristics.
3. Set the Active characteristic associated to the Camera Event Recording Management service to "1" (On/Active) with a timed write.
4. Set the value of the Recording Audio Active characteristic to "1" (Enabled) with a timed write.
5. Read the Supported Audio Recording Configuration characteristic. In the Details view of the Characteristic Read Completed, note the bitRate under the audioCodecRecordingParameters.
6. Configure the accessory to record by using the Build TLV button associated with the Selected Camera Recording Configuration characteristic, and using a 4000msec prebuffer duration, 4000msec fragment duration, 800kbps video bit rate, 4000 i-Frame Rate Interval, and the bitRate value from the read audioCodecRecordingParameters.
7. Select the "Write" button in the Write TLV panel.
8. Navigate to the Data Stream Transport Management service and select Send Start Command, then choose Connect.
9. Enter the same integer in both Stream ID fields under Camera Recording.
10. Select Send Start Request and use the HDS trace view to ensure the accessory begins to send video data.
11. Navigate to the Camera RTP Stream Management Service.
12. Select the "Select Stream Parameters" button, select supported parameters, and then select "Configure".
13. Select"Negotiate" and then select "Start Streaming".
14. Verify that the accessory successfully displays a live stream and continues to send binary data via HDS.
15. Navigate to the Data Stream Transport Management service.
16. Select Send Close Event with reason "0".
17. Select Save Recording.
18. Open and view the recording and verify that it is playable.
19. Remove all pairings from the accessory via HAT.

TCR005 **Verify that the accessory can successfully stream at either 1280x960 (with 4:3 or 3:4 aspect ratio) or 1280x720 (with 16:9 or 9:16 aspect ratio) while actively recording at either 1600x1200 (with 4:3 or 3:4 aspect ratio) or 1920x1080 (with 16:9 or 9:16 aspect ratio).**

Applies to accessories that support Camera Event Recording. Perform this test case with HAT using the steps below.

1. Perform Pair Setup adding the accessory to HAT.

2. Discover the accessory's services and characteristics.
3. Navigate to the Camera RTP Stream Management Service.
4. Set the Active characteristic associated to the Camera RTP Stream Management service to "1" (Active) with a timed write.
5. Navigate to the Camera RTP Stream Management Service.
6. Select the "Select Stream Parameters" button, select either 1280x960 (with 4:3 or 3:4 aspect ratio) or 1280x720 (with 16:9 or 9:16 aspect ratio) @30fps or 24fps, and then select "Configure".
7. Select"Negotiate" and then select "Start Streaming".
8. Navigate to the Camera Event Recording Management service.
9. Set the value of the Recording Audio Active characteristic to "1" (Enabled) with a timed write.
10. Read the Supported Audio Recording Configuration characteristic. In the Details view of the Characteristic Read Completed, note the bitRate under the audioCodecRecordingParameters.
11. Navigate to the Camera Event Recording Management service.
12. Set the Active characteristic associated to the Camera Event Recording Management service to "1" (On/Active) with a timed write.
13. Configure the accessory to record by using the Build TLV button associated with the Selected Camera Recording Configuration characteristic, and using either 1600x1200 (with 4:3 or 3:4 aspect ratio) or 1920x1080 (with 16:9 or 9:16 aspect ratio) resolution, 30fps, 4000msec prebuffer duration, 4000msec fragment duration, 800kbps video bit rate, 4000 i-Frame Rate Interval, and the bitRate value from the read audioCodecRecordingParameters.
14. Select the "Write" button in the Write TLV panel.
15. Navigate to the Data Stream Transport Management service and select Send Start Command, then choose Connect.
16. Enter the same integer in both Stream ID fields under Camera Recording.
17. Select Send Start Request.
18. Wait at least 5 seconds.
19. Select Send Close Event with reason "0".
20. Select Save Recording.
21. Open and view the recording and verify that it is playable.
22. Verify that the live stream succeeded and was uninterrupted.
23. Remove all pairings from the accessory via HAT.

TCR008 **The accessory must reject any Start Stream request over HDS when the "Active" characteristic on the "Camera Event Recording Management" service is set to "0" (Inactive).**

Applies to accessories that support Camera Event Recording. Perform this test case with HAT using the steps below.

1. Perform Pair Setup adding the accessory to HAT.
2. Discover the accessory's services and characteristics.
3. Set the Active characteristic associated to the Camera Event Recording Management service to "0" (Inactive) with a timed write.
4. Set the value of the Recording Audio Active characteristic to "1" (Enabled) with a timed write.
5. Read the Supported Audio Recording Configuration characteristic. In the Details view of the Characteristic Read Completed, note the bitRate under the audioCodecRecordingParameters.
6. Configure the accessory to record by using the Build TLV button associated with the Selected Camera Recording Configuration characteristic, and using a 4000msec prebuffer duration, 4000msec fragment duration, 800kbps video bit rate, 4000 i-Frame Rate Interval, and the bitRate value from the read audioCodecRecordingParameters.
7. Select the "Write" button in the Write TLV panel.
8. Navigate to the Data Stream Transport Management service and select Send Start Command, then choose Connect.
9. Enter the same integer 1 in both Stream ID fields under Camera Recording.
10. Select Send Start Request.
11. Using the HDS view in the trace, verify that the accessory responds with Status: 6 (Protocol specific error) Status: 1 (Not allowed).
12. Remove all pairings from the accessory via HAT.

TCR009 **The accessory must reject any request to start a stream when Active is set to False and respond to any read/write to the Setup Endpoints characteristic and the Selected RTP Stream Configuration characteristic with HTTP 207 Multi-Status response including HAP Status Code -70412 (Not allowed in the current state).**

Applies to accessories that support Camera Event Recording. Perform this test case with HAT using the steps below.

1. Perform Pair Setup adding the accessory to HAT.
2. Discover the accessory's services and characteristics.
3. Set the Active characteristics associated to the Camera RTP Stream Management services to "1" (Active) with a timed write.
4. Navigate to the Camera RTP Stream Management Service.
5. Select the "Select Stream Parameters" button, select supported parameters, and then select "Configure".
6. Select"Negotiate".
7. Set the Active characteristics associated to the Camera RTP Stream Management services to "0" (Inactive) with a timed write.

8. Navigate to the Camera RTP Stream Management Service and select “Start Streaming”.
9. Verify the accessory responds with HTTP Status Code 207 Multi-Status indicating HAP Status Code -70412 (Not allowed in the current state).
10. Navigate to the Camera RTP Stream Management Service.
11. Select the “Select Stream Parameters” button, select supported parameters, and then select “Configure”.
12. Select “Negotiate”.
13. Verify the accessory responds with HTTP Status Code 207 Multi-Status indicating HAP Status Code -70412 (Not allowed in the current state).
14. Remove all pairings from the accessory via HAT.

TCR010 **Verify that the accessory responds with HTTP Status Code 207 Multi-Status indicating HAP Status Code -70412 when Event Snapshots Active is set to False and a controller issues a snapshot request.**

Applies to accessories that support Camera Event Recording. Perform this test case with HAT using the steps below.

1. Perform Pair Setup adding the accessory to HAT.
2. Discover the accessory’s services and characteristics.
3. Navigate to the Camera Operating Mode service.
4. Set the Event Snapshots Active characteristic to “0” (Disable Snapshots).
5. Set the Periodic Snapshots Active characteristic to “1” (Enable).
6. Navigate to the Camera RTP Stream Management service.
7. Scroll down to the Snapshot Management panel.
8. Choose Take Snapshot without specifying a 0 or 1.
9. Verify that the accessory responds with HTTP Status Code 207 Multi-Status indicating HAP Status Code -70412.
10. Enter “0” into the Reason field, and choose Take Snapshot.
11. Verify that the accessory responds with a valid snapshot.
12. Specify reason 1, and choose Take Snapshot.
13. Verify that the accessory responds with HTTP Status Code 207 Multi-Status indicating HAP Status Code -70412.
14. Set the Event Snapshots Active characteristic to “1” (Enabled).
15. Choose Take Snapshot without specifying a 0 or 1.
16. Verify that the accessory responds with a valid snapshot.
17. Enter “0” into the Reason field, and choose Take Snapshot.

18. Verify that the accessory responds with a valid snapshot.
19. Specify reason "1", and choose Take Snapshot.
20. Verify that the accessory responds with a valid snapshot.
21. Remove all pairings from the accessory via HAT.

TCR013 **Verify that snapshots are enabled when the accessory is set to stream and record.**

Applies to accessories that support Camera Event Recording. Perform this test case with HAT using the steps below.

1. Perform Pair Setup adding the accessory to HAT.
2. Discover the accessory's services and characteristics.
3. Set the Event Snapshots Active characteristic to "0" (Disable Snapshots).
4. Set the Periodic Snapshots Active characteristic to "1" (Enable Snapshots).
5. Set the Active characteristic associated to the Camera RTP Stream Management service to "1" (Active) with a timed write.
6. Set the Active characteristic associated to the Camera Event Recording Management service to "1" (Active) with a timed write.
7. Navigate to the Camera RTP Stream Management Service.
8. Choose Take Snapshot without specifying a Reason, e.g. leave the Reason field blank.
9. Verify that the accessory responds with HTTP Status Code 207 Multi-Status indicating HAP Status Code -70412.
10. Enter "0" into the Reason field, and choose Take Snapshot.
11. Verify that a snapshot is received.
12. Enter "1" into the Reason field, and choose Take Snapshot.
13. Verify that the accessory responds with HTTP Status Code 207 Multi-Status indicating HAP Status Code -70412.
14. Set the Event Snapshots Active characteristic to "1" (Enable Snapshots).
15. Navigate to the Camera RTP Stream Management Service.
16. Choose Take Snapshot without specifying a Reason, e.g. leave the Reason field blank.
17. Verify that a snapshot is received.
18. Enter "0" into the Reason field, and choose Take Snapshot.
19. Verify that a snapshot is received.
20. Enter "1" into the Reason field, and choose Take Snapshot.
21. Verify that a snapshot is received.

22. Remove all pairings from the accessory via HAT.

TCR014 **The accessory must reject any write to the "Active" characteristic associated with the "Camera Event Recording" service from a non-admin controller with HTTP Status Code 207 Multi-Status indicating HAP Status Code -70401 (Request denied due to insufficient privileges).**

Applies to accessories that support Camera Event Recording. Perform this test case with HAT using the steps below.

1. Perform Pair Setup adding the accessory to HAT.
2. Discover the accessory's services and characteristics.
3. Using the admin controller, set the Active characteristic to "0" (Inactive) on the Recording Event Recording Management service.
4. Add an additional IP controller.
5. Add the new controller as a non-admin controller.
6. Using the non-admin controller, attempt to set the "Active" characteristic associated to the "Camera Event Recording Management" service to "1" (Active) with a timed write.
7. Verify accessory responds to the write request with HTTP Status Code 207 Multi-Status indicating HAP Status Code -70401.
8. Using the non-admin controller, attempt to set the "Active" characteristic associated to the "Camera Event Recording Management" service to "0" (Inactive) with a timed write.
9. Verify accessory responds to the write request with HTTP Status Code 207 Multi-Status indicating HAP Status Code -70401.
10. Remove all pairings from the accessory via HAT.

TCR015 **The accessory must reject any write to the Active characteristics associated with any Camera RTP Management services from non-admin controllers with HTTP Status Code 207 Multi-Status indicating HAP Status Code -70401 (Request denied due to insufficient privileges).**

Applies to accessories that support Camera Event Recording. Perform this test case with HAT using the steps below.

1. Perform Pair Setup adding the accessory to HAT.
2. Discover the accessory's services and characteristics.
3. Using the admin controller, set the Active characteristic on all Camera RTP Management services to "0" (Inactive).
4. Add an additional IP controller.
5. Add the new controller as a non-admin controller.
6. Using the non-admin controller, attempt to set the Active Characteristic associated to each Camera RTP Stream Management service to "1" (Active) with a timed write.

7. Verify accessory responds to the write request with HTTP Status Code 207 Multi-Status indicating HAP Status Code -70401.
8. Using the non-admin controller, attempt to set the Active characteristic associated with each Camera RTP Stream Management service to "0" (Inactive) with a timed write.
9. Verify accessory responds to the write request with HTTP Status Code 207 Multi-Status indicating HAP Status Code -70401.
10. Remove all pairings from the accessory via HAT.

TCR016 **The accessory must be able to handle consecutive HDS data transfer operations.**

Applies to accessories that support Camera Event Recording. Perform this test case with HAT using the steps below.

1. Perform Pair Setup adding the accessory to HAT.
2. Discover the accessory's services and characteristics.
3. Set the HomeKit Camera Active characteristic in the Camera Operating Mode service to "1" (On) with a timed write.
4. Set the Active characteristic associated to the Camera Event Recording Management service to "1" (Active) with a timed write.
5. Set the value of the Recording Audio Active characteristic is "1" (Enabled) with a timed write.
6. Read the Supported Audio Recording Configuration characteristic. In the Details view of the Characteristic Read Completed, note the bitRate under the audioCodecRecordingParameters.
7. Configure the accessory to record by using the Build TLV button associated with the Selected Camera Recording Configuration characteristic, and using a 4000msec prebuffer duration, 4000msec fragment duration, 800kbps video bit rate, 4000 i-Frame Rate Interval, and the bitRate value from the read audioCodecRecordingParameters.
8. Select the "Write" button in the Write TLV panel.
9. Navigate to the Data Stream Transport Management service and select Send Start Command, then choose Connect.
10. Enter the integer 1 in both Stream ID fields under Camera Recording.
11. Select Send Start Request.
12. Wait 5 seconds.
13. Select Send Close Event with reason "0".
14. Select Save Recording.
15. Enter the integer 2 in both Stream ID fields under Camera Recording.
16. Select Send Start Request.
17. Wait 5 seconds.

18. Select Send Close Event with reason “0”.
19. Select Save Recording.
20. Enter the integer 3 in both Stream ID fields under Camera Recording.
21. Select Send Start Request.
22. Wait 5 seconds.
23. Select Send Close Event with reason “0”.
24. Select Save Recording.
25. Open and view all of the recordings and verify that they are playable.
26. Remove all pairings from the accessory via HAT.

TCR017 **Verify accessory supports the required corresponding service for each Event Trigger Type specified in the accessory’s Supported Camera Recording Configuration.**

Applies to accessories that support Camera Event Recording. Perform this test case with HAT using the steps below.

1. Perform Pair Setup adding the accessory to HAT.
2. Discover the accessory’s services and characteristics.
3. Read the Supported Camera Recording Configuration, and using the Events view in the trace, notate each eventTriggerType, e.g. bit 0 (Motion) , bit 1 (Doorbell).
4. For each eventTriggerType, verify accessory includes the correct corresponding service(s), e.g. for bit 0 (Motion), verify the presence of a Motion Sensor service. For bit 1 (Doorbell), verify the presence of a Doorbell service.
5. Remove all pairings from the accessory via HAT.

TCR019 **The accessory must respond to the start data stream request within 5 seconds.**

Applies to accessories that support Camera Event Recording. Perform this test case with HAT using the steps below.

1. Perform Pair Setup adding the accessory to HAT.
2. Discover the accessory’s services and characteristics.
3. Set the HomeKit Camera Active characteristic in the Camera Operating Mode service to “1” (On) with a timed write.
4. Set the Active characteristic associated to the Camera Event Recording Management service to “1” (Active) with a timed write.
5. Read the Supported Audio Recording Configuration characteristic. In the Details view of the Characteristic Read Completed, note the bitRate under the audioCodecRecordingParameters.

6. Configure the accessory to record by using the Build TLV button associated with the Selected Camera Recording Configuration characteristic, and using a 4000msec prebuffer duration, 4000msec fragment duration, 800kbps video bit rate, 4000 i-Frame Rate Interval, and the bitRate value from the read audioCodecRecordingParameters.
7. Select the "Write" button in the Write TLV panel.
8. Navigate to the "Data Stream Transport Management" service and select "Send Start Command", and then Connect.
9. Enter the same integer in both Stream ID fields under Camera Recording.
10. Select Send Start Request.
11. Using the HDS view in the trace, verify that the accessory responds to the Data Start Request within 5 seconds.
12. Remove all pairings from the accessory via HAT.

TCR020 **The accessory must include the requested prebuffer duration's worth of video at the beginning of the video data sent over HDS.**

Applies to accessories that support Camera Event Recording. Perform this test case with HAT using the steps below.

1. Perform Pair Setup adding the accessory to HAT.
2. Discover the accessory's services and characteristics.
3. Set the HomeKit Camera Active characteristic in the Camera Operating Mode service to "1" (On) with a timed write.
4. Set the Active characteristic associated to the Camera Event Recording Management service to "1" (Active) with a timed write.
5. Navigate to Supported Camera Recording Configuration.
6. Read the characteristic value.
7. View the response in the Event view of the trace, and note the value of the supported prebuffer duration in seconds, and verify it is at least 4000msec.
8. Select Camera Event Recording Management.
9. Read the Supported Audio Recording Configuration characteristic. In the Details view of the Characteristic Read Completed, note the bitRate under the audioCodecRecordingParameters.
10. Using the maximum supported Prebuffer Duration specified in the Supported Camera Recording Configuration from step 6, configure the accessory to record by using the Build TLV button associated with the Selected Camera Recording Configuration characteristic, and using a 4000msec fragment duration, 800kbps video bit rate, 4000 i-Frame Rate Interval, and the bitRate value from the read audioCodecRecordingParameters.
11. Select the "Write" button in the Write TLV panel.

12. Navigate to Data Stream Transport Management service and select Send Start Command, then choose Connect.
13. Enter the same integer in both Stream ID fields under Camera Recording.
14. Point the accessory at a stopwatch.
15. Start the stopwatch and choose Send Start Request.
16. Wait 10 seconds in addition to the prebuffer duration requested in step 8.
17. Select Send Close Event.
18. Save, open, and play video to verify that the length of the prebuffer duration included in the video is equal to the requested prebuffer duration from step 9.
19. Remove all pairings from the accessory via HAT.

TCR021 **If a previously advertised audio or video recording configuration becomes unsupported (e.g., after a user-triggered configuration change via the accessory app), the accessory shall update the respective Supported Video Recording Configuration and/or Audio Recording Configuration, send a notification to all controllers, and locally discard the Selected Recording Configuration if it is no longer compatible for the next recording session. Do not discard the Selected Recording Configuration if a session is currently active (i.e., the accessory is still sending fragments with the previous configuration for the current session). In this case, the Selected Recording Configuration reflects the parameters for the next recording session, but the recording currently in progress shall continue using the previous parameters. If the controller tries to read the Selected Recording Configuration prior to writing to it, the accessory should return HAP error -70402 in case it is no longer compatible and was locally discarded.**

Applies to accessories that support Camera Event Recording. Applies to camera accessories with accessory apps that provide the ability to change the Supported Video and/or Audio Recording Configuration(s) while paired to HAT. Perform this test case using HAT and an iOS device running the accessory app.

1. Using an iOS device, setup the accessory in the accessory app.
2. Pair and discover the accessory in HAT.
3. Navigate to the Camera RTP Stream Management Service.
4. Set the Active characteristic associated to the Camera RTP Stream Management service to “1” (Active) with a timed write.
5. Navigate to the Camera Event Recording Management Service.
6. Set the value of the Recording Audio Active characteristic to “1” (Enabled) with a timed write.
7. Read the Supported Audio Recording Configuration characteristic. In the Details view of the Characteristic Read Completed, note the bitRate under the audioCodecRecordingParameters.
8. Navigate to both the Supported Audio Recording Configuration and Supported Video Recording Configuration characteristics, then select “Enable” for Event Notifications.
9. Read the Selected Camera Recording Configuration characteristic.

10. Using the HTTP view in the trace, verify that the accessory responds to the read request with HAP error -70402.

11. Read and notate the values of both the Supported Audio Recording Configuration and Supported Video Recording Configuration characteristics.

12. Configure the accessory to record by using the "Build TLV" button associated with the Selected Camera Recording Configuration characteristic, and using a 4000msec prebuffer duration, 4000msec fragment duration, 800kbps video bit rate, 4000 i-Frame Rate Interval, and the bitRate value from the read audioCodecRecordingParameters.

13. Select the "Write" button in the Write TLV panel.

14. Notate the parameters used in step 11.

15. Using the accessory app, change the Supported Audio Recording Configuration and/or Supported Video Recording Configuration so that all or some of the parameters used in step 11 are no longer supported.

16. Read and notate the values of both the Supported Audio Recording Configuration and Supported Video Recording Configuration characteristics, and verify that the values are updated to reflect the changes from step 14.

17. Read the Selected Camera Recording Configuration characteristic, and using the HTTP view in the trace, verify that the accessory responds to the read request with HAP error -70402.

18. Using the new parameters from step 14, configure the accessory to record by using the "Build TLV" button associated with the Selected Camera Recording Configuration characteristic, and using a 4000msec prebuffer duration, 4000msec fragment duration, 800kbps video bit rate, 4000 i-Frame Rate Interval, and the bitRate value from the read audioCodecRecordingParameters.

19. Navigate to the Data Stream Transport Management service and select Send Start Command, then select Connect.

20. Enter "1" in both Stream ID fields under Camera Recording, and select Send Start Request.

21. Using the HDS trace view, verify that the accessory begins sending video data via HDS.

22. Using the accessory app, change the Supported Audio Recording Configuration and/or Supported Video Recording Configuration so that all or some of the parameters used in step 17 are no longer supported.

23. Verify that Controller 1 receives a notification for Supported Audio Recording Configuration and Supported Video Recording Configuration with their new values.

24. Using the new recording parameters set in the accessory app in step 21, configure the accessory to record by using the "Build TLV" button associated with the Selected Camera Recording Configuration characteristic, and using a 4000msec prebuffer duration, 4000msec fragment duration, 800kbps video bit rate, 4000 i-Frame Rate Interval, and the bitRate value from the read audioCodecRecordingParameters.

25. Read the value of the Selected Camera Recording Configuration characteristic and verify it reflects the parameters set in step 23.

26. Using the HDS view in the trace, verify that the accessory continues to send video data.

27. Navigate to the Data Stream Transport Management service and select Send Close Event with reason 0.
28. Save and open video, and verify that the video continued to use the parameters from step 17.
29. Navigate to the Data Stream Transport Management service and choose Send Start Command, then choose Connect.
30. Enter "2" in both Stream ID fields under Camera Recording, and select Send Start Request.
31. Wait 5 seconds.
32. Select "Send Close Event" with reason 0.
33. Save and open video, and verify that the video continued to use the parameters from step 23.

TCR022 **Reconfiguring recording parameters while recording is ongoing must not cause the recording to be interrupted. Changes to the configuration must only be applied to the next recording session.**

Applies to accessories that support Camera Event Recording. Perform this test case with HAT using the steps below.

1. Perform Pair Setup adding the accessory to HAT.
2. Discover the accessory's services and characteristics.
3. In the Controllers window, select "+" to create a new IP Controller 2.
4. Under Controller 1, select the accessory name, under "Add Additional Controllers" panel, select "Controller 2" as Controller, select 'on' for Admin, then select the "Add Controller" button.
5. On the left pane of the Controllers window, select the accessory name under Controller 2, select the "Start" button, then select the "Discover" button.
6. Using Controller 2, navigate to the Selected Camera Recording Configuration characteristic and select "Enable" for Event Notifications.
7. Using Controller 1, set the Active characteristic associated to the Camera Event Recording Management service to "1" (Active) with a timed write.
8. Read the Supported Audio Recording Configuration characteristic. In the Details view of the Characteristic Read Completed, note the bitRate under the audioCodecRecordingParameters.
9. Configure the accessory to record by using the Build TLV button associated with the Selected Camera Recording Configuration characteristic to write a valid TLV with the resolution of 1920x1080 @ 15fps, 24fps, or 30fps to the Selected Camera Recording Configuration characteristic, and using a 4000msec prebuffer duration, 4000msec fragment duration, 800kbps video bit rate, 4000 i-Frame Rate Interval, and the bitRate value from the read audioCodecRecordingParameters.
10. Select the "Write" button in the Write TLV panel.
11. Enable Event Notifications for the Selected Camera Recording Configuration characteristic.
12. Navigate to Data Stream Transport Management service and select Send Start Command, then choose Connect.

13. Enter the same integer, e.g. 1, in both Stream ID fields under the Camera Recording pane.
14. Select Send Start Request.
15. Wait 5 seconds.
16. Configure the accessory to record by using the Build TLV button associated with the Selected Camera Recording Configuration characteristic to write a valid TLV with the resolution of 1280x720 @ 15fps, 24fps or 30fps to the Selected Camera Recording Configuration characteristic, and using a 4000msec prebuffer duration, 4000msec fragment duration, 800kbps video bit rate, 4000 i-Frame Rate Interval, and the bitRate value from the read audioCodecRecordingParameters.
17. Select the "Write" button in the Write TLV panel.
18. Verify that the accessory sends a notification to Controller 2 that the Selected Video Recording Configuration has been updated with the new configuration from step 15.
19. Verify that by using the HDS trace view the accessory continues to send data to the controller.
20. Navigate to the Data Stream Transport Management service.
21. Select Send Close Event.
22. Select Save Recording.
23. Open and view the recording and verify that it is playable and is at the correct selected resolution of 1920x1080 (or other 1080p-equivalent select in Step 8).
24. Navigate to the Data Stream Transport Management service and select Send Start Command, then choose Connect.
25. Enter the same integer, e.g. 2, in both Stream ID fields under Camera Recording pane.
26. Select Send Start Request.
27. Wait 5 seconds.
28. Select Send Close Event.
29. Select Save Recording.
30. Open and view the recording and verify that it is playable and is at the newly selected resolution of 1280x720 (or other 720p-equivalent select in Step 15).
31. Remove all pairings from the accessory via HAT.

TCR023 **Verify Camera Event Recording functionality in the Home app continues after enabling 3rd party recording services (if applicable) via the accessory app.**

Applies to accessories that support Camera Event Recording. Perform this test case using an iOS device running the accessory app and the Home app.

1. Perform Pair Setup adding the accessory to Controller A using iCloud account A.
2. Verify that Controller A completes the add and can stream from the accessory.

3. Enable Home Hub on a supported iPad or Apple TV, or use a HomePod, running the latest version of iOS using the same iCloud account.
4. Using Controller A, change the accessory settings to enable streaming and enable recording for all motion.
5. Create a Motion or Doorbell event to cause recording to being.
6. Stop creating Motion or Doorbell events to cause recording to stop.
7. Wait for 30s.
8. Verify the event is captured in the Home app as a viewable clip.
9. Add the accessory app to the device and add the accessory to the accessory app (while maintaining the HomeKit pairing).
10. Configure recording services for the accessory app.
11. Create a Motion or Doorbell event to cause recording to begin.
12. Stop creating Motion or Doorbell events to cause recording to stop.
13. Wait for 30s.
14. Verify recording succeeds and the clip can be viewed (in Home app).

TCR024 **After first enabling 3rd party recording services via the accessory app, enable Camera Event Recording functionality in the Home app and verify Home app receives recorded video clips.**

Applies to accessories that support Camera Event Recording. Applies to camera accessories that support recording within the accessory app. Perform this test case using an iOS device running the accessory app and the Home app.

1. Add the accessory to the Home app on Controller A, and ensure the accessory settings are set to Stream only, with Recoding disabled.
2. Verify the accessory can successfully display a live stream in the Home app.
3. Add the accessory app to the device and add the accessory to the accessory app (while maintaining the HomeKit pairing).
4. Configure recording services for the accessory app.
5. Enable HomeHub on a supported iPad or Apple TV, or use a HomePod, running the latest version of iOS using the same iCloud account.
6. Using the Home app on Controller A, change the accessory settings to enable Streaming and enable Recording for all motion.
7. Create a Motion or Doorbell event to cause recording to begin.
8. Stop creating Motion or Doorbell events to cause recording to stop.
9. Wait for 30s.
10. Verify the event is captured in the Home app as a viewable clip.

TCR025 **The accessory must reject any write to the Selected Camera Recording Configuration characteristic on the Camera Event Recording Management service from non-admin controllers with HTTP Status Code 207 Multi-Status indicating HAP Status Code -70401 (Request denied due to insufficient privileges).**

Applies to accessories that support Camera Event Recording. Perform this test case with HAT using the steps below.

1. Perform Pair Setup adding the accessory to HAT.
2. Discover the accessory's services and characteristics.
3. Read the Supported Audio Recording Configuration characteristic. In the Details view of the Characteristic Read Completed, note the bitRate under the audioCodecRecordingParameters.
4. Configure the accessory to record by using the Build TLV button associated with the Selected Camera Recording Configuration characteristic, and using a 4000msec prebuffer duration, 4000msec fragment duration, 800kbps video bit rate, 4000 i-Frame Rate Interval, and the bitRate value from the read audioCodecRecordingParameters.
5. Select the "Write" button in the Write TLV panel.
6. Add an additional IP controller.
7. Add the new controller as a non-admin controller.
8. Using the non-admin controller repeat step 3 to attempt to write a TLV to the Selected Camera Recording Configuration.
9. Verify that the accessory responds to the request with HTTP Status Code 207 Multi-Status indicating HAP Status Code -70401.
10. Remove all pairings from the accessory via HAT.

TCR026 **The accessory shall support multiple HDS connections, but must only send video content to one connection at a time. If the accessory is already sending video content, it should respond to the new Start request with a status of Busy.**

Applies to accessories that support Camera Event Recording. Perform this test case with HAT using the steps below.

1. Perform Pair Setup adding the accessory to HAT.
2. Create additional IP Controllers in HAT (for a total of 2).
3. Select Add Additional Controllers as admin.
4. Discover the accessory's services and characteristics on both of the HAT controllers.
5. Set the HomeKit Camera Active characteristic in the Camera Operating Mode service to "1" (On) with a timed write.
6. Set the Active characteristic associated to the Camera Event Recording Management service to "1" (Active) with a timed write.

7. Read the Supported Audio Recording Configuration characteristic. In the Details view of the Characteristic Read Completed, note the bitRate under the audioCodecRecordingParameters.
8. Configure the accessory to record by using the Build TLV button associated with the Selected Camera Recording Configuration characteristic, and using a 4000msec prebuffer duration, 4000msec fragment duration, 800kbps video bit rate, 4000 i-Frame Rate Interval, and the bitRate value from the read audioCodecRecordingParameters.
9. Select the "Write" button in the Write TLV panel.
10. Select Send Start Command and then Connect to set up an HDS session on the Data Stream Transport Management service.
11. Using the HDS view on the trace, verify "Data Stream Connected" event is seen for the controller.
12. Using Controller 1, enter in the integer 1 in both Stream ID fields under Camera Recording.
13. Using Controller 1, navigate to Data Stream Transport Management and select Send Start Request.
14. Using Controller 2, enter in the integer 2 in both Stream ID fields under Camera Recording.
15. Select Send Start Command and then Connect to set up an HDS session on the Data Stream Transport Management service.
16. Using the HDS view on the trace, verify that the Data Stream Connected event is seen for the controller.
17. Using the HDS view in the trace, verify that both controller's HDS sessions are still connected (e.g., no Data Stream Disconnected messages are seen).
18. Using Controller 2, select Send Start Request.
19. Using the HomeKit Data Stream trace view, verify the accessory responds to the Start Data Stream request with a response that has a header that includes dataSend as the protocol, Open as the topic, 6 as the Status set (Protocol Specific Error), and a message with properties for streamID with a value that matches the streamID the controller chose, and 2 as the Status set(Busy).
20. Remove all pairings from the accessory via HAT.

TCR027 **An accessory supporting HomeKit Camera Event Recording must include the Active characteristic as part of the Camera RTP Stream Management service.**

Applies to accessories that support Camera Event Recording. Perform this test case with HAT using the steps below.

1. Pair and discover the accessory.
2. In left side bar of Controllers window, verify the Active characteristic is included within the Camera RTP Stream Management service.

TCR032 **Verify when the HomeKit Camera Active characteristic and/or Active characteristics for Camera RTP Stream Management Services is set to False while a live stream is on-going, that the accessory stops the live stream.**

Applies to accessories that support Camera Event Recording. Perform this test case with HAT using the steps below.

1. Perform Pair Setup adding the IP Camera to HAT.
2. Discover the accessory's services and characteristics.
3. Select Camera RTP Stream Management Service.
4. Select the "Select Stream Parameters" button, select supported parameters, and then select "Configure".
5. Select"Negotiate" and then select "Start Streaming".
6. After the accessory begins the live stream, set the Active characteristic associated with the Camera RTP Stream Management service that is actively streaming to "0" (Inactive) using a Timed Write.
7. Verify that the stream stops.
8. Wait 5 seconds.
9. Set the Active characteristic from step 6 to "1" (Active) using a Timed Write.
10. Navigate to the Camera RTP Stream Management Service, then select the "Stop Streaming" button.
11. Select the "Select Stream Parameters" button, select supported parameters, and then select "Configure".
12. Select"Negotiate" and then select "Start Streaming".
13. Verify the live stream functions properly.
14. Select the "Stop Streaming" button.
15. Repeat steps 3-12 for each Camera RTP Stream Management Service.
16. Remove all pairings from the accessory via HAT.

TCR034 **Verify that the LED Indicator reflects the accessory's current state and configuration.**

Applies to accessories that support Camera Event Recording. Perform this test case with HAT using the steps below.

1. Perform Pair Setup adding the accessory to HAT.
2. Discover the accessory's services and characteristics.
3. Read the Supported Audio Recording Configuration characteristic. In the Details view of the Characteristic Read Completed, note the bitRate under the audioCodecRecordingParameters.
4. Configure the accessory to record by using the Build TLV button associated with the "Selected Camera Recording Configuration" characteristic, using a 4000msec prebuffer duration, 4000msec fragment duration, 800kbps video bit rate, 4000 i-Frame Rate Interval, and the bitRate value from the read audioCodecRecordingParameters.
5. Set the "HomeKit Camera Active" characteristic to "0" (Off).

6. Set the Active characteristic associated to the “Camera Event Recording Management” service to “0” (Inactive) with a timed write.
7. Verify that the LED is Off.
8. Set the “HomeKit Camera Active” characteristic to “1” (On).
9. Set the “Active” characteristic associated to the “Camera Event Recording Management” service to “1” (Active) with a timed write.
10. Verify that the LED is Red.
11. Set the “Active” characteristic associated to the “Camera Event Recording Management” service to “0” (Inactive) with a timed write.
12. Verify that the LED is Blue.
13. Select the “Camera RTP Stream Management” service.
14. Select the “Select Stream Parameters” button, select supported parameters, then select “Configure”.
15. Select“Negotiate” and “Start Streaming”.
16. Verify that the LED is Red.
17. Select “Stop Streaming”.
18. Verify that the LED is Blue.
19. Set the “Active” characteristic associated to the “Camera Event Recording Management” service to “1” (Active) with a timed write.
20. Select the “Write” button in the Write TLV panel.
21. Navigate to the “Data Stream Transport Management” service and select “Send Start Command”, then choose “Connect”.
22. Enter the same integer in both Stream ID fields under Camera Recording.
23. Select “Send Start Request”.
24. Verify that the LED is Red.
25. Select the “Camera RTP Stream Management” service.
26. Select the “Select Stream Parameters” button, select supported parameters, and then select “Configure”.
27. Select“Negotiate” and “Start Streaming”.
28. Verify that the LED is still Red.
29. Select “Stop Streaming”.
30. Verify that the LED is still Red.
31. Navigate to the “Data Stream Transport Management” service and select “Send Close Event”.
32. Verify that the LED is still Red.

33. Remove all pairings from the accessory via HAT.

TCR035 **Verify that the accessory rejects any HDS Start Request when the Camera Event Recording Management service's Active characteristic is set to False with a response that contains Status 1 = Not Allowed.**

Applies to accessories that support Camera Event Recording. Perform this test case with HAT using the steps below.

1. Perform Pair Setup adding the accessory to HAT.
2. Discover the accessory's services and characteristics.
3. Select Camera Event Recording Management.
4. Read the Supported Audio Recording Configuration characteristic. In the Details view of the Characteristic Read Completed, note the bitRate under the audioCodecRecordingParameters.
5. Configure the accessory to record by using the Build TLV button associated with the Selected Camera Recording Configuration characteristic, and using a 4000msec prebuffer duration, 4000msec fragment duration, 800kbps video bit rate, 4000 i-Frame Rate Interval, and the bitRate value from the read audioCodecRecordingParameters.
6. Select the "Write" button in the Write TLV panel.
7. Navigate to the Data Stream Transport Management service and enter the same integer in both Stream ID fields under Camera Recording.
8. Select Send Start Command and then Connect to setup an HDS session on the Data Stream Transport Management service.
9. Set the Active characteristic associated with Camera Event Recording Management Service to "0" (Inactive) with a timed write.
10. Select Send Start Request.
11. Using the HomeKit Data Stream trace view, verify that the accessory responds to the Start Data Stream request with a response that has a header that includes dataSend as the protocol, Open as the topic, Status set to 6 (Protocol Specific Error), and message with properties for steamID with a value that matches the streamID the controller chose, and Status set to 1 (Not Allowed).
12. Remove all pairings from the accessory via HAT.

TCR037 **Verify the presence and functionality of the Third Party Camera Active characteristic if the accessory supports streaming and/or recording outside of HomeKit.**

Applies to accessories that support Camera Event Recording. Applies to accessories that support streaming and/or recording outside of HomeKit. Perform this test case using an iOS device running the accessory app and the Home app.

1. Add accessory to the accessory app with Controller A while signed into iCloud.
2. Verify that Controller A completes the add and can stream from the accessory.

3. Enable Home Hub on a supported iPad or Apple TV, or use a HomePod, running the latest version of iOS using the same iCloud account.
4. Using Controller A, add accessory to the Home App and configure the accessory to enable streaming and enable recording for all motion.
5. Verify the collection of a clip and that it can be played locally.
6. Verify the presence, and functionality of the Third Party Camera Active in the accessory app.
7. Remove all pairings from the accessory via iOS.

TCR038 **When Supported Video Recording Configuration is updated via accessory app, verify a notification is sent to applicable controllers, and accessory discards the previously configured configuration. If a recording is ongoing, the recording remains uninterrupted, and the subsequent recordings will use the newly defined configuration.**

Applies to accessories that support Camera Event Recording. Perform this test case using an iOS device running the accessory app and the Home app.

1. Add accessory to the accessory app with Controller A while signed into iCloud.
2. After adding the accessory to the accessory app, verify the accessory can successfully display a live stream.
3. Enable Home Hub on a supported iPad or Apple TV, or use a HomePod, running the latest version of iOS using the same iCloud account.
4. Using Controller A, add accessory to the Home app and configure the accessory to enable streaming and enable recording for all motion.
5. Verify the collection of a clip and that it can be played locally.
6. While a clip is being recorded, use the accessory app to modify the supported video configuration (change from 1080p to 720p).
7. Verify that the recording session does not immediately terminate, as the clip recording in progress must continue.
8. Cover the accessory lens for 10s to cease recordings of significant clips.
9. Uncover the accessory.
10. Verify that all new clips succeed and are recorded at the newly selected resolution.
11. Remove all pairings from the accessory via HAT and iOS.

TCR039 **If a previously advertised audio or video recording configuration becomes unsupported (e.g., after a user-triggered configuration change via the accessory app), the accessory shall locally discard the Selected Recording Configuration if it is no longer compatible for the next recording session. Do not discard the Selected Recording Configuration if a session is currently active (i.e., the accessory is still sending fragments with the previous configuration for the current session). In this case, the recording currently in progress shall continue using the previous parameters.**

Applies to accessories that support Camera Event Recording. Applies to camera accessories with accessory apps that provide the ability to change the Supported Video and/or Audio Recording Configuration(s), but cannot be paired to HAT at the same time. Perform this test case using an iOS device running the accessory app and the Home app.

1. Add accessory to the Home app on an iOS device signed into iCloud, and verify the accessory can successfully display a live stream.
2. Enable Home Hub on a supported iPad or Apple TV, or use a HomePod, running the latest version of iOS using the same iCloud account.
3. Using the Home app, change the accessory settings to enable both streaming and recording for all motion.
4. Perform action to create Motion and/or Doorbell events, causing recording to begin.
5. Verify that the iOS device receives a notification for the event.
6. Stop creating Motion and/or Doorbell events, causing recording to stop.
7. Verify the clip is available in the Home app.
8. Download the clip, and notate the parameters used by using the Inspector feature in QuickTime Player or similar tool.
9. Perform actions to create Motion and/or Doorbell events, causing recording to begin.
10. Verify that the iOS device receives a notification for the event.
11. While continuously performing actions from step 9, use the accessory app or alternative method to change the video recording settings to no longer allow the previously used parameters for recordings.
12. Wait 10 seconds, then stop creating Motion and/or Doorbell events, causing recording to stop.
13. Verify the clip is available in the Home app, and was not interrupted by the settings change in step 11. e.g. recording continued throughout the settings change without interruption.
14. Download the clip, and verify the parameters used match those of step 8 by using the Inspector feature in QuickTime Player or similar tool.
15. Perform action to create Motion and/or Doorbell events, causing recording to begin.
16. Verify that the iOS device receives a notification for the event.
17. Stop creating Motion and/or Doorbell events, causing recording to stop.
18. Download the clip, and verify the new parameters from step 11 were used by using the Inspector feature in QuickTime Player or similar tool.

TCR040 **Verify the values for the Event Snapshot Active, HomeKit Camera Active, and Third Party Camera Active (if applicable) characteristics on the Camera Operating Mode service are correctly set by default, after reboot after factory reset, and after pairings are removed.**

Applies to accessories that support Camera Event Recording. Perform this test case with HAT using the steps below.

1. Perform Pair Setup adding the accessory to HAT.
2. Discover the accessory's services and characteristics.
3. Add the accessory to the accessory app.
4. Verify the (default) value of the Event Snapshot Active characteristic is "1" (Enable snapshots).
5. Verify the (default) value of the Periodic Snapshot Active characteristic is "1" (Enable snapshots).
6. Verify the (default) value of the HomeKit Camera Active characteristic is "1" (On).
7. Verify the (default) value of the Third Party Camera Active characteristic is "1" (On).
8. Reboot the accessory.
9. Verify the values of steps 4-7 persist.
10. Set the value of the Event Snapshot Active characteristic to "0" (Disable snapshots).
11. Set the value of the Periodic Snapshot Active characteristic to "0" (Disable snapshots).
12. Set the value of the HomeKit Camera Active characteristic to "0" (Off).
13. Using the accessory app, set the value of the Third Party Camera Active characteristic to "0" (Off).
14. Reboot the accessory.
15. Verify the value of the Event Snapshot Active characteristic is "0" (Disable snapshots).
16. Verify the value of the Periodic Snapshot Active characteristic is "0" (Disable snapshots).
17. Verify the value of the HomeKit Camera Active characteristic is "0" (Off).
18. Verify the value of the Third Party Camera Active characteristic is "0" (Off).
19. Reboot the accessory.
20. Cleanly remove the HomeKit pairings from the accessory in HAT by choosing Remove Pairing under Pairing.
21. Perform Pair Setup adding the IP Camera to HAT.
22. Discover the accessory's services and characteristics.
23. Verify the (default) value of the Event Snapshot Active characteristic is "1" (Enable snapshots).
24. Verify the (default) value of the Periodic Snapshot Active characteristic is "1" (Enable snapshots).
25. Verify the (default) value of the HomeKit Camera Active characteristic is "1" (On).
26. Verify the (default) value of the Third Party Camera Active characteristic is "0" (Off).
27. Remove all pairings from the accessory via HAT.

TCR041 **Verify the values of the Active characteristic for the Camera Event Recording Management service are correctly set by default, after reboot, after factory reset, and after pairings are removed.**

Applies to accessories that support Camera Event Recording. Perform this test case with HAT using the steps below.

1. Perform Pair Setup adding the accessory to HAT.
2. Discover the accessory's services and characteristics.
3. Verify the (default) value of the Active characteristic associated to the Camera Event Recording Management Service is "0" (Inactive).
4. Reboot the accessory.
5. Verify the values persist.
6. Write "1" (Active) to the the Active characteristic associated to the Camera Event Recording Management service.
7. Verify the value of the Active is set to "1" (Active).
8. Cleanly remove the HomeKit pairings from the accessory in HAT by choosing Remove Pairing under Pairing.
9. Perform Pair Setup adding the accessory to HAT.
10. Discover the accessory's services and characteristics.
11. Verify the (default) value of the Active characteristic associated to the Camera Event Recording Management Service is "0" (Inactive).
12. Remove all pairings from the accessory via HAT.
13. Factory reset the accessory.
14. Perform Pair Setup adding the accessory to HAT.
15. Discover the accessory's services and characteristics.
16. Verify the (default) value of the Active characteristic associated to the Camera Event Recording Management Service is "0" (Inactive).
17. Remove all pairings from the accessory via HAT.

TCR042 **Verify that when the accessory is already sending video content, it responds to the new Start Request message with a Busy status in the Start Response message.**

Applies to accessories that support Camera Event Recording. Perform this test case with HAT using the steps below.

1. Perform Pair Setup adding the accessory to HAT.
2. Discover the accessory's services and characteristics.
3. Create an additional controller.
4. Using Controller 1, add the secondary controller pairing as admin.
5. Set the HomeKit Camera Active characteristic, and the Active characteristic associated to the Camera Event Recording Management service to "1" (On/Active) with a timed write.
6. Read the Supported Audio Recording Configuration characteristic. In the Details view of the Characteristic Read Completed, note the bitRate under the audioCodecRecordingParameters.

7. Configure the accessory to record by using the Build TLV button associated with the Selected Camera Recording Configuration characteristic, and using a 4000msec prebuffer duration, 4000msec fragment duration, 800kbps video bit rate, 4000 i-Frame Rate Interval, and the bitRate value from the read audioCodecRecordingParameters.
8. Select the "Write" button in the Write TLV panel.
9. Using HAT Controller 1, navigate to the Data Stream Transport Management service and select Send Start Command, then select Connect.
10. Enter the same integer in both Stream ID fields under Camera Recording.
11. Select Send Start Request.
12. Using the HDS view in the trace, verify that the accessory responds with Status: 0 (Success) and begins to transfer data via HDS.
13. Using Controller 2, discover the accessory's services and characteristics and navigate to the Data Stream Transport Management service and select Send Start Command, then select Connect.
14. Enter in the same integer in both Stream ID fields under Camera Recording.
15. Select Send Start Request.
16. Using the HDS view in the trace, verify that the accessory responds with Status: 2 (Busy / max number of transport sessions reached) and that the accessory continues to only send data via HDS to Controller 1.
17. Remove all pairings from the accessory via HAT.

TCR045 **Verify that the accessory returns HAP error -70402 when the controller attempts to read the value of Selected Recording Configuration before anything has been written to the Selected Recording Configuration.**

Applies to accessories that support Camera Event Recording. Perform this test case with HAT using the steps below.

1. Perform Pair Setup adding the IP Camera to HAT.
2. Discover the accessory's services and characteristics.
3. Set the HomeKit Camera Active characteristic, and the Active characteristic associated to the Camera Event Recording Management service to "1" (On) with a timed write.
4. Select Selected Camera Recording Configuration.
5. Read the value of the characteristic.
6. Verify that the accessory returns HAP error status -70402.
7. Remove all pairings from the accessory via HAT.

TCR046 **Verify that the prebuffer length is less than or equal to the value reported by the accessory in the Supported Camera Recording Configuration characteristic.**

Applies to accessories that support Camera Event Recording. Perform this test case with HAT using the steps below.

1. Perform Pair Setup adding the accessory to HAT.
2. Discover the accessory's services and characteristics.
3. Set the HomeKit Camera Active characteristic in the Camera Operating Mode service to "1" (On) with a timed write.
4. Set the Active characteristic associated to the Camera Event Recording Management service to "1" (Active) with a timed write.
5. Navigate to the Supported Camera Recording Configuration characteristic, and read the value.
6. Using the Events view in the trace, inspect the read response details and verify that the PreBuffer Duration is at least 4000msec.
7. Read the Supported Audio Recording Configuration characteristic. In the Details view of the Characteristic Read Completed, note the bitRate under the audioCodecRecordingParameters.
8. Configure the accessory to record by using the Build TLV button associated with the Selected Camera Recording Configuration characteristic, and using a 4000msec prebuffer duration, 4000msec fragment duration, 800kbps video bit rate, 4000 i-Frame Rate Interval, and the bitRate value from the read audioCodecRecordingParameters.
9. Select the "Write" button in the Write TLV panel.
10. Navigate to the Data Stream Transport Management service and select Send Start Command, then choose Connect.
11. Enter the same integer in both Stream ID fields under Camera Recording.
12. Point the accessory at a stopwatch.
13. Start the stopwatch and let it run for 10 seconds.
14. Select Send Start Request.
15. Wait 10 seconds.
16. Select Send Close Event.
17. Save, open, and view the recording and verify that it is playable.
18. Verify that the length of the Prebuffer Duration in the saved video is 4000ms from before the Send Start Request was sent.
19. Remove all pairings from the accessory via HAT.

TCR047 **Verify that the accessory responds with HAP Status Code -70410 when the controller attempts to write an incomplete value (tlv) to the Selected Camera Recording Configuration.**

Applies to accessories that support Camera Event Recording. Perform this test case with HAT using the steps below.

1. Perform Pair Setup adding the IP Camera to HAT.
2. Discover the accessory's services and characteristics.
3. Set the HomeKit Camera Active characteristic, and the Active characteristic associated to the Camera Event Recording Management service to "1" (On) with a timed write.
4. Navigate to the Selected Camera Recording Configuration characteristic within the Camera Event Recording Management service. Attempt to incorrectly configure the accessory to record by leaving some of fields blank and then select the Build TLV button.
5. Verify that the accessory responds with HAP Status Code -70410.
6. Remove all pairings from the accessory via HAT.

TCR048 **Verify that the value written to the recording configuration persists across reboots of the accessory.**

Applies to accessories that support Camera Event Recording. Perform this test case with HAT using the steps below.

1. Perform Pair Setup adding the accessory to HAT.
2. Discover the accessory's services and characteristics.
3. Set the HomeKit Camera Active characteristic to "1" (On) with a timed write.
4. Set the Active characteristic associated to the Camera Event Recording Management service to "1" (On) with a timed write.
5. Read the Supported Audio Recording Configuration characteristic. In the Details view of the Characteristic Read Completed, note the bitRate under the audioCodecRecordingParameters.
6. Configure the accessory to record by using the Build TLV button associated with the Selected Camera Recording Configuration characteristic, and using a 4000msec prebuffer duration, 4000msec fragment duration, 800kbps video bit rate, 4000 i-Frame Rate Interval, and the bitRate value from the read audioCodecRecordingParameters.
7. Select the "Write" button in the Write TLV panel.
8. Select Read on the value of the TLV of the Selected Camera Recording Configuration characteristic.
9. Note the value of the TLV.
10. Reboot the accessory.
11. When the accessory is back online discover its services and characteristics.
12. Navigate to the Selected Camera Recording Configuration characteristic.
13. Select Read on the value of the TLV of the Selected Camera Recording Configuration characteristic.
14. Verify that the value of the TLV is the same as it was before the reboot.

15. Remove all pairings from the accessory via HAT.

TCR055 **Verify that the accessory responds with HTTP Status Code 207 Multi-Status indicating HAP Status Code -70401 when a controller attempts to write to the Event Snapshot Active characteristic if the Admin bit is not set to 1.**

Applies to accessories that support Camera Event Recording. Perform this test case with HAT using the steps below.

1. Perform Pair Setup adding the accessory to HAT.
2. Discover the accessory's services and characteristics.
3. Write any supported value to the Event Snapshots Active.
4. Add an additional IP controller.
5. Add the new controller as a non-admin controller.
6. Write any supported value to the Event Snapshots Active from the non-Admin controller.
7. Verify that the accessory rejects it with HTTP Status Code 207 Multi-Status indicating HAP Status Code -70401 (Request denied due to insufficient privileges).
8. Remove all pairings from the accessory via HAT.

TCR056 **Verify that the accessory responds with HTTP Status Code 207 Multi-Status indicating HAP Status Code -70401 when a controller attempts to write to the HomeKit Camera Active characteristic if the Admin bit is not set to 1.**

Applies to accessories that support Camera Event Recording. Perform this test case with HAT using the steps below.

1. Perform Pair Setup adding the accessory to HAT.
2. Discover the accessory's services and characteristics.
3. Write any supported value to the HomeKit Camera Active.
4. Add an additional IP controller.
5. Add the new controller as a non-admin controller.
6. Write any supported value to the HomeKit Camera Active from the non-Admin controller.
7. Verify that the accessory rejects it with HTTP Status Code 207 Multi-Status indicating HAP Status Code -70401 (Request denied due to insufficient privileges).
8. Remove all pairings from the accessory via HAT.

TCR057 **Verify that the accessory responds with HTTP Status Code 207 Multi-Status indicating HAP Status Code -70401 when a controller attempts to write to the Camera Operating Mode Indicator characteristic if the Admin bit is not set to 1.**

Applies to accessories that support Camera Event Recording. Perform this test case with HAT using the steps below.

1. Perform Pair Setup adding the accessory to HAT.
2. Discover the accessory's services and characteristics.
3. Write any supported value to the Camera Operating Mode Indicator.
4. Add an additional IP controller.
5. Add the new controller as a non-admin controller.
6. Write any supported value to the Camera Operating Mode Indicator from the non-Admin controller.
7. Verify that the accessory rejects it with HTTP Status Code 207 Multi-Status indicating HAP Status Code -70401 (Request denied due to insufficient privileges).
8. Remove all pairings from the accessory via HAT.

TCR058 **Verify that the selected recording configuration persists when changing the state of the characteristic to inactive/active.**

Applies to accessories that support Camera Event Recording. Perform this test case with HAT using the steps below.

1. Perform Pair Setup adding the accessory to HAT.
2. Discover the accessory's services and characteristics.
3. Set the HomeKit Camera Active characteristic in the Camera Operating Mode service to "1" (On) with a timed write.
4. Set the Active characteristic associated to the Camera Event Recording Management service to "1" (Active) with a timed write.
5. Read the Supported Audio Recording Configuration characteristic. In the Details view of the Characteristic Read Completed, note the bitRate under the audioCodecRecordingParameters.
6. Configure the accessory to record by using the Build TLV button associated with the Selected Camera Recording Configuration characteristic, and using a 4000msec prebuffer duration, 4000msec fragment duration, 800kbps video bit rate, 4000 i-Frame Rate Interval, and the bitRate value from the read audioCodecRecordingParameters.
7. Select the "Write" button in the Write TLV panel.
8. Select Read on the value of the TLV of the Selected Camera Recording Configuration characteristic.
9. Note the value.
10. Set all Active characteristics associated to the Camera Event Recording Management service and the Camera RTP Stream Management service to "0" (Inactive) with a timed write.

11. Set all Active characteristics associated to the Camera Event Recording Management service and the Camera RTP Stream Management service to "1" (Active) with a timed write.
12. Select Read on the value of the TLV of the Selected Camera Recording Configuration characteristic.
13. Verify that the value of the TLV is the same as in step 5.
14. Remove all pairings from the accessory via HAT.

TCR059 **Verify that the Active characteristics associated to the Camera RTP Stream Management service, and Camera Event Recording Management service retain their value when HomeKit Camera Active is set to Off.**

Applies to accessories that support Camera Event Recording. Perform this test case with HAT using the steps below.

1. Perform Pair Setup adding the accessory to HAT.
2. Discover the accessory's services and characteristics.
3. Set the HomeKit Camera Active characteristic in the Camera Operating Mode service to "1" (On) with a timed write.
4. Set the Active characteristic associated to the Camera Event Recording Management service to "1" (Active) with a timed write.
5. Read the Supported Audio Recording Configuration characteristic. In the Details view of the Characteristic Read Completed, note the bitRate under the audioCodecRecordingParameters.
6. Configure the accessory to record by using the Build TLV button associated with the Selected Camera Recording Configuration characteristic, and using a 4000msec prebuffer duration, 4000msec fragment duration, 800kbps video bit rate, 4000 i-Frame Rate Interval, and the bitRate value from the read audioCodecRecordingParameters.
7. Select the "Write" button in the Write TLV panel.
8. Select Read on the value of the TLV of the Selected Camera Recording Configuration characteristic.
9. Set all Active characteristics associated to the Camera Event Recording Management Service and Camera RTP Stream Management Service to "1" (Active) with a timed write.
10. Navigate to the HomeKit Camera Active characteristic.
11. Write "0" (Off) to HomeKit Camera Active.
12. Write "1" (On) to HomeKit Camera Active.
13. Navigate to the Active characteristic associated with the Camera RTP Stream Management service(s).
14. Select Read.
15. Verify the value returned from the accessory is "1" (Active).
16. Navigate to the Active characteristic associated with Camera Event Recording Management service.
17. Select Read.

18. Verify that the value returned from the accessory is "1" (Active).
19. Remove all pairings from the accessory via HAT.

TCR060 **Verify that the Active characteristic associated to the Camera RTP Stream Management, and Camera Event Recording Management services retains their values when Third Party Camera Active is set to Off.**

Applies to accessories that support Camera Event Recording. Perform this test case using HAT and an iOS device running the accessory app.

1. Add the accessory in the accessory app (outside of HomeKit).
2. Perform Pair Setup adding the accessory to HAT.
3. Discover the accessory's services and characteristics.
4. Read the Supported Audio Recording Configuration characteristic. In the Details view of the Characteristic Read Completed, note the bitRate under the audioCodecRecordingParameters.
5. Configure the accessory to record by using the Build TLV button associated with the Selected Camera Recording Configuration characteristic, and using a 4000msec pre-buffer duration, 4000msec fragment duration, 800kbps video bit rate, 4000 i-Frame Rate Interval, and the bitRate value from the read audioCodecRecordingParameters.
6. Select the "Write" button in the Write TLV panel.
7. Navigate to the Active characteristic associated with Camera RTP Stream Management service and set it to "0".
8. Navigate to the Active characteristic associated with Camera Event Recording service and set it to "1".
9. Launch the accessory app and navigate to the Third Party Camera Active characteristic and set it to "0".
10. In HAT read the Active characteristic associated to Camera RTP Stream Management.
11. Verify that the value returned by the accessory is "0" (Inactive).
12. Navigate to the Active characteristic associated to Camera Event Recording Management service.
13. Read the value of the Active characteristic associated to Camera Event Recording Management service.
14. Verify that the value returned by the accessory is "1" (Active).
15. Remove all pairings from the accessory via HAT and iOS.

TCR061 **Verify that the accessory rejects any start request over HDS if the HomeKit Camera Active characteristic or Active characteristic for Camera Event Recording Management service is set to "0" (Off) by responding to the dataSend.open request with the Protocol Specific Error, "Not Allowed". If the accessory is actively sending HDS data to the controller, and the HomeKit Camera Active characteristic or Active characteristic for Camera Event Recording Management service is set to "0" (Off), the accessory shall send a Close Event with the reason key set to 1.**

Applies to accessories that support Camera Event Recording. Perform this test case with HAT using the steps below.

1. Perform Pair Setup adding the accessory to HAT.
2. Discover the accessory's services and characteristics
3. Set the HomeKit Camera Active characteristic in the Camera Operating Mode service to "1" (On) with a timed write.
4. Set the Active characteristic associated to the Camera Event Recording Management service to "1" (Active) with a timed write.
5. Select Camera Event Recording Management.
6. Read the Supported Audio Recording Configuration characteristic. In the Details view of the Characteristic Read Completed, note the bitRate under the audioCodecRecordingParameters.
7. Configure the accessory to record by using the Build TLV button associated with the Selected Camera Recording Configuration characteristic, and using a 4000msec prebuffer duration, 4000msec fragment duration, 800kbps video bit rate, 4000 i-Frame Rate Interval, and the bitRate value from the read audioCodecRecordingParameters.
8. Select the "Write" button in the Write TLV panel.
9. Navigate to the HomeKit Camera Active characteristic.
10. Write "0" (Inactive) to the HomeKit Camera Active characteristic.
11. Navigate to the Data Stream Transport Management service and select Send Start Command, and then select Connect.
12. Enter the value "1" in both Stream ID fields under Camera Recording and select Send Start Request.
13. Using the HomeKit Data Stream trace view, verify that the accessory responds to the Start Data Stream request with a response that has a header that includes "dataSend" as the protocol, "Open" as the topic, "6" as the "Status" set (Protocol Specific Error), an "ID" that matches the ID of the Start Data Stream request, and a message with "1" as the "Status" set (Not Allowed).
14. Navigate to the HomeKit Camera Active characteristic.
15. Write "1" (Active) to the HomeKit Camera Active characteristic.
16. Navigate to the Active characteristic associated to the Camera Event Recording Management service.
17. Write "0" (Inactive) to the Active characteristic.
18. Navigate to the Data Stream Transport Management service and select Send Start Command.
19. Enter the value "2" in both Stream ID fields under Camera Recording and select Send Start Request.
20. Using the HomeKit Data Stream trace view, verify the accessory responds to the Start Data Stream request with a response that has a header that includes "dataSend" as the protocol, "Open" as the topic, "Status" set to "6" (Protocol Specific Error), "ID" that matches the ID of the Start Data Stream request, and message with "Status" set to "1" (Not Allowed).
21. Navigate to the Active characteristic associated to the Camera Event Recording Management service.

22. Write “1” (Active) to the Active characteristic.
23. Navigate to the Data Stream Transport Management service, select Send Start Command, then select Connect.
24. Enter the value “3” in both Stream ID fields under Camera Recording.
25. Select Send Start Request.
26. After the accessory begins sending data over HDS, navigate to the HomeKit Camera Active characteristic.
27. Write “0” (Inactive) to the HomeKit Camera Active characteristic.
28. Using the HDS view in the trace, verify that the accessory sends a Close Event with the reason set to “1”.
29. Write “1” (Active) to the HomeKit Camera Active characteristic.
30. Navigate to the Data Stream Transport Management service.
31. Enter the value “4” in both Stream ID fields under Camera Recording and select Send Start Request.
32. After the accessory begins sending data over HDS, navigate to the Active characteristic associated to the Camera Event Recording Management service.
33. Write “0” (Inactive) to the Active characteristic.
34. Using the HDS view in the trace, verify accessory sends a Close Event with the reason set to “1”.
35. Write “1” (Active) to the Active characteristic.
36. Navigate to the Data Stream Transport Management service.
37. Enter the value “5” in both Stream ID fields under Camera Recording and select Send Start Request.
38. Wait 5 seconds.
39. Select Send Close Event.
40. Select Save Recording.
41. Open and view the recording and verify that it is playable.
42. Remove all pairings from the accessory via HAT.

TCR063 **Verify that if the accessory exposes a Sensor service, such as a Motion Sensor or Occupancy sensor service, that it disables the sensor when HomeKit Camera Active is set to Off by setting the Status Active characteristic on the corresponding Sensor Service to False.**

Applies to accessories that support Camera Event Recording. Perform this test case with HAT using the steps below.

1. Perform Pair Setup adding the accessory to HAT.
2. Discover the accessory’s services and characteristics.

3. Navigate to any Status Active characteristics associated with the Motion Sensor and/or Occupancy Sensor service.
4. Read the characteristic and verify the value is “1” (Active).
5. Navigate to HomeKit Camera Active characteristic.
6. Write “0” (Off) to HomeKit Camera Active.
7. Navigate to all Motion Detected and/or Occupancy Detected characteristic(s), and enable Event Notifications.
8. Navigate to any Status Active characteristic(s) associated with the Motion Sensor and/or Occupancy Sensor service(s).
9. Choose read.
10. Verify that the accessory returns a ‘0” (Inactive) for each ‘Status Active’’ characteristic.
11. Cause a motion event in front of the accessory.
12. Verify no Motion or Occupancy event notifications are received.
13. Remove all pairings from the accessory via HAT.

TCR066 **Verify that the accessory responds with HTTP Status Code 207 Multi-Status indicating HAP Status Code -70412 when HomeKit Camera Active is set to Off and a controller issues a snapshot request.**

Applies to accessories that support Camera Event Recording. Perform this test case with HAT using the steps below.

1. Perform Pair Setup adding the accessory to HAT.
2. Discover the accessory’s services and characteristics.
3. Set the HomeKit Camera Active characteristic to “0” (Off).
4. Navigate to the Camera RTP Stream Management Service.
5. Select the initial resolution of any supported resolution and fps.
6. Choose Take Snapshot.
7. Verify that the accessory responds with HTTP Status Code 207 Multi-Status indicating HAP Status Code -70412.
8. Remove all pairings from the accessory via HAT.

TCR067 **Verify that the accessory responds to any read/write to the Setup Endpoints request with HTTP 207 Multi-Status response including HAP Status Code -70412 when the Active characteristic for the Camera RTP Stream Management service is set to false.**

Applies to accessories that support Camera Event Recording. Perform this test case with HAT using the steps below.

1. Perform Pair Setup adding the accessory to HAT.
2. Discover the accessory's services and characteristics.
3. Set the Active characteristic associated to the Camera RTP Stream Management service to "0" (Inactive) with a timed write.
4. Navigate to the Setup Endpoints characteristic associated with the same Camera RTP Stream Management Service.
5. Read the value of the Setup Endpoints characteristic.
6. Verify that the accessory with HTTP 207 Multi-Status response including HAP Status Code- 70412.
7. Navigate to the Camera RTP Stream Management Service.
8. Choose Negotiate.
9. Verify that the accessory with HTTP 207 Multi-Status response including HAP Status Code- 70412.
10. Remove all pairings from the accessory via HAT.

TCR068 **Verify that when HomeKit Camera Active is set to Inactive, that the accessory responds to any read/write to the Setup Endpoints characteristic with HTTP 207 Multi-Status response including HAP Status Code -70412.**

Applies to accessories that support Camera Event Recording. Perform this test case with HAT using the steps below.

1. Perform Pair Setup adding the accessory to HAT.
2. Discover the accessory's services and characteristics.
3. Navigate to the HomeKit Camera Active characteristic.
4. Write "0" (Inactive) to the above characteristic.
5. Navigate to the Camera RTP Stream Management Service.
6. Choose Negotiate.
7. Verify that the accessory with HTTP 207 Multi-Status response including HAP Status Code -70412.
8. Navigate to the Setup Endpoints characteristic associated with the same Camera RTP Stream Management Service.
9. Read the value of the Setup Endpoints characteristic.
10. Verify that the accessory with HTTP 207 Multi-Status response including HAP Status Code -70412.
11. Remove all pairings from the accessory via HAT.

TCR076 **Request an unencrypted snapshot while recording and streaming are ongoing.**

Applies to accessories that support Camera Event Recording. Perform this test case with HAT using the steps below.

1. Perform Pair Setup, adding the accessory to HAT.
2. Discover the accessory's services and characteristics.
3. Navigate to the Camera RTP Stream Management service on the first controller.
4. Select the "Select Stream Parameters" button, select supported parameters, then select "Configure".
5. Select"Negotiate", then select "Start Streaming".
6. Set the Active characteristic associated to the Camera Event Recording Management service to "1" (Active) with a timed write.
7. Read the Supported Audio Recording Configuration characteristic. In the Details view of the Characteristic Read Completed, note the bitRate under the audioCodecRecordingParameters.
8. Configure the accessory to record by using the Build TLV button associated with the Selected Camera Recording Configuration characteristic. Use a 4000msec prebuffer duration, 4000msec fragment duration, 800kbps video bit rate, 4000 i-Frame Rate Interval, and the bitRate value from the read audioCodecRecordingParameters.
9. Select the "Write" button in the Write TLV panel.
10. Navigate to the Data Stream Transport Management service and select Send Start Command, then choose Connect.
11. Enter the same integer in both Stream ID fields under Camera Recording.
12. Select Send Start Request.
13. Add a second controller to the accessory that is a non-admin controller and discover the services and characteristics.
14. Navigate to the Camera RTP Stream Management service.
15. Select Take Unsecured Snapshot.
16. Verify that the streaming is uninterrupted on Controller 1's live stream.
17. Select Send Close Event on Controller 1.
18. Select Save Recording.
19. Open and view the recording and verify that it is playable.
20. Remove all pairings from the accessory via HAT.

TCR079 **Verify the required Audio Bit Rate configurations are supported.**

Applies to accessories that support Camera Event Recording. Perform this test case with HAT using the steps below.

1. Perform Pair Setup adding the accessory to HAT.
2. Discover the accessory's services and characteristics.
3. Navigate to the Supported Audio Recording Configuration characteristic associated with the Camera Event Recording Management Service.

4. Read the value of the Supported Audio Recording Configuration characteristic.
5. Inspect the Events view in the HAT Trace.
6. Verify values listed in the parsed TLV are valid.
7. Remove all pairings from the accessory via HAT.

TCR080 **Verify that the Periodic Snapshots Active characteristic controls the accessory's ability to capture and deliver snapshots for motion and doorbell notifications.**

Applies to accessories that support Camera Event Recording. Perform this test case with HAT using the steps below.

1. Perform Pair Setup adding the accessory to HAT.
2. Discover the accessory's services and characteristics.
3. Set the Periodic Snapshots Active characteristic to "0" (Disable Snapshots).
4. Read the value of the Periodic Snapshots Active characteristic.
5. Verify the value returned is "0" (Disable Snapshots).
6. Navigate to the Camera RTP Stream Management Service.
7. Choose Take Snapshot.
8. Verify that the snapshot is rejected with HTTP Status Code 207 Multi-Status indicating HAP Status Code -70412 (Not allowed in the current state).
9. Set the Periodic Snapshots Active characteristic to "1" (Enable Snapshots).
10. Read the value of the Periodic Snapshots Active characteristic.
11. Verify the value returned is "1" (Enable Snapshots).
12. Navigate to the Camera RTP Stream Management Service.
13. Choose Take Snapshot.
14. Verify that the snapshot is received and can be viewed.
15. Remove all pairings from the accessory via HAT.

TCR082 **Verify that the accessory responds with HTTP Status Code 207 Multi-Status indicating HAP Status Code -70412 when Periodic Snapshots Active is set to "False" and a controller issues a snapshot request.**

Applies to accessories that support Camera Event Recording. Perform this test case with HAT using the steps below.

1. Perform Pair Setup adding the accessory to HAT.
2. Discover the accessory's services and characteristics.
3. Set the Periodic Snapshots Active characteristic to "0" (Disable Snapshots).

4. Navigate to the Camera RTP Stream Management Service.
5. Choose Take Snapshot.
6. Verify that the accessory responds with HTTP Status Code 207 Multi-Status indicating HAP Status Code -70412
7. For accessories that use the Doorbell service, enable event notifications on the Programmable Switch Event characteristic.
8. For accessories that use the Motion Sensor service, enable event notifications on the Motion Detected characteristic.
9. For doorbells, press doorbell button.
10. Verify that a notification is generated in the Events Traffic View.
11. For accessory, cause motion in front of the accessory.
12. Verify that a notification is generated in the Events Traffic View.
13. Navigate to the Camera RTP Stream Management Service and choose Take Snapshot.
14. Verify that the accessory responds with HTTP Status Code 207 Multi-Status indicating HAP Status Code -70412.
15. Remove all pairings from the accessory via HAT.

TCR083 **Verify that the accessory responds with HTTP Status Code 207 Multi-Status indicating HAP Status Code -70401 when a controller attempts to write to the Periodic Snapshot Active characteristic if the Admin bit is not set to 1.**

Applies to accessories that support Camera Event Recording. Perform this test case with HAT using the steps below.

1. Perform Pair Setup adding the accessory to HAT.
2. Discover the accessory's services and characteristics.
3. Write any supported value to the Periodic Snapshots Active.
4. Add an additional IP controller.
5. Add the new controller as a non-admin controller.
6. Write any supported value to the Periodic Snapshots Active from the non-Admin controller that is different than its current value.
7. Verify that the accessory rejects it with HTTP Status Code 207 Multi-Status indicating HAP Status Code -70401 (Request denied due to insufficient privileges).
8. Read the Periodic Snapshot Active characteristic did not change.
9. Remove all pairings from the accessory via HAT.

TCR085 **Verify the accessory advertises the Timed Write permission for the Active characteristics on the Camera RTP Stream Management and Camera Event Recording Management services.**

Applies to accessories that support Camera Event Recording. Perform this test case with HAT using the steps below.

1. Perform Pair Setup adding the accessory to HAT.
2. Discover the accessory's services and characteristics.
3. Verify the presence of the Active characteristic on the Camera RTP Stream Management and Camera Event Recording Management services.
4. Verify the Active characteristics advertise the Timed Write permission.
5. Remove all pairings from the accessory via HAT.

TCR086 **Verify that the accessory accepts reasons of 0 or 1 for Event Snapshot Active and the accessory returns the a valid snapshot for each respective reason.**

Applies to accessories that support Camera Event Recording. Perform this test case with HAT using the steps below.

1. Perform Pair Setup adding the accessory to HAT.
2. Discover the accessory's services and characteristics.
3. Navigate to the Camera RTP Stream Management Service.
4. Set the Active characteristic(s) associated to the Camera RTP Stream Management to "1" (Active).
5. Select the initial resolution of any supported resolution and fps.
6. Choose Take Snapshot and specify reason as "1".
7. Verify that a snapshot is received.
8. Choose Take Snapshot and specify reason as "0".
9. Verify that a snapshot is received.
10. Remove all pairings from the accessory via HAT.

TCR087 **Verify that the accessory responds with HTTP Status Code 207 Multi-Status indicating HAP Status Code -70412 when Periodic Snapshots Active is set to False and a controller issues a snapshot request.**

Applies to accessories that support Camera Event Recording. Perform this test case with HAT using the steps below.

1. Perform Pair Setup adding the accessory to HAT.
2. Discover the accessory's services and characteristics.
3. Set the Event Snapshots Active characteristic to "1" (Enable Snapshots).

4. Set the Periodic Snapshots Active characteristic to “0” (Disable).
5. Navigate to the Camera RTP Stream Management Service.
6. Choose Take Snapshot.
7. Verify that the accessory responds with HTTP Status Code 207 Multi-Status indicating HAP Status Code -70412.
8. Specify reason “0”, and choose Take Snapshot.
9. Verify that the accessory responds with HTTP Status Code 207 Multi-Status indicating HAP Status Code -70412.
10. Specify reason “1”, and choose Take Snapshot.
11. Verify this succeeds.
12. Remove all pairings from the accessory via HAT.

TCR089 **Verify the correct format, permissions, and valid values, for the Manually Disabled characteristic.**

Applies to accessories that support Camera Event Recording. Perform this test case with HAT using the steps below.

1. Perform Pair Setup adding the accessory to HAT.
2. Verify the presence of the Manually Disabled characteristic (UUID 00000227).
3. Navigate to the Manually Disabled characteristic.
4. Verify the permissions are exactly set to the following: Paired Read and Notify
5. If the accessory supports reenabling of the hardware switch via a software mechanism, verify the permissions are exactly set to the following: Paired Read, Paired Write, and Notify.
6. Verify the format is listed as uint8.
7. Verify the valid values are: 0 (Manually Enabled) and 1 (Manually Disabled)
8. Manually switch the the accessory to “Disabled”.
9. Read the characteristic and verify it returns “1”.
10. Manually switch the the accessory to “Enabled”.
11. Read the characteristic and verify it returns “0”.
12. Remove all pairings from the accessory via HAT.

TCR090 **Verify that if accessory is turned off manually (e.g via a physical button on the accessory), that it overrides both HomeKit and Third Party Camera active.**

Applies to accessories that support Camera Event Recording. Perform this test case using HAT and an iOS device running the accessory app.

1. Perform Pair Setup adding the accessory to HAT.
2. Discover the accessory's services and characteristics.
3. Set the Periodic Snapshots Active characteristic to "1" (Enable).
4. Manually switch the the accessory to Disabled via the hardware switch.
5. Read the characteristic for Manually Disabled' and verify it returns "1".
6. Read the characteristic for Third Party Active' and verify it returns "0".
7. Verify that the accessory responds to any read/write to the Setup Endpoints characteristic and the Selected RTP Stream Configuration characteristic with HTTP 207 Multi-Status response including HAP Status Code-70412 (Not allowed in the current state).
8. Navigate to the Camera RTP Stream Management Service.
9. Choose Take Snapshot and specify reason as "1".
10. Verify that the accessory responds with HTTP Status Code 207 Multi-Status indicating HAP Status Code -70412 (Not allowed in the current state).
11. Verify the streaming and recording in the accessory app is still prevented.
12. In HAT Enable HomeKit Camera Active.
13. Verify streaming and recording is still prevent via HAP.
14. Manually switch the the accessory to "Enabled".
15. Read the characteristic and verify it returns "0".
16. Using the accessory app Enable Third Party Active.
17. Verify the streaming and recording in the accessory app is working.
18. Verify streaming and recording is working via HAP.
19. Remove all pairings from the accessory via HAT.

TCR091 **Verify that if the accessory can be re-enabled via software, the accessory must advertise Paired Write for the Manually Disabled characteristic.**

Applies to accessories that support Camera Event Recording. Perform this test case with HAT using the steps below.

1. Perform Pair Setup adding the accessory to HAT.
2. Discover the accessory's services and characteristics.
3. Manually switch the accessory to "Disabled".
4. Read the characteristic and verify it returns "1".
5. If the characteristic supports Paired Write, use HAT to perform a write operation to switch the the accessory to Enabled.

6. Read the characteristic and verify it returns “0”.
7. Verify that the accessory advertises the Paired Write permission for the Manually Disabled Characteristic.
8. Verify streaming and recording is working via HAT.
9. Remove all pairings from the accessory via HAT.

TCR093 **The accessory must not include audio in the mp4 fragments sent over HDS when the Recording Audio Active characteristic is set to 0 (Disable audio in recordings).**

Applies to accessories that support Camera Event Recording. Perform this test case with HAT using the steps below.

1. Perform Pair Setup adding the accessory to HAT.
2. Discover the accessory’s services and characteristics.
3. Set the HomeKit Camera Active characteristic in the Camera Operating Mode service to “1” (On) with a timed write.
4. Set the Active characteristic associated to the Camera Event Recording Management service to “1” (Active) with a timed write.
5. Navigate to the Recording Audio Active characteristic.
6. Set the Recording Audio Active characteristic to “1” (Active) with a timed write.
7. Read the Supported Audio Recording Configuration characteristic. In the Details view of the Characteristic Read Completed, note the bitRate under the audioCodecRecordingParameters.
8. Configure the accessory to record by using the Build TLV button associated with the Selected Camera Recording Configuration characteristic, and using a 4000msec prebuffer duration, 4000msec fragment duration, 800kbps video bit rate, 4000 i-Frame Rate Interval, and the bitRate value from the read audioCodecRecordingParameters.
9. Select the “Write” button in the Write TLV panel.
10. Navigate to the Data Stream Transport Management service and choose Send Start Command, then select Connect.
11. Enter the integer “1” in both Stream ID fields under Camera Recording.
12. Select Send Start Request.
13. Wait 5 seconds.
14. Select Send Close Event with reason “0”.
15. Select Save Recording.
16. Open and view the recording and verify that it is playable and valid audio can be heard.
17. Navigate to the Recording Audio Active characteristic.
18. Set the Recording Audio Active characteristic to “0” (Inactive) with a timed write.

19. Navigate to the Data Stream Transport Management service and select Send Start Command, then select Connect.
20. Enter the integer “2” in both Stream ID fields under Camera Recording.
21. Select Send Start Request.
22. Wait 5 seconds.
23. Select Send Close Event with reason “0”.
24. Select Save Recording.
25. Open and view the recording and verify that it is playable but that there is no audio.
26. Remove all pairings from the accessory via HAT.

TCR094 **The accessory must reject any write to the Recording Audio Active characteristic if the Admin bit is not set to 1 in the request with HTTP Status Code 207 Multi-Status indicating HAP Status Code -70401 (Request denied due to insufficient privileges).**

Applies to accessories that support Camera Event Recording. Perform this test case with HAT using the steps below.

1. Perform Pair Setup adding the accessory to HAT.
2. Discover the accessory’s services and characteristics.
3. Set the Recording Audio Active characteristic to false.
4. Add an additional IP controller.
5. Add the new controller as a non-admin controller.
6. Using the non-admin controller attempt to set the Recording Audio Active Characteristic to “1” (true) with a timed write.
7. Verify that the accessory rejects it with HTTP Status Code 207 Multi-Status indicating HAP Status Code -70401.
8. Using the non-admin controller attempt to set the Recording Audio Active Characteristic to false or 0 with a timed write.
9. Verify that the accessory rejects it with HTTP Status Code 207 Multi-Status indicating HAP Status Code -70401.
10. Remove all pairings from the accessory via HAT.

TCR095 **Verify that the accessory rejects any periodic snapshot request or snapshot request without a (valid) reason field when the Periodic Snapshots Active characteristic is set to Disabled with HTTP Status Code 207 Multi-Status indicating HAP Status Code -70412 (Not allowed in the current state).**

Applies to accessories that support Camera Event Recording. Perform this test case with HAT using the steps below.

1. Perform Pair Setup adding the accessory to HAT.
2. Discover the accessory's services and characteristics.
3. Set the Periodic Snapshots Active characteristic to "1" (Enable Snapshots).
4. Navigate to the Camera RTP Stream Management Service.
5. Choose Take Snapshot without specifying a reason.
6. Verify that the accessory responds with a valid snapshot.
7. Set the Periodic Snapshots Active characteristic to "0" (Disable Snapshots).
8. Navigate to the Camera RTP Stream Management Service.
9. Specify reason "0", and choose Take Snapshot.
10. Verify that the accessory responds with HTTP Status Code 207 Multi-Status indicating HAP Status Code -70412.
11. Navigate to the Camera RTP Stream Management Service.
12. Specify reason "1", and choose Take Snapshot.
13. Verify that the accessory responds with a valid snapshot.
14. Choose Take Snapshot without specifying a reason.
15. Verify that the accessory responds with HTTP Status Code 207 Multi-Status indicating HAP Status Code -70412.
16. Remove all pairings from the accessory via HAT.

TCR096 **Check the default values of audio recording active for all four instances (default, pairing reset, factory reset, and reboot).**

Applies to accessories that support Camera Event Recording. Perform this test case with HAT using the steps below.

1. Perform Pair Setup adding the accessory to HAT.
2. Discover the accessory's services and characteristics.
3. Verify the (default) value of the Recording Audio Active characteristic is "0" (Disabled).
4. Reboot the accessory.
5. Verify the values of step 3 persists.
6. Set the value of the Recording Audio Active characteristic is "1" (Enabled) with a timed write.
7. Reboot the accessory.
8. Verify the values of step 6 persists.
9. Cleanly remove the HomeKit pairings from the accessory in HAT by choosing Remove Pairing under Pairing.

10. Perform Pair Setup adding the accessory to HAT.
11. Discover the accessory's services and characteristics.
12. Verify the value of the Recording Audio Active characteristic is "0" (Disabled).
13. Set the value of the Recording Audio Active characteristic is "1" (Enabled) with a timed write.
14. Perform factory reset on the accessory.
15. Perform Pair Setup adding the accessory to HAT.
16. Discover the accessory's services and characteristics.
17. Verify the value of the Recording Audio Active characteristic is "0" (Disabled).
18. Remove all pairings from the accessory via HAT.

TCR097 **Verify that Audio Recording Active is set to 0 (Inactive) by default, after factory reset, after pairings are removed, and whenever the characteristic is set to 0 the video recording does not contain audio.**

Applies to accessories that support Camera Event Recording. Perform this test case with HAT using the steps below.

1. Perform Pair Setup adding the accessory to HAT.
2. Discover the accessory's services and characteristics.
3. Set the HomeKit Camera Active characteristic on the Camera Operating Mode service to "1" (On) with a timed write.
4. Set the Active characteristic on the Camera Event Recording Management service to "1" (Active) with a timed write.
5. Read the Supported Audio Recording Configuration characteristic. In the Details view of the Characteristic Read Completed, note the bitRate under the audioCodecRecordingParameters.
6. Configure the accessory to record by using the Build TLV button associated with the Selected Camera Recording Configuration characteristic, and using a 4000msec prebuffer duration, 4000msec fragment duration, 800kbps video bit rate, 4000 i-Frame Rate Interval, and the bitRate value from the read audioCodecRecordingParameters.
7. Select the "Write" button in the Write TLV panel.
8. Read the Recording Audio Active to verify that it is set to "0".
9. Navigate to the Data Stream Transport Management service and select Send Start Command, then select Connect.
10. Enter the integer 1 in both Stream ID fields under Camera Recording.
11. Select Send Start Request.
12. Wait 5 seconds.
13. Select Send Close Event with reason "0".

14. Select Save Recording.
15. Open and view the recording and verify that it is playable and that there is no audio.
16. Set the value of the Recording Audio Active characteristic as "1" (Enabled) with a timed write.
17. Enter the integer "2" in both Stream ID fields under Camera Recording.
18. Select Send Start Request.
19. Wait 5 seconds.
20. Select Send Close Event with reason "0".
21. Select Save Recording.
22. Open and view the recording and verify that it is playable and that there is audio.
23. Remove pairing from the accessory.
24. Perform Pair Setup adding the accessory to HAT.
25. Discover the accessory's services and characteristics.
26. Set the HomeKit Camera Active characteristic to "1" (On) with a timed write.
27. Set the Active characteristic associated to the Camera Event Recording Management service to "1" (On) with a timed write.
28. Configure the accessory to record by using the Build TLV button associated with the Selected Camera Recording Configuration characteristic, and using a 4000msec prebuffer duration, 4000msec fragment duration, 800kbps video bit rate, 4000 i-Frame Rate Interval, and the bitRate value from the read audioCodecRecordingParameters.
29. Select the "Write" button in the Write TLV panel.
30. Read the Recording Audio Active to verify it is set to "0".
31. Navigate to the Data Stream Transport Management service and select Send Start Command, then select Connect.
32. Enter the integer "3" in both Stream ID fields under Camera Recording.
33. Select Send Start Request.
34. Wait 5 seconds.
35. Select Send Close Event with reason "0".
36. Select Save Recording.
37. Open and view the recording and verify that it is playable and that there is no audio.
38. Remove all pairings from the accessory via HAT.

TCR098 **The accessory must reject the Start Data Stream request when the Manually Disabled characteristic is set to "1" (Manually Disabled) with Protocol Specific Error, Status: 1 (Now Allowed).**

Applies to accessories that support Camera Event Recording. Perform this test case with HAT using the steps below.

1. Perform Pair Setup adding the accessory to HAT.
2. Discover the accessory's services and characteristics.
3. Set the value of the Recording Audio Active characteristic to "1" (Enabled) with a timed write.
4. Read the Supported Audio Recording Configuration characteristic. In the Details view of the Characteristic Read Completed, note the bitRate under the audioCodecRecordingParameters.
5. Enable event notifications associated with the Motion Detected characteristic.
6. Configure the accessory to record by using the Build TLV button associated with the Selected Camera Recording Configuration characteristic, and using a 4000msec prebuffer duration, 4000msec fragment duration, 800kbps video bit rate, 4000 i-Frame Rate Interval, and the bitRate value from the read audioCodecRecordingParameters.
7. Select the "Write" button in the Write TLV panel.
8. Navigate to the Active characteristic associated with Camera RTP Stream Management service and set it to "1".
9. Navigate to the Active characteristic associated with Camera Event Recording service and set it to "1".
10. Navigate to the HomeKit Camera Active characteristic and set it to "1".
11. Manually switch the accessory to Disabled via the Hardware switch.
12. Navigate to the Data Stream Transport Management service and select Send Start Command, then select Connect.
13. Enter the integer "1" in both Stream ID fields under Camera Recording.
14. Select Send Start Request.
15. Wait 5 seconds.
16. Using the HomeKit Data Stream trace view, verify that the accessory responds to the Start Data Stream request with a response that has a header that includes "dataSend" as the protocol, "Open" as the topic, "6" as the "Status" set (Protocol Specific Error), an "ID" that matches the ID of the Start Data Stream request, and a message with "1" as the "Status" set (Not Allowed).
17. Remove all pairings from the accessory via HAT.

TCR099 **Verify that the accessory accepts read/writes to characteristics, excluding Manually Disabled, for the Camera RTP Stream Management, Data Stream Transport Management, Camera Event Recording Management, and Camera Operating Mode services when Manually Disabled characteristic is set to 1 (Manually Disabled) unless otherwise specified, and reflect the changes once Manually Disabled characteristic is set to 0 (Manually Enabled).**

Applies to accessories that support Camera Event Recording. Perform this test case with HAT using the steps below.

1. Perform Pair Setup adding the accessory to HAT.
2. Discover the accessory's services and characteristics.

3. Manually switch the the accessory to Disabled via the Hardware switch.
4. Set the value of the Event Snapshot Active characteristic to "1" (Enabled).
5. Set the value of the Periodic Snapshot Active characteristic to "1" (Enabled).
6. Set the value of the HomeKit Camera Active characteristic to "1" (Enabled).
7. Set the value of the Active characteristic associated to the Camera RTP Stream Management Service to "1" (Enabled).
8. Read the value of the Event Snapshot Active characteristic and verify it is set to "1" (Enabled).
9. Read the value of the Periodic Snapshot Active characteristic and verify it is set to "1" (Enabled).
10. Read the value of the HomeKit Camera Active characteristic and verify it is set to "1" (Enabled).
11. Read the value of the Active characteristic associated to the Camera RTP Stream Management Service and verify it is set to "1" (Enabled).
12. Manually switch the the accessory to Disabled via the Hardware switch.
13. Read the value of the Event Snapshot Active characteristic and verify it is set to "1" (Enabled).
14. Read the value of the Periodic Snapshot Active characteristic and verify it is set to "1" (Enabled).
15. Read the value of the HomeKit Camera Active characteristic and verify it is set to "1" (Enabled).
16. Read the value of the Active characteristic associated to the Camera RTP Stream Management Service and verify it is set to "1" (Enabled).
17. Remove all pairings from the accessory via HAT.

TCR100 **Verify that the accessory supports concurrent live streams from each of its Camera RTP Stream Management services.**

Applies to accessories that support Camera Event Recording. Perform this test case with HAT using the steps below.

1. Perform Pair Setup, adding the accessory to HAT.
2. Discover the accessory's services and characteristics.
3. Navigate to the Camera RTP Stream Management service.
4. Select the "Select Stream Parameters" button, select supported parameters, then select "Configure".
5. Select"Negotiate", then select "Start Stream".
6. Verify that the accessory successfully begins live streaming.
7. Repeat steps 3-5 for each of the Camera RTP Stream Management services on the accessory, verifying that previously started live streams are uninterrupted.
8. Navigate to the Camera Event Recording Management service.
9. Set the Active characteristic associated to the Camera Event Recording Management service to "1" (On/Active) with a timed write.

10. Set the value of the Recording Audio Active characteristic to “1” (Enabled) with a timed write.
11. Read the Supported Audio Recording Configuration characteristic. In the Details view of the Characteristic Read Completed, note the bitRate under the audioCodecRecordingParameters.
12. Configure the accessory to record by using the Build TLV button associated with the Selected Camera Recording Configuration characteristic, and using a 4000msec prebuffer duration, 4000msec fragment duration, 800kbps video bit rate, 4000 i-Frame Rate Interval, and the bitRate value from the read audioCodecRecordingParameters.
13. Select the “Write” button in the Write TLV panel.
14. Navigate to the Data Stream Transport Management service and select Send Start Command, then select Connect.
15. Enter the same integer in both Stream ID fields under Camera Recording.
16. Select Send Start Request.
17. Wait 5 seconds.
18. Select Send Close Event with reason “0”.
19. Select Save Recording.
20. Open and view the recording and verify that it is playable.
21. Remove all pairings from the accessory via HAT.

TCR101 **Verify Audio and Video recording requirements.**

Applies to accessories that support Camera Event Recording. Perform this test case with HAT using the steps below.

1. Pair and discover accessory.
2. Set the HomeKit Camera Active characteristic in the Camera Operating Mode service to “1”(On) with a timed write.
3. Set the Active characteristic associated to the Camera Event Recording Management service to “1” (Active) with a timed write.
4. Set the value of the Recording Audio Active characteristic to “1” (Enabled).
5. Configure the accessory to record by using the Build TLV button associated with the Selected Camera Recording Configuration characteristic. Use a 4000msec pre-buffer duration, 4000msec fragment duration, 4000 i-Frame Rate Interval, and 64kbps audio bit rate and set Image Width, Height, Video Frame Rate, Video Bit Rate, Audio Sample Rate from the table.
6. Select the “Write” button in the Write TLV panel.
7. Navigate to Data Stream Transport Management service. Choose Send Start Command and then Connect.
8. Enter in the same integer in both Stream ID fields under Camera Recording.

9. Choose Send Start Request.
10. Wait 5 seconds.
11. Choose Send Close Event with reason “0”.
12. Choose Save Recording.
13. Open and view the recording and verify that it is playable.
14. Verify the following for each combination of Image resolution, Video frame rate, and Video bit rate from the table.
    - Video recording initiates successfully in HAT.
    - There are no visual artifacts in the video clip recorded.
    - There are no audio artifacts in the HAT audio downlink.
    - There are no audio artifacts in the video clip recorded.
    - Video recorded is smooth, is not jittery, and does not lag.
    - Camera is encoding at the correct resolution.
    - Framerate does not exceed target framerate.
    - Operating Bitrate is within the Target Bitrate.
    - Audio and Video are synced on the clip recorded.
    - Verify Mandatory resolutions for corresponding aspect ratio are supported.
    - Attempt to start a recording with unsupported resolution, and verify that the accessory responds with “HTTP 207” with HAP status code “70410”.

Verify for each of the following combinations (mandatory resolutions are required for aspect ratios advertised by the accessory):

Table 1.10: Test Cases for IP Camera Audio and Video recording requirements

Test Case	Resolution	HAP Specification	Video Frame Rate	Video Bit Rate	Audio Sample Rate	Aspect Ratio
TCR101	1920 x 1080	All	Highest Supported	800k	16kHz and/or 24kHz	16:9
TCR102	1920 x 1080	R16 or later	Highest Supported	2M	16kHz and/or 24kHz	16:9
TCR103	1280 x 720	All	Highest Supported	800k	16kHz and/or 24kHz	16:9
TCR104	1280 x 720	R16 or later	Highest Supported	2M	16kHz and/or 24kHz	16:9
TCR105	1080 x 1920	R16 or later	Highest Supported	2M	16kHz and/or 24kHz	9:16
TCR106	720 x 1280	R16 or later	Highest Supported	2M	16kHz and/or 24kHz	9:16
TCR107	1600 x 1200	All	Highest Supported	800k	16kHz and/or 24kHz	4:3
TCR108	1600 x 1200	R16 or later	Highest Supported	2M	16kHz and/or 24kHz	4:3
TCR109	1440 x 1080	R16 or later	Highest Supported	2M	16kHz and/or 24kHz	4:3
TCR110	1280 x 960	All	Highest Supported	800k	16kHz and/or 24kHz	4:3
TCR111	1280 x 960	R16 or later	Highest Supported	2M	16kHz and/or 24kHz	4:3
TCR112	1024 x 768	All	Highest Supported	800k	16kHz and/or 24kHz	4:3
TCR113	1024 x 768	R16 or later	Highest Supported	2M	16kHz and/or 24kHz	4:3
TCR114	1200 x 1600	R16 or later	Highest Supported	2M	16kHz and/or 24kHz	3:4
TCR115	1080 x 1440	R16 or later	Highest Supported	2M	16kHz and/or 24kHz	3:4
TCR116	960 x 1280	R16 or later	Highest Supported	2M	16kHz and/or 24kHz	3:4
TCR117	768 x 1024	R16 or later	Highest Supported	2M	16kHz and/or 24kHz	3:4

Table 1.11: Mandatory IP Camera Audio and Video recording requirements (mandatory resolutions are required for aspect ratios advertised by the accessory)

Test Case	Requirement
TCR101	Mandatory for R15 or earlier
TCR102	Mandatory for R16 or later
TCR103	Mandatory for R15 or earlier
TCR105	Mandatory for R16 or later
TCR108	Mandatory for R16 or later
TCR114	Mandatory for R16 or later

1.20 HomeKit Data Stream

TCHDS001: Verify that if no message is received before the 10 second timeout, or if the accessory received any other message from the controller, that the accessory closes the socket.

TCHDS002: Verify that the accessory can support multiple HDS connections. If the accessory supports more HAP over TCP sessions than it does HDS over TCP connections, accessory must respond to the Start Command with TLV item 0x01 (Status) with a value of "2" (Busy) when the max number of HDS session is reached.

TCHDS003: Accessories that support HomeKit Data Stream over Wi-FI or Ethernet must be able to establish a Data Stream Transport session.

TCHDS004: If the accessory has multiple pending connections, and there is ambiguity in which controller is connecting, (e.g. it provides the same port in all setup requests), it must attempt decrypting the first packet with each key that is a possible match. If none match, then it must close the socket since it received a connection that did not have a valid hello request as its first message. If a key is tested but is not a match, the nonce for that key must not be changed, so that when the actual controller with that key connects, that key is still usable.

TCHDS005: The TCP port range for HomeKit Data Stream must be >= 32768.

TCHDS006: Accessories that support HomeKit Data Stream must include the "Data Stream Transport Management" service and the required characteristics.

TCHDS007: Accessories that support HomeKit Data Stream over Wi-FI or Ethernet must claim support as part of the "Supported Data Stream Transport Configuration" characteristic TLV value.

TCHDS008: Accessories that support HomeKit Data Stream over HAP must claim support as part of the "Supported Data Stream Transport Configuration" characteristic TLV value.

TCHDS009: Accessories that support HomeKit Data Stream over HAP must be able to establish a Data Stream Transport session.

TCHDS011: Verify that the accessory sends frame sizes with a value between 1KB to 900KB as the chunk size for each data chunk.

TCHDS012: Verify that the first "Data Sequence Number" is "1", and if present, the first "Data Chunk Sequence Number" is also "1", and subsequent Data or Chunk Sequence Numbers increment correctly.

TCHDS001 **Verify that if no message is received before the 10 second timeout, or if the accessory received any other message from the controller, that the accessory closes the socket.**

Applies to accessories that support HomeKit Data Stream (HDS) over Ethernet or Wi-Fi. Perform this test case with HAT using the steps below.

1. Pair and discover accessory.
2. Select the "Data Stream Transport Management" service and select the "Send Start Command" button.
3. Using the HTTP traffic view, verify the accessory responds to the Start Command with status "0".
4. Wait 11 seconds from sending the initial Start Command.

5. Under the "Data Stream Transport Management" service, select the "Connect" button.
6. Using the HDS Frames traffic view, verify a "Data Stream Error" message is received. Select the "Details" button and verify the error message contains "The operation couldn't be completed. Connection refused."
7. Verify the "Data Steam Error" message is followed by a "Data Stream Disconnected" message.
8. Select the "Data Stream Transport Management" service and select the "Send Start Command" button.
9. Using the HTTP traffic view, verify the accessory responds to the Start Command with status "0".
10. Select the "Connect" button with the "Send Hello after connection established" checkbox disabled.
11. Within 10 seconds of Connecting, send an HDS message other than Hello, e.g. for Apple TV Remote accessories, enter a Target Control Identifier integer and send a Target Control Identifier event. For Camera Event Recording accessories, enter a Stream ID and select "Send Start Request".
12. Verify the accessory immediately closes the HDS connection after the message is received.

TCHDS002 **Verify that the accessory can support multiple HDS connections. If the accessory supports more HAP over TCP sessions than it does HDS over TCP connections, accessory must respond to the Start Command with TLV item 0x01 (Status) with a value of "2" (Busy) when the max number of HDS session is reached.**

Applies to accessories that support HomeKit Data Stream (HDS) over Ethernet or Wi-Fi. Perform this test case with HAT using the steps below.

1. Pair and discover accessory.
2. Using the "+" on the bottom left corner, add additional IP controllers that equal the amount of HDS connections the accessory can support. (e.g. If the accessory can support 8 HDS connections, add an additional 8 controllers, for a total of 9 controllers.) If the max amount of HDS over TCP connections exceeds the amount of pairing relationships that the accessory supports, add additional controllers until the pairing relationship max is met.
3. Using Controller 1, choose "Add Additional Controllers" as non-admins.
4. Select the "Data Stream Transport Management" service and select the "Send Start Command" button.
5. Using the HTTP traffic view, verify the accessory responds to the Start Command with a value field containing "AQEAA" (Base64 equivalent of TLV item 0x01 (Status) with a value of "0" (Success).
6. Under the "Data Stream Transport Management" service, select the "Connect" button.
7. Using the HDS Frames traffic view, verify the "Data Stream Connected" and "Hello Response Received" messages are received by the controller, and that no "Data Stream Disconnected" messages are seen.
8. Select the next controller and discover the accessory.

9. Repeat steps 4-9 with each additional controller until the accessory responds to the Start Command in step 4 with a value field consisting of "AQEC" (Base64 equivalent of TLV item 0x01 (Status) with a value of "2" (Busy) in the HTTP traffic view. Please note: If the max amount of HDS over TCP connections exceeds the amount of pairing relationships that the accessory supports, all additional controllers are expected to successfully establish an HDS connection and a "Busy" status will not apply.

TCHDS003 **Accessories that support HomeKit Data Stream over Wi-FI or Ethernet must be able to establish a Data Stream Transport session.**

Applies to accessories that support HomeKit Data Stream (HDS) over Ethernet or Wi-Fi. Perform this test case with HAT using the steps below.

1. Pair and discover accessory.
2. Navigate to the "Data Stream Transport Management" service.
3. Select the "Send Start Command" button.
4. In the HTTP traffic view, verify the accessory Start response contains "status": 0.
5. Within 10 seconds of sending the Start Command, select "Connect" button.
6. Using the HDS Frames view, verify "Data Stream Connected" message is seen.
7. Select the "Hello Request Sent" event, and show details. Notate the Identifier number.
8. Verify a Hello response is received.
9. Select the "Hello Response Received" event, and show the details.
10. Verify the Identifier matches the Identifier from step 7, and the connection remains open. (e.g, no "Data Stream Disconnected" message is seen)

TCHDS004 **If the accessory has multiple pending connections, and there is ambiguity in which controller is connecting, (e.g. it provides the same port in all setup requests), it must attempt decrypting the first packet with each key that is a possible match. If none match, then it must close the socket since it received a connection that did not have a valid hello request as its first message. If a key is tested but is not a match, the nonce for that key must not be changed, so that when the actual controller with that key connects, that key is still usable.**

Applies to accessories that support HomeKit Data Stream (HDS) over Ethernet or Wi-Fi. Perform this test case with HAT using the steps below.

1. Pair and discover accessory.
2. Using the "+" in the bottom left corner, add an additional IP controller.
3. Using Controller 1, choose "Add Additional Controllers" as non-admin.
4. Select Controller 2 and select the "Discover" button.
5. Perform steps 6-13 within 10 seconds.

6. From Controller 1, select the "Data Stream Transport Management" service and select the "Send Start Command" button.
7. Select the "Connect" button with the "Send Hello after connection established" checkbox disabled.
8. Note the Port value given for Controller 1.
9. From Controller 2, select the "Data Stream Transport Management" service, enter the Port value from step 8.
10. Select the "Connect" button with the "Send Hello after connection established" checkbox enabled.
11. Verify the accessory closes the HDS connection for Controller 2.
12. From Controller 1, select the "Data Stream Transport Management" service and select the "Send Hello" button.
13. Verify successful Hello response is received to Controller 1 and connection remains open after the 10 second connection timer expires.
14. Perform steps 15-20 within 10 seconds.
15. From Controller 1, select the "Data Stream Transport Management" service and select the "Send Start Command" button.
16. Select the "Connect" button with the "Send Hello after connection established" check box disabled.
17. Note the Port value given for Controller 1.
18. From Controller 2, select the "Data Stream Transport Management" service and enter the Port value from step 17.
19. Select the "Connect" button with the "Send Hello after connection established" check box enabled.
20. Verify the accessory closes the HDS connection for Controller 2.
21. Verify connection to Controller 1 is closed after 10 seconds timer expires.

TCHDS005 **The TCP port range for HomeKit Data Stream must be >= 32768.**

Applies to accessories that support HomeKit Data Stream (HDS) over Ethernet or Wi-Fi. Perform this test case automatically with HCA. Below are steps that can be used to perform this test case manually in HAT.

1. Pair and discover accessory.
2. In the left side of the controller's, window select the Data Stream Management service.
3. In the HomeKit Data Stream panel, select the "Send Start Command" button.
4. Verify that the port populated in the text field below is greater than or equal to 32768.
5. If applicable, enable "Send Hello after connection established" checkbox.
6. In the HomeKit Data Stream panel, select the "Connect" button.
7. Wait until the "Status" in the HomeKit Data Stream panel is "Connected".
8. In the HomeKit Data Stream panel, select the "Disconnect" button.

9. Repeat Step 3-8 100 times.

TCHDS006 **Accessories that support HomeKit Data Stream must include the "Data Stream Transport Management" service and the required characteristics.**

Applies to all accessories that support HomeKit Data Stream (HDS). Perform this test case with HAT using the steps below.

1. Pair and discover accessory.
2. In the left sidebar of Controllers window, see each of the accessory's services.
3. Verify that the Data Stream Transport Management service is present.
4. Verify that the Data Stream Transport Management service includes the required characteristics.
   Required characteristics for HDS over Wi-FI or Ethernet:
   - Supported Data Stream Transport Configuration (r)
   - Setup Data Stream Transport (w/r/wr)
   - Version (r)

   Required characteristics for HDS over BLE or Thread:
   - Supported Data Stream Transport Configuration (r)
   - Setup Data Stream Transport (w/r/wr)
   - Version (r)
   - Data Stream HAP Transport (w/r)
   - Data Stream HAP Transport Interrupt (r)

TCHDS007 **Accessories that support HomeKit Data Stream over Wi-FI or Ethernet must claim support as part of the "Supported Data Stream Transport Configuration" characteristic TLV value.**

Applies to accessories that support HomeKit Data Stream (HDS) over Ethernet or Wi-Fi. Perform this test case with HAT using the steps below.

1. Pair and discover accessory.
2. Navigate to the "Data Stream Transport Management" service and read the "Supported Data Stream Transport Configuration" characteristic.
3. In the Events view of the Trace window, verify that the response value contains Transport Type = "0" (HomeKit Data Stream over Wi-FI or Ethernet).

TCHDS008 **Accessories that support HomeKit Data Stream over HAP must claim support as part of the "Supported Data Stream Transport Configuration" characteristic TLV value.**

Applies to accessories that support HomeKit Data Stream (HDS) over HAP. Perform this test case with HAT using the steps below.

1. Pair and discover accessory.

2. Navigate to the "Data Stream Transport Management" service and read the "Supported Data Stream Transport Configuration" characteristic.
3. In the Events view of the Trace window, verify the response value contains Transport Type = "1" (HomeKit Data Stream over HAP).

TCHDS009 **Accessories that support HomeKit Data Stream over HAP must be able to establish a Data Stream Transport session.**

Applies to accessories that support HomeKit Data Stream (HDS) over HAP. Perform this test case with HAT using the steps below.

1. Pair and discover accessory.
2. For HAP over BLE, enable the "Pair Resume Keep Alive Enabled'' checkbox in the Connection panel.
3. Navigate to the "Data Stream Transport Management" service.
4. Select the "Send Start Command" button.
5. In the HomeKit Data Stream panel, enable the "Send Hello after session starts" checkbox.
6. Within 10 seconds of sending the Start Command, select the "Start Session" button.
7. Using the HDS Frames view, select the "Hello Request Sent" event, and show details. Note the Identifier number.
8. Verify a Hello response is received.
9. Select the "Hello Response Received" event, and show the details.
10. Verify the Identifier matches the Identifier from step 7.
11. In the HomeKit Data Stream panel, select the "End Session" button.

TCHDS011 **Verify that the accessory sends frame sizes with a value between 1KB to 900KB as the chunk size for each data chunk.**

Applies to accessories that support HomeKit Data Stream (HDS) over Ethernet or Wi-Fi. Perform this test case with HAT using the steps below.

1. Pair and discover the accessory with HAT.
2. In the left sidebar of Controllers window, select the "Data Stream Management" service.
3. In the "HomeKit Data Stream" panel, select the "Send Start Command" button and then select "Connect".
4. For accessories that support camera event recording, set the "HomeKit Camera Active" characteristic and the "Active" characteristic associated to the "Camera Event Recording Management" service to "1" (On) with a timed write, and then configure the accessory to record.
5. For remotes for Apple TV accessories that support Siri, configure the accessory to send Siri audio to the controller.

6. Navigate to the “Data Stream Transport Management” service.
7. For accessories that support camera event recording, enter a stream ID into both of the “Stream ID” fields under the “Camera Recording” pane. Select “Send Start Request”, wait at least 10 seconds, enter “0” into the “Close Reason” field, and then select “Send Close Event”.
8. For remote accessories that support Siri, press and hold the Siri button on the accessory for at least 10 seconds while speaking any uterance, and then release the Siri button.
9. In the HDS Frames trace view, verify each “Binary Data Event” with data type “Media Fragment” packet size is between 1024 and 921600 bytes (1KB-900KB). An exception to the 1 KB minimum is the first packet, a.k.a MediaInitialization, which can be less than 1 KB. The other exception to this, is the last packet of a Data Sequence, if the accessory has less than 1 KB to be sent.

TCHDS012 **Verify that the first “Data Sequence Number” is “1”, and if present, the first “Data Chunk Sequence Number” is also “1”, and subsequent Data or Chunk Sequence Numbers increment correctly.**

Applies to accessories that support HomeKit Data Stream (HDS) over Ethernet or Wi-Fi. Perform this test case with HAT using the steps below.

1. Pair and discover the accessory with HAT.
2. In the left side of the controller’s, window select the “Data Stream Management” service.
3. In the “HomeKit Data Stream” panel, select the “Send Start Command” button, and then select “Connect”.
4. For accessories that support camera event recording, set the “HomeKit Camera Active” characteristic and the “Active” characteristic associated to the “Camera Event Recording Management” service to “1” (On) with a timed write, and then configure the accessory to record.
5. For remotes for Apple TV accessories that support Siri, configure the accessory to send Siri audio to the controller.
6. Navigate to the “Data Stream Transport Management” service.
7. For accessories that support camera event recording, enter a stream ID into both of the “Stream ID” fields under the “Camera Recording” pane, and select “Send Start Request”, wait at least 10 seconds, enter “0” into the “Close Reason” field, and then select “Send Close Event”.
8. For remote accessories that support Siri, press and hold the Siri button on the accessory for at least 10 seconds while speaking any uterance, and then release the Siri button.
9. In the HDS Frames trace view, select the first “Binary Data Event” and show the details.
10. If the “metadata” field is present and contains a “Data Sequence Number” field, verify that the value is set to “1”.
11. If the “metadata” field is present and contains a “Data Chunk Sequence Number” field, verify that the value is set to “1”.
12. If present, navigate to the next “Binary Data Event”.
13. If the “metadata” field is present and contains a “Data Sequence Number” field, verify that the value is the same as it was in the previous Binary Data Event, or has incremented by only 1.

14. If the “metadata” field is present and contains a “Data Chunk Sequence Number” field, and the “Data Sequence Number” is the same as it was in the previous Binary Data Event, verify that the “Data Chunk Sequence Number” has incremented by only 1.

15. If the “metadata” field is present and contains a “Data Chunk Sequence Number” field, and the “Data Sequence Number” has incremented, verify that the “Data Chunk Sequence Number” is set to “1”.

16. Repeat steps 12-15 for each “Binary Data Event”.

1.21 Thread

TCT001: Verify that a thread accessory can be paired over BLE and moved to thread network.

TCT002: Verify that the accessory CoAP application level retries are set to a minimum of 2.

TCT003: Verify that the accessory supports a CoAP packet containing multiple PDU requests.

TCT004: Accessories must stop advertising within 10 minutes of the last pairing attempt while still in unpaired state. Accessories must require a user action such as a power cycle, button press or other explicit user action for the accessory to re-enter pairing mode and advertise as an unpaired accessory.

TCT005: Verify that the thread accessory implements a MTD (Minimum Thread Device) role if battery powered, a FTD (Full Thread Device) role if AC-powered, and that the "Node Capabilities" characteristic reflects this capability.

TCT006: Verify the accessory advertises on the network via Bonjour using "_hap._udp".

TCT008: Verify that "Pair Setup" can be completed successfully over Thread transport.

TCT009: Verify that "Pair Verify" completes successfully and uses the "/2" URL to establish a new security session.

TCT010: Verify that the "Thread Role" characteristic value accurately reflects the current role of an accessory in the thread network.

TCT011: Verify that FTD-type accessories are able to receive and process HAP requests at any time after the HAP service has been registered, and are able to send back responses and notifications immediately.

TCT012: Verify that MTD-type accessories can receive and process HAP requests when they wake up, but still must send status change notifications immediately, regardless of state.

TCT013: Verify that a thread accessory supports refreshing of security session when a controller performs a pair-verify with the accessory again and that any existing active session is terminated.

TCT014: Verify that a thread accessory has a "Thread Transport" service and all required characteristics.

TCT015: Verify the functionality of the Thread Control Point characteristic's "Set Thread Parameters" with "Forming Allowed" set to 1, for an Always-On accessory.

TCT016: Verify that the characteristics within "Thread Transport Service" can only be read from or written to by an admin user and not by a shared non-admin user.

TCT017: Verify that the functionality of the Thread Control Point characteristic's "Set Thread Parameters" with "Forming Allowed" set to 0.

TCT019: Verify the functionality of the Thread Control characteristic's "Set Thread Parameters" when the accessory is already connected to a thread network.

TCT020: The HAP accessory object, with the instance ID of "1", must be the primary HAP accessory object. For bridges, the primary HAP accessory object must be the bridge itself.

TCT021: Verify that when a sleepy accessory is unpaired, it must switch to wakeful state until it goes over the pair-setup and pair-verify flow and that it can go back to its intended wake up frequency once paired.

TCT022: When a paired sleepy accessory receives a pair-verify request (e.g., keys expired, or controller is establishing a new session), verify that it switches to wakeful state (of SleepInterval=500 milliseconds) until the pair-verify process is complete and that the queued requests are completed on its next wake up cycle.

TCT023: For a sleepy accessory, verify that an accessory switching to wakeful state due to a Pair-Verify, implements a "graceful" delay of 2 seconds to process any potential commands from Controller before going back to sleep.

TCT026: Verify the accessory can successfully remove all pairings, and perform Pair-Setup again while on the Thread network.

TCT028: If the controller provides the credentials using the Thread Transport service, verify that the accessory keeps its BLE connection open and at the same time tries to join the network immediately.

TCT029: Once the accessory successfully joins the Thread network, verify accessory disables its bluetooth functionality and switches to Thread only mode.

TCT030: If attaching to a Thread network is unsuccessful, the Thread network is not viable, or the accessory gets detached from the Thread network, verify that the accessory switches back to BLE mode and periodically attempts to rejoin the Thread network.

TCT031: If an accessory that was connected to a Thread network gets detached, verify that it waits 30 seconds before attempting to switch back to BLE (to help overcome temporary network glitches and avoid unnecessary overhead).

TCT032: Verify that the transition from BLE to Thread, and vice versa, is fully transparent to users.

TCT033: When removing the last pairing of an accessory, verify that the accessory transitions into unpaired state but remains connected to the Thread network. In case of a sleepy device, it should change its sleep interval to 500ms during the first 5 minutes to improve the response times in case user wants to pair it again. After the 5 minutes the device can revert back to its original sleep period.

TCT034: For a Sleepy Accessory, when removing the last pairing of an accessory, verify that the accessory transitions into unpaired state but remains connected to the Thread network. It should change its sleep interval to 500ms during the first 5 minutes to improve the response times in case user wants to pair it again. After the 5 minutes the device may revert back to its original sleep period.

TCT035: Verify that Paired accessories use a DNS-Advertisement expiration timeout of 1 hour.

TCT036: Verify that Unpaired accessories use a DNS-Advertisement expiration timeout of 30 seconds.

TCT038: The minimum number of pairing relationships that an accessory must support is 16.

TCT039: Accessories must be able to support 8 controllers simultaneously.

TCT041: The accessory must support event notifications for multiple controllers.

TCT043: The accessory must support writing values to one or more characteristics via a single CoAP request.

TCT044: Services contained within the HAP accessory must be colocated. For example, a fan with a light on it would expose a single HAP accessory with 3 services: the required accessory information service, a fan service, and a lightbulb service. Conversely, a bridge that bridges 2 independent lights, which may be in different physical locations, must expose a HAP accessory object for each independent light.

TCT045: Accessories must supply required properties in each service object: Type, Instance ID, Characteristics, Hidden Services (Conditional), Primary Services (Optional), and Linked Services (Optional).

TCT046: The accessory must include the following properties for each characteristic that supports paired read: Type, Instance ID, Permissions, Value, Format.

TCT047: The accessory must include the following properties for each characteristic that does not support paired read: Type, Instance ID, Permissions, and Format.

TCT048: For characteristics that do not support notifications, verify correct error response when trying to enable notifications.

TCT049: The notification event must be generated on the characteristic that notifications are enabled on and not generated on another characteristic.

TCT050: The name of the Bonjour service (i.e., the user-visible name of accessory) must match the accessory name.

TCT051: The required Bonjour TXT keys must be supplied in the accessory's Bonjour advertisement.

TCT052: The Device ID must persist across a reboot and is randomly generated when accessory is factory reset.

TCT053: The following device information must persist across reboots and power cycles: Device ID (id), Configuration number (c#), and Accessory category identifier (ci).

TCT054: If supported, accessory's configuration number must increment when a service or characteristic is added to or removed from accessory server.

TCT055: For characteristics that don't support paired write, write attempts should fail with the correct error.

TCT056: For characteristics that don't support paired read, read attempts should fail with the correct error.

TCT057: If an accessory has characteristics that have minimum value and maximum value metadata, writing values below the minimum value and above the maximum value must not be accepted by accessory.

TCT058: Accessories must supply required properties inside each accessory: Accessory Instance ID and Services.

TCT061: An accessory must support timed writes to all characteristics even if the characteristic does not require it.

TCT064: Prepare-write. Wait for TTL to expire. Prepare-write. Execute-write.

TCT067: Prepare-write. Execute-write. Execute-write again.

TCT068: If the accessory receives an Execute Write Request after the TTL has expired, it must respond with "Invalid Request 0x06".

TCT069: If the accessory receives a standard write request on a characteristic which requires timed write, the accessory must respond with "Invalid Request 0x06".

TCT070: Verify that the accessory accepts consecutive Prepare Write Requests in the same session.

TCT071: Prepare-write from Controller 1 and Execute-write from Controller 2

TCT073: Accessory must expose a single instance of the Pairing Service with the following required characteristics: Pair Setup, Pair Verify, Features, and Pairings.

TCT077: If an accessory receives a HAP PDU with an opcode that it does not support, it shall reject the PDU and respond with a status code "Unsupported PDU (0x01)" in its HAP response.

TCT078: In case a single CoAP packet contains several PDU requests, verify that the accessory processes these PDUs in the order in which they are received and that responses are stored sequentially in CoAP response.

TCT079: Verify that the accessory supports a minimum of 1024 bytes of incoming PDU data.

TCT081: If authTag verification of encrypted data fails during pair-verify, the accessory must respond with "M4" and "kTLVError_Authentication 0x02".

TCT083: Accessories must support multiple iterations of Pair Verify on a single connection.

TCT087: Accessory must indicate that Security Class characteristics require HAP-Characteristic-Timed-Write using the HAP Characteristic Properties Descriptor.

TCT091: Accessories must implement a 10 second HAP procedure timeout. All HAP procedures including Pair-Verify must complete within 10 seconds.

TCT092: Accessory must support connected events.

TCT093: Accessory must always successfully deliver event notifications for every characteristic that supports them when a single client has subscribed multiple times.

TCT095: When Thread parameters are cleared, the accessory should fall back to BLE. When the accessory is connected to a thread network again, verify that the Accessory can Pair-Verify successfully.

TCT097: Verify that Successful CoAP responses use the 2.04 status code and that Unsuccessful CoAP responses use 4.04.

TCT099: When Thread parameters are overwritten to an accessory while it is on BLE, verify that the Accessory can successfully connect to the new network.

TCT100: When the last pairing has been removed, accessory must change the status flag within 5 seconds after remove pairing is completed, and subsequent pairing attempts with accessory must succeed.

TCT001 **Verify that a thread accessory can be paired over BLE and moved to thread network.**

Applies to accessories using the HAP over Thread transport. Perform this test case with HAT using the steps below.

1. Select the "+" at the bottom of left sidebar and select "Create Shared Key Store".
2. Select the "+" at the bottom of left sidebar and select "Create BLE Controller" to make a new virtual BLE controller. In the Summary panel of the controller, assign the Shared Key Store created in Step 1, and click on the "Start" button to begin discovering BLE accessories.
3. Select the "+" at the bottom of left sidebar and select "Create Thread Controller" to make a new virtual Thread controller. In the Summary panel of the controller, assign the Shared Key Store created in Step 1, and click on the "Start" button to begin discovering Thread accessories.
4. Pair and discover the accessory using BLE Controller.
5. With the Thread network present, navigate to "Thread Transport" service, select "Thread Control Point" characteristic, click "Build TLV" within "Write [tlv8]" panel, select "Set Thread Parameters" from the drop-down, enter the details of the thread border router network, enter "0" in the "Forming Allowed" field. Click "Build TLV".
6. Select "Write" to send the TLV to the accessory and check Thread Discovery view of HAT Trace. Accessory must connect to the Border Router and show its Bonjour advertisement in Thread Discovery view of HAT trace.
7. Discover the accessory using Thread Controller and verify Events View in HAT Trace shows the "Discovered Accessories" response. Click on the Details button to verify that the response contains Services and Characteristics of the thread accessory.

TCT002 **Verify that the accessory CoAP application level retries are set to a minimum of 2.**

Applies to accessories using the HAP over Thread transport.

1. Select the "+" at the bottom of left sidebar and select "Create Thread Controller" to make a new virtual Thread controller, Controller 1. In the Summary panel of the controller, click the "Start'' button to begin discovering Thread accessories. Repeat this step to create a second Thread controller, Controller 2.
2. Connect accessory to Thread network and ensure the accessory's advertisements are present within the "Thread Discovery'' view in the trace window.
3. Pair and discover accessory using Controller 1.
4. Select accessory name under Controller1 and enable notifications on all characteristics that support them.
5. In Accessory Servers view, select the accessory, navigate to Options panel and check the "Skip CoAP Acknowledgement on Events" option.
6. In Controllers window, under "Add Additional Controllers" panel, select "Controller 2" as Controller, and select the "Add Controller" button.
7. In left sidebar, select the Controller 2, select the accessory name, and click the "Discover" button.
8. Select a characteristic that supports notifications. If the selected characteristic is writable, perform a write operation on that characteristic from Controller 2 that will trigger notifications. Or, if applicable, trigger a status change on the hardware that will result in notifications.
9. In CoAP Traffic View of HAT Trace, verify that at least three notification attempts "CoAP Request (Event)" are received for Controller 1.

TCT003 **Verify that the accessory supports a CoAP packet containing multiple PDU requests.**

Applies to accessories using the HAP over Thread transport.

1. Select the "+" at the bottom of left sidebar and select "Create Thread Controller" to make a new virtual Thread controller. In the Summary panel of the controller, click the "Start'' button to begin discovering Thread accessories.
2. Connect accessory to Thread network and ensure the accessory's advertisements are present within the "Thread Discovery'' view in the trace window.
3. Pair and discover the Thread accessory using Thread Controller.
4. In Controllers window, select any characteristic that supports paired write.
5. In the Queued Writing Panel, select "Enable Queue".
6. Write value in the Queue Characteristic Panel and select Queue button.
7. Select another characteristic that supports paired write.
8. Write value in the Queue Characteristic Panel and select Queue button.
9. Repeat Steps 7 and 8 to queue two more write actions for characteristics that support paired write.
10. In the pop-up window, select the "Send" button to send the queue of requests.

11. Verify in CoAP Traffic view of HAT Trace that the queue is sent as a single CoAP request and receives a single CoAP response. Accessories that sleep have an exception to this - it is acceptable for Sleepy accessories to receive CoAP responses for each of the retransmissions attempted by the HAT Controller while the accessory is asleep.

12. Verify in HAP Traffic view that the accessory responds with "Status (0x00)" for each successful write request.

TCT004 **Accessories must stop advertising within 10 minutes of the last pairing attempt while still in unpaired state. Accessories must require a user action such as a power cycle, button press or other explicit user action for the accessory to re-enter pairing mode and advertise as an unpaired accessory.**

Applies to accessories using the HAP over Thread transport. Not Applicable to accessories that clear thread credentials when last admin pairing is removed. Perform this test case automatically with HCA. Below are steps that can be used to perform this test case manually in HAT.

1. Select the "+" at the bottom of left sidebar and select "Create Thread Controller" to make a new virtual Thread controller. In the Summary panel of the controller, click the "Start'' button to begin discovering Thread accessories.
2. Connect the accessory to a Thread network and ensure the accessory's advertisements are present within the "Thread Discovery'' view in the trace window.
3. Pair and discover the Thread accessory using Thread Controller.
4. After the Discover operation completes, select "Remove Pairing".
5. Using the "Thread Discovery'' view in the Trace window, verify the accessory continues to advertise.
6. Wait 11 minutes.
7. Select the controller and then select the "Stop" button to stop discovering accessory servers. Then select "Start" to begin again.
8. Using the "Thread Discovery'' view in the Trace window, verify the accessory is no longer advertising.
9. Place the accessory back into pairing mode by performing a power cycle, button press or other explicit user action.
10. Using the "Thread Discovery'' view in the Trace window, verify the accessory begins advertising again.
11. In the left sidebar of the Controllers window, select the accessory and then select "Start Pairing".
12. After the pairing prompt appears, select "Stop" to dismiss the pairing prompt and to abort the pairing process.
13. Select the "Disconnect" button.
14. Wait 11 minutes.
15. Select the controller and then select the "Stop" button to stop discovering accessory servers. Then select "Start" to begin again.
16. Using the "Thread Discovery'' view in the Trace window, verify the accessory is no longer advertising.

17. Place the accessory back into pairing mode by performing a power cycle, button press or other explicit user action.
18. Using the "Thread Discovery'' view in the Trace window, verify the accessory begins advertising again.
19. Wait 11 minutes.
20. Select the controller and then select the "Stop" button to stop discovering accessory servers. Then select "Start" to begin again.
21. Using the "Thread Discovery'' view in the Trace window, verify the accessory is no longer advertising.

TCT005 **Verify that the thread accessory implements a MTD (Minimum Thread Device) role if battery powered, a FTD (Full Thread Device) role if AC-powered, and that the "Node Capabilities" characteristic reflects this capability.**

Applies to accessories using the HAP over Thread transport. Perform this test case with HAT using the steps below.

1. Select the "+" at the bottom of left sidebar and select "Create Thread Controller" to make a new virtual Thread controller. In the Summary panel of the controller, click the "Start'' button to begin discovering Thread accessories.
2. Connect accessory to Thread network and ensure the accessory's advertisements are present within the "Thread Discovery'' view in the trace window.
3. Pair and discover the accessory on thread network using Thread Controller.
4. Navigate to "Thread Transport" service, select "Node Capabilities" characteristic, and read the value.
5. Navigate to Events view in HAT trace. Verify in Details view of the "Characteristic Read Completed". Verify the value of the characteristic read above is at least "8", implying that it is a Full Thread Device, if the accessory is plugged into AC power-source. If the accessory is battery powered, the value can be lower than "8", implying that it is a Minimal Thread Device (Minimal End Device or Sleepy End Device).

TCT006 **Verify the accessory advertises on the network via Bonjour using "_hap._udp".**

Applies to accessories using the HAP over Thread transport. Perform this test case with HAT using the steps below.

1. Select the "+" at the bottom of left sidebar and select "Create Thread Controller" to make a new virtual Thread controller. In the Summary panel of the controller, click the "Start'' button to begin discovering Thread accessories.
2. Connect accessory to Thread network and ensure the accessory's advertisements are present within the "Thread Discovery'' view in the trace window.
3. Navigate to the "Thread Discovery" view in the HAT Trace and verify the accessory's advertisement(s) are present.

TCT008 **Verify that "Pair Setup" can be completed successfully over Thread transport.**

Applies to accessories using the HAP over Thread transport. Not Applicable to accessories that clear thread credentials when last admin pairing is removed. Perform this test case with HAT using the steps below.

1. Select the "+" at the bottom of left sidebar and select "Create Thread Controller" to make a new virtual Thread controller. In the Summary panel of the controller, click the "Start'' button to begin discovering Thread accessories.
2. Connect accessory to Thread network and ensure the accessory's advertisements are present within the "Thread Discovery'' view in the trace window.
3. Pair and discover the accessory on thread network.
4. In left sidebar of Controllers window, select accessory's name.
5. Navigate to Pairing panel and click on "Remove Pairing" to unpair the accessory from Thread Controller.
6. Pair the accessory over thread.
7. In HAP Traffic view of HAT trace, select each "Pair Setup" response and verify that it uses "url: /1".

TCT009 **Verify that "Pair Verify" completes successfully and uses the "/2" URL to establish a new security session.**

Applies to accessories using the HAP over Thread transport. Perform this test case with HAT using the steps below.

1. Select the "+" at the bottom of left sidebar and select "Create Thread Controller" to make a new virtual Thread controller. In the Summary panel of the controller, click the "Start'' button to begin discovering Thread accessories.
2. Connect accessory to Thread network and ensure the accessory's advertisements are present within the "Thread Discovery'' view in the trace window.
3. Pair and discover the accessory.
4. Navigate to the Accessory's Connection panel and click on "Pair Verify" button.
5. Using the HAP Traffic view of the trace window, select the accessory's Pair-Verify responses, select "Details" to show the details, and verify the messages use the "/2" URL.
6. In Events view of HAT Trace, verify the presense of "Pair-Verify - M1" through "Pair-Verify - M4" trasactions and "Pair-Verify Completed" response.

TCT010 **Verify that the "Thread Role" characteristic value accurately reflects the current role of an accessory in the thread network.**

Applies to accessories using the HAP over Thread transport. Perform this test case with HAT using the steps below.

1. Select the "+" at the bottom of left sidebar and select "Create Thread Controller" to make a new virtual Thread controller. In the Summary panel of the controller, click the "Start'' button to begin discovering Thread accessories.
2. Connect accessory to Thread network and ensure the accessory's advertisements are present within the "Thread Discovery'' view in the trace window.
3. Pair and discover the accessory on thread network using Thread Controller.
4. Navigate to "Thread Transport" service, select "Thread Role" characteristic and read the value.
5. In Events view of the trace window, select the read response, select "Details" to show the details, and verify the the value corresponds to one of the supported thread roles, such as "64" (Border Router), "32" (Leader), "16" (Router), "8" (Child), "4" (Joining), "2" (Detached) or, "1" (Disabled).

TCT011 **Verify that FTD-type accessories are able to receive and process HAP requests at any time after the HAP service has been registered, and are able to send back responses and notifications immediately.**

Applies to accessories using the HAP over Thread transport. Applies to AC-powered Thread accessories - Full Thread Device (FTD) only. Perform this test case with HAT using the steps below.

1. Select the "+" at the bottom of left sidebar and select "Create Thread Controller" to make a new virtual Thread controller. In the Summary panel of the controller, click the "Start'' button to begin discovering Thread accessories.
2. Connect accessory to Thread network and ensure the accessory's advertisements are present within the "Thread Discovery'' view in the trace window.
3. Pair and discover the accessory on thread network using Thread Controller.
4. In left side bar of Controllers window, select accessory's name. Under Summary, select Paired Identify button.
5. Verify from "CoAP Traffic" section of HAT trace that the accessory response to the Identify Request is within 200 milliseconds of the Request.
6. Enable Notifications on a characteristic that supports notifications, and while looking at the Events view in HAT trace, trigger a status change on the hardware. Verify that a notification is received immediately (approximately, within 1 second.)

TCT012 **Verify that MTD-type accessories can receive and process HAP requests when they wake up, but still must send status change notifications immediately, regardless of state.**

Applies to accessories using the HAP over Thread transport. Applies to battery powered Thread accessories - Minimal Thread Device (MTD) only. Perform this test case with HAT using the steps below.

1. Select the "+" at the bottom of left sidebar and select "Create Thread Controller" to make a new virtual Thread controller. In the Summary panel of the controller, click the "Start'' button to begin discovering Thread accessories.
2. Connect accessory to Thread network and ensure the accessory's advertisements are present within the "Thread Discovery'' view in the trace window.

3. Pair and discover the accessory on thread network using Thread Controller.
4. Navigate to the "Accessory Runtime Information" Service and perform a read on the "Sleep Interval" characteristic. Note the value for use in the next step.
5. In left side bar of the Controllers window, select the accessory's name. Under the Summary panel, select the "Paired Identify'' button. Verify from "CoAP Traffic" view of HAT trace that the accessory response to the Identify Request is within (SleepInterval + 200 milliseconds) of the Request.
6. Enable Notifications on a characteristic that supports notifications, and while observing the Events view in HAT trace, trigger a status change on the hardware, if applicable. Verify that a notification is received immediately (approximately, within 1 second, subjectively.) Accessory should send the notification immediately, when status is changed without delaying it until it wakes up from sleep state.

TCT013 **Verify that a thread accessory supports refreshing of security session when a controller performs a pair-verify with the accessory again and that any existing active session is terminated.**

Applies to accessories using the HAP over Thread transport. Perform this test case with HAT using the steps below.

1. Select the "+" at the bottom of left sidebar and select "Create Thread Controller" to make a new virtual Thread controller. In the Summary panel of the controller, click the "Start'' button to begin discovering Thread accessories.
2. Connect accessory to Thread network and ensure the accessory's advertisements are present within the "Thread Discovery'' view in the trace window.
3. Pair and discover the accessory on Thread network using Thread Controller.
4. View the HAT trace and verify that the controller is able to successfully establish a Pair-Verify session with accessory.
5. Quit and relaunch HAT Application.
6. Discover the accessory using Thread controller in HAT.
7. View the HAT trace and verify that the controller is able to successfully establish a Pair-Verify session with accessory.

TCT014 **Verify that a thread accessory has a "Thread Transport" service and all required characteristics.**

Applies to accessories using the HAP over Thread transport. Perform this test case with HAT using the steps below.

Required characteristics:

- Node Capabilities (r)
- Thread Role (r/ev*)
- Thread Control Point (r/w/wr)
- OpenThread Version (r)

- Current Transport (r)

Optional characteristics:

- Received Signal Strength Indication (r)
- Signal-to-Noise-Ratio (r)
- Transmit Power (r)
- Transmit Power Maximum (r)
- Receiver Sensitivity (r)
- CCA Signal Detect Threshold (r)
- CCA Energy Detect Threshold (r)
- MAC Retransmission Maximum (r)
- MAC Transmission Counters (r)
- Event Transmission Counters (r)
- Event Retransmission Maximum (r)

* Notify (ev) for BLE encompasses Indicate, Indicate (disconnected), and Broadcast. Please note, characteristics with TLV8 or String formats must not support Broadcast Events.

1. Select the "+" at the bottom of left sidebar and select "Create Shared Key Store".
2. Select the "+" at the bottom of left sidebar and select "Create BLE Controller" to make a new virtual BLE controller. In the Summary panel of the controller, assign the Shared Key Store created in Step 1, and click on the "Start'' button to begin discovering BLE accessories.
3. Select the "+" at the bottom of left sidebar and select "Create Thread Controller" to make a new virtual Thread controller. In the Summary panel of the controller, assign the Shared Key Store created in Step 1, and click on the "Start'' button to begin discovering Thread accessories.
4. Pair and discover accessory using BLE controller.
5. In left side bar of Controllers window, see each of accessory's services.
6. Verify required characteristics are included for each supported service type.
7. Using the Read and Write buttons in Controllers window, read each paired Read characteristic and write to each paired Write characteristic.
8. Verify proper values are returned for all Read characteristics and that all Writes characteristics properly update accessory's current state.
9. With the Thread network present, navigate to "Thread Transport" service, select "Thread Control Point" characteristic, click "Build TLV" under "Write [tlv8]" panel, select "Set Thread Parameters" from the drop-down, enter the details of the thread border router network, enter "0" in the "Forming Allowed" field, and click "Build TLV".
10. Select "Write" to send the TLV to the accessory and check Thread Discovery view of HAT Trace. Accessory must connect to the Border Router immediately and show its Bonjour advertisement in Thread Discovery view.

11. Pair and discover the accessory on thread network using Thread Controller.
12. Repeat steps 5 through 8 using Thread Controller.

TCT015 **Verify the functionality of the Thread Control Point characteristic's "Set Thread Parameters" with "Forming Allowed" set to 1, for an Always-On accessory.**

Applies to accessories using the HAP over Thread transport. Applies to AC-powered Thread accessories - Full Thread Device (FTD) only. Perform this test case with HAT using the steps below.

1. Select the "+" at the bottom of left sidebar and select "Create Shared Key Store".
2. Select the "+" at the bottom of left sidebar and select "Create BLE Controller" to make a new virtual BLE controller. In the Summary panel of the controller, assign the Shared Key Store created in Step 1, and click on the "Start" button to begin discovering BLE accessories.
3. Select the "+" at the bottom of left sidebar and select "Create Thread Controller" to make a new virtual Thread controller. In the Summary panel of the controller, assign the Shared Key Store created in Step 1, and click on the "Start" button to begin discovering Thread accessories.
4. Pair discover the accessory using BLE Controller.
5. In left side bar of Controllers window, view the accessory's services and select Thread Transport service. Under "Write" panel of Thread Control Point Characteristic, select "Build TLV". Click on the drop-down to select "Read Thread Parameters" and click on "Build TLV". In "Options" panel, select the "Write with response" option.
6. Select "Write" to send the TLV to the accessory and check Events view in HAT Trace to verify that the "Response Value" returned contains a Raw Value of "0" bytes.
7. With the Thread network present, navigate to "Thread Transport" service, select "Thread Control Point" characteristic, click "Build TLV" under "Write [tlv8]" panel, select "Set Thread Parameters" from the drop-down, enter the details of the Thread border router network, enter "1" in the "Forming Allowed" field, and click "Build TLV".
8. Select "Write" to send the TLV to the accessory and check Thread Discovery view of HAT Trace. Accessory must connect to the Border Router immediately and show its Bonjour advertisement in Thread Discovery view.
9. Discover the accessory using Thread Controller to verify that the accessory has joined the Thread network.
10. In left side bar of Controllers window, view the accessory's services and select the "Thread Control Characteristic" under "Thread Transport Service".
11. Within the "Write" panel of Thread Control Point characteristic, click "Build TLV", select "Read Thread Parameters" and click on "Build TLV". In "Options" panel, select the "Write with Response" option and Write the TLV. Verify that the response in HAP Traffic view in HAT Trace contains Thread Parameters entered in Step 7 above.
12. Navigate to Thread Transport service, select Thread Control Point characteristic, click "Build TLV" under "Write [tlv8]" panel, select "Clear Thread Parameters" option from the drop-down and click "Build TLV".

13. Select "Write" to send the TLV to the accessory and then check Thread Discovery view of HAT Trace. Wait for the accessory to disconnect from the Border Router.

14. Look for a BLE advertisement from accessory in BLE Discovery of HAT Trace and verify accessory sends a BLE advertisement within 65 seconds after the Write request in previous Step.

15. Repeat Steps 1 through 8 while the Thread router used in Step 7 is powered-off, or removed and verify that, after the Write request in Step 8, the accessory cannot find the thread network, falls back to BLE and sends a BLE advertisement within 65 seconds.

TCT016 **Verify that the characteristics within "Thread Transport Service" can only be read from or written to by an admin user and not by a shared non-admin user.**

Applies to accessories using the HAP over Thread transport. Perform this test case with HAT using the steps below.

1. Select the "+" at the bottom of left sidebar and select "Create Shared Key Store".

2. Select the "+" at the bottom of left sidebar and select "Create BLE Controller" to make a new virtual BLE controller, Controller1. In the Summary panel of the controller, assign the Shared Key Store created in Step 1, and click the "Start" button to begin discovering BLE accessories.

3. Select the "+" at the bottom of left sidebar and select "Create Thread Controller" to make a new virtual Thread controller, Controller2. In the Summary panel of the controller, assign the Shared Key Store created in Step 1, and click the "Start" button to begin discovering Thread accessories.

4. Pair discover the accessory using BLE Controller.

5. With the Thread network present, navigate to "Thread Transport" service, select "Thread Control Point" characteristic, click "Build TLV" under "Write [tlv8]" panel, select "Set Thread Parameters" from the drop-down menu, enter the details of the thread border router network, enter "0" in the "Forming Allowed" field, and click "Build TLV".

6. Select "Write" to send the TLV to the accessory and check Thread Discovery view of HAT Trace. Accessory must connect to the Border Router immediately and show its Bonjour advertisement in Thread Discovery view.

7. Once accessory is connected to the Thread network, discover the accessory on the thread network using Thread Controller2 on HAT.

8. Add a second thread controller in HAT, Controller3. On Controller2, navigate to the accessory on the left, uncheck Admin option in "Add Additional Controller" panel for Controller 3, and click "Add Controller" to add it as a non-admin controller.

9. Using Controller3, navigate to Thread Transport Service, select "Open Thread Version" characteristic. Read the value of the characteristic.

10. View HAP Traffic and verify that accessory responds with "Status: Invalid Request (0x06)".

11. Repeat Steps 9 and 10 for each of the characteristics under "Thread Transport Service", and verify that the accessory responds with "Status: Invalid Request (0x06)" for read requests on these characteristics, except "Current Transport" characteristic where accessory should return a "Success(0x00". Non-admin Controller should be able to read "Current Transport" characteristic.

12. For the above list of characteristics under “Thread Transport Service”, attempt Writes using Controller3. Verify writes result in “Status: Invalid Request (0x06)”, if the characteristic has Write permissions. If the characteristic does not have Write permissions, it should respond with “Status = Unsupported PDU 0x01”.

13. For the list of characteristics within Thread Transport Service, attempt reads and writes using Controller2 and verify that all read/write attempts succeed.

TCT017 **Verify that the functionality of the Thread Control Point characteristic’s “Set Thread Parameters” with “Forming Allowed” set to 0.**

Applies to accessories using the HAP over Thread transport. Perform this test case with HAT using the steps below.

1. Select the “+” at the bottom of left sidebar and select “Create Shared Key Store”.
2. Select the “+” at the bottom of left sidebar and select “Create BLE Controller” to make a new virtual BLE controller. In the Summary panel of the controller, assign the Shared Key Store created in Step 1, and click on the “Start” button to begin discovering BLE accessories.
3. Select the “+” at the bottom of left sidebar and select “Create Thread Controller” to make a new virtual Thread controller. In the Summary panel of the controller, assign the Shared Key Store created in Step 1, and click on the “Start” button to begin discovering Thread accessories.
4. Pair discover the accessory using BLE Controller.
5. On the left side bar of Controllers window, view the accessory’s services and select Thread Transport service. Under “Write” panel of Thread Control Point Characteristic, select “Build TLV”. Click on the drop-down to select “Read Thread Parameters” and click on “Build TLV”. In “Options” panel, select the “Write with response” option.
6. Select “Write” to send the TLV to the accessory and check Events view in HAT Trace to verify that the “Response Value” returned contains a Raw Value of “0” bytes.
7. With the Thread network present, navigate to “Thread Transport” service, select “Thread Control Point” characteristic, click “Build TLV” under “Write [tlv8]” panel, select “Set Thread Parameters” from the drop-down, enter the details of the Thread border router network, enter “0” in the “Forming Allowed” field, and click “Build TLV”.
8. Select “Write” to send the TLV to the accessory and check Thread Discovery view of HAT Trace. Accessory must connect to the Border Router immediately and show its Bonjour advertisement in Thread Discovery view of HAT Trace.
9. Discover the accessory using Thread Controller to verify that the accessory has joined the Thread network.
10. In left side bar of Controllers window, view the accessory’s services and select the “Thread Control Characteristic” under “Thread Transport Service”.
11. Within the “Write” panel of Thread Control Point characteristic, click “Build TLV”, select “Read Thread Parameters” and click on “Build TLV”. In “Options” panel, select the “Write with Response” option and Write the TLV. Verify that the response in HAP Traffic view in HAT Trace contains Thread Parameters entered in Step 7 above.

12. Navigate to Thread Transport service, select Thread Control Point characteristic, click “Build TLV” under “Write [tlv8]” panel, select “Clear Thread network credentials” option from the drop-down and click “Build TLV”.

13. Select “Write” to send the TLV to the accessory and check Thread Discovery view of HAT Trace. For this request, note the timestamp of the accessory response in Events view of HAT Trace.

14. Look for a BLE advertisement from accessory in BLE Discovery View of HAT Trace and verify accessory sends a BLE advertisement within 65 seconds after the Write request in previous Step.

15. Repeat Steps 1 through 8 while the Thread router used in Step 7 is powered-off/disabled/removed. View BLE Discovery in HAT Trace and verify that advertisements from this accessory are seen within 65 seconds from time stamp noted in Step 13.

TCT019 **Verify the functionality of the Thread Control characteristic’s “Set Thread Parameters” when the accessory is already connected to a thread network.**

Applies to accessories using the HAP over Thread transport. Perform this test case with HAT using the steps below.

1. Select the “+” at the bottom of left sidebar and select “Create Thread Controller” to make a new virtual Thread controller. In the Summary panel of the controller, click the “Start’’ button to begin discovering Thread accessories.

2. Connect accessory to Thread network and ensure the accessory’s advertisements are present within the “Thread Discovery’’ view in the trace window.

3. Pair and discover the accessory on Thread network using Thread Controller.

4. In left side bar of Controllers window, view the accessory’s services and select the “Thread Control” characteristic under “Thread Transport Service”.

5. Under “Write” panel of Thread Control Point Characteristic, click “Build TLV”. Select “Read Thread Parameters” and click on “Build TLV”. In “Options” panel, select the “Write with response” option. Write the TLV and verify the response in HAP Traffic view of HAT Trace contains the Parameters of the thread network the accessory is connected to.

6. Bring up a second border router that uses different Thread credentials.

7. Navigate to Thread Transport Service. Select Thread Control Point characteristic and click “Build TLV” under “Write [tlv8]” panel. Select “Set Thread Parameters” from the drop-down, enter the details of the second Thread border router network, enter “0” in the Forming Allowed field, and click “Build TLV”.

8. Select “Write” to send the TLV to the accessory. Verify in HAP Traffic View, that the accessory responds with a “Status: Invalid Request (0x06)” in the response for the Write request.

9. Navigate to Thread Transport Service. Select Thread Control Point characteristic and click “Build TLV” under “Write [tlv8]” panel. Select “Initiate Joiner Mode” from the drop-down and click “Build TLV”.

10. Select “Write” to send the TLV to the accessory. Verify in HAP Traffic View, that the accessory responds with a “Status: Invalid Request (0x06)” in response to the Write request.

TCT020 **The HAP accessory object, with the instance ID of "1", must be the primary HAP accessory object. For bridges, the primary HAP accessory object must be the bridge itself.**

Applies to accessories using the HAP over Thread transport. Perform this test case with HAT using the steps below.

1. Select the "+" at the bottom of left sidebar and select "Create Thread Controller" to make a new virtual Thread controller. In the Summary panel of the controller, click the "Start'' button to begin discovering Thread accessories.
2. Connect accessory to Thread network and ensure the accessory's advertisements are present within the "Thread Discovery'' view in the trace window.
3. Pair and discover the accessory on thread network using Thread Controller.
4. In the Events traffic view, select Discovered Accessories packet.
5. Select the Details button.
6. See Accessories and scroll down to locate "Primary: Yes."
7. Verify that the primary accessory object has instance ID "1."
8. If accessory is a bridge, verify that the bridge itself is the primary accessory object.

TCT021 **Verify that when a sleepy accessory is unpaired, it must switch to wakeful state until it goes over the pair-setup and pair-verify flow and that it can go back to its intended wake up frequency once paired.**

Applies to accessories using the HAP over Thread transport. Applies to battery powered Thread accessories - Minimal Thread Device (MTD) only. Not Applicable to accessories that clear thread credentials when last admin pairing is removed. Perform this test case with HAT using the steps below.

1. Select the "+" at the bottom of left sidebar and select "Create Thread Controller" to make a new virtual Thread controller. In the Summary panel of the controller, click the "Start'' button to begin discovering Thread accessories.
2. Connect accessory to Thread network and ensure the accessory's advertisements are present within the "Thread Discovery'' view in the trace window.
3. Pair and discover the accessory on Thread network using Thread Controller.
4. In the Controllers window, select the "Remove Pairing" button.
5. In left side bar of the Controllers window, select the accessory's name. Under the Summary panel, click the "Unpaired Identify" button and verify from "CoAP Traffic" view of HAT trace that the accessory responds to the request within 700 milliseconds.

TCT022 **When a paired sleepy accessory receives a pair-verify request (e.g., keys expired, or controller is establishing a new session), verify that it switches to wakeful state (of SleepInterval=500 milliseconds) until the pair-verify process is complete and that the queued requests are completed on its next wake up cycle.**

Applies to accessories using the HAP over Thread transport. Applies to battery powered Thread accessories - Minimal Thread Device (MTD) only. Perform this test case with HAT using the steps below.

1. Select the "+" at the bottom of left sidebar and select "Create Thread Controller" to make a new virtual Thread controller. In the Summary panel of the controller, click the "Start'' button to begin discovering Thread accessories.
2. Connect accessory to Thread network and ensure the accessory's advertisements are present within the "Thread Discovery'' view in the trace window.
3. Pair and discover the accessory on Thread network using Thread Controller.
4. In the Accessory Summary View, press "Disconnect"to close the current session.
5. In left side bar of the Controllers window, select the accessory's name. Under the Summary panel, select the "Paired Identify'' button.
6. Verify from "CoAP Traffic" view of HAT trace that the response to the Identify request is within 700 milliseconds after "Pair-Verify Completed".

TCT023 **For a sleepy accessory, verify that an accessory switching to wakeful state due to a Pair-Verify, implements a "graceful" delay of 2 seconds to process any potential commands from Controller before going back to sleep.**

Applies to accessories using the HAP over Thread transport. Applies to battery powered Thread accessories - Minimal Thread Device (MTD) only. Perform this test case with HAT using the steps below.

1. Select the "+" at the bottom of left sidebar and select "Create Thread Controller" to make a new virtual Thread controller. In the Summary panel of the controller, click the "Start'' button to begin discovering Thread accessories.
2. Connect accessory to Thread network and ensure the accessory's advertisements are present within the "Thread Discovery'' view in the trace window.
3. Pair and discover the accessory on thread network using Thread Controller.
4. In the Accessory Summary View, press "Disconnect"to close the current session.
5. With Events view in HAT Trace open, perform Step 7 within 2 seconds of performing Step 6. The idea is to send a request within 2 seconds of receiving "Pair-Verify Completed" in Step 6.
6. In the Accessory Connection View, press "Pair-Verify"to open a session.
7. In left side bar of the Controllers window, select the accessory's name. Under the Summary panel, select the "Paired Identify'' button.
8. Verify from "HAP Traffic" section of HAT trace that the accessory responds to the "Identify" request within 700 milliseconds, if the "Identify" request was received by the accessory within 2 seconds after the Pair-Verify completed.

TCT026 **Verify the accessory can successfully remove all pairings, and perform Pair-Setup again while on the Thread network.**

Applies to accessories using the HAP over Thread transport. Not Applicable to accessories that clear thread credentials when last admin pairing is removed. Perform this test case with HAT using the steps below.

1. Select the "+" at the bottom of left sidebar and select "Create Thread Controller" to make a new virtual Thread controller. In the Summary panel of the controller, click the "Start'' button to begin discovering Thread accessories.
2. Connect accessory to Thread network and ensure the accessory's advertisements are present within the "Thread Discovery'' view in the trace window.
3. Pair and discover the accessory on thread network using Thread Controller.
4. In the Accessory View, under Pairing panel, Select the "Remove Pairing" button.
5. After the pairing has been removed, verify accessory can successfully pair to thread controller again. Discover the accessory using Thread Controller and verify in Events View that Pairing and Discovery is successful.

TCT028 **If the controller provides the credentials using the Thread Transport service, verify that the accessory keeps its BLE connection open and at the same time tries to join the network immediately.**

Applies to accessories using the HAP over Thread transport. Perform this test case with HAT using the steps below.

1. Select the "+" at the bottom of left sidebar and select "Create Shared Key Store".
2. Select the "+" at the bottom of left sidebar and select "Create BLE Controller" to make a new virtual BLE controller. In the Summary panel of the controller, assign the Shared Key Store created in Step 1, and click on the "Start'' button to begin discovering BLE accessories.
3. Select the "+" at the bottom of left sidebar and select "Create Thread Controller" to make a new virtual Thread controller. In the Summary panel of the controller, assign the Shared Key Store created in Step 1, and click on the "Start'' button to begin discovering Thread accessories.
4. Pair discover the accessory using BLE Controller.
5. With a border router powered off, navigate to "Thread Transport" service, select "Thread Control Point" characteristic, click "Build TLV" under "Write [tlv8]" panel, select "Set Thread Parameters" from the drop-down, enter the details of the thread border router network, enter "0" in the "Forming Allowed" field, and click "Build TLV".
6. Select "Write" to send the TLV to the accessory and wait for 65 seconds while the accessory attempts to connect to the Border Router.
7. Using BLE Controller, in left side bar of the Controllers window, select the accessory's name. Under the Summary panel, select the "Paired Identify'' button. In HAP Procedures view of HAT Trace, verify the accessory responds to the Identify Request.
8. Power ON the Border Router and observe the Thread Discovery view of HAT Trace. Wait until a Bonjour Advertisement is seen from the accessory.
9. Once the accessory's advertisement is seen by the Thread Controller, discover the accessory using Thread Controller.

10. Using the Thread Controller, in left side bar of the Controllers window, select the accessory's name. Under the Summary panel, select the "Paired Identify'' button.

11. Using BLE Controller, in left side bar of the Controllers window, select the accessory's name. Under the Summary panel, select the "Paired Identify'' button. In HAP Procedures view of HAT Trace, verify the HAT controller shows "Connecting" attempt to the accessory which should not be successful.

TCT029 **Once the accessory successfully joins the Thread network, verify accessory disables its bluetooth functionality and switches to Thread only mode.**

Applies to accessories using the HAP over Thread transport. Perform this test case with HAT using the steps below.

1. Select the "+" at the bottom of left sidebar and select "Create Shared Key Store".

2. Select the "+" at the bottom of left sidebar and select "Create BLE Controller" to make a new virtual BLE controller. In the Summary panel of the controller, assign the Shared Key Store created in Step 1, and click on the "Start'' button to begin discovering BLE accessories.

3. Select the "+" at the bottom of left sidebar and select "Create Thread Controller" to make a new virtual Thread controller. In the Summary panel of the controller, assign the Shared Key Store created in Step 1, and click on the "Start'' button to begin discovering Thread accessories.

4. Pair discover the accessory using BLE Controller.

5. With the Thread network present, navigate to "Thread Transport" service, select "Thread Control Point" characteristic, click "Build TLV" under "Write [tlv8]" panel, select "Set Thread Parameters" from the drop-down, enter the details of the thread border router network, enter "0" in the "Forming Allowed" field, and click "Build TLV".

6. Select "Write" to send the TLV to the accessory and check Thread Discovery view of HAT Trace. Accessory must connect to the Border Router immediately and show its Bonjour advertisement in Thread Discovery view. Note the timestamp of accessory's first Bonjour advertisement after it joins Thread network.

7. Discover the accessory on the thread network using Thread Controller on HAT.

8. Using Thread Controller, in left side bar of the Controllers window, select the accessory's name. Under the Summary panel, select the "Paired Identify'' button. Verify in CoAP Traffic view that CoAP response from accessory shows "Code: 2.04" for a successful response.

9. In BLE Discovery view of HAT trace, verify that no BLE advertisements are seen from accessory after timestamps noted in Step 6.

TCT030 **If attaching to a Thread network is unsuccessful, the Thread network is not viable, or the accessory gets detached from the Thread network, verify that the accessory switches back to BLE mode and periodically attempts to rejoin the Thread network.**

Applies to accessories using the HAP over Thread transport. Perform this test case with HAT using the steps below.

1. Create a Shared Key Store.
2. Create a BLE Controller in HAT. Select the Shared Key Store from Step 1. Click on “Start” to start the Controller.
3. Create a Thread Controller, using the same Shared Key Store from Step 1. Click on “Start” to start the Controller.
4. Pair discover the accessory using BLE Controller.
5. With the border router powered off, navigate to “Thread Transport” service, select “Thread Control Point” characteristic, click “Build TLV” under “Write [tlv8]” panel, select “Set Thread Parameters” from the drop-down, enter the details of the thread border router network, enter “0” in the “Forming Allowed” field, and click “Build TLV”.
6. Select “Write” to send the TLV to the accessory and wait for more than 65 seconds for accessory to attempt connecting to the Border Router and fall back to BLE.
7. Verify it falls back to BLE by looking for a BLE advertisement in BLE Discovery after the above step.
8. Power on the Border Router and check Thread Discovery view of HAT Trace. Accessory must connect to the Border Router and show its Bonjour advertisement in Thread Discovery view within 6 hours.
9. Discover the accessory on thread network using Thread Controller.
10. Navigate to the Accessory’s Connection panel and click on “Pair Verify” button.
11. In Events view of HAT Trace, verify the presense of “Pair-Verify - M1” through “Pair-Verify - M4” trasactions and “Pair-Verify Completed” response.
12. Power off the Border Router and and wait for 65 seconds for accessory to fall back to BLE.
13. Verify it falls back to BLE by looking for a BLE advertisement in BLE Discovery after the above step.
14. Power on the Border Router after 60 minutes and check Thread Discovery view of HAT Trace. Accessory must connect to the Border Router and show its Bonjour advertisement in Thread Discovery view within 6 hours after Border Router has been powered up.
15. Discover the accessory on thread network using Thread Controller.
16. Navigate to the Accessory’s Connection panel and click on “Pair Verify” button.
17. In Events view of HAT Trace, verify the presense of “Pair-Verify - M1” through “Pair-Verify - M4” trasactions and “Pair-Verify Completed” response.

TCT031 **If an accessory that was connected to a Thread network gets detached, verify that it waits 30 seconds before attempting to switch back to BLE (to help overcome temporary network glitches and avoid unnecessary overhead).**

Applies to accessories using the HAP over Thread transport. Perform this test case with HAT using the steps below.

1. Select the “+” at the bottom of left sidebar and select “Create BLE Controller” to make a new virtual BLE controller. In the Summary panel of the controller, click the “Start’’ button to begin discovering BLE accessories.

2. Select the "+" at the bottom of left sidebar and select "Create Thread Controller" to make a new virtual Thread controller. In the Summary panel of the controller, click the "Start'' button to begin discovering Thread accessories.
3. Pair and discover the accessory on Thread network using Thread Controller.
4. Power off Border Router and then power it back on within 30 seconds.
5. Once the Border Router is online, navigate to the Accessory's Connection panel and click on "Pair Verify" button.
6. In Events view of HAT Trace, verify the presense of "Pair-Verify - M1" through "Pair-Verify - M4" trasactions and "Pair-Verify Completed" response.
7. Verify within BLE Discovery view of HAT Trace that there are no advertisements from accessory after during or after Step 4.

TCT032 **Verify that the transition from BLE to Thread, and vice versa, is fully transparent to users.**

Applies to accessories using the HAP over Thread transport. Perform this test case with HAT using the steps below.

1. Select the "+" at the bottom of left sidebar and select "Create Shared Key Store".
2. Select the "+" at the bottom of left sidebar and select "Create BLE Controller" to make a new virtual BLE controller. In the Summary panel of the controller, assign the Shared Key Store created in Step 1, and click on the "Start'' button to begin discovering BLE accessories.
3. Select the "+" at the bottom of left sidebar and select "Create Thread Controller" to make a new virtual Thread controller. In the Summary panel of the controller, assign the Shared Key Store created in Step 1, and click on the "Start'' button to begin discovering Thread accessories.
4. Pair and discover the accessory using BLE Controller.
5. With the Thread network present, navigate to "Thread Transport" service, select "Thread Control Point" characteristic, click "Build TLV" within "Write [tlv8]" panel, select "Set Thread Parameters" from the drop-down, enter the details of the thread border router network, enter "0" in the "Forming Allowed" field, and click "Build TLV".
6. Select "Write" to send the TLV to the accessory and check Thread Discovery view of HAT Trace. Accessory must connect to the Border Router immediately and show its Bonjour advertisement in Thread Discovery view of HAT trace.
7. Once accessory is connected to the Thread network, discover the accessory on thread network using Thread Controller on HAT.
8. Power off Border Router.
9. Wait for 65s for the accessory to fall back to BLE.
10. Look for a BLE advertisement in BLE Discovery view of HAT trace to verify that it falls back to BLE. Note the time stamps.

11. Power on the Border Router and wait for Accessory to join the thread network. Accessory must connect to the Border Router and show its Bonjour advertisement in Thread Discovery view within 6 hours after Border Router has been powered up. Note the time stamps.
12. Verify time stamps in BLE Discovery View of HAT trace that BLE Advertisements for the accessory are stopped as soon as Bonjour advertisements start showing up in Thread Discovery view in Step 11.
13. Using Thread Controller, on the left side bar of the Controllers window, select the accessory's name. Under the Summary panel, select the Disconnect button.
14. Navigate to the Accessory's Connection panel and click on "Pair Verify" button.
15. In Events view of HAT Trace, verify the presense of "Pair-Verify - M1" through "Pair-Verify - M4" trasactions and "Pair-Verify Completed" response.

TCT033 **When removing the last pairing of an accessory, verify that the accessory transitions into unpaired state but remains connected to the Thread network. In case of a sleepy device, it should change its sleep interval to 500ms during the first 5 minutes to improve the response times in case user wants to pair it again. After the 5 minutes the device can revert back to its original sleep period.**

Applies to accessories using the HAP over Thread transport. Not Applicable to accessories that clear thread credentials when last admin pairing is removed. Perform this test case with HAT using the steps below.

1. Select the "+" at the bottom of left sidebar and select "Create Shared Key Store".
2. Select the "+" at the bottom of left sidebar and select "Create BLE Controller" to make a new virtual BLE controller. In the Summary panel of the controller, assign the Shared Key Store created in Step 1, and click on the "Start" button to begin discovering BLE accessories.
3. Select the "+" at the bottom of left sidebar and select "Create Thread Controller" to make a new virtual Thread controller. In the Summary panel of the controller, assign the Shared Key Store created in Step 1, and click on the "Start" button to begin discovering Thread accessories.
4. Pair and discover the accessory using BLE Controller.
5. With the Thread network present, navigate to "Thread Transport" service, select "Thread Control Point" characteristic, click "Build TLV" within "Write [tlv8]" panel, select "Set Thread Parameters" from the drop-down, enter the details of the thread border router network, enter "0" in the "Forming Allowed" field, and click "Build TLV".
6. Select "Write" to send the TLV to the accessory and check Thread Discovery view of HAT Trace. Accessory must connect to the Border Router immediately and show its Bonjour advertisement in Thread Discovery view of HAT trace.
7. Once accessory is connected to the Thread network, discover the accessory on thread network using Thread Controller on HAT.
8. Navigate to Pairing panel and click on "Remove Pairing" to unpair the accessory using thread controller in HAT and verify that the accessory show its Bonjour advertisement in Thread Discovery view of HAT trace, with "Status Flags: 1"
9. Verify BLE Discovery view of HAT Trace. There must not be an advertisement sent by the accessory over BLE for 10 minutes following Step 8.

10. Power cycle the accessory.
11. Attempt to pair and discover using Thread controller within 10 minutes after above step. Pairing should be successful.

TCT034 **For a Sleepy Accessory, when removing the last pairing of an accessory, verify that the accessory transitions into unpaired state but remains connected to the Thread network. It should change its sleep interval to 500ms during the first 5 minutes to improve the response times in case user wants to pair it again. After the 5 minutes the device may revert back to its original sleep period.**

Applies to accessories using the HAP over Thread transport. Not Applicable to accessories that clear thread credentials when last admin pairing is removed. Perform this test case with HAT using the steps below.

1. Select the "+" at the bottom of left sidebar and select "Create Shared Key Store".
2. Select the "+" at the bottom of left sidebar and select "Create BLE Controller" to make a new virtual BLE controller. In the Summary panel of the controller, assign the Shared Key Store created in Step 1, and click on the "Start'' button to begin discovering BLE accessories.
3. Select the "+" at the bottom of left sidebar and select "Create Thread Controller" to make a new virtual Thread controller. In the Summary panel of the controller, assign the Shared Key Store created in Step 1, and click on the "Start'' button to begin discovering Thread accessories.
4. Pair and discover the accessory using BLE Controller.
5. With the Thread network present, navigate to "Thread Transport" service, select "Thread Control Point" characteristic, click "Build TLV" within "Write [tlv8]" panel, select "Set Thread Parameters" from the drop-down, enter the details of the thread border router network, enter "0" in the "Forming Allowed" field, and click "Build TLV".
6. Select "Write" to send the TLV to the accessory and check Thread Discovery view of HAT Trace. Accessory must connect to the Border Router immediately and show its Bonjour advertisement in Thread Discovery view of HAT trace.
7. Once accessory is connected to the Thread network, discover the accessory on thread network using Thread Controller on HAT.
8. Navigate to Pairing panel and click on "Remove Pairing" to unpair the accessory using thread controller in HAT and verify that the accessory show its Bonjour advertisement in Thread Discovery view of HAT trace, with "Status Flags: 1"
9. Verify BLE Discovery view of HAT Trace. There must not be an advertisement sent by the accessory over BLE for 10 minutes following Step 8.
10. Power cycle the accessory.
11. Perform Unpaired Identify on the accessory. Verify in HAP Traffic view of HAT Trace that a response for the "Unpaired Identify" request is received within 700ms.
12. Pair and discover using Thread controller within 5 minutes after Step 10. Pairing should be successful.

TCT035 **Verify that Paired accessories use a DNS-Advertisement expiration timeout of 1 hour.**

Applies to accessories using the HAP over Thread transport. Perform this test case automatically with HCA. Below are steps that can be used to perform this test case manually in HAT.

1. Select the "+" at the bottom of left sidebar and select "Create Thread Controller" to make a new virtual Thread controller. In the Summary panel of the controller, click the "Start'' button to begin discovering Thread accessories.
2. Connect accessory to Thread network and ensure the accessory's advertisements are present within the "Thread Discovery'' view in the trace window.
3. Pair and discover the accessory on thread network using Thread Controller.
4. Power cycle the Accessory and check Thread Discovery view of HAT Trace. Once the accessory connect to the Border Router, note the time stamp of its Bonjour advertisement in Thread Discovery view.
5. Power off the Accessory.
6. Wait for 55 minutes after Step 4 and verify using HAT under thread controller that the accessory is still listed under "Accessory Servers".
7. Wait for more than 60 minutes after Step 4. Verify using HAT under thread controller that the accessory is NOT listed under "Accessory Servers". HAT may show "Accessory may not be present".

TCT036 **Verify that Unpaired accessories use a DNS-Advertisement expiration timeout of 30 seconds.**

Applies to accessories using the HAP over Thread transport. Not Applicable to accessories that clear thread credentials when last admin pairing is removed. Perform this test case with HAT using the steps below.

1. Select the "+" at the bottom of left sidebar and select "Create Thread Controller" to make a new virtual Thread controller. In the Summary panel of the controller, click the "Start'' button to begin discovering Thread accessories.
2. Connect accessory to Thread network and ensure the accessory's advertisements are present within the "Thread Discovery'' view in the trace window.
3. Pair and discover the accessory on thread network using Thread Controller.
4. Navigate to Pairing panel and click on "Remove Pairing" to unpair the accessory using thread controller in HAT and verify that the accessory is still listed under "Accessory Servers".
5. Power cycle or reboot the Accessory and check Thread Discovery view of HAT Trace. Once the accessory connect to the Border Router, note the time stamp of its Bonjour advertisement in Thread Discovery view.
6. Power off the Accessory.
7. Wait for more than 30 seconds after Step 6. Verify using HAT under thread controller that the accessory is NOT listed under "Accessory Servers".
8. Power on the Accessory.

9. Wait for accessory to boot up and connect to the Border Router. Verify its Bonjour advertisement in Thread Discovery view of HAT Trace and that the accessory is listed under "Accessory Servers" under a thread controller in HAT.

10. Wait for more than 10 minutes after Step 8. Verify using HAT under thread controller that the accessory is NOT listed under "Accessory Servers".

11. Power cycle or reboot the Accessory.

12. Wait for accessory to boot up and connect to the Border Router. Verify its Bonjour advertisement in Thread Discovery view of HAT Trace and that the accessory is listed under "Accessory Servers" under a thread controller in HAT.

TCT038 **The minimum number of pairing relationships that an accessory must support is 16.**

Applies to accessories using the HAP over Thread transport. Perform this test case automatically with HCA. Below are steps that can be used to perform this test case manually in HAT.

1. Select the "+" at the bottom of left sidebar and select "Create Thread Controller" to make a new virtual Thread controller. In the Summary panel of the controller, click the "Start'' button to begin discovering Thread accessories.

2. Connect accessory to Thread network and ensure the accessory's advertisements are present within the "Thread Discovery'' view in the trace window.

3. Pair and discover the accessory on thread network using Thread Controller 1.

4. Select the "+" at the bottom of left sidebar.

5. Select "Create Thread Controller" to create 15 additonal controllers, for a total of 16 controllers.

6. On the left pane of the Controllers window, using admin Controller 1, add pairings to each of the 15 secondary controllers.

7. Select "List Pairings" button in the Controllers window.

8. In the Event view of the Trace window, verify the response to "List Pairings Completed" event shows all 16 pairings.

9. On the left pane of the Controllers window, select Controller 2, select the "Start" button, select the accessory name, and select the "Discover" button.

10. In the Event view of the Trace window, verify for Controller 2., that the "Discovered Accessories" response shows the accessory's Services and Characteristics successfully.

11. Repeat Steps 9-10 for each additional controller.

TCT039 **Accessories must be able to support 8 controllers simultaneously.**

Applies to accessories using the HAP over Thread transport. Perform this test case automatically with HCA. Below are steps that can be used to perform this test case manually in HAT.

1. Select the "+" at the bottom of the left sidebar and select "Create Thread Controller" to make a new virtual Thread controller. In the Summary panel of the controller, click the "Start'' button to begin discovering Thread accessories.
2. Connect the accessory to the Thread network and ensure that the accessory's advertisements are present within the "Thread Discovery'' view in the trace window.
3. Pair and discover the accessory on the Thread network using Thread Controller 1.
4. Select the "+" at the bottom of the left sidebar.
5. Select "Create Thread Controller" to create 7 additional controllers, for a total of 8 controllers.
6. On the left pane of the Controllers window, using admin Controller 1, add pairings to each of the 7 secondary controllers.
7. In left sidebar, select Controller 2. Select the "Start" button, select the accessory, and then select the "Discover" button. Verify that the "Discovered Accessories" response for Controller 2 shows the accessory's Services and Characteristics successfully.
8. Repeat step 7 with each additional controller. Verify that the "Discovered Accessories" response for each controller shows the accessory's Services and Characteristics successfully.
9. In left side bar of the Controllers window, select Controller 1, then select the accessory's name. Under the Summary panel, select the "Paired Identify'' button. In then HAP Traffic view of HAT Trace, verify the accessory responds to the Identify Request with a "Status: Success (0x00)". The accessory must not require the Controller to perform a Pair-Verify (other than the Pair-Verify performed during step 8) to respond to the Identify request.
10. Repeat step 9 for each additional controller.

TCT041 **The accessory must support event notifications for multiple controllers.**

Applies to accessories using the HAP over Thread transport. Perform this test case with HAT using the steps below.

1. Select the "+" at the bottom of left sidebar and select "Create Thread Controller" to make a new virtual Thread controller. In the Summary panel of the controller, click the "Start'' button to begin discovering Thread accessories.
2. Connect accessory to Thread network and ensure the accessory's advertisements are present within the "Thread Discovery'' view in the trace window.
3. Pair and discover the accessory on thread network using Thread Controller 1.
4. Select the "+" at the bottom of left sidebar and then select "Create Thread Controller" to make a new controller.
5. In left sidebar, select the Controller 1 and select the accessory name.
6. Under the "Add Additional Controllers" panel, select "Controller 2" as Controller, check the box "on" for Admin, and select the "Add Controller" button.
7. In left sidebar, select the Controller 2, select the accessory name, and select the "Discover" button.

8. Select the "+" at the bottom of left sidebar and then select "Create Thread Controller" to make a new controller.
9. In left sidebar, select the Controller 1, and select the accessory name.
10. In Controllers window, under "Add Additional Controllers" panel, select "Controller 3" as Controller, check the box "on" for Admin, and select the "Add Controller" button.
11. In left sidebar, select the Controller 3, select the accessory name, and select the "Discover" button.
12. Select accessory name under Controller 1, navigate to each characteristic that supports the "Notify" permission, and select the "Enable" button to enable event notifications for each characteristic.
13. Repeat step 12 for Controllers 2 and 3.
14. Using Controller 1, select the first characteristic that supports the "Notify" permission.
15. If the characteristic supports the "Paired Write" permission, write a valid value other than the current value to the characteristic.
16. Using the HAP Traffic view of HAT trace window, verify "Notifications" were only sent to Controller 2 and Controller 3.
17. Click on Details and verify the notification contains the correct value, characteristic IID, and AID.
18. If the characteristic supports the "Paired Read" permission, and the characteristic's state can be changed outside of HAP (e.g., by physical means of interaction, remote control, or app) change the state of the accessory once (e.g., turn light on).
19. Verify that "Notifications" were sent to all controllers; Controller 1, Controller 2, and Controller 3.
20. Verify the notifications contain the correct value, and IID.
21. Repeat Steps 14-18 for each characteristic that supports the "Notify" permission.

TCT043 **The accessory must support writing values to one or more characteristics via a single CoAP request.**

Applies to accessories using the HAP over Thread transport. Perform this test case with HAT using the steps below.

1. Select the "+" at the bottom of left sidebar and select "Create Thread Controller" to make a new virtual Thread controller. In the Summary panel of the controller, click the "Start'' button to begin discovering Thread accessories.
2. Connect accessory to Thread network and ensure the accessory's advertisements are present within the "Thread Discovery'' view in the trace window.
3. Pair and discover the accessory on thread network using Thread Controller.
4. In Controllers window, select any characteristic.
5. In the Queued Writing Panel, Enable Queue State.
6. Write value in the Queue Characteristic Panel and select Queue button.
7. Select another characteristic.

8. Write value in the Queue Characteristic Panel to a read-only characteristic and select Queue button.
9. In the pop-up window, select the Write button.
10. View accessory response in the HAP traffic view and select Details button.
11. Verify the accessory responds with Status: Success (0x00) for a successful write and with a status: Unsupported PDU (0x01) for an unsuccessful write.

TCT044 **Services contained within the HAP accessory must be colocated. For example, a fan with a light on it would expose a single HAP accessory with 3 services: the required accessory information service, a fan service, and a lightbulb service. Conversely, a bridge that bridges 2 independent lights, which may be in different physical locations, must expose a HAP accessory object for each independent light.**

Applies to accessories using the HAP over Thread transport. Perform this test case with HAT using the steps below.

1. Select the "+" at the bottom of left sidebar and select "Create Thread Controller" to make a new virtual Thread controller. In the Summary panel of the controller, click the "Start'' button to begin discovering Thread accessories.
2. Connect accessory to Thread network and ensure the accessory's advertisements are present within the "Thread Discovery'' view in the trace window.
3. Pair and discover the accessory on thread network using Thread Controller.
4. Locate accessory's response to Discover accessories in the Events view.
5. Select the Details button.
6. See Attribute Database Object. Use disclosure arrows to show and hide details.
7. If accessory server contains 1 physical accessory with more than 1 service, verify that its services are located within 1 accessory object.
8. If accessory server contains more than 1 physical accessory, verify there is 1 accessory object for each physical accessory.

TCT045 **Accessories must supply required properties in each service object: Type, Instance ID, Characteristics, Hidden Services (Conditional), Primary Services (Optional), and Linked Services (Optional).**

Applies to accessories using the HAP over Thread transport. Perform this test case with HAT using the steps below.

1. Select the "+" at the bottom of left sidebar and select "Create Thread Controller" to make a new virtual Thread controller. In the Summary panel of the controller, click the "Start'' button to begin discovering Thread accessories.
2. Connect accessory to Thread network and ensure the accessory's advertisements are present within the "Thread Discovery'' view in the trace window.
3. Pair and discover the accessory on thread network using Thread Controller.

4. In the Events view of the trace window, select the "Discovered Accessories" response.
5. Select the "Details" button to reveal the details within the response.
6. Within each service object, verify that the following properties are provided: Type, Instance ID, Characteristics, Hidden Services (Conditional), Primary Services (Optional), and Linked Services (Optional).
7. Under the type property, verify descriptive names are shown for Apple-defined UUIDs (e.g., "Accessory Information Service" is listed for UUID "0000003E-0000-1000-8000-0026BB765291").

TCT046 **The accessory must include the following properties for each characteristic that supports paired read: Type, Instance ID, Permissions, Value, Format.**

Applies to accessories using the HAP over Thread transport. Perform this test case with HAT using the steps below.

1. Select the "+" at the bottom of left sidebar and select "Create Thread Controller" to make a new virtual Thread controller. In the Summary panel of the controller, click the "Start'' button to begin discovering Thread accessories.
2. Connect accessory to Thread network and ensure the accessory's advertisements are present within the "Thread Discovery'' view in the trace window.
3. Pair and discover the accessory on thread network using Thread Controller.
4. Locate accessory's response to Discover accessories in the Events view.
5. Select the Details button.
6. Within each characteristic object, verify that the following properties are provided: type, instance ID, permissions, value, and format.
7. Under the type property, verify descriptive names are shown for Apple-defined UUIDs (e.g., "Identify" Characteristic is listed for UUID "00000014-0000-1000-8000-0026BB765291").

TCT047 **The accessory must include the following properties for each characteristic that does not support paired read: Type, Instance ID, Permissions, and Format.**

Applies to accessories using the HAP over Thread transport. Perform this test case with HAT using the steps below.

1. Select the "+" at the bottom of left sidebar and select "Create Thread Controller" to make a new virtual Thread controller. In the Summary panel of the controller, click the "Start'' button to begin discovering Thread accessories.
2. Connect accessory to Thread network and ensure the accessory's advertisements are present within the "Thread Discovery'' view in the trace window.
3. Pair and discover the accessory on thread network using Thread Controller.
4. Locate accessory's response to Discover accessories in the Events view.
5. Select the Details button.

6. Within each characteristic object, verify that the following properties are provided: type, instance ID, permissions, and format.

7. Under the type property, verify descriptive names are shown for Apple-defined UUIDs (e.g., "Identify" Characteristic is listed for UUID "00000014-0000-1000-8000-0026BB765291").

TCT048 **For characteristics that do not support notifications, verify correct error response when trying to enable notifications.**

Applies to accessories using the HAP over Thread transport. Perform this test case with HAT using the steps below.

1. Select the "+" at the bottom of left sidebar and select "Create Thread Controller" to make a new virtual Thread controller. In the Summary panel of the controller, click the "Start'' button to begin discovering Thread accessories.

2. Connect accessory to Thread network and ensure the accessory's advertisements are present within the "Thread Discovery'' view in the trace window.

3. Pair and discover the accessory on thread network using Thread Controller.

4. From the Controllers Window, locate each characteristic that does not support "ev" (i.e., event notifications) in the permissions key. Select the enable notifications button.

5. In the HAP traffic view, verify accessory responds with "Invalid Request (0x06)".

TCT049 **The notification event must be generated on the characteristic that notifications are enabled on and not generated on another characteristic.**

Applies to accessories using the HAP over Thread transport. Perform this test case with HAT using the steps below.

1. Select the "+" at the bottom of left sidebar and select "Create Thread Controller" to make a new virtual Thread controller. In the Summary panel of the controller, click the "Start'' button to begin discovering Thread accessories.

2. Connect accessory to Thread network and ensure the accessory's advertisements are present within the "Thread Discovery'' view in the trace window.

3. Pair and discover the accessory on thread network using Thread Controller.

4. In the Controllers window, locate characteristics that support Event Notifications and select enable.

5. Make a change from accessory to each of the characteristics that has notifications enabled.

6. Verify that the notification event is generated on each characteristic that the notifications are enabled on.

7. Verify that notifications are not generated on other characteristics.

TCT050 **The name of the Bonjour service (i.e., the user-visible name of accessory) must match the accessory name.**

Applies to accessories using the HAP over Thread transport. Perform this test case with HAT using the steps below.

1. Select the "+" at the bottom of left sidebar and select "Create Thread Controller" to make a new virtual Thread controller. In the Summary panel of the controller, click the "Start'' button to begin discovering Thread accessories.
2. Connect accessory to Thread network and ensure the accessory's advertisements are present within the "Thread Discovery'' view in the trace window.
3. Pair and discover the accessory on thread network using Thread Controller.
4. Select accessory's name in left sidebar of Controllers window.
5. To locate name of the HAP accessory object, see Accessory Name under Summary Panel.
6. To locate Bonjour name, check the Bonjour Record of the accessory in Thread Discovery section of HAT Trace. This should be listed under "Bonjour Info" details as "DNS Name."
7. Verify that these 2 values match.

TCT051 **The required Bonjour TXT keys must be supplied in the accessory's Bonjour advertisement.**

Applies to accessories using the HAP over Thread transport. Perform this test case with HAT using the steps below.

1. Select the "+" at the bottom of left sidebar and select "Create Thread Controller" to make a new virtual Thread controller. In the Summary panel of the controller, click the "Start'' button to begin discovering Thread accessories.
2. Connect accessory to Thread network and ensure the accessory's advertisements are present within the "Thread Discovery'' view in the trace window.
3. Pair and discover the accessory on thread network using Thread Controller.
4. Go to the Bonjour Discovery traffic view, select the Bonjour record for the accessory, and open the details sidebar.
5. Verify that the required Bonjour TXT keys are present.
6. Verify the presence of the Device ID (id) key.
7. Verify the presense of the Model (md) key.
8. Verify the presense of the Protocol version (pv) key.
9. Verify the presense of the Configuration number (c#) key.
10. Verify the presense of the Status flags (sf) key, required if non-zero value.
11. Verify the presense of the Accessory category identifier (ci) key.
12. Verify the presense of the State number (s#) key.

13. Verify the presense of the Setup Hash (sh) key, required if the the accessory supports enhanced setup payload information.
14. Verify the presense of the Pairing Feature flags (ff) key, required if non-zero.

TCT052 **The Device ID must persist across a reboot and is randomly generated when accessory is factory reset.**

Applies to accessories using the HAP over Thread transport. Perform this test case with HAT using the steps below.

1. Select the "+" at the bottom of left sidebar and select "Create Thread Controller" to make a new virtual Thread controller. In the Summary panel of the controller, click the "Start'' button to begin discovering Thread accessories.
2. Connect accessory to Thread network and ensure the accessory's advertisements are present within the "Thread Discovery'' view in the trace window.
3. Pair and discover the accessory on thread network using Thread Controller.
4. In Controllers window, note the Device ID in the Advertisement Information Panel.
5. In the Summary panel, select the "Disconnect" button.
6. Power cycle accessory.
7. Select the Discover button in the Summary Panel.
8. Verify the Device ID did not change.
9. Perform factory reset on accessory.
10. In the BLE Discovery traffic view, look for the most recent accessory advertisement. Select details button and verify Device ID is randomly generated and differs from Device ID from Step 8.
11. Add accessory back to the thread network and wait for it to begin advertising over Bonjour.
12. In the Thread Discovery traffic view, look for the most recent accessory advertisement. Select details button and verify Device ID is same as the one from Device ID in Step 10.
13. Power cycle the accessory.
14. Wait for the accessory to begin advertising over Bonjour. In Controllers window, note the Device ID in the Advertisement Information Panel. Verify the Device ID not change from Step 10.

TCT053 **The following device information must persist across reboots and power cycles: Device ID (id), Configuration number (c#), and Accessory category identifier (ci).**

Applies to accessories using the HAP over Thread transport. Perform this test case with HAT using the steps below.

1. Select the "+" at the bottom of left sidebar and select "Create Thread Controller" to make a new virtual Thread controller. In the Summary panel of the controller, click the "Start'' button to begin discovering Thread accessories.

2. Connect accessory to Thread network and ensure the accessory's advertisements are present within the "Thread Discovery'' view in the trace window.
3. Pair and discover the accessory on Thread network using Thread Controller.
4. Select accessory's name in left sidebar of Controllers window.
5. Using the "Advertisement Information" panel on the main accessory server view, note the current Device ID, configuration number, and accessory category identifier values.
6. In the Summary panel, select the "Disconnect" button.
7. Power cycle the accessory.
8. Select Discover in Controllers window.
9. Repeat steps 4-6.
10. Verify that the values found in Step 5 did not change.

TCT054 **If supported, accessory's configuration number must increment when a service or characteristic is added to or removed from accessory server.**

Applies to accessories using the HAP over Thread transport. Perform this test case with HAT using the steps below.

1. Select the "+" at the bottom of left sidebar and select "Create Thread Controller" to make a new virtual Thread controller. In the Summary panel of the controller, click the "Start'' button to begin discovering Thread accessories.
2. Connect accessory to Thread network and ensure the accessory's advertisements are present within the "Thread Discovery'' view in the trace window.
3. Pair and discover the accessory on thread network using Thread Controller.
4. Select accessory's name in left sidebar of Controllers window.
5. See the Advertisement Information panel in the Controllers window.
6. Note the current configuration number.
7. Add a service or characteristic to accessory or remove a service or characteristic from accessory.
8. Note the configuration number.
9. Verify that the configuration number has incremented.

TCT055 **For characteristics that don't support paired write, write attempts should fail with the correct error.**

Applies to accessories using the HAP over Thread transport. Perform this test case with HAT using the steps below.

1. Select the "+" at the bottom of left sidebar and select "Create Thread Controller" to make a new virtual Thread controller. In the Summary panel of the controller, click the "Start'' button to begin discovering Thread accessories.

2. Connect accessory to Thread network and ensure the accessory's advertisements are present within the "Thread Discovery'' view in the trace window.
3. Pair and discover the accessory on thread network using Thread Controller.
4. In Controllers window, select a characteristic that does not support paired write.
5. Write a value to this characteristic.
6. See the HAP traffic view in HAT Trace.
7. Verify that the accessory responds with "Status: Unsupported PDU (0x01)".

TCT056 **For characteristics that don't support paired read, read attempts should fail with the correct error.**

Applies to accessories using the HAP over Thread transport. Perform this test case with HAT using the steps below.

1. Select the "+" at the bottom of left sidebar and select "Create Thread Controller" to make a new virtual Thread controller. In the Summary panel of the controller, click the "Start'' button to begin discovering Thread accessories.
2. Connect accessory to Thread network and ensure the accessory's advertisements are present within the "Thread Discovery'' view in the trace window.
3. Pair and discover the accessory on thread network using Thread Controller.
4. In Controllers window, select a characteristic that does not support a paired read.
5. Read this characteristic.
6. See the HAP traffic view in HAT Trace.
7. Verify that the accessory responds with "Status: Unsupported PDU (0x01)".

TCT057 **If an accessory has characteristics that have minimum value and maximum value metadata, writing values below the minimum value and above the maximum value must not be accepted by accessory.**

Applies to accessories using the HAP over Thread transport. Perform this test case with HAT using the steps below.

1. Select the "+" at the bottom of left sidebar and select "Create Thread Controller" to make a new virtual Thread controller. In the Summary panel of the controller, click the "Start'' button to begin discovering Thread accessories.
2. Connect accessory to Thread network and ensure the accessory's advertisements are present within the "Thread Discovery'' view in the trace window.
3. Pair and discover the accessory on thread network using Thread Controller.
4. In left sidebar of Controllers window, see each of the accessory's services.
5. Using the Write button in Controllers window, write to each paired Read/Write characteristic with a value that is below the minValue if the value to be written is valid for the specified format (e.g., for a format of uint8 and a minValue of 0, this step can be skipped).

6. Verify in HAP Traffic view of HAT Trace that Accessory responds to out of range values with "Status: Invalid Request (0x06)."
7. Using the Write button in Controllers window, write to each paired Read/Write characteristic with a value that is above the maxValue if the value to be written is valid for the specified format (e.g., for a format of uint8 and a maxValue of 255, this step can be skipped).
8. Verify in HAP Traffic view of HAT Trace that Accessory responds to out of range values with "Status: Invalid Request (0x06)."

TCT058 **Accessories must supply required properties inside each accessory: Accessory Instance ID and Services.**

Applies to accessories using the HAP over Thread transport. Perform this test case with HAT using the steps below.

1. Select the "+" at the bottom of left sidebar and select "Create Thread Controller" to make a new virtual Thread controller. In the Summary panel of the controller, click the "Start'' button to begin discovering Thread accessories.
2. Connect accessory to Thread network and ensure the accessory's advertisements are present within the "Thread Discovery'' view in the trace window.
3. Pair and discover the accessory on thread network using Thread Controller.
4. Locate accessory's Discovery response in the Events view in HAT Trace.
5. Select Details button to reveal Details sidebar.
6. Within each accessory, verify that the following properties are provided: Accessory instance ID and Services.

TCT061 **An accessory must support timed writes to all characteristics even if the characteristic does not require it.**

Applies to accessories using the HAP over Thread transport. Perform this test case with HAT using the steps below.

1. Select the "+" at the bottom of left sidebar and select "Create Thread Controller" to make a new virtual Thread controller. In the Summary panel of the controller, click the "Start'' button to begin discovering Thread accessories.
2. Connect accessory to Thread network and ensure the accessory's advertisements are present within the "Thread Discovery'' view in the trace window.
3. Pair and discover accessory.
4. In Controllers window, select a characteristic that supports paired write but does not require timed write.
5. In the Prepare Write Panel, Set TTL to 200 (to set a TTL of 200x100 milliseconds = 20 seconds). Enter a value (i.e turn on light) and select Prepare Write. Perform the next step within 20 seconds of performing this step.

6. In the Execute Write Panel, select Execute Write.
7. Verify in HAP Traffic view of HAT Trace that the accessory responds with "Status: Success (0x00)" for the "Execute Write" request.

TCT064 **Prepare-write. Wait for TTL to expire. Prepare-write. Execute-write.**

Applies to accessories using the HAP over Thread transport. Perform this test case with HAT using the steps below.

1. Select the "+" at the bottom of left sidebar and select "Create Thread Controller" to make a new virtual Thread controller. In the Summary panel of the controller, click the "Start'' button to begin discovering Thread accessories.
2. Connect accessory to Thread network and ensure the accessory's advertisements are present within the "Thread Discovery'' view in the trace window.
3. Pair and discover accessory.
4. In Controllers window, select a characteristic that supports paired write.
5. In the Prepare Write Panel, set TTL to 1 (=1x100 milliseconds = 0.1 seconds). Enter a value (i.e turn on light) and select Prepare Write. Wait for 1 second or more, for the TTL to expire.
6. In the Prepare Write Panel, set TTL to 200 (=200x100 milliseconds = 20 seconds). Enter a value (i.e turn on light) and select Prepare Write. Perform the next steps within 20 seconds of performing this step.
7. In the Execute Write Panel, select Execute write.
8. Verify in HAP Traffic view of HAT Trace that the accessory responds with "Status: Success (0x00)" for the "Execute Write" request.

TCT067 **Prepare-write. Execute-write. Execute-write again.**

Applies to accessories using the HAP over Thread transport. Perform this test case with HAT using the steps below.

1. Select the "+" at the bottom of left sidebar and select "Create Thread Controller" to make a new virtual Thread controller. In the Summary panel of the controller, click the "Start'' button to begin discovering Thread accessories.
2. Connect accessory to Thread network and ensure the accessory's advertisements are present within the "Thread Discovery'' view in the trace window.
3. Pair and discover accessory.
4. In Controllers window, select a characteristic that supports paired write.
5. In the Prepare Write Panel, set TTL to 200 (=200x100 milliseconds = 20 seconds). Enter a value (i.e turn on light) and select Prepare Write. Perform the next steps within 20 seconds of performing this step.

6. In the Execute Write Panel, select Execute write.
7. Verify accessory completes the write without error.
8. In the Execute Write Panel, select Execute write.
9. Verify in HAP Traffic view of HAT Trace that the accessory responds with "Status: Invalid Request 0x06" for the "Execute Write" request.

TCT068 **If the accessory receives an Execute Write Request after the TTL has expired, it must respond with "Invalid Request 0x06".**

Applies to accessories using the HAP over Thread transport. Perform this test case with HAT using the steps below.

1. Select the "+" at the bottom of left sidebar and select "Create Thread Controller" to make a new virtual Thread controller. In the Summary panel of the controller, click the "Start'' button to begin discovering Thread accessories.
2. Connect accessory to Thread network and ensure the accessory's advertisements are present within the "Thread Discovery'' view in the trace window.
3. Pair and discover accessory.
4. In Controllers window, select a characteristic that supports paired write.
5. In the Prepare Write Panel, set TTL to 1 (=1x100 milliseconds). Enter a value (i.e turn on light) and select Prepare Write. Wait for at least 2 seconds before performing the next step.
6. In the Execute Write Panel, select Execute write.
7. Verify in HAP Traffic view of HAT Trace that the accessory responds with "Status: Invalid Request 0x06" for the "Execute Write" request.

TCT069 **If the accessory receives a standard write request on a characteristic which requires timed write, the accessory must respond with "invalid Request 0x06".**

Applies to accessories using the HAP over Thread transport. Perform this test case with HAT using the steps below.

1. Select the "+" at the bottom of left sidebar and select "Create Thread Controller" to make a new virtual Thread controller. In the Summary panel of the controller, click the "Start'' button to begin discovering Thread accessories.
2. Connect accessory to Thread network and ensure the accessory's advertisements are present within the "Thread Discovery'' view in the trace window.
3. Pair and discover accessory.
4. In Controllers window, select a characteristic that requires timed write.
5. Write a value to this characteristic.

6. Verify in HAP Traffic view of HAT Trace that the accessory responds with "Status: Invalid Request 0x06" for the write request.

TCT070 **Verify that the accessory accepts consecutive Prepare Write Requests in the same session.**

Applies to accessories using the HAP over Thread transport. Perform this test case with HAT using the steps below.

1. Select the "+" at the bottom of left sidebar and select "Create Thread Controller" to make a new virtual Thread controller. In the Summary panel of the controller, click the "Start'' button to begin discovering Thread accessories.
2. Connect accessory to Thread network and ensure the accessory's advertisements are present within the "Thread Discovery'' view in the trace window.
3. Pair and discover accessory.
4. In Controllers window, select a characteristic that supports paired write.
5. In the Prepare Write Panel, set TTL to 200 (=200x100 milliseconds = 20 seconds). Enter a value (i.e turn on light) and select Prepare Write. Perform the next step within 20 seconds of performing this step.
6. In the Prepare Write Panel, set TTL to 200 (=200x100 milliseconds = 20 seconds). Enter a value (i.e turn on light) and select Prepare Write. Perform the next steps within 20 seconds of performing this step.
7. In the Execute Write Panel, select Execute write.
8. Verify in HAP Traffic view of HAT Trace that the accessory responds with "Status: Success (0x00)" for the write request.

TCT071 **Prepare-write from Controller 1 and Execute-write from Controller 2**

Applies to accessories using the HAP over Thread transport. Perform this test case with HAT using the steps below.

1. Select the "+" at the bottom of left sidebar and select "Create Thread Controller" to make a new virtual Thread controller. In the Summary panel of the controller, click the "Start'' button to begin discovering Thread accessories.
2. Connect accessory to Thread network and ensure the accessory's advertisements are present within the "Thread Discovery'' view in the trace window.
3. Add two controllers in HAT, Controller 1 and Controller 2.
4. Pair and discover accessory using Controller 1.
5. In Controllers window, under "Add Additional Controllers" panel, select "Controller 2" as Controller and then select the "Add Controller" button.
6. In left sidebar, select the Controller 2, select the accessory name, and select the "Discover" button.

7. From Controller 1, in Controllers window, select a characteristic that supports paired write.
8. In the Prepare Write Panel, Set TTL to 200 (i.e., =200x100 milliseconds = 20 seconds). Enter a value (e.g., turn on light) and select Prepare Write. Perform the next step within 20 seconds of performing this step.
9. From Controller 2, in Controllers window, select the same characteristic as in Step 5. In the Execute Write Panel, select Execute write.
10. Verify in HAP Traffic view of HAT Trace that the accessory responds with "Status: Invalid Request 0x06" for the write request.

TCT073 **Accessory must expose a single instance of the Pairing Service with the following required characteristics: Pair Setup, Pair Verify, Features, and Pairings.**

Applies to accessories using the HAP over Thread transport. Perform this test case with HAT using the steps below.

1. Select the "+" at the bottom of left sidebar and select "Create Thread Controller" to make a new virtual Thread controller. In the Summary panel of the controller, click the "Start'' button to begin discovering Thread accessories.
2. Connect accessory to Thread network and ensure the accessory's advertisements are present within the "Thread Discovery'' view in the trace window.
3. Pair and discover the accessory on thread network using Thread Controller.
4. In left sidebar of Controllers window, see accessory's Pairing Service.
5. Verify that the required characteristics are included in the Pairing Service.

TCT077 **If an accessory receives a HAP PDU with an opcode that it does not support, it shall reject the PDU and respond with a status code "Unsupported PDU (0x01)" in its HAP response.**

Applies to accessories using the HAP over Thread transport. Perform this test case with HAT using the steps below.

1. Select the "+" at the bottom of left sidebar and select "Create Thread Controller" to make a new virtual Thread controller. In the Summary panel of the controller, click the "Start'' button to begin discovering Thread accessories.
2. Connect accessory to Thread network and ensure the accessory's advertisements are present within the "Thread Discovery'' view in the trace window.
3. Pair and discover the accessory on thread network using Thread Controller.
4. Select any characteristic.
5. In the Controllers window, under Options panel, select "Send Unsupported Opcode".
6. In the HAP Traffic view, verify accessory responds with "Unsupported PDU (0x01)".

TCT078 **In case a single CoAP packet contains several PDU requests, verify that the accessory processes these PDUs in the order in which they are received and that responses are stored sequentially in CoAP response.**

Applies to accessories using the HAP over Thread transport. Perform this test case with HAT using the steps below.

1. Select the "+" at the bottom of left sidebar and select "Create Thread Controller" to make a new virtual Thread controller. In the Summary panel of the controller, click the "Start'' button to begin discovering Thread accessories.
2. Connect accessory to Thread network and ensure the accessory's advertisements are present within the "Thread Discovery'' view in the trace window.
3. Pair and discover the accessory on thread network using Thread Controller.
4. In Controllers window, select any characteristic that supports paired write.
5. In the Queued Writing Panel, Enable Queue State.
6. Write value in the Queue Characteristic Panel and select Queue button.
7. Select another characteristic that supports paired write.
8. Write value in the Queue Characteristic Panel and select Queue button.
9. Select a read-only characteristic.
10. Write value in the Queue Characteristic Panel and select Queue button.
11. Repeat Steps 7 and 8 to add two more write actions to the queue for characteristics that support paired write.
12. In the pop-up window, select the Send button to send the queue of requests.
13. Verify in CoAP Traffic view of HAT Trace that the queue is sent as a single CoAP request and receives a single CoAP response. Accessories that sleep have an exception to this - it is acceptable for Sleepy accessories to receive CoAP responses for each of the retransmissions attempted by the HAT Controller while the accessory is asleep.
14. Verify in HAP Traffic view that the accessory responds with "Status (0x00)" for each successful write request and "Unsupported PDU (0x01)" for each unsuccessful write request. Accessory must not process or respond to Write requests queued in Step 11.

TCT079 **Verify that the accessory supports a minimum of 1024 bytes of incoming PDU data.**

Applies to accessories using the HAP over Thread transport. Perform this test case with HAT using the steps below.

1. Select the "+" at the bottom of left sidebar and select "Create Thread Controller" to make a new virtual Thread controller. In the Summary panel of the controller, click the "Start'' button to begin discovering Thread accessories.
2. Connect accessory to Thread network and ensure the accessory's advertisements are present within the "Thread Discovery'' view in the trace window.

3. Pair and discover the accessory on thread network using Thread Controller.

4. In Controllers window, select the Firmware Version characteristic.

5. Enter a value that is 1024 bytes long in the Write panel.
E.g. “0123456789012345678901234567890123456789012345678901234567890
12345678901234567890123456789012345678901234567890123456789012345678
90123456789012345678901234567890123456789012345678901234567890123456
78901234567890123456789012345678901234567890123456789012345678901234
56789012345678901234567890123456789012345678901234567890123456789012
34567890123456789012345678901234567890123456789012345678901234567890
12345678901234567890123456789012345678901234567890123456789012345678
90123456789012345678901234567890123456789012345678901234567890123456
78901234567890123456789012345678901234567890123456789012345678901234
56789012345678901234567890123456789012345678901234567890123456789012
34567890123456789012345678901234567890123456789012345678901234567890
12345678901234567890123456789012345678901234567890123456789012345678
90123456789012345678901234567890123456789012345678901234567890123456
78901234567890123456789012345678901234567890123456789012345678901234
56789012345678901234567890123456789012345678901234567890123456789012
34567890123”.

6. Write the value to the Characteristic.

7. Check HAP Traffic View in HAT trace and verify that the write responds with “Unsupported PDU (0x01).”

TCT081 **If authTag verification of encrypted data fails during pair-verify, the accessory must respond with “M4” and “kTLVError_Authentication 0x02”.**

Applies to accessories using the HAP over Thread transport. Perform this test case automatically with HCA. Below are steps that can be used to perform this test case manually in HAT.

1. Select the “+” at the bottom of left sidebar and select “Create Thread Controller” to make a new virtual Thread controller. In the Summary panel of the controller, click the “Start’’ button to begin discovering Thread accessories.

2. Connect accessory to Thread network and ensure the accessory’s advertisements are present within the “Thread Discovery’’ view in the trace window.

3. Pair and discover the accessory on thread network using Thread Controller.

4. Click “Disconnect” in the Controllers window or wait for the accessory to disconnect.

5. In the “Options” section of the accessory server view, enable the “Use Bad Auth Tag during Pair-Setup and Pair-Verify” check box.

6. Click “Pair Verify” in the Connection section of the Controllers window.

7. See HAP Traffic view.

8. Verify accessory response in Pair Verify value is “M4” and “`kAuthenticationErr`”. Verify in Events view that the M4 results in “Authentication Error”.

9. In the "Options" section of the accessory server view, disable the "Use Bad Auth Tag during Pair-Setup, Pair-Verify and Pair-Resume" check box.

TCT083 **Accessories must support multiple iterations of Pair Verify on a single connection.**

Applies to accessories using the HAP over Thread transport. Perform this test case with HAT using the steps below.

1. Select the "+" at the bottom of left sidebar and select "Create Thread Controller" to make a new virtual Thread controller. In the Summary panel of the controller, click the "Start'' button to begin discovering Thread accessories.
2. Connect accessory to Thread network and ensure the accessory's advertisements are present within the "Thread Discovery'' view in the trace window.
3. Pair and discover the accessory on thread network using Thread Controller.
4. Verify that Pair Verify payloads are seen in the Events Traffic view.
5. Read or write any characteristic.
6. Under the Connection panel, select "Pair Verify" button.
7. In the Events traffic view of HAT Trace, verify that the message "Pair-Verify Completed" is received.
8. Under the Connection panel, select "Pair Verify" button again.
9. In the Events traffic view, Verify the message "Pair-Verify Completed" is received and the Details show no errors.

TCT087 **Accessory must indicate that Security Class characteristics require HAP-Characteristic-Timed-Write using the HAP Characteristic Properties Descriptor.**

Applies to accessories using the HAP over Thread transport. Perform this test case with HAT using the steps below.

1. Select the "+" at the bottom of left sidebar and select "Create Thread Controller" to make a new virtual Thread controller. In the Summary panel of the controller, click the "Start'' button to begin discovering Thread accessories.
2. Connect accessory to Thread network and ensure the accessory's advertisements are present within the "Thread Discovery'' view in the trace window.
3. Pair and discover the accessory on thread network using Thread Controller.
4. See Discovered Accessories in the Events traffic view.
5. Select "Details" and view Attribute Database.
6. Verify Security Class characteristics use the Timed-Write permission.
7. Write a value to the characteristics that support Timed-Write (e.g., lock or unlock).
8. Verify physical action completed (e.g., deadbolt locked or unlocked).

TCT091 **Accessories must implement a 10 second HAP procedure timeout. All HAP procedures including Pair-Verify must complete within 10 seconds.**

Applies to accessories using the HAP over Thread transport. Perform this test case with HAT using the steps below.

1. Select the "+" at the bottom of left sidebar and select "Create Thread Controller" to make a new virtual Thread controller. In the Summary panel of the controller, click the "Start'' button to begin discovering Thread accessories.
2. Connect accessory to Thread network and ensure the accessory's advertisements are present within the "Thread Discovery'' view in the trace window.
3. Pair and discover the accessory on thread network using Thread Controller.
4. In the Connection Panel, select the Pair Verify button.
5. In Events Traffic view, locate Pair Verify Write Requests/Responses and select details.
6. Verify the Pair Verify procedure, from state value 1 (M1) through state value 4 (M4), completes within 10 seconds.
7. In the Connection Panel, select the Pair Verify button.
8. In HAP traffic view, locate Pair Verify Requests/Responses and select details.
9. Verify the Pair Verify procedure, from state value 1 (M1) through state value 4 (M4), completes within 10 seconds.

TCT092 **Accessory must support connected events.**

Applies to accessories using the HAP over Thread transport. Perform this test case with HAT using the steps below.

1. Select the "+" at the bottom of left sidebar and select "Create Thread Controller" to make a new virtual Thread controller. In the Summary panel of the controller, click the "Start'' button to begin discovering Thread accessories.
2. Connect accessory to Thread network and ensure the accessory's advertisements are present within the "Thread Discovery'' view in the trace window.
3. Pair and discover the accessory on thread network using Thread Controller.
4. Navigate to each characteristic that supports Notify and select Enable for event notifications.
5. Using a value different than the current value, write to a characteristic that contains both the paired write and notify permissions.
6. Verify the controller does not receive notifications for the characteristic that was written to.
7. While connected to the accessory, physically toggle a characteristic that contains the notify permission. Note: If the accessory disconnects, Event notifications must be re-enabled for each characteristic.

8. Using the Events traffic view in the trace, verify that the accessory sends a notification for the correct service and characteristic that was toggled.
9. Using the HAP Traffic view, verify the value in the notification received contains the correct value.
10. Repeat Steps 4-9 for each characteristic that contains the notify permissions.

TCT093 **Accessory must always successfully deliver event notifications for every characteristic that supports them when a single client has subscribed multiple times.**

Applies to accessories using the HAP over Thread transport. Perform this test case with HAT using the steps below.

1. Select the "+" at the bottom of left sidebar and select "Create Thread Controller" to make a new virtual Thread controller. In the Summary panel of the controller, click the "Start'' button to begin discovering Thread accessories.
2. Connect accessory to Thread network and ensure the accessory's advertisements are present within the "Thread Discovery'' view in the trace window.
3. Pair and discover the accessory on thread network using Thread Controller.
4. Navigate to each characteristic that supports connected Event Notifications. Subscribe multiple times to Event Notifications by selecting "Enable" button 3 times.
5. For characteristics that provide physical means of interaction, physically toggle each applicable characteristic on the accessory.
6. Verify that the controller receives only a single notification for each state change.

TCT095 **When Thread parameters are cleared, the accessory should fall back to BLE. When the accessory is connected to a thread network again, verify that the Accessory can Pair-Verify successfully.**

Applies to accessories using the HAP over Thread transport. Perform this test case with HAT using the steps below.

1. Select the "+" at the bottom of left sidebar and select "Create Shared Key Store".
2. Select the "+" at the bottom of left sidebar and select "Create BLE Controller" to make a new virtual BLE controller. In the Summary panel of the controller, assign the Shared Key Store created in Step 1, and click on the "Start'' button to begin discovering BLE accessories.
3. Select the "+" at the bottom of left sidebar and select "Create Thread Controller" to make a new virtual Thread controller. In the Summary panel of the controller, assign the Shared Key Store created in Step 1, and click on the "Start'' button to begin discovering Thread accessories.
4. Pair discover the accessory using BLE Controller.
5. With the Thread network present, navigate to "Thread Transport" service, select "Thread Control Point" characteristic, click "Build TLV" under "Write [tlv8]" panel, select "Set Thread Parameters" from the drop-down, enter the details of the thread border router network, enter "0" in the "Forming Allowed" field, and click "Build TLV".

6. Select “Write” to send the TLV to the accessory and check Thread Discovery view of HAT Trace. Accessory must connect to the Border Router immediately and show its Bonjour advertisement in Thread Discovery view.

7. Pair and discover accessory over Thread.

8. Navigate to “Thread Transport” service, select “Thread Control Point” characteristic, click “Build TLV” under “Write [tlv8]” panel, select “Clear Thread Parameters” from the drop-down. Click “Build TLV”.

9. Select “Write” to send the TLV to the accessory and wait for accessory to fall back to BLE. Verify it falls back to BLE by looking for a BLE advertisement in BLE Discovery.

10. Discover accessory over BLE using the BLE Controller in HAT.

11. Bring up a border router that uses the same set of thread parameters or a different set of thread parameters.

12. Navigate to the “Thread Transport” service, select the “Thread Control Point” characteristic, select “Build TLV” under “Write [tlv8]” panel, select “Set Thread Parameters” from the menu, enter the details of the thread border router network brought up in Step 11, enter “0” in the “Forming Allowed” field. Select “Build TLV”.

13. Select “Write” to send the TLV to the accessory and check Thread Discovery view of HAT Trace. Accessory must connect to the Border Router immediately and show its Bonjour advertisement in Thread Discovery view.

14. In Controllers window, attempt a Pair-Verify and verify in Events view of HAT Trace that “Pair-Verify Completed”.

TCT097 **Verify that Successful CoAP responses use the 2.04 status code and that Unsuccessful CoAP responses use 4.04.**

Applies to accessories using the HAP over Thread transport. Perform this test case with HAT using the steps below.

1. Select the “+” at the bottom of left sidebar and select “Create Thread Controller” to make a new virtual Thread controller. In the Summary panel of the controller, click the “Start’’ button to begin discovering Thread accessories.

2. Connect accessory to Thread network and ensure the accessory’s advertisements are present within the “Thread Discovery’’ view in the trace window.

3. Pair and discover the accessory on thread network using Thread Controller.

4. In left side bar of the Controllers window, select the accessory’s name. Under the Summary panel, select the “Paired Identify’’ button. Verify in CoAP Traffic view that CoAP response from accessory shows “Code: 2.04” for a successful response.

5. Invalidate the existing session by power-cycling the accessory.

6. Without performing a Pair-Verify or Discover, in left side bar of the Controllers window, select the accessory’s name. Under the Summary panel, select the “Paired Identify’’ button. Verify the response from the accessory in CoAP Traffic View shows a “Code: 4.04” for an unsuccessful response.

7. Navigate to Connection panel, click on “Pair Verify”, and verify that the CoAP Traffic view in HAT Trace shows “Code: 2.04” for each of the CoAP Responses of the Pair-Verify transaction.

TCT099 **When Thread parameters are overwritten to an accessory while it is on BLE, verify that the Accessory can successfully connect to the new network.**

Applies to accessories using the HAP over Thread transport. Perform this test case with HAT using the steps below.

1. Select the “+” at the bottom of left sidebar and select “Create Shared Key Store”.
2. Select the “+” at the bottom of left sidebar and select “Create BLE Controller” to make a new virtual BLE controller. In the Summary panel of the controller, assign the Shared Key Store created in Step 1, and click on the “Start” button to begin discovering BLE accessories.
3. Select the “+” at the bottom of left sidebar and select “Create Thread Controller” to make a new virtual Thread controller. In the Summary panel of the controller, assign the Shared Key Store created in Step 1, and click on the “Start” button to begin discovering Thread accessories.
4. Pair discover the accessory using BLE Controller.
5. With the Thread network present, navigate to “Thread Transport” service, select “Thread Control Point” characteristic, click “Build TLV” under “Write [tlv8]” panel, select “Set Thread Parameters” from the drop-down, enter the details of the thread border router network, enter “0” in the “Forming Allowed” field. Click “Build TLV”.
6. Select “Write” to send the TLV to the accessory and check Thread Discovery view of HAT Trace. Accessory must connect to the Border Router immediately and show its Bonjour advertisement in Thread Discovery view.
7. Pair and discover accessory over Thread.
8. Unplug the border router(s) which have thread credentials used in Step 5.
9. Verify accessory falls back to BLE by looking for a BLE advertisement in BLE Discovery view of HAT Trace.
10. Discover accessory over BLE using the BLE Controller in HAT.
11. Bring up a border router with a different set of thread parameters.
12. Navigate to “Thread Transport” service, select “Thread Control Point” characteristic, click “Build TLV” under “Write [tlv8]” panel, select “Set Thread Parameters” from the drop-down, enter the details of the new thread border router network, enter “0” in the “Forming Allowed” field. Click “Build TLV”.
13. Select “Write” to send the TLV to the accessory and check Thread Discovery view of HAT Trace. Accessory must connect to the Border Router and show its Bonjour advertisement in Thread Discovery view.
14. In Connection panel, click on “Pair Verify”. Verify that “Pair-Verify Completed” is shown in Events view of HAT Trace.

TCT100 **When the last pairing has been removed, accessory must change the status flag within 5 seconds after remove pairing is completed, and subsequent pairing attempts with accessory must succeed.**

Applies to accessories using the HAP over Thread transport. Not Applicable to accessories that clear thread credentials when last admin pairing is removed. Perform this test case with HAT using the steps below.

1. Select the "+" at the bottom of left sidebar and select "Create Thread Controller" to make a new virtual Thread controller. In the Summary panel of the controller, click the "Start'' button to begin discovering Thread accessories.
2. Connect accessory to Thread network and ensure the accessory's advertisements are present within the "Thread Discovery'' view in the trace window.
3. Pair and discover the accessory on thread network using Thread Controller.
4. Select the Remove Pairing button and note the timestamp of the "Remove Pairing Completed" response in Events traffic view.
5. Check the "Add/Update Bonjour Record" message sent in Thread Discovery view and verify Status Flags is set to 1. Note the timestamp of this record.
6. Verify the delta between timestamps noted in Step 5 and Step 6 is within 5 seconds.
7. In the Controllers window, in the Advertisement Information Panel, verify that the Status Flag changed to "0x01".
8. After the pairing has been removed, verify accessory can successfully pair to controller again.
9. In Controllers window, select "Start Pairing."
10. Enter setup code.
11. Discover accessory.
12. Verify that Pair Setup and Pair Verify complete successfully.

1.22 Accessory Diagnostics

TCADX001: Any Diagnostics services must include the required characteristics.

TCADX002: An accessory that advertises the Diagnostics service must support the HomeKit Data Stream (HDS) service.

TCADX003: Verify that the accessory supports HDS dataSend messages with the type "diagnostics.snapshot".

TCADX004: When the accessory has a diagnostics snapshot stream open to any controller, the accessory must respond to additional diagnostics.snapshot open requests with the "Busy" status.

TCADX005: If the accessory includes the optional metadata fields in the first data packet, verify that the proper format is used.

TCADX006: Verify that the file transferred via HDS is in ZIP or TEXT format and less than 5Mb in size.

TCADX007: Verify that the accessory claims support for valid diagnostic snapshot types as part of the "Supported Diagnostics Snapshot" characteristic value.

TCADX009: If a diagnostic snapshot transfer over HDS is interrupted, verify accessory can establish a new HDS session and successfully complete the diagnostic snapshot transfer.

TCADX010: Requests for diagnostic snapshots by non-admin controllers must respond to the HDS Start Request with status "6" (Protocol-specific error) and reason "1" (Not Allowed).

TCADX001 **Any Diagnostics services must include the required characteristics.**

Applies to accessories that support Accessory Diagnostics. Applies to accessories that use HAP over Ethernet or Wi-Fi. Applies to accessories that use HAP over BLE. Applies to accessories using the HAP over Thread transport. Perform this test case with HAT using the steps below.

Required characteristics:

- Supported Diagnostics Snapshot (r)

* Notify (ev) for BLE encompasses Indicate, Indicate (disconnected), and Broadcast. Please note, characteristics with TLV8 or String formats must not support Broadcast Events.

1. Pair and discover accessory.
2. In left sidebar of Controllers window, see each of accessory's services.
3. Verify "Supported Diagnostics Snapshot" characteristic is included in the "Diagnostics" service.
4. For HAP over IP accessories and HAP over BLE accessories, perform a Paired Read on the "Supported Diagnostics Snapshot" characteristic and verify the Read Completed without errors.
5. For HAP over Thread accessories, perform a Paired Read on the "Supported Diagnostics Snapshot" characteristic and verify the Read Response shows no errors on HAP traffic view of trace under Thread.

TCADX002 **An accessory that advertises the Diagnostics service must support the HomeKit Data Stream (HDS) service.**

Applies to accessories that support Accessory Diagnostics. Applies to accessories that use HAP over Ethernet or Wi-Fi. Applies to accessories that use HAP over BLE. Applies to accessories using the HAP over Thread transport. Perform this test case with HAT using the steps below.

1. Pair and discover accessory.
2. In the sidebar of the Controllers window, verify accessory contains the "Data Stream Transport Management" service.

TCADX003 **Verify that the accessory supports HDS dataSend messages with the type "diagnostics.snapshot".**

Applies to accessories that support Accessory Diagnostics. Applies to accessories that use HAP over Ethernet or Wi-Fi. Applies to accessories that use HAP over BLE. Applies to accessories using the HAP over Thread transport. Perform this test case with HAT using the steps below.

1. Pair and discover accessory.
2. For HAP over IP accessories, in the sidebar of the Controllers window, select "Data Stream Transport Management", select "Send Start Command", then select "Connect".
3. For HAP over BLE accessories, enable the "Pair Resume Keep Alive" checkbox with a 27-second interval.
4. For HAP over BLE accessories or HAP over Thread accessories, in the sidebar of the Controllers window, select the "Data Stream HAP Transport Interrupt" characteristic under the "Data Stream Transport Management" service and enable Event Notifications.
5. For HAP over BLE accessories or HAP over Thread accessories, select "Data Stream Transport Management", then select "Send Start Command".
6. Enter "12" into the Stream ID field under the "Diagnostics Snapshot" pane, select correct "Snapshot Type" from the menu, then select "Send Diagnostics Snapshot Request".
7. Using the HDS Frames trace view, verify the accessory responds to the Start Data Stream request with a response that has a header that includes "dataSend" as the protocol, "open" as the topic, Status set to "0" (Success), ID with a value that matches the ID the controller chose in the Data Start Request, and no message.
8. Verify that the accessory sends Binary Data Event(s) that have a header which includes "dataSend" as the protocol, "data" as the topic, and a message with Stream ID set to "12" and packet(s) which contain the diagnostic data. (Note: Accessory may optionally include Data Sequence Number and/or URL Parameters fields within the packet metadata).
9. Verify that the last Binary Data Event from the accessory contains a header with "End Of Stream" set to "true".
10. After the controller sends a Data Acknowledgement Event, verify accessory sends a response which includes a header containing "dataSend" as the protocol, "close" as the topic, and a message with Stream ID set to "12' and Reason set to "0" (Normal).

11. Save and open the diagnostic snapshot file, and verify the ZIP or Text contains diagnostic snapshots.

TCADX004 **When the accessory has a diagnostics snapshot stream open to any controller, the accessory must respond to additional diagnostics.snapshot open requests with the "Busy" status.**

Applies to accessories that support Accessory Diagnostics. Applies to accessories that use HAP over Ethernet or Wi-Fi. Applies to accessories using the HAP over Thread transport. Perform this test case with HAT using the steps below.

1. Pair and discover accessory.
2. Add an additional IP controller, Controller 2.
3. Add Controller 2 as an admin controller, select Controller 2, choose the accessory, then select "Discover".
4. For HAP over IP accessories, using Controller 1, in the sidebar of the Controllers window, select the "Data Stream Transport Management" service, select "Send Start Command", then select "Connect".
5. For HAP over IP accessories, using Controller 2, in the sidebar of the Controllers window, select the "Data Stream Transport Management" service, select "Send Start Command", then select "Connect".
6. For HAP over IP accessories, using Controller 1, enter "11" into the Stream ID field under the "Diagnostics Snapshot" pane, select correct "Snapshot Type" from the menu, then select "Send Diagnostics Snapshot Request".
7. For HAP over IP accessories, using Controller 2, enter "13" into the Stream ID field under the "Diagnostics Snapshot" pane, select correct "Snapshot Type" from the menu, then select "Send Diagnostics Snapshot Request".
8. For HAP over Thread accessories, as Controller 1, in the sidebar of the Controllers window, select the "Data Stream HAP Transport Interrupt" characteristic under "Data Stream Transport Management" service, then enable Event Notifications.
9. For HAP over Thread accessories, as Controller 1, select the "Data Stream Transport Management" service, then select "Send Start Command".
10. For HAP over Thread accessories, as Controller 2, in the sidebar of the Controllers window, select the "Data Stream HAP Transport Interrupt" characteristic under the "Data Stream Transport Management" service, then enable Event Notifications.
11. For HAP over Thread accessories, as Controller 2, select the "Data Stream Transport Management" service, then select "Send Start Command".
12. For HAP over Thread accessories, using Controller 1, enter "11" into the Stream ID field under the "Diagnostics Snapshot" pane, select correct "Snapshot Type" from the menu, then select "Send Diagnostics Snapshot Request".
13. For HAP over Thread accessories, using Controller 2, enter "13" into the Stream ID field under the "Diagnostics Snapshot" pane, select correct "Snapshot Type" from the menu, then select "Send Diagnostics Snapshot Request".
14. Using the HDS Frames trace view, verify the accessory responds to the Start Data Stream request with a response that has a header that includes dataSend as the protocol, Open as the topic, Status set to

6 (Protocol Specific Error), ID with a value that matches the ID the controller chose in the Data Start Request, and a message with a Status set to 2 (Busy).

TCADX005 **If the accessory includes the optional metadata fields in the first data packet, verify that the proper format is used.**

Applies to accessories that support Accessory Diagnostics. Applies to accessories that use HAP over Ethernet or Wi-Fi. Applies to accessories that use HAP over BLE. Applies to accessories using the HAP over Thread transport. Perform this test case with HAT using the steps below.

1. Pair and discover accessory.
2. For HAP over IP accessories, in the sidebar of the Controllers window, select "Data Stream Transport Management", select "Send Start Command", then select "Connect".
3. For HAP over BLE accessories, enable the "Pair Resume Keep Alive" checkbox with a 27-second interval.
4. For HAP over BLE accessories or HAP over Thread accessories, in the sidebar of the Controllers window, select the "Data Stream HAP Transport Interrupt" characteristic under the "Data Stream Transport Management" service, then enable Event Notifications.
5. For HAP over BLE accessories or HAP over Thread accessories, select "Data Stream Transport Management", then select "Send Start Command".
6. Enter "15" into the Stream ID field under the "Diagnostics Snapshot" pane, select the correct "Snapshot Type" from the menu, then select "Send Diagnostics Snapshot Request".
7. Using the HDS Frames trace view, inspect each Binary Data Event.
8. If the "metadata" field is present and contains a "urlParameters" dictionary key, verify that only the first data packet contains this key.
9. If the "metadata" field is present and contains a "Data Sequence Number" field, verify that the first Binary Data Event has this field set to "1" and, if applicable, that subsequent Data Sequence Numbers increment by only 1.

TCADX006 **Verify that the file transferred via HDS is in ZIP or TEXT format and less than 5Mb in size.**

Applies to accessories that support Accessory Diagnostics. Applies to accessories that use HAP over Ethernet or Wi-Fi. Applies to accessories that use HAP over BLE. Applies to accessories using the HAP over Thread transport. Perform this test case with HAT using the steps below.

1. Pair and discover accessory.
2. For HAP over IP accessories, in the sidebar of the Controllers window, select "Data Stream Transport Management", select "Send Start Command", then select "Connect".
3. For HAP over BLE accessories, enable the "Pair Resume Keep Alive" checkbox with a 27-second interval.

4. For HAP over BLE accessories or HAP over Thread accessories, in the sidebar of the Controllers window, select the "Data Stream HAP Transport Interrupt" characteristic under the "Data Stream Transport Management" service, then enable Event Notifications.
5. For HAP over BLE accessories or HAP over Thread accessories, select "Data Stream Transport Management", then select "Send Start Command".
6. Enter "16" into the Stream ID field under the "Diagnostics Snapshot" pane, select correct "Snapshot Type" from the menu, then select "Send Diagnostics Snapshot Request".
7. Verify the output file is in ZIP or Text format and is less than 5Mb in size.

TCADX007 **Verify that the accessory claims support for valid diagnostic snapshot types as part of the "Supported Diagnostics Snapshot" characteristic value.**

Applies to accessories that support Accessory Diagnostics. Applies to accessories that use HAP over Ethernet or Wi-Fi. Applies to accessories that use HAP over BLE. Applies to accessories using the HAP over Thread transport. Perform this test case with HAT using the steps below.

1. Pair and discover accessory.
2. In the sidebar of the Controllers window, select the "Supported Diagnostics Snapshot" characteristic.
3. Read the value of the "Supported Diagnostics Snapshot" characteristic.
4. For HAP over BLE accessories or HAP over IP accessories, under the Events view of the Trace on Read Completed response, verify the TLV type "1" (Format) has values of either "0" (zip) or "1" (text).
5. Verify that the TLV type "2" (Snapshot Type) has values of either "Bit 0" set (Manufacturer snapshot) or "Bit 1" set (ADK snapshot).
6. For HAP over Thread accessories, under the HAP Traffic (Thread) view of Trace, on the Read Response, verify the TLV type "1" (Format) has values of either "0" (zip) or "1" (text).
7. Verify that the TLV type "2" (Snapshot Type) has values of either "Bit 0" set (Manufacturer snapshot) or "Bit 1" set (ADK snapshot).

TCADX009 **If a diagnostic snapshot transfer over HDS is interrupted, verify accessory can establish a new HDS session and successfully complete the diagnostic snapshot transfer.**

Applies to accessories that support Accessory Diagnostics. Applies to accessories that use HAP over Ethernet or Wi-Fi. Applies to accessories that use HAP over BLE. Applies to accessories using the HAP over Thread transport. Perform this test case with HAT using the steps below.

1. Pair and discover accessory.
2. For HAP over IP accessories, in the sidebar of the Controllers window, select "Data Stream Transport Management", select "Send Start Command", then select "Connect".
3. For HAP over BLE accessories, enable the "Pair Resume Keep Alive" checkbox with a 27-second interval.

4. For HAP over BLE accessories or HAP over Thread accessories, in the sidebar of the Controllers window, select the “Data Stream HAP Transport Interrupt” characteristic under the “Data Stream Transport Management” service, then enable Event Notifications.
5. For HAP over BLE accessories or HAP over Thread accessories, select “Data Stream Transport Management”, then select “Send Start Command”.
6. Enter “16” into the Stream ID field under the “Diagnostics Snapshot” pane, select correct “Snapshot Type” from the menu, then select “Send Diagnostics Snapshot Request”.
7. Before the diagnostic snapshot transfer completes, immediately select the “End Session” button in the “HomeKit Data Stream” pane to close the HDS session.
8. Using the HDS Frames trace view, verify that “Data Stream Disconnected” is displayed.
9. Select “Data Stream Transport Management”, then select “Send Start Command”.
10. Enter “17” into the Stream ID field under the “Diagnostics Snapshot” pane, select correct “Snapshot Type” from the menu, then select “Send Diagnostics Snapshot Request”.
11. After the transfer completes successfully, save and open the diagnostic snapshot Zip or Text file and verify that it contains valid diagnostic snapshots.

TCADX010 **Requests for diagnostic snapshots by non-admin controllers must respond to the HDS Start Request with status “6” (Protocol-specific error) and reason “1” (Not Allowed).**

Applies to accessories that support Accessory Diagnostics. Applies to accessories that use HAP over Ethernet or Wi-Fi. Applies to accessories that use HAP over BLE. Applies to accessories using the HAP over Thread transport. Perform this test case with HAT using the steps below.

1. Pair and discover accessory.
2. Create a new controller (Controller 2) and add the additional controller pairing as non-admin.
3. Discover the accessory as Controller 2.
4. For HAP over IP accessories, as Controller 2 (non-admin), in the sidebar of the Controllers window, select the “Data Stream Transport Management” service, select “Send Start Command”, then select “Connect”.
5. For HAP over BLE accessories, enable the “Pair Resume Keep Alive” checkbox with a 27-second interval.
6. For HAP over BLE accessories or HAP over Thread accessories, as Controller 2 (non-admin), in the sidebar of the Controllers window, select the “Data Stream HAP Transport Interrupt” characteristic under the “Data Stream Transport Management” service, then enable Event Notifications.
7. For HAP over BLE accessories or HAP over Thread accessories, as Controller 2 (non-admin), select the “Data Stream Transport Management” service, then select “Send Start Command”.
8. As Controller 2 (non-admin), enter “20” into the Stream ID field under the “Diagnostics Snapshot” pane, select the correct “Snapshot Type” from the menu, then select “Send Diagnostics Snapshot Request”.

9. Under HDS Frames of the Trace window, verify the accessory responds to the Start Request with a response that has a header that includes “dataSend” as the protocol, “Open” as the topic, “Status” set to 6 (Protocol Specific Error), ID with a value that matches the ID the controller chose in the Data Start Request, and a message with a Status set to “1” (Not allowed).

1.23 Light Shift

TCLS001: An accessory supporting the Light Shift feature must include the "Characteristic Value Transition Control" characteristic, the "Supported Characteristic Value Transition Configuration" characteristic, and the "Characteristic Value Active Transition Count" characteristic as part of the "Light Bulb" service.

TCLS002: An accessory supporting Light Shift must include the "Color Temperature" characteristic as part of the "Light Bulb" service.

TCLS003: When the value of the "Color Temperature" characteristic changes, the accessory must update the values of "Hue" and "Saturation" characteristics accordingly and notify the controller.

TCLS004: When the values of the "Hue" and "Saturation" characteristics change, the accessory must update the value of the "Color Temperature" characteristic accordingly.

TCLS005: An accessory supporting the Light Shift feature for "Color Temperature" must support and advertise a "Color Temperature" minimum value of at least 200 mirek and a maximum value of at least 370 mirek.

TCLS006: If the accessory is power cycled, it must continue transitioning from the point in the transition within 10 minutes of where it lastly was before it powered down.

TCLS007: The accessory must be able to accept a new transition after being power cycled.

TCLS008: Verify that the accessory supports "Transition Start" operations for the supported transition types for all supported characteristics, and the correct values are returned in Write Responses.

TCLS009: Verify the accessory can successfully perform a Fetch Operation and that it returns the expected parameters after a Transition has been applied.

TCLS010: Verify that the "Characteristic Value Transition Control" characteristic returns the expected values before and after transitions have been applied.

TCLS011: Verify that the contents of the Supported Transition characteristic read response.

TCLS012: Accessory must be able to support up to 52 Transition Points in a single Linear Derived transition for Color Temperature. Accessory must reject write requests, with the correct status code, when the number of transition points specified in the Transition are more than the accessory can support.

TCLS013: When the "Start Delay Duration" TLV item is absent in the "Transition Start" request for a "Linear Derived Transition", that accessory assumes a start delay duration of "0".

TCLS014: Verify that the millisecond calculation accuracy on the accessory for the "Characteristic Value Transition Control" characteristic's "Target Completion Duration" TLV item.

TCLS015: For characteristics that support Linear Derived transitions, verify that the transition repeats if "End Behavior" is set to "Loop", or stops once the transition completes if set to "No Change".

TCLS016: Verify that the permissions and formats for the "Characteristic Value Transition Control", "Supported Characteristic Value Transition Configuration", and "Characteristic Value Active Transition Count" characteristics.

TCLS017: Verify the accessory rejects a "Transition Start" operation when an Instance ID that does not support Transitions is used as the target characteristic in the request.

TCLS018: For characteristics that support Linear Transitions, verify the accessory rejects "Transition Start" operations when unsupported Target Values are used.

TCLS019: Verify the accessory rejects "Transition Start" operations when unsupported Lower and Upper bound values are used.

TCLS021: After setting the "Brightness" value below the transition's Lower Bound, verify that setting the "Brightness" value below the transition's Lower Bound value doesn't change "Color Temperature" value.

TCLS022: After setting the "Brightness" value above the transition's Upper Bound, verify that setting the "Brightness" value above the transition's Upper Bound value doesn't change "Color Temperature" value.

TCLS023: Verify the accessory repeats a Linear Transition for the "Brightness" characteristic with an "End Behavior" set to "Loop".

TCLS024: Verify the accessory does not repeat a Linear Transition for the "Brightness" characteristic with an "End Behavior" set to "No Change".

TCLS025: For characteristics that support Linear transitions, the accessory must be able to support transition points with the first transition point's "Target Completion Duration" set to "0" ms, and subsequent transition points set to at least "100" ms.

TCLS026: For characteristics that support Linear transitions, verify the accessory sends notifications to the controller only at the increments set in the "Notify Value Change Threshold" TLV item.

TCLS027: For characteristics that support Linear transitions, verify the accessory sends Broadcast Event Notifications to the controller only at the increments set in the "Notify Value Change Threshold" TLV item.

TCLS028: For characteristics that support Linear transitions, verify the accessory sends notifications to the controller only at the increments set in the "Notify Time Interval Threshold" TLV item.

TCLS029: For characteristics that support Linear transitions, verify the accessory sends Broadcast Event Notifications to the controller only at the increments set in the "Notify Time Interval Threshold" TLV item.

TCLS030: Verify the accessory responds to "Fetch" operations for characteristics that do not support transitions with the correct HAP status code.

TCLS031: Verify the accessory successfully ends an active "Linear Derived" transition when the accessory receives a "Transition Start" operation to the "Characteristic Value Transition Control" characteristic containing the HAP Instance ID of the characteristic without any transitions specified.

TCLS032: Verify the accessory successfully ends an active "Linear" transition when the accessory receives a "Transition Start" operation to the "Characteristic Value Transition Control" characteristic containing the HAP Instance ID of the characteristic without any transitions specified.

TCLS033: Verify the accessory successfully ends an active "Linear Derived" transition when the accessory receives a write request to the characteristic undergoing the transition.

TCLS034: Verify the accessory successfully ends an active "Linear" transition when the accessory receives a write request to the characteristic undergoing the transition.

TCLS035: For characteristics that support Linear transitions, verify the accessory updates the characteristic value only at the increments set in the "Value Update Time Interval" TLV item.

TCLS036: Verify when a "Linear Derived" transition is active for the "Color Temperature" characteristic using the "Brightness" characteristic as the source value, that the "Color Temperature" value updates according to the set "Brightness" value.

TCLS037: For characteristics that support Linear transitions, accessory must assume a "Value Update Time Interval" of 1 minute when the controller does not specify a "Value Update Time Interval" in the "Transition Start" operation.

TCLS038: For services with characteristics that support Linear and Linear Derived transitions, verify the accessory only ends the Linear Derived transition of the characteristic specified in the HAP Instance ID TLV item of the "Transition Start" operation, without any transitions specified, when multiple transitions are currently active.

TCLS039: For services with characteristics that support both Linear and Linear Derived transitions, verify the accessory only ends the Linear transition of the characteristic specified in the HAP Instance ID TLV item of the "Transition Start" operation, without any transitions specified, when multiple transitions are currently active.

TCLS040: For characteristics that support Linear Derived transitions, the accessory must be able to support transition points with the first transition point's "Target Completion Duration" set to "0" ms, and subsequent transition points set to at least "100" ms.

TCLS041: For characteristics that support Linear Derived transitions, accessory must assume a "Value Update Time Interval" of 1 minute when the controller does not specify a "Value Update Time Interval" in the "Transition Start" operation.

TCLS042: For characteristics that support Linear Derived transitions, verify the accessory updates the characteristic value only at the increments set in the "Value Update Time Interval" TLV item.

TCLS043: For characteristics that support Linear Derived transitions, verify the accessory sends notifications to the controller only at the increments set in the "Notify Value Change Threshold" TLV item unless the "Brightness" characteristic is changed, in which case the "Notify Value Change Threshold" is ignored.

TCLS044: For characteristics that support Linear Derived transitions, verify the accessory sends Broadcast Event Notifications to the controller only at the increments set in the "Notify Value Change Threshold" TLV item.

TCLS045: For characteristics that support Linear Derived transitions, verify the accessory sends notifications to the controller only at the increments set in the "Notify Time Interval Threshold" TLV item.

TCLS046: For characteristics that support Linear Derived transitions, verify the accessory sends Broadcast Event Notifications to the controller only at the increments set in the "Notify Time Interval Threshold" TLV item.

TCLS047: Verify accessories with "Color Temperature" are accurately displaying the intended "Color Temperature" within a degree of 5 percent.

TCLS048: Verify that the accessory rejects "Transition Start" operation with a scale and offset that results in the target value falling outside of the supported range.

TCLS049: An accessory supporting Light Shift must stop all transitions when the accessory is reset to factory settings.

TCLS001 **An accessory supporting the Light Shift feature must include the "Characteristic Value Transition Control" characteristic, the "Supported Characteristic Value Transition Configuration" characteristic, and the "Characteristic Value Active Transition Count" characteristic as part of the "Light Bulb" service.**

Applies to Light Bulb accessories that support Light Shift. Perform this test case with HAT using the steps below.

1. Pair and discover accessory.
2. In the sidebar of the Controllers window locate each "Light Bulb" service.
3. Verify that the "Characteristic Value Transition Control", "Supported Characteristic Value Transition Configuration", and "Characteristic Value Active Transition Count" characteristics are included as part of the "Light Bulb" service.

4. Repeat steps 2-3 for each “Light Bulb” service that supports Light Shift.

TCLS002 **An accessory supporting Light Shift must include the “Color Temperature” characteristic as part of the “Light Bulb” service.**

Applies to Light Bulb accessories that support Light Shift. Perform this test case with HAT using the steps below.

1. Pair and discover accessory.
2. In the sidebar of the Controllers window locate each “Light Bulb” service.
3. Verify that the “Color Temperature” characteristic is included as part of the “Light Bulb” service.
4. Repeat steps 2-3 for each “Light Bulb” service that supports Light Shift.

TCLS003 **When the value of the “Color Temperature” characteristic changes, the accessory must update the values of “Hue” and “Saturation” characteristics accordingly and notify the controller.**

Applies to Light Bulb accessories that support Light Shift. Applies to Light Bulb accessories that support Hue and Saturation along with the Color Temperature characteristic. Perform this test case with HAT using the steps below.

1. Pair and discover accessory.
2. For HAP over BLE accessories, enable the “Pair Resume Keep Alive” checkbox with a 27-second interval.
3. In the sidebar of the Controllers window, under the “Light Bulb” service, select the “Hue” characteristic.
4. Select “Read”, notate the value, and select “Enable” to enable Event Notifications.
5. In the sidebar of the Controllers window, select “Saturation”.
6. Select “Read”, notate the value, and select “Enable” to enable Event Notifications.
7. In the sidebar of the Controllers window, select the “Color Temperature” characteristic.
8. Write a value of “250”.
9. In the Events view of Trace, verify that the accessory accepts the write request, and notifications were received for both “Hue” and “Saturation”.
10. Notate and verify that the values for both “Hue” and “Saturation” differ from the ones notated in step 4 and step 6.
11. In the sidebar of the Controllers window, select the “Color Temperature” characteristic.
12. Write a value of “350”.
13. In the Events view of Trace, verify that the accessory accepts the write request, and notifications were received for both “Hue” and “Saturation”.

14. Verify that the values for both “Hue” and “Saturation” differ from the ones notated in step 10.
15. Repeat step 3-15 for each “Light Bulb” service that supports Light Shift.

TCLS004 **When the values of the “Hue” and “Saturation” characteristics change, the accessory must update the value of the “Color Temperature” characteristic accordingly.**

Applies to Light Bulb accessories that support Light Shift. Applies to Light Bulb accessories that support Hue and Saturation along with the Color Temperature characteristic. Perform this test case with HAT using the steps below.

1. Pair and discover accessory.
2. For HAP over BLE accessories, enable “Pair-Resume keep alive” with a 27-second interval.
3. In the sidebar of the Controllers window, select “Color Temperature”.
4. Select the “Enable” button to subscribe to Event Notifications.
5. Select “Read” and notate the value.
6. In the sidebar of the Controllers window, select “Hue”
7. Perform a valid write operation to change “Hue” to a new value.
8. In the sidebar of the Controllers window, select “Saturation”.
9. Perform a valid write operation to change “Saturation” to a new value.
10. In the Events view of the Trace, verify that a notification was received that the “Color Temperature” has been updated.
11. In the sidebar of the Controllers window, select “Color Temperature”.
12. Select “Read” and verify that the value has updated to match the notification value from step 9.
13. Perform valid write operations to the “Hue” and “Saturation” characteristics to values that do not map to a supported “Color Temperature” value.
14. In the Events view of the Trace, verify that a notification was received that the “Color Temperature” characteristic has been updated.
15. Read the value of “Color Temperature” and verify that the value has updated to its minimum supported value.
16. Repeat steps 3-14 for each “Light Bulb” service that supports “Hue” and “Saturation” along with the “Color Temperature” characteristic.

TCLS005 **An accessory supporting the Light Shift feature for “Color Temperature” must support and advertise a “Color Temperature” minimum value of at least 200 mirek and a maximum value of at least 370 mirek.**

Applies to Light Bulb accessories that support Light Shift. Perform this test case with HAT using the steps below.

1. Pair and discover accessory.

2. In the sidebar of the Controllers window, under the “Light Bulb” service, select the “Color Temperature” characteristic.
3. In the Summary panel, verify that the Minimum Value is less than or equal to 200, and the Maximum Value is greater than or equal to 370.
4. Write a value of “200” then select “Read”.
5. In the Events view of Trace, verify that the accessory accepts the write request, and the Read Response contains a value of 200.
6. Write a value of “370” then select “Read”.
7. In the Events view of Trace, verify that the accessory accepts the write requests and the Read Response contains a value of 370.

TCLS006 **If the accessory is power cycled, it must continue transitioning from the point in the transition within 10 minutes of where it lastly was before it powered down.**

Applies to Light Bulb accessories that support Light Shift. Perform this test case with HAT using the steps below.

1. Pair and discover accessory.
2. For HAP over BLE accessories, enable the “Pair Resume Keep Alive” checkbox with a 27-second interval.
3. In the sidebar of the Controllers window, under the “Light Bulb” service, select the “Characteristic Value Transition Control” characteristic.
4. Select the “Build TLV” button, and then select the “Transition Start (Presets)” option.
5. In the “Linear or Linear Derived” menu, choose “Linear Derived”.
6. Leave the “Notify Value Change Threshold”, “Value Update Time Interval”, and “Notify Time Interval Threshold” fields blank.
7. In the “End Behavior” menu, choose “Loop”.
8. Set the “Upper Bound” to “100” and the “Lower Bound” to “0”.
9. Set the “Target Completion Duration Per Point” to “3600000”.
10. Set the “Total Number of Transition Points” to “11”.
11. Select “Build TLV”, then select “Write” to write the TLV value.
12. In the sidebar of the Controllers window, select the “Color Temperature” characteristic.
13. Wait 10 seconds, then select “Read” and notate the value.
14. Manually power off the accessory.
15. In the Accessory Server, select “Disconnect”, and wait 20 minutes.
16. Power the accessory back on and wait for it to begin advertising again.

17. In the sidebar of the Controllers window, select the “Color Temperature” characteristic.
18. Select “Read” and verify that the value is higher than that noted in step 13.
19. Write a value of “200” to abort the current transition.

TCLS007 **The accessory must be able to accept a new transition after being power cycled.**

Applies to Light Bulb accessories that support Light Shift. Perform this test case with HAT using the steps below.

1. Pair and discover accessory.
2. For HAP over BLE accessories, enable the “Pair Resume Keep Alive” checkbox with a 27-second interval.
3. In the sidebar of the Controllers window, under the “Light Bulb” service, select the “Characteristic Value Transition Control” characteristic.
4. Select the “Build TLV” button, and then select the “Transition Start (Presets)” option.
5. In the “Linear or Linear Derived” menu, choose “Linear Derived”.
6. Leave the “Notify Value Change Threshold”, “Value Update Time Interval”, and “Notify Time Interval Threshold” fields blank.
7. In the “End Behavior” menu, choose “Loop”.
8. Set the “Upper Bound” to “100” and the “Lower Bound” to “0”.
9. Set the “Target Completion Duration Per Point” to “60000”.
10. Set the “Total Number of Transition Points” to “11”.
11. Select “Build TLV”, then select “Write” to write the TLV value.
12. In the Accessory Server, select “Disconnect”.
13. Manually power off the accessory and wait 1 minute.
14. Power the accessory back on, and wait for it to begin advertising again.
15. Select the “Build TLV” button, and then select the “Transition Start (Presets)” option.
16. In the “Linear or Linear Derived” menu, choose “Linear Derived”.
17. Leave the “Notify Value Change Threshold”, “Value Update Time Interval”, and “Notify Time Interval Threshold” fields blank.
18. In the “End Behavior” menu, choose “Loop”.
19. Set the “Upper Bound” to “100” and the “Lower Bound” to “0”.
20. Set the “Target Completion Duration Per Point” to “60000”.
21. Set the “Total Number of Transition Points” to “20”.
22. Select “Build TLV”, then select “Write” to write the TLV value.

23. Select "Read" for the Characteristic Value Transition Control characteristic.
24. In the Events view of Trace, verify that the accessory accepts the write request and the Read Response reflects the contents of the TLV written in Step 22.
25. In the sidebar of the Controllers window, select the "Color Temperature" characteristic.
26. Write a value of "200" to abort the current transition.

TCLS008 **Verify that the accessory supports "Transition Start" operations for the supported transition types for all supported characteristics, and the correct values are returned in Write Responses.**

Applies to Light Bulb accessories that support Light Shift. Perform this test case with HAT using the steps below.

1. Pair and discover accessory.
2. For HAP over BLE accessories, enable the "Pair Resume Keep Alive" checkbox with a 27-second interval.
3. In the sidebar of the Controllers window, under the "Light Bulb" service, select the "Color Temperature" characteristic.
4. Select "Enable" to enable Event Notifications.
5. In the sidebar of the Controllers window, select the "Supported Characteristic Value Transition Control" characteristic.
6. Select "Read".
7. In the Events view of the Trace, verify that the Read Response contains "Supported Transition TLV" item(s) for each supported characteristic, with sub-TLV item that includes a "HAP Instance ID" item with the supported characteristic IID, and a "Supported Characteristic Value Transition Types" item with a bit mask value of "bit:0 (Linear)" and/or "bit 1: (Linear Derived)". Notate the IID(s) and Transition Type(s).
8. If the accessory supports Linear Derived transitions, locate the Controllers window, and select "Characteristic Value Transition Control".
9. For HAP over BLE accessories, enable the "Write with Response" checkbox.
10. Select the "Build TLV" button, and then select the "Transition Start (Presets)" option.
11. In the "Linear or Linear Derived" menu, choose "Linear Derived".
12. Leave the "Notify Value Change Threshold", "Value Update Time Interval", and "Notify Time Interval Threshold" fields blank.
13. In the "End Behavior" menu, choose "Loop".
14. Set the "Upper Bound" to "100" and the "Lower Bound" to "0".
15. Set the "Target Completion Duration Per Point" to "60000".
16. Set the "Total Number of Transition Points" to "11".

17. Select “Build TLV”, then select “Write” to write the TLV value.
18. In the Events view of Trace, verify that the Write Response contains a “Transition State” TLV item, with a sub-TLV that contains “Active Transition Contexts”, with a sub-TLV that contains “HAP Instance ID” item which matches the “Color Temperature” IID, a “Controller Context” item that matches the Controller Context found in the Write Request, and “Time Elapsed Since Start” item which shows time since the transition began in milliseconds.
19. Verify notifications for “Color Temperature” changes are received.
20. In the sidebar of the Controllers window, select the “Brightness” characteristic.
21. Write a new value to the “Brightness” characteristic.
22. In the Events view of Trace, verify that a notification was received for the updated “Color Temperature” value. Note the value.
23. If the accessory supports Linear transitions, locate the Controllers window, and select “Characteristic Value Transition Control”.
24. For HAP over BLE accessories, enable the “Write with Response” checkbox.
25. Select the “Build TLV” button, and then select the “Transition Start (Manual)” option.
26. Set the “HAP Instance ID” to the IID of the “Brightness” characteristic, the “Controller Context” to any hex value, and the “End Behavior” to “Loop” .
27. Leave the “Notify Value Change Threshold”, “Value Update Time Interval”, and “Notify Time Interval Threshold” fields blank.
28. Create 10 Transition Points, each using a “Target Completion Duration” of “60000”, leaving the “Start Delay Duration” blank.
29. Set the “Target Value” of Transition Points in the following order: the first “0A”, the second “14”, the third “1E”, the fourth “28”, the fifth “32”, the sixth “3C”, the seventh “46”, the eighth “50”, the ninth “5A, and the tenth “64”.
30. Set the “Start Condition” to “None” then select “Add Transition”.
31. Select “Build TLV”, then select “Write” to write the TLV value.
32. In the Events view of the Trace, verify that the Write Response contains a Transition State” TLV item, with a sub-TLV that contains Active Transition Contexts”, with a sub-TLV that contains HAP Instance ID” item which matches the “Brightness” IID, a Controller Context” item that matches the Controller Context found in the Write Request, and “Time Elapsed Since Start” item which shows time since the transition began in milliseconds.
33. After 1 minute, read the value of “Color Temperature” and verify it is different from step 22.
34. In the sidebar of the Controllers window, select the “Color Temperature” characteristic.
35. Write a value of “200” to abort the current transition.

TCLS009 **Verify the accessory can successfully perform a Fetch Operation and that it returns the expected parameters after a Transition has been applied.**

Applies to Light Bulb accessories that support Light Shift. Perform this test case with HAT using the steps below.

1. Pair and discover accessory.
2. For HAP over BLE accessories, enable the "Pair Resume Keep Alive" checkbox with a 27-second interval.
3. In the sidebar of the Controllers window, under the "Light Bulb" service, select "Characteristic Value Transition Control".
4. In the Options panel, enable the "Write with Response" checkbox.
5. Select the "Build TLV" button and choose the "Transition Fetch" option.
6. Enter the IID of the "Color Temperature" characteristic, select "Build TLV", and then write the value.
7. In the Events view of the Trace, verify that the write response to the Transition Fetch operation contains a value of "0 bytes".
8. In the sidebar of the Controllers window, select "Characteristic Value Transition Control".
9. In the "Linear or Linear Derived" menu, choose "Linear Derived".
10. Leave the "Notify Value Change Threshold", "Value Update Time Interval", and "Notify Time Interval Threshold" fields blank.
11. In the "End Behavior" menu, choose "Loop".
12. Set the "Upper Bound" to "100" and the "Lower Bound" to "0".
13. Set the "Target Completion Duration Per Point" to "60000".
14. Set the "Total Number of Transition Points" to "11".
15. Select "Build TLV", then select "Write" to write the TLV value.
16. Select the "Build TLV" button and choose the "Transition Fetch" option.
17. Enter the IID of the "Color Temperature" characteristic, select "Build TLV", and then write the value.
18. In the Events view of Trace, verify that the Write Response to the Transition Fetch operation contains all of the transition TLV items and values used in step 15.
19. In the sidebar of the Controllers window, select "Characteristic Value Transition Control", and select "Read".
20. In the Events view of Trace, verify that the Read Response contains a TLV item "Active Transition Contexts", with sub-TLV items "HAP Instance ID" with the instance ID of the "Color Temperature" characteristic, "Controller Context" that matches the context used in the write request, and "Time Elapsed Since Start" which shows the time since the transition started in milliseconds.
21. In the sidebar of the Controllers window, select the "Color Temperature" characteristic.
22. Write a value of "200" to abort the current transition.

TCLS010 **Verify that the "Characteristic Value Transition Control" characteristic returns the expected values before and after transitions have been applied.**

Applies to Light Bulb accessories that support Light Shift. Perform this test case with HAT using the steps below.

1. Pair and discover accessory.
2. For HAP over BLE accessories, enable the "Pair Resume Keep Alive" checkbox with a 27-second interval.
3. In the sidebar of the Controllers window, under the "Light Bulb" service, select "Characteristic Value Transition Control".
4. In the Options panel, enable the "Write with Response" checkbox, then select "Read" to read the current value.
5. In the Events view of Trace, verify that the Read Response contains a value of 0 bytes.
6. Select the "Build TLV" button, and then select the "Transition Start (Presets)" option, and in the "Linear or Linear Derived" menu, choose "Linear Derived".
7. Leave the "Notify Value Change Threshold", "Value Update Time Interval", and "Notify Time Interval Threshold" fields blank.
8. In the "End Behavior" menu, choose "Loop".
9. Set the "Upper Bound" to "100" and the "Lower Bound" to "0".
10. Set the "Target Completion Duration Per Point" to "60000".
11. Set the "Total Number of Transition Points" to "11".
12. Select "Build TLV", then select "Write" to write the TLV value.
13. Select "Read" to read the current value of the "Characteristic Value Transition Control" characteristic.
14. In the Events view of Trace, verify that the "Color Temperature" immediately changes to the curve. Verify that the Read Response contains a TLV item "Active Controller Contexts", with sub-TLV items "HAP Instance ID" with the instance ID of the "Color Temperature" characteristic, "Controller Context" that matches the context used in the write request, and "Time Elapsed Since Start" which shows the time since the transition started in milliseconds.
15. In the sidebar of the Controllers window, select the "Color Temperature" characteristic.
16. Write a value of "200" to abort the current transition.

TCLS011 **Verify that the contents of the Supported Transition characteristic read response.**

Applies to Light Bulb accessories that support Light Shift. Perform this test case with HAT using the steps below.

1. Pair and discover accessory.
2. In the sidebar of the Controllers window, under the "Light Bulb" service, select the "Supported Characteristic Value Transition Control" characteristic.

3. Select “Read” to read the value.
4. In the Events view of Trace, verify that the Read Response contains Supported Transition TLV item(s) for each supported characteristic, with sub-TLV item that includes a “HAP Instance ID” item with the supported characteristic IID and a “Supported Characteristic Value Transition Types” item with a bit mask value of bit:0 (Linear) and/or bit 1: (Linear Derived).

TCLS012 **Accessory must be able to support up to 52 Transition Points in a single Linear Derived transition for Color Temperature. Accessory must reject write requests, with the correct status code, when the number of transition points specified in the Transition are more than the accessory can support.**

Applies to Light Bulb accessories that support Light Shift. Perform this test case with HAT using the steps below.

1. Pair and discover accessory.
2. For HAP over BLE accessories, enable the “Pair Resume Keep Alive” checkbox with a 27-second interval.
3. In the sidebar of the Controllers window, under the “Light Bulb” service, select “Characteristic Value Transition Control”.
4. In the Options panel, enable the “Write with Response” checkbox.
5. Select the “Build TLV” button and then select the “Transition Start (Presets)” option. In the “Linear or Linear Derived” menu, choose “Linear Derived”.
6. Leave the “Notify Value Change Threshold”, “Value Update Time Interval”, and “Notify Time Interval Threshold” fields blank.
7. In the “End Behavior” menu, choose “No Change”.
8. Set the “Upper Bound” to “100” and the “Lower Bound” to “0”.
9. Set the “Target Completion Duration Per Point” to “1661538”.
10. Set the “Total Number of Transition Points” to “52”.
11. Select “Build TLV”, then select “Write” to write the TLV value.
12. In the Events view of Trace, verify that the accessory accepts the write request.
13. In the sidebar of the Controllers window, select “Characteristic Value Transition Control”.
14. Select the “Build TLV” button, and then select the “Transition Start (Presets)” option, and in the “Linear or Linear Derived” menu, choose “Linear Derived”.
15. Leave the “Notify Value Change Threshold”, “Value Update Time Interval”, and “Notify Time Interval Threshold” fields blank.
16. In the “End Behavior” menu, choose “No Change”.
17. Set the “Upper Bound” to “100” and the “Lower Bound” to “0”.
18. Set the “Target Completion Duration Per Point” to “60000”.

19. Set the "Total Number of Transition Points" to a value greater than the accessory can support.
20. Select "Build TLV", then select "Write" to write the TLV value.
21. In the Events view of the Trace, verify that the write request was rejected. For HAP over Wi-Fi or Ethernet accessories the Status Code must be "70407", and for HAP over BLE accessories the Status Code must be "0x07".

TCLS013 **When the "Start Delay Duration" TLV item is absent in the "Transition Start" request for a "Linear Derived Transition", that accessory assumes a start delay duration of "0".**

Applies to Light Bulb accessories that support Light Shift. Perform this test case with HAT using the steps below.

1. Pair and discover accessory.
2. For HAP over BLE accessories, enable the "Pair Resume Keep Alive" checkbox with a 27-second interval.
3. In the sidebar of the Controllers window, select the "Brightness" characteristic.
4. Perform a valid write operation to set the "Brightness" value to "80".
5. Select the "Color Temperature" characteristic.
6. Perform a valid write operation to set the "Color Temperature" value to "300", then select "Enable" to enable Event Notifications.
7. In the sidebar of the Controllers window, select the "Characteristic Value Transition Control" characteristic.
8. In the Options panel, enable the "Write with Response" checkbox.
9. Select "Build TLV" and choose the "Transition Start (Manual)" option.
10. Enter the instance ID for "Color Temperature" into the "HAP Instance ID" field.
11. Set the "End Behavior" to "No Change".
12. In the "Linear or Linear Derived" menu, choose "Linear Derived".
13. Create a new transition point using the "+" icon.
14. Set the Scale to "2.01", Offset to "1.01", Target Completion Duration to "0", and "Start Delay Duration" to "10000".
15. Create a second transition point with the Scale set to "3.65", Offset to "1.01", and Target Completion Duration set to "240000". Leave the "Start Delay Duration" field blank.
16. Set the "Source Instance ID" to the "Brightness" characteristic's Instance ID, and set the Lower Bound to "0a" (10) and the Upper Bound to "5a" (90).
17. Select "Add Transition", select "Build TLV", and then write the value.
18. Verify that the accessory accepts the write request and does not change the "Color Temperature" immediately.

19. Verify that the “Color Temperature” changes after 10 seconds.
20. In the Events view of the Trace, verify that the first notification for the Color Temperature’s updated value is received 10 seconds after writing the value in step 17.
21. Select “Build TLV”, select the “Transition Fetch” option, and set the “HAP Instance ID” to the IID of the “Color Temperature” characteristic.
22. Select “Build TLV” and write the value before the 4-minute transition completes.
23. Verify that the write response to the “Fetch” operation includes “Transition Point” TLV items for both transition points, with the second transition’s “Start Delay Duration” item set to “0”.
24. In the Events view of the Trace, verify that the first notification for the Color Temperature’s next updated value is received 1 minute after the notification from step 19.
25. In the sidebar of the Controllers window, select the “Color Temperature” characteristic.
26. Write a value of “300” to abort the current transition.
27. In the sidebar of the Controllers window, select the “Characteristic Value Transition Control” characteristic.
28. Select “Build TLV” and choose the “Transition Start (Manual)” option.
29. Enter the instance ID for “Color Temperature” into the “HAP Instance ID” field.
30. Set the “End Behavior” to “No Change”.
31. In the “Linear or Linear Derived” menu, choose “Linear Derived”.
32. Create a new transition point using the “+” icon.
33. Set the Scale to “2.49”, Offset to “1.01”, and Target Completion Duration to “0”. Leave the “Start Delay Duration” field blank.
34. Create a second transition point with the Scale set to “3.65”, Offset to “1.01”, Target Completion Duration set to “240000”, and set the “Start Delay Duration” to “60000’.
35. Set the “Source Instance ID” to the “Brightness” characteristic’s Instance ID, and set the Lower Bound to “0a” (10) and the Upper Bound to “5a” (90).
36. Select “Add Transition”, select “Build TLV”, and then write the value.
37. Verify that the accessory accepts the write request and immediately changes the “Color Temperature”.
38. In the Events view of the Trace, verify that the first notification for the Color Temperature’s updated value is received immediately.
39. Verify that the “Color Temperature” changes again after 2 minutes.
40. In the Events view of the Trace, verify that the next notification for the Color Temperature’s updated value is received 2 minutes after the notification found in step 38.
41. Select “Build TLV”, select the “Transition Fetch” option, and set “HAP Instance ID” to the IID of the “Color Temperature” characteristic.
42. Select “Build TLV” and write the value before the 4-minute transition is completed.

43. Verify that the write response to the "Fetch" operation includes "Transition Point" TLV items for both transition points, where "Start Delay Duration" item is set to "0" for the first and "60000" for the second.

44. In the sidebar of the Controllers window, select the "Color Temperature" characteristic.

45. Write a value of "300" to abort the current transition.

TCLS014 **Verify that the millisecond calculation accuracy on the accessory for the "Characteristic Value Transition Control" characteristic's "Target Completion Duration" TLV item.**

Applies to Light Bulb accessories that support Light Shift. Perform this test case with HAT using the steps below.

1. Pair and discover accessory.
2. For HAP over BLE accessories, enable the "Pair Resume Keep Alive" checkbox with a 27-second interval.
3. In the sidebar of the Controllers window, select the "Color Temperature" characteristic.
4. Select the "Enable" button to subscribe to Event Notifications.
5. In the sidebar of the Controllers window, select the "Characteristic Value Transition Control" characteristic.
6. Select "Build TLV" and then select the "Transition Start (Presets) option. In the "Linear or Linear Derived" menu, choose "Linear Derived".
7. Set the "Upper Bound" to "100" and the "Lower Bound" to "0".
8. Set the "End Behavior" to "Loop", set the "Target Completion Duration Per Point" to "60000", and set the "Number of Transition Points" to "11". Leave the "Notify Value Change Threshold", "Notify Time Interval Threshold", and "Value Update Time Interval" fields blank.
9. Select "Build TLV", write the value, and then verify that the accessory accepts the write request and starts the "Color Temperature" transition.
10. In the Events view of the Trace, wait for the first notification from Color Temperature.
11. Wait 60 seconds, and verify that each subsequent notification arrives in 60 seconds intervals.
12. Continue to monitor for notifications for 5 minutes, and verify that they continue to arrive in 60 second intervals.
13. In the sidebar of the Controllers window, select the "Color Temperature" characteristic.
14. Write a value of "200" to abort the current transition.

TCLS015 **For characteristics that support Linear Derived transitions, verify that the transition repeats if "End Behavior" is set to "Loop", or stops once the transition completes if set to "No Change".**

Applies to Light Bulb accessories that support Light Shift. Perform this test case with HAT using the steps below.

1. Pair and discover accessory.
2. In the sidebar of the Controllers window, select Characteristic Value Transition Control.
3. Select “Build TLV” and choose the “Transition Start (Presets)” option.
4. In the “Linear or Linear Derived” menu, choose “Linear Derived”.
5. Set the “End Behavior” to “Loop”, set the “Target Completion Duration Per Point” to “60000”, and set the “Total Number of Transition Points” to “5”.
6. Select “Build TLV” and write the value.
7. Verify that the accessory successfully accepts the write request and begins to transition the “Color Temperature” characteristic.
8. After 5 minutes, verify that the “Color Temperature” transition repeats.
9. In the sidebar of the Controllers window, select Characteristic Value Transition Control.
10. Select “Build TLV” and choose the “Transition Start (Presets)” option.
11. In the “Linear or Linear Derived” menu, choose “Linear Derived”.
12. Set the “End Behavior” to “No Change”, set the “Target Completion Duration Per Point” to “60000”, and set the “Total Number of Transition Points” to “5”.
13. Select “Build TLV” and write the value.
14. Verify that the accessory accepts the write request and begins to transition the “Color Temperature” characteristic every 60 seconds.
15. After 5 minutes, verify that the “Color Temperature” value remains the same, and the transitions have stopped.
16. Verify that the TLV is written and that the accessory starts the “Color Temperature” transition.
17. In the sidebar of the Controllers window, select Characteristic Value Active Transition Count.
18. Select “Read” to read the current value and verify its “0”.

TCLS016 **Verify that the permissions and formats for the “Characteristic Value Transition Control”, “Supported Characteristic Value Transition Configuration”, and “Characteristic Value Active Transition Count” characteristics.**

Applies to Light Bulb accessories that support Light Shift. Perform this test case with HAT using the steps below.

1. Pair and discover accessory.
2. In the sidebar of the Controllers window, under the “Light Bulb” service, select “Characteristic Value Transition Control”.
3. For HAP over Wi-Fi or Ethernet accessories, verify that the permissions are: Paired Read, Paired Write, and Write Response. For HAP over BLE, verify that the permissions are: Paired Read and Paired Write.
4. Verify that the format is “TLV8” and there is no unit set.

5. In the sidebar of the Controllers window, select “Supported Characteristic Value Transition Configuration”.
6. Verify that the permissions are: Paired Read only.
7. Verify that the format is “TLV8” and there is no unit set.
8. In the sidebar of the Controllers window, select “Characteristic Value Active Transition Count”.
9. For HAP over Wi-Fi or Ethernet, verify that the permissions are: Paired Read and Notify. For HAP over BLE, verify that the permissions are: Paired Read, Indicate, Indicate (Disconnected), and Broadcast.
10. Verify that the format is “uint8” and there is no unit set.

TCLS017 **Verify the accessory rejects a “Transition Start” operation when an Instance ID that does not support Transitions is used as the target characteristic in the request.**

Applies to Light Bulb accessories that support Light Shift. Perform this test case with HAT using the steps below.

1. Pair and discover accessory.
2. For HAP over BLE accessories, enable the “Pair Resume Keep Alive” checkbox with a 27-second interval.
3. In the sidebar of the Controllers window, under the “Light Bulb” service, select “Characteristic Value Transition Control”.
4. Select “Build TLV” and then choose the “Transition Start (Manual)” option.
5. Set the “HAP Instance ID” to the IID of the “On” characteristic, set the “End Behavior” to “Loop”, in the “Linear or Linear Derived” menu, choose “Linear”.
6. In the first transition point enter “0” for the “Target Value”, and “60000” for the “Target Completion Duration”.
7. Select the “+” icon to create a second transition point.
8. Enter “1” for the “Target Value”, and “60000” for the “Target Completion Duration”, and then select “Add Transition”.
9. Select “Build TLV” and then write the value.
10. In the Events view of Trace, verify that the accessory rejects the write request.
11. For HAP over Wi-Fi or Ethernet, in Events view of the Trace, verify that the accessory responds with HAP status code -70410.
12. For HAP over BLE, in the HAP Transactions view, verify that the accessory responds with 0x06 (Invalid Request).

TCLS018 **For characteristics that support Linear Transitions, verify the accessory rejects “Transition Start” operations when unsupported Target Values are used.**

Applies to Light Bulb accessories that support Light Shift. Perform this test case with HAT using the steps below.

1. Pair and discover accessory.
2. For HAP over BLE accessories, enable the "Pair Resume Keep Alive" checkbox with a 27-second interval.
3. In the sidebar of the Controllers window, under the "Light Bulb" service, select "Characteristic Value Transition Control".
4. Select the "Build TLV" button, and then select the "Transition Start (Manual)" option. Set the target "HAP Instance ID" to a characteristic that supports transitions, set the "End Behavior" to "Loop", select "Linear" in the "Linear or Linear Derived" menu, and select "+" to add the first transition point.
5. In the "Target Value" field, enter a hex value that the target characteristic does not support. E.g. For brightness, use "96" (hex representation of 150). Set the "Target Completion Duration" to "1000".
6. Select "+" to add another transition point.
7. In the "Target Value" field, enter another hex value that the target characteristic does not support. E.g. For brightness, use "C8" (hex representation of 200). Set the "Target Completion Duration" to "1000".
8. Select "Add Transition", select "Build TLV", and then write the value.
9. In the Events view of Trace, verify that the accessory rejects the write request.
10. For HAP over Wi-Fi or Ethernet, in Events view of the Trace, verify that the accessory responds with HAP status code -70410. For
11. HAP over BLE, in the HAP Transactions view, verify that the accessory responds with 0x06 (Invalid Request).

TCLS019 **Verify the accessory rejects "Transition Start" operations when unsupported Lower and Upper bound values are used.**

Applies to Light Bulb accessories that support Light Shift. Perform this test case with HAT using the steps below.

1. Pair and discover accessory.
2. For HAP over BLE accessories, enable the "Pair Resume Keep Alive" checkbox with a 27-second interval.
3. In the sidebar of the Controllers window, under the "Light Bulb" service, select "Characteristic Value Transition Control".
4. Select the "Build TLV" button, and choose the "Transition Start (Presets)" option.
5. In the "Linear or Linear Derived" menu, choose "Linear Derived".
6. Configure the preset's Lower and Upper Bounds to values that are unsupported by the source characteristic. E.g. for Brightness, use values outside of the supported 0-100 range.
7. In the Events view of Trace, verify that the accessory rejects the write request.

8. For HAP over Wi-Fi or Ethernet, in Events view of the Trace, verify that the accessory responds with HAP status code -70410. For
9. HAP over BLE, in the HAP Transactions view, verify that the accessory responds with 0x06 (Invalid Request).

TCLS021 **After setting the "Brightness" value below the transition's Lower Bound, verify that setting the "Brightness" value below the transition's Lower Bound value doesn't change "Color Temperature" value.**

Applies to Light Bulb accessories that support Light Shift. Perform this test case with HAT using the steps below.

1. Pair and discover accessory.
2. In the sidebar of the Controllers window, select the "Characteristic Value Transition Control" characteristic.
3. In the Options panel, enable the "Write with Response" checkbox.
4. Select the "Build TLV" button, and then select the "Transition Start (Presets)" option, and in the "Linear or Linear Derived" menu, choose "Linear Derived".
5. Configure the preset with Lower Bound equal to 80 and Upper Bound equal to 90, Total Number of Transition Points set to "48", and Target Completion Duration Per Point set to "1800000".
6. Select "Build TLV" and write the value.
7. In Events view of the Trace, verify that a write response was received.
8. Set the "Brightness" to 80.
9. Read and notate the "Color Temperature" value.
10. Set the "Brightness" to 70.
11. Read the "Color Temperature" characteristic, and verify that the value is equal to the previously read value.
12. In the sidebar of the Controllers window, select the "Color Temperature" characteristic.
13. Write a value of "200" to abort the current transition.

TCLS022 **After setting the "Brightness" value above the transition's Upper Bound, verify that setting the "Brightness" value above the transition's Upper Bound value doesn't change "Color Temperature" value.**

Applies to Light Bulb accessories that support Light Shift. Perform this test case with HAT using the steps below.

1. Pair and discover accessory.
2. In the sidebar of the Controllers window, select the "Characteristic Value Transition Control" characteristic.
3. In the Options panel, enable the "Write with Response" checkbox.

4. Select the "Build TLV" button, and then select the "Transition Start (Presets)" option, and in the "Linear or Linear Derived" menu, choose "Linear Derived".
5. Configure the preset with Lower Bound equal to 80 and Upper Bound equal to 90, Total Number of Transition Points set to "48", and Target Completion Duration Per Point set to "1800000".
6. Select "Build TLV" and write the value.
7. In the Events view of the Trace, verify that a write response was received.
8. Set the "Brightness" to 90.
9. Read and notate the "Color Temperature" characteristic value.
10. Set the "Brightness" to 100.
11. Read the "Color Temperature" characteristic, and verify that the value is equal to the previously read value.
12. In the sidebar of the Controllers window, select the "Color Temperature" characteristic.
13. Write a value of "200" to abort the current transition.

TCLS023 **Verify the accessory repeats a Linear Transition for the "Brightness" characteristic with an "End Behavior" set to "Loop".**

Applies to Light Bulb accessories that support Light Shift. Perform this test case with HAT using the steps below.

1. Pair and discover accessory.
2. For HAP over BLE accessories, enable the "Pair Resume Keep Alive" checkbox with a 27-second interval.
3. In the sidebar of the Controllers window, under the "Light Bulb" service, select the "Brightness" characteristic.
4. Write a value of 0.
5. In the sidebar of the Controllers window, select Characteristic Value Transition Control.
6. Select the "Build TLV" button, and then select the "Transition Start (Manual)" option. Set the target "HAP Instance ID" to the "Brightness" characteristic Instance ID, set the "End Behavior" to "Loop", and select "Linear" in the "Linear or Linear Derived" menu.
7. In the "Target Value" field, enter the value "0A" (hex representation of 10), set the "Target Completion Duration" to "60000", and "Start Delay Duration" to 0.
8. Select "+" to add another transition point.
9. In the "Target Value" field, enter "64" (hex representation of 100), set the "Target Completion Duration" to "60000", and "Start Delay Duration" to "0".
10. Select "Add Transition", select "Build TLV", and then write the value.
11. In the Events view of Trace, verify that the accessory accepts the write request and begins cycling through "Brightness" values every 2 minutes.

12. In the sidebar of the Controllers window, select the "Brightness" characteristic.
13. Write a value of "0" to abort the current transition.

TCLS024 **Verify the accessory does not repeat a Linear Transition for the "Brightness" characteristic with an "End Behavior" set to "No Change".**

Applies to Light Bulb accessories that support Light Shift. Perform this test case with HAT using the steps below.

1. Pair and discover accessory.
2. In the sidebar of the Controllers window, under the Lightbulb service, select Brightness.
3. Write a value of 0.
4. In the sidebar of the Controllers window, select Characteristic Value Transition Control.
5. Select the "Build TLV" button, and then select the "Transition Start (Manual)" option. Set the target "HAP Instance ID" to the "Brightness" characteristic Instance ID, set the "End Behavior" to "No Change", and select "Linear" in the "Linear or Linear Derived" menu.
6. In the "Target Value" field, enter the value 0A (hex representation of 10), set the "Target Completion Duration" to 60000, and "Start Delay Duration" to 0.
7. Select "+" to add another transition point.
8. In the "Target Value" field, enter 64 (hex representation of 100), set the "Target Completion Duration" to 60000, and set "Start Delay Duration" to 0.
9. Select "Add Transition", select "Build TLV", and then write the value.
10. Verify that the accessory accepts the write request and changes "Brightness" from 10 to 100 within 2 minutes. Once the transition completes, verify that the "Brightness" value remains at 100. E.g. transition does not repeat.
11. In the sidebar of the Controllers window, select the "Brightness" characteristic.
12. Write a value of "0" to abort the current transition.

TCLS025 **For characteristics that support Linear transitions, the accessory must be able to support transition points with the first transition point's "Target Completion Duration" set to "0" ms, and subsequent transition points set to at least "100" ms.**

Applies to Light Bulb accessories that support Light Shift. Perform this test case with HAT using the steps below.

1. Pair and discover accessory.
2. In the sidebar of the Controllers window, select the "Brightness" characteristic under the "Light Bulb" service.
3. Select "Enable" to enable Event Notifications for the "Brightness" characteristic.

4. In the sidebar of the Controllers window, select the "Characteristic Value Transition Control" characteristic.
5. Select the "Build TLV" button, and then select the "Transition Start (Presets)" option.
6. In the "Linear or Linear Derived" menu, choose "Linear".
7. In the "Transition Preset Style" menu, choose "Increasing".
8. in the "Start Condition" menu, choose "None".
9. In the "End Behavior" menu, choose "Loop".
10. Enter "1000" into the "Target Completion Duration Per Point" field.
11. Enter "11" into the "Total Number of Transition Points" field.
12. Leave the "Notify Value Change Threshold" and "Notify Time Interval Threshold" fields blank.
13. Set the "Value Update Time Interval" to "1000".
14. Select "Build TLV" and then select "Write" to write the value.
15. Verify that the accessory accepts the write request and begins to flash.
16. In the Events view of the Trace, verify that a notification is received every second indicating a "Brightness" value change.
17. In the sidebar of the Controllers window, select the "Brightness" characteristic.
18. Write a value of "0" to abort the current transition.

TCLS026 **For characteristics that support Linear transitions, verify the accessory sends notifications to the controller only at the increments set in the "Notify Value Change Threshold" TLV item.**

Applies to Light Bulb accessories that support Light Shift. Perform this test case with HAT using the steps below.

1. Pair and discover accessory.
2. For HAP over BLE accessories, enable the "Pair Resume Keep Alive" checkbox with a 27-second interval.
3. In the sidebar of the Controllers window, select the "Brightness" characteristic under the "Light Bulb" service.
4. Write a value of "100".
5. Select "Enable" to enable Event Notifications for the "Brightness" characteristic.
6. In the sidebar of the Controllers window, select the "Characteristic Value Transition Control" characteristic.
7. Select the "Build TLV" button, and then select the "Transition Start (Presets)" option.
8. In the "Linear or Linear Derived" menu, choose "Linear".
9. Leave the "Value Update Time Interval" and "Notify Time Interval Threshold" fields blank.

10. Set the “Notify Value Change Threshold” to “10”.
11. In the “Transition Preset Style” menu, choose “Increasing”.
12. In the “Start Condition” menu, choose “None”.
13. In the “End Behavior” menu, choose “Loop”.
14. Set the “Target Completion Duration Per Point” to “60000”.
15. Set the “Total Number of Transition Points” to “11”.
16. Select “Build TLV”, and then write the value.
17. Verify that the accessory accepts the write request and immediately changes the “Brightness” value to “0”.
18. Wait 5 minutes.
19. Using the Events view of the Trace, verify the accessory sends notifications only when the value changes by 10. (E.g. 10, 20, 30, 40, etc.)
20. In the sidebar of the Controllers window, select the “Brightness” characteristic.
21. Write a value of “0” to abort the current transition.

TCLS027 **For characteristics that support Linear transitions, verify the accessory sends Broadcast Event Notifications to the controller only at the increments set in the “Notify Value Change Threshold” TLV item.**

Applies to Light Bulb accessories that support Light Shift. Applies to accessories that use HAP over BLE. Perform this test case with HAT using the steps below.

1. Pair and discover accessory.
2. Enable the “Pair Resume Keep Alive” checkbox with a 27-second interval.
3. Under the “HAP Protocol Information” service, select the “Service Signature” characteristic, and then in the “Protocol Configuration” panel, enable the “Set Advertising Identifier”, “Get all params”, and “Generate Broadcast Keys” checkboxes.
4. Under “Desired Advertising Identifier”, enter “AABBCCDDEEFF”, and then select “Send”.
5. Select the “Brightness” characteristic under the “Lightbulb” service.
6. Write a value of “100”.
7. In the “Characteristic Configuration” panel, enter “1” for broadcast interval and select “Enable Broadcast Notifications”.
8. In the sidebar of the Controllers window, select the “Characteristic Value Transition Control” characteristic.
9. Select the “Build TLV” button, and then select the “Transition Start (Presets)” option.
10. In the “Linear or Linear Derived” menu, choose “Linear”.
11. Set the “Value Update Time Interval” to “1”.

12. Set the “Notify Time Interval Threshold” to “1”.
13. Set the “Notify Value Change Threshold” to “10”.
14. In the “End Behavior” menu, choose “Loop”.
15. In the “Start Condition” menu, choose “None”.
16. In the “Transition Preset Style” menu, choose “Increasing”.
17. Set the “Target Completion Duration Per Point” to “60000”.
18. Set the “Total Number of Transition Points” to “11”.
19. Select “Build TLV”, and then write the value.
20. After the write completes, select the “Disconnect” button on the main accessory server view.
21. Wait 5 minutes.
22. Using the BLE Discovery view in the trace window, verify the accessory sends Encrypted Broadcast notifications only when the value changes by 10. (E.g. 10, 20, 30, etc.)
23. In the sidebar of the Controllers window, select the “Brightness” characteristic.
24. Write a value of “0” to abort the current transition.

TCLS028 **For characteristics that support Linear transitions, verify the accessory sends notifications to the controller only at the increments set in the “Notify Time Interval Threshold” TLV item.**

Applies to Light Bulb accessories that support Light Shift. Perform this test case with HAT using the steps below.

1. Pair and discover accessory.
2. For HAP over BLE accessories, enable the “Pair Resume Keep Alive” checkbox with a 27-second interval.
3. In the sidebar of the Controllers window, select the “Brightness” characteristic under the “Lightbulb” service.
4. Write a value of “100”.
5. Select “Enable” to enable Event Notifications for the “Brightness” characteristic.
6. In the sidebar of the Controllers window, select the “Characteristic Value Transition Control” characteristic.
7. Select the “Build TLV” button, and then select the “Transition Start (Presets)” option.
8. In the “Linear or Linear Derived” menu, choose “Linear”.
9. Set the “Value Update Time Interval” to “1”.
10. Set the “Notify Time Interval Threshold” to “120000”.
11. Set the “Notify Value Change Threshold” to “1”.

12. In the “End Behavior” menu, choose “Loop”.
13. In the “Start Condition” menu, choose “None”.
14. In the “Transition Preset Style” menu, choose “Increasing”.
15. Set the “Target Completion Duration Per Point” to “60000”.
16. Set the “Total Number of Transition Points” to “11”.
17. Select “Build TLV”, and then write the value.
18. Verify that the accessory accepts the write request and immediately changes the “Brightness” value.
19. In the Events view of the Trace, verify that the first notification for the Brightness’s updated value is received.
20. Wait 5 minutes.
21. Verify the accessory sends subsequent notifications in no less than 120 second (2 minute) intervals.
22. In the sidebar of the Controllers window, select the “Brightness” characteristic.
23. Write a value of “0” to abort the current transition.

TCLS029 **For characteristics that support Linear transitions, verify the accessory sends Broadcast Event Notifications to the controller only at the increments set in the “Notify Time Interval Threshold” TLV item.**

Applies to Light Bulb accessories that support Light Shift. Applies to accessories that use HAP over BLE. Perform this test case with HAT using the steps below.

1. Pair and discover accessory.
2. Enable the “Pair Resume Keep Alive” checkbox with a 27-second interval.
3. Under the “HAP Protocol Information” service, select the “Service Signature” characteristic, and then in the “Protocol Configuration” panel, enable the “Set Advertising Identifier”, “Get all params”, and “Generate Broadcast Keys” checkboxes.
4. Under “Desired Advertising Identifier”, enter “AABBCCDDEEFF”, and then select “Send”.
5. Select the “Brightness” characteristic under the “Lightbulb” service.
6. Write a value of “100”.
7. In the “Characteristic Configuration” panel, enter “1” for broadcast interval and select “Enable Broadcast Notifications”.
8. In the sidebar of the Controllers window, select the “Characteristic Value Transition Control” characteristic.
9. Select the “Build TLV” button, and then select the “Transition Start (Presets)” option.
10. In the “Linear or Linear Derived” menu, choose “Linear”.
11. Set the “Value Update Time Interval” to “1”.
12. Set the “Notify Time Interval Threshold” to “120000’.

13. Set the “Notify Value Change Threshold” to “1”.
14. In the “End Behavior” menu, choose “Loop”.
15. In the “Start Condition” menu, choose “None”.
16. In the “Transition Preset Style” menu, choose “Increasing”.
17. Set the “Target Completion Duration Per Point” to “60000”.
18. Set the “Total Number of Transition Points” to “11”.
19. Select “Build TLV”, and then write the value.
20. After the write completes, select the “Disconnect” button on the main accessory server view.
21. Wait 5 minutes.
22. Using the BLE Discovery view in the trace window, verify the accessory sends Encrypted Broadcast notifications only every 120 seconds (2 minutes).
23. In the sidebar of the Controllers window, select the “Brightness” characteristic.
24. Write a value of “0” to abort the current transition.

TCLS030 **Verify the accessory responds to “Fetch” operations for characteristics that do not support transitions with the correct HAP status code.**

Applies to Light Bulb accessories that support Light Shift. Perform this test case with HAT using the steps below.

1. Pair and discover accessory.
2. In the sidebar of the Controllers window, select the “Characteristic Value Transition Control” characteristic.
3. In the Options panel, enable the “Write with Response” checkbox.
4. Select the “Build TLV” button, and then select the “Transition Fetch” option.
5. Enter an IID of an existing characteristic that does not support transitions into the “HAP Instance ID” field, e.g. the IID of the “Identify” characteristic.
6. Select “Build TLV” and then select “Write” to write the value.
7. In the Events view of the Trace, select the write response and select “Details” to show the details.
8. For HAP over Wi-Fi or Ethernet accessories, verify that the accessory responds with HAP status code -70402.
9. For HAP over BLE accessories, verify that the accessory responds with HAP status code 0x06.

TCLS031 **Verify the accessory successfully ends an active “Linear Derived” transition when the accessory receives a “Transition Start” operation to the “Characteristic Value Transition Control” characteristic containing the HAP Instance ID of the characteristic without any transitions specified.**

Applies to Light Bulb accessories that support Light Shift. Perform this test case with HAT using the steps below.

1. Pair and discover accessory.
2. For HAP over BLE accessories, enable the "Pair Resume Keep Alive" checkbox with a 27-second interval.
3. In the sidebar of the Controllers window, select the "Color Temperature" characteristic.
4. Select the "Enable" button to subscribe to Event Notifications.
5. Select the "Characteristic Value Active Transition Count" characteristic.
6. Select the "Enable" button to subscribe to Event Notifications.
7. In the sidebar of the Controllers window, select the "Characteristic Value Transition Control" characteristic.
8. In the Options panel, enable the "Write with Response" checkbox.
9. Select "Build TLV" and then select the "Transition Start (Presets)" option, and in the "Linear or Linear Derived" menu, choose "Linear Derived".
10. Leave the "Notify Value Change Threshold" field blank.
11. Set the "Notify Time Interval Threshold" to "1".
12. Set the "Value Update Time Interval" to "1".
13. Set the "End Behavior" to "Loop".
14. Set the "Upper Bound" to "100" and the "Lower Bound" to "0".
15. Set the "Target Completion Duration Per Point" to "1800000".
16. Set the "Number of Transition Points" to "11",
17. Select "Build TLV", write the value, and then verify that the accessory accepts the write request.
18. In the Events view of the Trace, verify that a notification for the "Characteristic Value Active Transition Count" characteristic is received, and contains a value of "1".
19. Verify that the first notification for the Color Temperature's updated value is received.
20. In the sidebar of the Controllers window, select the "Brightness" characteristic.
21. Perform a valid write operation to change the "Brightness" value.
22. In the Events view of the Trace, verify that a notification for the Color Temperature's updated value is received.
23. In the sidebar of the Controllers window, select the "Brightness" characteristic.
24. Perform a valid write operation to change the "Brightness" value again.
25. In the Events view of the Trace, verify that a notification for the Color Temperature's updated value is received.

26. In the sidebar of the Controllers window, select the "Characteristic Value Transition Control" characteristic.
27. Select the "Build TLV" button, and then select the "Transition Start (Manual)" option.
28. Set the target "HAP Instance ID" to the "Color Temperature" characteristic Instance ID.
29. Select the "Add Transition" button, select "Build TLV", and then write the value to abort the current transition.
30. In the Events view of the Trace, verify that the write response contains the "Transition State" TLV item, with a 0 length value.
31. Verify that a notification for the "Characteristic Value Active Transition Count" characteristic is received, and contains a value of "0".
32. Verify that the "Color Temperature" immediately stops transitioning.
33. In the sidebar of the Controllers window, select the "Color Temperature" characteristic.
34. Select the "Read" button to read the current value. Notate the current value.
35. In the sidebar of the Controllers window, select the "Brightness" characteristic.
36. Perform a valid write operation to change the "Brightness" value.
37. In the Events view of the Trace, verify that a notification for the Color Temperature's is not received.
38. Wait 1 minute.
39. In the sidebar of the Controllers window, select the "Color Temperature" characteristic.
40. Select the "Read" button to read the current value.
41. Verify that the value is the same as it was in step 34.

TCLS032 **Verify the accessory successfully ends an active "Linear" transition when the accessory receives a "Transition Start" operation to the "Characteristic Value Transition Control" characteristic containing the HAP Instance ID of the characteristic without any transitions specified.**

Applies to Light Bulb accessories that support Light Shift. Perform this test case with HAT using the steps below.

1. Pair and discover accessory.
2. For HAP over BLE accessories, enable the "Pair Resume Keep Alive" checkbox with a 27-second interval.
3. In the sidebar of the Controllers window, select the "Brightness" characteristic.
4. Select the "Enable" button to subscribe to Event Notifications.
5. Select the "Characteristic Value Active Transition Count" characteristic.
6. Select the "Enable" button to subscribe to Event Notifications.
7. In the sidebar of the Controllers window, select the "Characteristic Value Transition Control" characteristic.

8. In the Options panel, enable the “Write with Response” checkbox.
9. Select “Build TLV” and then select the “Transition Start (Presets) option, and in the “Linear or Linear Derived” menu, choose “Linear”.
10. Leave the “Notify Value Change Threshold”, “Value Update Time Interval”, and “Notify Time Interval Threshold” fields blank.
11. Set the “End Behavior” to “Loop”.
12. Set the “Start Condition” to “None”.
13. Set the “Transition Preset Style” to “Inverted U”.
14. Set the “Target Completion Duration Per Point” to “60000”.
15. Set the “Number of Transition Points” to “11”.
16. Select “Build TLV”, write the value, and then verify that the accessory accepts the write request.
17. In the Events view of the Trace, verify that a notification for the “Characteristic Value Active Transition Count” characteristic is received, and contains a value of “1”.
18. Verify that the first notification for the Brightness’s updated value is received.
19. Verify that the “Brightness” continues to change every 60 seconds.
20. In the sidebar of the Controllers window, select the “Characteristic Value Transition Control” characteristic.
21. Select the “Build TLV” button, and then select the “Transition Start (Manual)” option.
22. Set the target “HAP Instance ID” to the “Brightness” characteristic Instance ID.
23. Select the “Add Transition” button, select “Build TLV”, and then write the value to abort the current transition.
24. In the Events view of the Trace, verify that the write response contains the “Transition State” TLV item, with a 0 length value.
25. Verify that a notification for the “Characteristic Value Active Transition Count” characteristic is received, and contains a value of “0”.
26. Verify that the “Brightness” immediately stops transitioning.
27. In the sidebar of the Controllers window, select the “Brightness” characteristic.
28. Select “Read” to read the current value. Notate the value.
29. In the Events view of the Trace, verify notifications for “Brightness” are no longer received.
30. Wait 1 minute.
31. Select “Read” to read the current value.
32. Verify that the “Brightness” value is the same as it was in step 31.

TCLS033 **Verify the accessory successfully ends an active "Linear Derived" transition when the accessory receives a write request to the characteristic undergoing the transition.**

Applies to Light Bulb accessories that support Light Shift. Perform this test case with HAT using the steps below.

1. Pair and discover accessory.
2. For HAP over BLE accessories, enable the "Pair Resume Keep Alive" checkbox with a 27-second interval.
3. In the sidebar of the Controllers window, select the "Color Temperature" characteristic.
4. Select the "Enable" button to subscribe to Event Notifications.
5. Select the "Characteristic Value Active Transition Count" characteristic.
6. Select the "Enable" button to subscribe to Event Notifications.
7. In the sidebar of the Controllers window, select the "Characteristic Value Transition Control" characteristic.
8. Select "Build TLV" and then select the "Transition Start (Presets) option, and in the "Linear or Linear Derived" menu, choose "Linear Derived".
9. Leave the "Notify Value Change Threshold", "Value Update Time Interval", and "Notify Time Interval Threshold" fields blank.
10. Set the "End Behavior" to "Loop".
11. Set the "Upper Bound" to "100" and the "Lower Bound" to "0".
12. Set the "Target Completion Duration Per Point" to "60000".
13. Set the "Number of Transition Points" to "11".
14. Select "Build TLV", write the value, and then verify that the accessory accepts the write request.
15. In the Events view of the Trace, verify that a notification for the "Characteristic Value Active Transition Count" characteristic is received, and contains a value of "1".
16. Verify that the first notification for the Color Temperature's updated value is received.
17. In the sidebar of the Controllers window, select the "Brightness" characteristic.
18. Perform a valid write operation to change the "Brightness" value.
19. In the Events view of the Trace, verify that a notification for the Color Temperature's updated value is received.
20. In the sidebar of the Controllers window, select the "Brightness" characteristic.
21. Perform a valid write operation to change the "Brightness" value again.
22. In the Events view of the Trace, verify that a notification for the Color Temperature's updated value is received.
23. In the sidebar of the Controllers window, select the "Color Temperature" characteristic.

24. Perform a valid write operation to set a new value.
25. Verify that a notification for the "Characteristic Value Active Transition Count" characteristic is received, and contains a value of "0".
26. Verify no further notifications for "Color Temperature" are received.
27. In the sidebar of the Controllers window, select the "Brightness" characteristic.
28. Perform a valid write operation to change the "Brightness" value.
29. In the Events view of the Trace, verify no notification for "Color Temperature" is received.

TCLS034 **Verify the accessory successfully ends an active "Linear" transition when the accessory receives a write request to the characteristic undergoing the transition.**

Applies to Light Bulb accessories that support Light Shift. Perform this test case with HAT using the steps below.

1. Pair and discover accessory.
2. For HAP over BLE accessories, enable the "Pair Resume Keep Alive" checkbox with a 27-second interval.
3. In the sidebar of the Controllers window, select the "Brightness" characteristic.
4. Select the "Enable" button to subscribe to Event Notifications.
5. Select the "Characteristic Value Active Transition Count" characteristic.
6. Select the "Enable" button to subscribe to Event Notifications.
7. In the sidebar of the Controllers window, select the "Characteristic Value Transition Control" characteristic.
8. Select "Build TLV" and then select the "Transition Start (Presets) option, and in the "Linear or Linear Derived" menu, choose "Linear".
9. Leave the "Notify Value Change Threshold", "Value Update Time Interval", and "Notify Time Interval Threshold" fields blank.
10. Set the "End Behavior" to "Loop".
11. Set the "Start Condition" to "None".
12. Set the "Transition Preset Style" to "Inverted U".
13. Set the "Target Completion Duration Per Point" to "60000".
14. Set the "Number of Transition Points" to "11".
15. Select "Build TLV", write the value, and then verify that the accessory accepts the write request.
16. In the Events view of the Trace, verify that a notification for the "Characteristic Value Active Transition Count" characteristic is received, and contains a value of "1".
17. Verify that the first notification for the Brightness's updated value is received.

18. In the sidebar of the Controllers window, select the "Brightness" characteristic.
19. Perform a valid write operation to set a new value.
20. Verify that the "Brightness" immediately stops transitioning.
21. Verify that a notification for the "Characteristic Value Active Transition Count" characteristic is received, and contains a value of "0".
22. In the Events view of the Trace, verify notifications for "Brightness" are no longer received.

TCLS035 **For characteristics that support Linear transitions, verify the accessory updates the characteristic value only at the increments set in the "Value Update Time Interval" TLV item.**

Applies to Light Bulb accessories that support Light Shift. Perform this test case with HAT using the steps below.

1. Pair and discover accessory.
2. For HAP over BLE accessories, enable the "Pair Resume Keep Alive" checkbox with a 27-second interval.
3. In the sidebar of the Controllers window, under the "Light Bulb" service, select the "Brightness" characteristic.
4. Write a value of "100", then select "Enable" to enable Event Notifications.
5. In the sidebar of the Controllers window, select the "Characteristic Value Transition Control" characteristic.
6. Select the "Build TLV" button, and then select the "Transition Start (Presets)" option.
7. In the "Linear or Linear Derived" menu, choose "Linear".
8. Set the "Value Update Time Interval" to "120000".
9. Leave the "Notify Value Change Threshold" and "Notify Time Interval Threshold" fields blank.
10. In the "End Behavior" menu, choose "Loop".
11. In the "Start Condition" menu, choose "None".
12. In the "Transition Preset Style" menu, choose "Inverted U".
13. Set the "Target Completion Duration Per Point" to "300000".
14. Set the "Total Number of Transition Points" to "11".
15. Select "Build TLV", then select "Write" to write the TLV value.
16. In the Events view of Trace, verify that the accessory accepts the write request and an initial Event Notification for the "Brightness" characteristic is received.
17. Wait at least 6 minutes.
18. Using the Events view of the Trace, verify the accessory sends notifications only every 120 seconds (2 minutes).

19. In the sidebar of the Controllers window, select the “Brightness” characteristic.
20. Write a value of “0” to abort the current transition.

TCLS036 **Verify when a “Linear Derived” transition is active for the “Color Temperature” characteristic using the “Brightness” characteristic as the source value, that the “Color Temperature” value updates according to the set “Brightness” value.**

Applies to Light Bulb accessories that support Light Shift. Perform this test case with HAT using the steps below.

1. Pair and discover accessory.
2. For HAP over BLE accessories, enable the “Pair Resume Keep Alive” checkbox with a 27-second interval.
3. In the sidebar of the Controllers window, select the “Brightness” characteristic under the “Lightbulb” service.
4. Write a value of “100”.
5. Select “Enable” to enable Event Notifications for the “Brightness” characteristic.
6. In the sidebar of the Controllers window, select the “Color Temperature” characteristic.
7. Select the “Enable” button to subscribe to Event Notifications.
8. Write a value of “300”.
9. In the sidebar of the Controllers window, select the “Characteristic Value Transition Control” characteristic.
10. Select the “Build TLV” button, and then select the “Transition Start (Presets)” option.
11. In the “Linear or Linear Derived” menu, choose “Linear Derived”.
12. Set the “Upper Bound” to “100” and the “Lower Bound” to “0”.
13. Set the “End Behavior” to “Loop”.
14. Set the “Target Completion Duration Per Point” to “1800000”.
15. Set the “Total Number Of Transition Points” to “11”.
16. Set the “Notify Value Change Threshold” to “1”.
17. Leave the “Notify Time Interval Threshold” field blank.
18. Leave the “Value Update Time Interval” field blank.
19. Select “Build TLV”, write the value, and then verify that the accessory accepts the write request.
20. In the Events view of the Trace, verify that a notification for the initial “Color Temperature” value is received.
21. In the sidebar of the Controllers window, select the “Brightness” characteristic.
22. Write a value of “20”.

23. In the Events view of the Trace, verify that a notification for the new "Color Temperature" value is received.
24. In the sidebar of the Controllers window, select the "Brightness" characteristic.
25. Write a value of "50".
26. In the Events view of the Trace, verify that a notification for the new "Color Temperature" value is received.
27. In the sidebar of the Controllers window, select the "Brightness" characteristic.
28. Write a value of "70".
29. In the Events view of the Trace, verify that a notification for the new "Color Temperature" value is received.
30. In the sidebar of the Controllers window, select the "Brightness" characteristic.
31. Write a value of "100".
32. In the Events view of the Trace, verify that a notification for the new "Color Temperature" value is received.
33. In the sidebar of the Controllers window, select the "Color Temperature' characteristic.
34. Write a value of "200" to abort the current transition.

TCLS037 **For characteristics that support Linear transitions, accessory must assume a "Value Update Time Interval" of 1 minute when the controller does not specify a "Value Update Time Interval" in the "Transition Start" operation.**

Applies to Light Bulb accessories that support Light Shift. Perform this test case with HAT using the steps below.

1. Pair and discover accessory.
2. For HAP over BLE accessories, enable the "Pair Resume Keep Alive" checkbox with a 27-second interval.
3. In the sidebar of the Controllers window, under the "Light Bulb" service, select the "Brightness" characteristic.
4. Write a value of "100", then select "Enable" to enable Event Notifications.
5. In the sidebar of the Controllers window, select the "Characteristic Value Transition Control" characteristic.
6. Select the "Build TLV" button, and then select the "Transition Start (Presets)" option.
7. In the "Linear or Linear Derived" menu, choose "Linear".
8. Leave the "Value Update Time Interval", "Notify Time Interval Threshold", and "Notify Value Change Threshold" fields blank.
9. In the "End Behavior" menu, choose "Loop".

10. In the “Start Condition” menu, choose “None”.
11. In the “Transition Preset Style” menu, choose “Inverted U”.
12. Set the “Target Completion Duration Per Point” to “300000”.
13. Set the “Total Number of Transition Points” to “11”.
14. Select “Build TLV”, then select “Write” to write the TLV value.
15. In the Events view of Trace, verify that the accessory accepts the write request, an initial Event Notification for the “Brightness” characteristic is received, with continuous notifications arriving in 60 second intervals.
16. Wait at least 5 minutes.
17. In the Events view of Trace, verify that the notifications for the updated “Brightness” values are continuing to arrive in 60 second intervals.
18. In the sidebar of the Controllers window, select the “Brightness” characteristic.
19. Write a value of “0” to abort the current transition.

TCLS038 **For services with characteristics that support Linear and Linear Derived transitions, verify the accessory only ends the Linear Derived transition of the characteristic specified in the HAP Instance ID TLV item of the “Transition Start” operation, without any transitions specified, when multiple transitions are currently active.**

Applies to Light Bulb accessories that support Light Shift. Perform this test case with HAT using the steps below.

1. Pair and discover accessory.
2. For HAP over BLE accessories, enable the “Pair Resume Keep Alive” checkbox with a 27-second interval.
3. In the sidebar of the Controllers window, select the “Color Temperature” characteristic.
4. Select the “Enable” button to subscribe to Event Notifications.
5. Select the “Color Temperature” characteristic.
6. Write a value of “300”.
7. Select the “Enable” button to subscribe to Event Notifications.
8. Select the “Brightness” characteristic.
9. Write a value of “100”.
10. Select the “Enable” button to subscribe to Event Notifications.
11. Select the “Characteristic Value Active Transition Count” characteristic.
12. Select the “Enable” button to subscribe to Event Notifications.
13. In the sidebar of the Controllers window, select the “Characteristic Value Transition Control” characteristic.

14. In the Options panel, enable the “Write with Response” checkbox.
15. Select “Build TLV” and then select the “Transition Start (Presets) option, and in the “Linear or Linear Derived” menu, choose “Linear and Linear Derived”.
16. Set the “End Behavior” to “Loop”.
17. Set the “Start Behavior” to “None”.
18. Set the “Transition Preset Style” to “Inverted U”.
19. Set the “Upper Bound” to “100” and the “Lower Bound” to “0”.
20. Set the “Target Completion Duration Per Point” for Linear to “60000”.
21. Set the “Target Completion Duration Per Point” for Linear Derived to “60000”.
22. Set the “Number of Transition Points” for Linear to “10”
23. Set the “Number of Transition Points” for Linear Derived to “10”
24. Select “Build TLV”, write the value, and then verify that the accessory accepts the write request.
25. In the Events view of the Trace, verify that the write response contains the “Transition State” for both the “Brightness” and “Color Temperature” transitions.
26. In the Events view of the Trace, verify that a notification for the “Characteristic Value Active Transition Count” characteristic is received, and contains a value of “2”.
27. In the sidebar of the Controllers window, select the “Characteristic Value Transition Control” characteristic.
28. Select the “Build TLV” button, and then select the “Transition Start (Manual)” option.
29. Set the target “HAP Instance ID” to the “Color Temperature” characteristic Instance ID.
30. Select the “Add Transition” button, select “Build TLV”, and then write the value to abort the current “Color Temperature” transition.
31. In the Events view of the Trace, verify that the write response contains the “Transition State” for only the “Brightness” transition.
32. Verify that a notification for the “Characteristic Value Active Transition Count” characteristic is received, and contains a value of “1”.
33. Verify that the “Color Temperature” immediately stops transitioning.
34. Verify “Brightness” continues to transition.
35. In the sidebar of the Controllers window, select the “Characteristic Value Transition Control” characteristic.
36. Select the “Build TLV” button, and then select the “Transition Start (Manual)” option.
37. Set the target “HAP Instance ID” to the “Brightness” characteristic Instance ID.
38. Select the “Add Transition” button, select “Build TLV”, and then write the value to abort the current “Brightness” transition.

39. In the Events view of the Trace, verify that the write response contains the “Transition State” TLV item, with a 0 length value.
40. Verify that a notification for the “Characteristic Value Active Transition Count” characteristic is received, and contains a value of “0”.
41. Verify that the “Brightness” immediately stops transitioning.

TCLS039 **For services with characteristics that support both Linear and Linear Derived transitions, verify the accessory only ends the Linear transition of the characteristic specified in the HAP Instance ID TLV item of the “Transition Start” operation, without any transitions specified, when multiple transitions are currently active.**

Applies to Light Bulb accessories that support Light Shift. Perform this test case with HAT using the steps below.

1. Pair and discover accessory.
2. For HAP over BLE accessories, enable the “Pair Resume Keep Alive” checkbox with a 27-second interval.
3. In the sidebar of the Controllers window, select the “Color Temperature” characteristic.
4. Select the “Enable” button to subscribe to Event Notifications.
5. Select the “Color Temperature” characteristic.
6. Write a value of “300”.
7. Select the “Enable” button to subscribe to Event Notifications.
8. Select the “Brightness” characteristic.
9. Write a value of “100”.
10. Select the “Enable” button to subscribe to Event Notifications.
11. Select the “Characteristic Value Active Transition Count” characteristic.
12. Select the “Enable” button to subscribe to Event Notifications.
13. In the sidebar of the Controllers window, select the “Characteristic Value Transition Control” characteristic.
14. In the Options panel, enable the “Write with Response” checkbox.
15. Select “Build TLV” and then select the “Transition Start (Presets) option, and in the “Linear or Linear Derived” menu, choose “Linear and Linear Derived”.
16. Set the “End Behavior” to “Loop”.
17. Set the “Start Behavior” to “None”.
18. Set the “Transition Preset Style” to “Inverted U”.
19. Set the “Upper Bound” to “100” and the “Lower Bound” to “0”.

20. Set the “Target Completion Duration Per Point” for Linear to “60000”.
21. Set the “Target Completion Duration Per Point” for Linear Derived to “60000”.
22. Set the “Number of Transition Points” for Linear to “10”
23. Set the “Number of Transition Points” for Linear Derived to “10”
24. Select “Build TLV”, write the value, and then verify that the accessory accepts the write request.
25. In the Events view of the Trace, verify that the write response contains the “Transition State” for both the “Brightness” and “Color Temperature” transitions.
26. In the Events view of the Trace, verify that a notification for the “Characteristic Value Active Transition Count” characteristic is received, and contains a value of “2”.
27. In the sidebar of the Controllers window, select the “Characteristic Value Transition Control” characteristic.
28. Select the “Build TLV” button, and then select the “Transition Start (Manual)” option.
29. Set the target “HAP Instance ID” to the “Brightness” characteristic Instance ID.
30. Select the “Add Transition” button, select “Build TLV”, and then write the value to abort the current “Brightness” transition.
31. In the Events view of the Trace, verify that the write response contains the “Transition State” for only the “Color Temperature” transition.
32. Verify that a notification for the “Characteristic Value Active Transition Count” characteristic is received, and contains a value of “1”.
33. Verify that the “Brightness” immediately stops transitioning.
34. Verify “Color Temperature” continues to transition.
35. In the sidebar of the Controllers window, select the “Characteristic Value Transition Control” characteristic.
36. Select the “Build TLV” button, and then select the “Transition Start (Manual)” option.
37. Set the target “HAP Instance ID” to the “Color Temperature” characteristic Instance ID.
38. Select the “Add Transition” button, select “Build TLV”, and then write the value to abort the current “Color Temperature” transition.
39. In the Events view of the Trace, verify that the write response contains the “Transition State” TLV item, with a 0 length value.
40. Verify that a notification for the “Characteristic Value Active Transition Count” characteristic is received, and contains a value of “0”.
41. Verify that the “Color Temperature” immediately stops transitioning.

TCLS040 **For characteristics that support Linear Derived transitions, the accessory must be able to support transition points with the first transition point’s “Target Completion Duration” set to “0” ms, and subsequent transition points set to at least “100” ms.**

Applies to Light Bulb accessories that support Light Shift. Perform this test case with HAT using the steps below.

1. Pair and discover accessory.
2. In the sidebar of the Controllers window, under the "Light Bulb" service, select the "Color Temperature" characteristic.
3. Write a value of "300", then select "Enable" to enable Event Notifications.
4. In the sidebar of the Controllers window, select the "Characteristic Value Transition Control" characteristic.
5. Select the "Build TLV" button, and then select the "Transition Start (Presets)" option.
6. In the "Linear or Linear Derived" menu, choose "Linear Derived".
7. Set the "Value Update Time Interval" to "10000".
8. Leave the "Notify Value Change Threshold" and "Notify Time Interval Threshold" fields blank.
9. In the "End Behavior" menu, choose "Loop".
10. Set the "Upper Bound" to "100" and the "Lower Bound" to "0".
11. Set the "Target Completion Duration Per Point" to "100".
12. Set the "Total Number of Transition Points" to "11".
13. Select "Build TLV", then select "Write" to write the TLV value.
14. In the Events view of Trace, verify that the accessory accepts the write request, an initial Event Notification for the "Color Temperature" characteristic is received, with continuous notifications arriving in 10 second intervals.
15. In the sidebar of the Controllers window, select the "Color Temperature" characteristic.
16. Select "Read" five times in quick succession.
17. In the Events view of Trace, verify that the values contained in each Read Response is different, indicating the "Color Temperature" is transitioning.
18. In the sidebar of the Controllers window, select the "Color Temperature" characteristic.
19. Write a value of "300" to abort the current transition.

TCLS041 **For characteristics that support Linear Derived transitions, accessory must assume a "Value Update Time Interval" of 1 minute when the controller does not specify a "Value Update Time Interval" in the "Transition Start" operation.**

Applies to Light Bulb accessories that support Light Shift. Perform this test case with HAT using the steps below.

1. Pair and discover accessory.
2. For HAP over BLE accessories, enable the "Pair Resume Keep Alive" checkbox with a 27-second interval.

3. In the sidebar of the Controllers window, under the "Light Bulb" service, select the "Color Temperature" characteristic.
4. Write a value of "300", then select "Enable" to enable Event Notifications.
5. In the sidebar of the Controllers window, select the "Characteristic Value Transition Control" characteristic.
6. Select the "Build TLV" button, and then select the "Transition Start (Presets)" option.
7. In the "Linear or Linear Derived" menu, choose "Linear Derived".
8. Leave the "Value Update Time Interval", "Notify Time Interval Threshold", and "Notify Value Change Threshold" fields blank.
9. In the "End Behavior" menu, choose "Loop".
10. Set the "Upper Bound" to "100" and the "Lower Bound" to "0".
11. Set the "Target Completion Duration Per Point" to "60000".
12. Set the "Total Number of Transition Points" to "10".
13. Select "Build TLV", then select "Write" to write the TLV value.
14. In the Events view of Trace, verify that the accessory accepts the write request, an initial Event Notification for the "Color Temperature" characteristic is received, with notifications arriving in multiple of 60 seconds (i.e 60 or 120 or 180).
15. Wait at least 5 minutes.
16. In the Events view of Trace, verify that the notifications for the Color Temperature's updated values are received in multiple of 60 seconds (i.e 60 or 120 or 180).
17. In the sidebar of the Controllers window, select the "Color Temperature" characteristic.
18. Write a value of "300" to abort the current transition.

TCLS042 **For characteristics that support Linear Derived transitions, verify the accessory updates the characteristic value only at the increments set in the "Value Update Time Interval" TLV item.**

Applies to Light Bulb accessories that support Light Shift. Perform this test case with HAT using the steps below.

1. Pair and discover accessory.
2. For HAP over BLE accessories, enable the "Pair Resume Keep Alive" checkbox with a 27-second interval.
3. In the sidebar of the Controllers window, under the "Light Bulb" service, select the "Color Temperature" characteristic.
4. Write a value of "300", then select "Enable" to enable Event Notifications.
5. In the sidebar of the Controllers window, select the "Characteristic Value Transition Control" characteristic.

6. Select the “Build TLV” button, and then select the “Transition Start (Presets)” option.
7. In the “Linear or Linear Derived” menu, choose “Linear Derived”.
8. Set the “Value Update Time Interval” to “120000”.
9. Leave the “Notify Value Change Threshold” and “Notify Time Interval Threshold” fields blank.
10. In the “End Behavior” menu, choose “Loop”.
11. Set the “Upper Bound” to “100” and the “Lower Bound” to “0”.
12. Set the “Target Completion Duration Per Point” to “60000”.
13. Set the “Total Number of Transition Points” to “11”.
14. Select “Build TLV”, then select “Write” to write the TLV value.
15. In the Events view of Trace, verify that the accessory accepts the write request and an initial Event Notification for the “Color Temperature” characteristic is received.
16. Wait at least 6 minutes.
17. In the Events view of Trace, verify notifications for the Color Temperature’s updated values arrived in 120 second intervals.
18. In the sidebar of the Controllers window, select the “Color Temperature” characteristic.
19. Write a value of “300” to abort the current transition.

TCLS043 **For characteristics that support Linear Derived transitions, verify the accessory sends notifications to the controller only at the increments set in the “Notify Value Change Threshold” TLV item unless the “Brightness” characteristic is changed, in which case the “Notify Value Change Threshold” is ignored.**

Applies to Light Bulb accessories that support Light Shift. Perform this test case with HAT using the steps below.

1. Pair and discover accessory.
2. For HAP over BLE accessories, enable the “Pair Resume Keep Alive” checkbox with a 27-second interval.
3. In the sidebar of the Controllers window, select the “Color Temperature” characteristic.
4. Write a value of “300”, then select “Enable” to enable Event Notifications.
5. In the sidebar of the Controllers window, select the “Characteristic Value Transition Control” characteristic.
6. Select the “Build TLV” button, and then select the “Transition Start (Presets)” option.
7. In the “Linear or Linear Derived” menu, choose “Linear Derived”.
8. Leave the “Value Update Time Interval” and “Notify Time Interval Threshold” fields blank.
9. Set the “Notify Value Change Threshold” to “50”.
10. In the “End Behavior” menu, choose “Loop”.

11. Set the “Upper Bound” to “100” and the “Lower Bound” to “0”.
12. Set the “Target Completion Duration Per Point” to “60000”.
13. Set the “Total Number of Transition Points” to “11”.
14. Select “Build TLV”, then select “Write” to write the TLV value.
15. In the Events view of Trace, verify that the accessory accepts the write request and an initial Event Notification for the “Color Temperature” characteristic is received.
16. Note the value of the Color Temperature.
17. Wait until another Event Notification for the “Color Temperature” characteristic is received. Check the “Color Temperature” value, and verify it has changed by more than 50.
18. In the sidebar of the Controllers window, select the “Brightness” characteristic.
19. Write a value of “1”.
20. Write a value of “5”.
21. Write a value of “10”.
22. Write a value of “1”.
23. In the Events view of Trace, verify notifications for the “Color Temperature” occur with each “Brightness” value change, including notifications where the change is less than 50.
24. In the sidebar of the Controllers window, select the “Color Temperature” characteristic.
25. Write a value of “300” to abort the current transition.

TCLS044 **For characteristics that support Linear Derived transitions, verify the accessory sends Broadcast Event Notifications to the controller only at the increments set in the “Notify Value Change Threshold” TLV item.**

Applies to Light Bulb accessories that support Light Shift. Applies to accessories that use HAP over BLE. Perform this test case with HAT using the steps below.

1. Pair and discover accessory.
2. Enable the “Pair Resume Keep Alive” checkbox with a 27-second interval.
3. Under the “HAP Protocol Information” service, select the “Service Signature” characteristic, and then in the “Protocol Configuration” panel, enable the “Set Advertising Identifier”, “Get all params”, and “Generate Broadcast Keys” checkboxes.
4. Under “Desired Advertising Identifier”, enter “AABBCCDDEEFF”, and then select “Send”.
5. In the sidebar of the Controllers window, select the “Brightness” characteristic.
6. Write a value of “100”.
7. In the sidebar of the Controllers window, select the “Color Temperature” characteristic.
8. Write a value of “300”, then select “Enable” to enable Event Notifications.

9. In the “Characteristic Configuration” panel, enter “1” for broadcast interval and select “Enable Broadcast Notifications”.
10. In the sidebar of the Controllers window, select the “Characteristic Value Transition Control” characteristic.
11. Select the “Build TLV” button, and then select the “Transition Start (Presets)” option.
12. In the “Linear or Linear Derived” menu, choose “Linear Derived”.
13. Leave the “Value Update Time Interval” and “Notify Time Interval Threshold” fields blank.
14. Set the “Notify Value Change Threshold” to “50”.
15. In the “End Behavior” menu, choose “Loop”.
16. Set the “Upper Bound” to “100” and the “Lower Bound” to “0”.
17. Set the “Target Completion Duration Per Point” to “15000”.
18. Set the “Total Number of Transition Points” to “11”.
19. Select “Build TLV”, then select “Write” to write the TLV value.
20. After the write completes, select the “Disconnect” button on the main accessory server view.
21. In the BLE Discovery view of Trace, verify that the accessory accepts the write request and an initial Event Notification for the “Color Temperature” characteristic is received.
22. Wait at least 5 minutes.
23. In the Events view of Trace, verify Encrypted Broadcast Notifications for the “Color Temperature” only occurred when the value changed by at least 50.
24. In the sidebar of the Controllers window, select the “Color Temperature” characteristic.
25. Write a value of “300” to abort the current transition.

TCLS045 **For characteristics that support Linear Derived transitions, verify the accessory sends notifications to the controller only at the increments set in the “Notify Time Interval Threshold” TLV item.**

Applies to Light Bulb accessories that support Light Shift. Perform this test case with HAT using the steps below.

1. Pair and discover accessory.
2. For HAP over BLE accessories, enable the “Pair Resume Keep Alive” checkbox with a 27-second interval.
3. In the sidebar of the Controllers window, select the “Color Temperature” characteristic.
4. Write a value of “300”, then select “Enable” to enable Event Notifications.
5. In the sidebar of the Controllers window, select the “Characteristic Value Transition Control” characteristic.
6. Select the “Build TLV” button, and then select the “Transition Start (Presets)” option.

7. In the “Linear or Linear Derived” menu, choose “Linear Derived”.
8. Leave the “Value Update Time Interval” and “Notify Value Change Threshold” fields blank.
9. Set the “Notify Time Interval Threshold” to “180000”.
10. In the “End Behavior” menu, choose “Loop”.
11. Set the “Upper Bound” to “100” and the “Lower Bound” to “0”.
12. Set the “Target Completion Duration Per Point” to “60000”.
13. Set the “Total Number of Transition Points” to “11”.
14. Select “Build TLV”, then select “Write” to write the TLV value.
15. In the Events view of Trace, verify that the accessory accepts the write request and an initial Event Notification for the “Color Temperature” characteristic is received.
16. Wait at least 10 minutes.
17. In the Events view of Trace, verify notifications for the Color Temperature’s updated values were delayed by at least 180 seconds.
18. In the sidebar of the Controllers window, select the “Color Temperature” characteristic.
19. Write a value of “300” to abort the current transition.

TCLS046 **For characteristics that support Linear Derived transitions, verify the accessory sends Broadcast Event Notifications to the controller only at the increments set in the “Notify Time Interval Threshold” TLV item.**

Applies to Light Bulb accessories that support Light Shift. Applies to accessories that use HAP over BLE. Perform this test case with HAT using the steps below.

1. Pair and discover accessory.
2. Enable the “Pair Resume Keep Alive” checkbox with a 27-second interval.
3. Under the “HAP Protocol Information” service, select the “Service Signature” characteristic, and then in the “Protocol Configuration” panel, enable the “Set Advertising Identifier”, “Get all params”, and “Generate Broadcast Keys” checkboxes.
4. Under “Desired Advertising Identifier”, enter “AABBCCDDEEFF”, and then select “Send”.
5. In the sidebar of the Controllers window, select the “Color Temperature” characteristic.
6. Write a value of “300”, then select “Enable” to enable Event Notifications.
7. In the “Characteristic Configuration” panel, enter “1” for broadcast interval and select “Enable Broadcast Notifications”.
8. In the sidebar of the Controllers window, select the “Characteristic Value Transition Control” characteristic.
9. Select the “Build TLV” button, and then select the “Transition Start (Presets)” option.

10. In the “Linear or Linear Derived” menu, choose “Linear Derived”.
11. Leave the “Value Update Time Interval” and “Notify Value Change Threshold” fields blank.
12. Set the “Notify Time Interval Threshold” to “180000”.
13. In the “End Behavior” menu, choose “Loop”.
14. Set the “Upper Bound” to “100” and the “Lower Bound” to “0”.
15. Set the “Target Completion Duration Per Point” to “60000”.
16. Set the “Total Number of Transition Points” to “11”.
17. Select “Build TLV”, then select “Write” to write the TLV value.
18. After the write completes, select the “Disconnect” button on the main accessory server view.
19. In the BLE Discovery view of Trace, verify that the accessory accepts the write request and an initial Event Notification for the “Color Temperature” characteristic is received.
20. Wait at least 10 minutes.
21. In the BLE Discovery view of Trace, verify Encrypted Broadcast Notifications for the Color Temperature’s updated values were delayed by at least 180 seconds.
22. In the sidebar of the Controllers window, select the “Color Temperature” characteristic.
23. Write a value of “300” to abort the current transition.

TCLS047 **Verify accessories with “Color Temperature” are accurately displaying the intended “Color Temperature” within a degree of 5 percent.**

Applies to Light Bulb accessories that support Light Shift. Perform this test case with HAT using the steps below.

1. Pair and discover accessory.
2. Place the accessory on a dark or black surface that measures less than 2 lux when recorded with a light meter. Ensure no other light sources are in the area that could interfere with measurements.
3. For HAP over BLE accessories, enable the “Pair Resume Keep Alive” checkbox with a 27-second interval.
4. In the sidebar of the Controllers window, under the “Light Bulb” service, select the “Color Temperature” characteristic.
5. Write a value of “200”.
6. In the sidebar of the Controllers window, select the “Brightness” characteristic.
7. Write a value of “100”.
8. Using a light meter, at a distance of about 24 inches, measure the “Color Temperature” output of the light emitted by the accessory.
9. The value should measure 5000K on the light meter.

10. Under the “Color Temperature” characteristic write a value of “227”.
11. Using a light meter, at a distance of about 24 inches, measure the updated “Color Temperature” output of the light emitted by the accessory.
12. The value should measure 4400K on the light meter.
13. Under the “Color Temperature” characteristic write a value of “256”.
14. Using a light meter, at a distance of about 24 inches, measure the updated “Color Temperature” output of the light emitted by the accessory.
15. The value should measure 3910K on the light meter.
16. Under the “Color Temperature” characteristic write a value of “304”.
17. Using a light meter, at a distance of about 24 inches, measure the updated “Color Temperature” output of the light emitted by the accessory.
18. The value should measure 3286K on the light meter.
19. Under the “Color Temperature” characteristic write a value of “310”.
20. Using a light meter, at a distance of about 24 inches, measure the updated “Color Temperature” output of the light emitted by the accessory.
21. The value should measure 3231K on the light meter.
22. Under the “Color Temperature” characteristic write a value of “331”.
23. Using a light meter, at a distance of about 24 inches, measure the updated “Color Temperature” output of the light emitted by the accessory.
24. The value should measure 3018K on the light meter.

TCLS048 **Verify that the accessory rejects “Transition Start” operation with a scale and offset that results in the target value falling outside of the supported range.**

Applies to Light Bulb accessories that support Light Shift. Perform this test case with HAT using the steps below.

1. Pair and discover accessory.
2. For HAP over BLE accessories, enable the “Pair Resume Keep Alive” checkbox with a 27-second interval.
3. In the sidebar of the Controllers window, under the “Light Bulb” service, select “Characteristic Value Transition Control”.
4. Select the “Build TLV” button, and then select the “Transition Start (Manual)” option. Set the target “HAP Instance ID” to a characteristic that supports transitions, set the “End Behavior” to “Loop”, select “Linear Derived” in the “Linear or Linear Derived” menu, and select “+” to add the first transition point.
5. Set the Scale to “0.51”, Offset to “1.01”, Target Completion Duration to “0”, and “Start Delay Duration” to “12000”.

6. Create a second transition point with the Scale set to “4.8”, Offset to “1.01”, and Target Completion Duration set to “120000”. Leave the “Start Delay Duration” field blank.
7. Create a third transition point with the Scale set to “9.8”, Offset to “1.01”, Target Completion Duration set to “120000”, and leave the “Start Delay Duration” field blank. (If the accessory supports a “Color Temperature” greater than this value, change the scale and offset to values that fall outside the capabilities of the accessory, e.g., a scale of 15.2 with an offset of 1.01 at 100 percent “Brightness” produces a “Color Temperature” of over 1500 mirek.)
8. Set the “Source Instance ID” to the “Brightness” characteristic’s Instance ID, and set the Lower Bound to “0a” (10) and the Upper Bound to “5a” (90).
9. Select “Add Transition”, select “Build TLV”, and then write the value.
10. In the Events view of Trace, verify that the accessory rejects the write request.
11. For HAP over Wi-Fi or Ethernet, in Events view of the Trace, verify that the accessory responds with HAP status code -70410.
12. For HAP over BLE, in the HAP Transactions view, verify that the accessory responds with 0x06 (Invalid Request).

TCLS049 **An accessory supporting Light Shift must stop all transitions when the accessory is reset to factory settings.**

Applies to Light Bulb accessories that support Light Shift. Perform this test case with HAT using the steps below.

1. Pair and discover accessory.
2. Select the “Characteristic Value Active Transition Count” characteristic.
3. Select the “Enable” button to subscribe to Event Notifications.
4. In the sidebar of the Controllers window, select the “Characteristic Value Transition Control” characteristic.
5. In the Options panel, enable the “Write with Response” checkbox.
6. Select “Build TLV” and then select the “Transition Start (Presets)” option, and in the “Linear or Linear Derived” menu, choose “Linear and Linear Derived”.
7. Set the “End Behavior” to “Loop”.
8. Set the “Start Behavior” to “None”.
9. Set the “Transition Preset Style” to “Inverted U”.
10. Set the “Upper Bound” to “100” and the “Lower Bound” to “0”.
11. Set the “Target Completion Duration Per Point” for Linear to “60000”.
12. Set the “Target Completion Duration Per Point” for Linear Derived to “60000”.
13. Set the “Number of Transition Points” for Linear to “10”

14. Set the “Number of Transition Points” for Linear Derived to “10”
15. Select “Build TLV”, write the value, and then verify that the accessory accepts the write request.
16. In the Events view of the Trace, verify that the write response contains the “Transition State” for both the “Brightness” and “Color Temperature” transitions.
17. In the Events view of the Trace, verify that a notification for the “Characteristic Value Active Transition Count” characteristic is received, and contains a value of “2”.
18. Restore the accessory to factory settings.
19. Pair and discover accessory.
20. Select the “Characteristic Value Active Transition Count” characteristic.
21. Read the value and verify it is set to “0”.

1.24 Wi-Fi Reconfiguration

TCWR001: Verify accessory can be reconfigured onto a new network using the "Simple Update" procedure.

TCWR002: If the accessory can support Wi-Fi configuration outside of the WAC proceedure (E.g. through the accessory's app), verify accessory can be reconfigured to a new network using the "Simple Update" procedure.

TCWR003: Verify all Wi-Fi accessories expose a single instance of the "Wi-Fi Transport" service, and includes the required characteristic(s).

TCWR004: Verify that the accessory handles a valid PPSK format. (E.g. If the length is 8 to 63 bytes, each being 32 to 126 decimal, then it's a plain-text password. Otherwise, it's expected to be a prehashed, 256-bit preshared key.)

TCWR005: Verify the "Current Transport" characteristic returns value "1" (True) when read over the Wi-Fi network.

TCWR006: Verify the "Current Transport" characteristic returns value "0" (False) when read over an Ethernet connection.

TCWR007: Verify accessory uses valid values for the "Wi-Fi Capabilities" characteristic.

TCWR008: Verify the accessory rejects reads and writes to the "Wi-Fi Configuration Control" characteristic from non-admin controllers.

TCWR009: Verify accessory removes its Wi-Fi configuration when recieving a "Simple Update" operation with an empty "Station Configuration" TLV, and enters WAC mode.

TCWR010: Verify accessory already configured for Wi-Fi can be reconfigured onto a new network using the "Fail-Safe Update" procedure.

TCWR011: Verify that when a "Fail-Safe" procedure fails due to an invalid SSID or PSK, on an accessory that is already configured for Wi-Fi, that the accessory returns the correct "Update Status" values and remains on the current network.

TCWR012: Verify accessory responds to Update Configuration requests with HAP status code -70403 if a Fail-Safe operation is already in progress.

TCWR013: Verify "Update Status" does not persist across reboots.

TCWR014: Verify accessory rejects "Update Configuration" operations that contain invalid SSID or PSK values with HAP status code -70410.

TCWR015: Verify accessory can be reconfigured onto an open and unsecured network.

TCWR016: If the "Operation Timeout" expires for a Fail-Safe operation, the accessory must clear the "Update Pending" flags and set the relevant error flags in the "Update Status".

TCWR001 **Verify accessory can be reconfigured onto a new network using the "Simple Update" procedure.**

Applies to accessories that implement the Wi-Fi Reconfiguration service.

1. Create two wireless networks that use different SSID's and PSK's (E.g. Network A and Network B).
2. Factory reset accessory.
3. Select the accessory, and in the "Wi-Fi Accessory Configuration" panel, select the "Join access point" button.

4. Once the accessory's first Bonjour advertisement is recieved, select "Start Pairing" and complete Pair-Setup.
5. In the "Wi-Fi Accessory Configuration" panel, enter the Wi-Fi network's SSID and PSK for Network A, and then select the "Send WAC Configuration" button.
6. Connect the Mac to Network A.
7. Once the accessory begins advertising via Bonjour on Network A, select the "Confirm WAC Configuration" button, and then select "Discover".
8. In the IP Discovery view of the trace window, select the most recent accessory advertisement and notate the "Configuration Number" value.
9. In the left sidebar, select the "Wi-Fi Configuration Control" characteristic.
10. Select the "Read" button.
11. In the Events view of the trace, verify the write response contains a "Cookie" value of "0" and "Update Status" value of "0".
12. In the "Prepare and Execute Timed Write" section, select "Build TLV".
13. In the "Wi-Fi Configuration Control" TLV builder, set the following parameters:
14. Set the "Operation" to "Update Configuration (Simple)".
15. Set the "Cookie" to any decimal value between 1 and 65535.
16. Set "SSID" to the SSID for Network B.
17. Set "Security Mode" to "WPA2-PSK".
18. Set "PSK" to the PSK for Network B.
19. Click the "Build TLV" button in the bottom right corner, and then select "Timed Write".
20. In the Events view of the trace, verify the write response contains a "Cookie" value that matches the cookie used in step 15, and "Update Status" value of "0".
21. Verify the accessory disconnects.
22. Connect the Mac running HAT to Network B.
23. In the IP Discovery view of the trace window, verify the accessory begins to advertise on the new network, and that the "Configuration Number" value incremented by 1.
24. Once the accessory begins advertising via Bonjour on Network B, select "Discover", and then verify Pair-Verify and the discover operation completes successfully.
25. Select the "Wi-Fi Configuration Control" characteristic and select the "Read" button.
26. In the Events view of the trace, verify the read response contains contains only the "Cookie" value that matches the cookie used in step 15, and "Update Status" value of "0". Verify no other TLV items are present.
27. Select the "Wi-Fi Configuration Control" characteristic.
28. In the "Prepare and Execute Timed Write" section, select "Build TLV".

29. Set the “Operation” to “Read Configuration”.
30. Select the “Build TLV” button in the bottom right corner, and then select “Timed Write”.
31. In the Events view of the trace, select the write response and select “Details” to show the details.
32. Verify the write response contains the “Cookie” value that matches the cookie used in step 15, “Update Status” value of “0”, a valid “Country Code Configuration” (this TLV item is optional), a “Station Configuration” with “SSID” set to the SSID of Network B, “Security Mode” set to “WPA2-PSK”, and “PSK” with a length of “0”.

TCWR002 **If the accessory can support Wi-Fi configuration outside of the WAC proceedure (E.g. through the accessory’s app), verify accessory can be reconfigured to a new network using the “Simple Update” procedure.**

Applies to accessories that implement the Wi-Fi Reconfiguration service. Applies to accessories that support HomeKit Accessory Protocol specification R16 or later.

1. Create two wireless networks that use different SSID’s and PSK’s (E.g. Network A and Network B).
2. Factory reset accessory.
3. Configure the accessory onto Network A outside of the WAC proceedure (E.g. through the accessory’s app).
4. Pair and discover the accessory.
5. In the left sidebar, select the “Wi-Fi Configuration Control” characteristic.
6. In the “Prepare and Execute Timed Write” section, select “Build TLV”.
7. In the “Wi-Fi Configuration Control” TLV builder, set the following parameters:
8. Set the “Operation” to “Update Configuration (Simple)”.
9. Set the “Cookie” to any decimal value between 1 and 65535.
10. Set “SSID” to the SSID for Network B.
11. Set “Security Mode” to “WPA2-PSK”.
12. Set “PSK” to the PSK for Network B.
13. Click the “Build TLV” button in the bottom right corner, and then select “Timed Write”.
14. Wait for the accesory to disconenct.
15. Connect the Mac to Network B.
16. Once the accessory begins advertising via Bonjour on Network B, select “Discover”, and then verify Pair-Verify and the discover operation completes successfully.

TCWR003 **Verify all Wi-Fi accessories expose a single instance of the “Wi-Fi Transport” service, and includes the required characteristic(s).**

Applies to accessories that implement the Wi-Fi Reconfiguration service. Applies to accessories that support HomeKit Accessory Protocol specification R16 or later.

1. Pair and discover accessory.
2. In the left sidebar of the Controllers window, verify only one "Wi-Fi Transport" service exists.
3. Verify the "Current Transport" characteristic is present, has the "Paired Read" permission, and uses a "bool" format.
4. Verify the "Wi-Fi Capabilities" characteristic is present, has the "Paired Read" permission, and uses a "uint32" format.
5. If the accessory supports reconfiguration of its Wi-Fi credentials, verify service includes the "Wi-Fi Configuration Control" characteristic.
6. If the the service includes the "Wi-Fi Configuration Control" characteristic, verify the characteristic has the permissions "Paired Read", "Paired Write", "Write Response", "Timed Write", and "Notify", and uses a "tlv8" format.

TCWR004 **Verify that the accessory handles a valid PPSK format. (E.g. If the length is 8 to 63 bytes, each being 32 to 126 decimal, then it's a plain-text password. Otherwise, it's expected to be a prehashed, 256-bit preshared key.)**

Applies to accessories that implement the Wi-Fi Reconfiguration service.

1. Create two wireless networks, with Network A using the SSID "networka" and Network B using the SSID "networkb".
2. Configure Network B to use a plain-text PSK of "TestPPSKPassphrase" and configure Network A to use a plain-text PSK of "11111111".
3. Factory reset the accessory.
4. Select the accessory, and in the "Wi-Fi Accessory Configuration" panel, select the "Join access point" button.
5. Once the accessory's first Bonjour advertisement is received, select "Start Pairing" and complete Pair-Setup.
6. In the "Wi-Fi Accessory Configuration" panel, enter the Wi-Fi network's SSID and PSK for Network A, and then select the "Send WAC Configuration" button.
7. Connect the Mac to Network A.
8. Once the accessory begins advertising via Bonjour on Network A, select the "Confirm WAC Configuration" button, and then select "Discover".
9. In the left sidebar, select the "Wi-Fi Configuration Control" characteristic.
10. In the "Prepare and Execute Timed Write" section, select "Build TLV".
11. In the "Wi-Fi Configuration Control" TLV builder, set the following parameters:
12. Set the "Operation" to "Update Configuration (Simple)".

13. Set the “Cookie” to any decimal value between 1 and 65535.
14. Set “SSID” to “networkb”.
15. Set “Security Mode” to “WPA2-PSK”.
16. Set “PSK” to the PSK for Network B.
17. Click the “Build TLV” button in the bottom right corner, then select “Timed Write”.
18. Wait for the accessory to disconnect.
19. Connect the Mac running HAT to Network B.
20. Once the accessory begins advertising via Bonjour on Network B, select “Discover”, and then verify that Pair-Verify and the discover operation complete successfully.
21. In the left sidebar, select the “Wi-Fi Configuration Control” characteristic.
22. In the “Prepare and Execute Timed Write” section, select “Build TLV”.
23. In the “Wi-Fi Configuration Control” TLV builder, set the following parameters:
24. Set the “Operation” to “Update Configuration (Simple)”.
25. Set the “Cookie” to any decimal value between 1 and 65535.
26. Set “SSID” to “networka”.
27. Set “Security Mode” to “WPA2-PSK”.
28. Set “PSK” to the PSK for Network A as “6987bfa06805c1c53884577b3cba951701b25bbe47ff1dbf890fd20b9bb193d7”.
29. Click the “Build TLV” button in the bottom right corner, and then select “Timed Write”.
30. Wait for the accessory to disconnect.
31. Connect the Mac running HAT to Network A.
32. Once the accessory begins advertising via Bonjour on Network A, select “Discover”, then verify that Pair-Verify and the discover operation completes successfully.

TCWR005 **Verify the “Current Transport” characteristic returns value “1” (True) when read over the Wi-Fi network.**

Applies to accessories that implement the Wi-Fi Reconfiguration service. Applies to accessories that support HomeKit Accessory Protocol specification R16 or later.

1. Ensure the accessory and the Mac running HAT are connected to the same Wi-Fi newtork.
2. If the accessory supports both Wi-Fi and Ethernet, ensure the Ethernet cable is unplugged.
3. Pair and discovery the accessory
4. In the left sidebar, select the “Current Transport” characteristic under the “Wi-Fi Transport” service.
5. Select “Read” to read the current value.

6. Verify the read response contains a value of "1" (True).

TCWR006 **Verify the "Current Transport" characteristic returns value "0" (False) when read over an Ethernet connection.**

Applies to accessories that implement the Wi-Fi Reconfiguration service. Applies to accessories that support HomeKit Accessory Protocol specification R16 or later.

1. Ensure the accessory is connected to the newtork only via Ethernet.
2. Pair and discovery the accessory.
3. In the left sidebar, select the "Current Transport" characteristic under the "Wi-Fi Transport" service.
4. Select "Read" to read the current value.
5. Verify the read response contains a value of "0" (False).

TCWR007 **Verify accessory uses valid values for the "Wi-Fi Capabilities" characteristic.**

Applies to accessories that implement the Wi-Fi Reconfiguration service.

1. Factory reset accessory.
2. In the trace window, navigate to the "WAC Discovery" view, select the accessory's WAC advertisement, and then select the "Details" button to show the details.
3. Notate the bits set under the "Flags" section.
4. Select the accessory in the Controllers window, and in the "Wi-Fi Accessory Configuration" panel, select the "Join access point" button.
5. Once the accessory's first Bonjour advertisement is recieved, select "Start Pairing" and complete Pair-Setup.
6. In the "Wi-Fi Accessory Configuration" panel, enter the Wi-Fi network's SSID and PSK for Network A, and then select the "Send WAC Configuration" button.
7. Ensure your Mac is on the network you expect the accessory to join, otherwise re-join the expected network.
8. Once the accessory begins advertising via Bonjour on the newly joined network, select the "Confirm WAC Configuration" button, and then select the "Discover" button.
9. In the left sidebar, select the "Wi-Fi Capabilities" characteristic.
10. Select the "Read" button to read the characteristic value.
11. In the Events view of the trace window, select the read response and select "Details" to show the details.
12. If bit 14 (Supports 2.4 GHz) was set in the WAC advertisment in step 3, verify "Wi-Fi Capabilities" characteristic has bit 0 (Supports 2.4 GHz) set.

13. If bit 15 (Supports 5 GHz) was set in the WAC advertisment in step 3, verify “Wi-Fi Capabilities” characteristic has bit 1 (Supports 5 GHz) set.
14. If bit 3 (Supports Wake on WLAN) was set in the WAC advertisment in step 3, verify “Wi-Fi Capabilities” characteristic has bit 2 (Supports Wake on WLAN) set.
15. Verify “Wi-Fi Capabilities” characteristic has bit 3 (Supports Station Mode) is set.

TCWR008 **Verify the accessory rejects reads and writes to the “Wi-Fi Configuration Control” characteristic from non-admin controllers.**

Applies to accessories that implement the Wi-Fi Reconfiguration service.

1. Pair and discover accessory.
2. On the bottom left corner of the controllers window, select the “+” and create a new IP controller.
3. Select the admin Controller 1, and under “Add Additional Controllers” panel, select “Controller 2” as Controller and select the “Add Controller” button, ensuring that the “admin” checkbox is deselected.
4. Select Controller 2, and then select “Discover”.
5. Select the “Wi-Fi Configuration Control” characteristic.
6. Select “Read”.
7. In the HTTP view of the trace, verify the read response returns HAP status code -70401.
8. In the “Prepare and Execute Timed Write” section, select “Build TLV”.
9. In the “Wi-Fi Configuration Control” TLV builder, set the following parameters:
10. Set the “Operation” to “Update Configuration (Simple)”.
11. Set the “Cookie” to any decimal value between 1 and 65535.
12. Set “SSID” to any valid value.
13. Set “Security Mode” to “WPA2-PSK”.
14. Set “PSK” to any valid value.
15. Click the “Build TLV” button in the bottom right corner, and then select “Timed Write”.
16. In the HTTP view of the trace, verify the write response returns HAP status code -70401.

TCWR009 **Verify accessory removes its Wi-Fi configuration when recieving a “Simple Update” operation with an empty “Station Configuration” TLV, and enters WAC mode.**

Applies to accessories that implement the Wi-Fi Reconfiguration service.

1. Pair and discover accessory.
2. In the IP Discovery view of the trace window, select the most recent accessory advertisement and notate the “Configuration Number” value.

3. In the left sidebar, select the “Wi-Fi Configuration Control” characteristic.
4. In the “Prepare and Execute Timed Write” section, select “Build TLV”.
5. In the “Wi-Fi Configuration Control” TLV builder, set the following parameters:
6. Set the “Operation” to “Update Configuration (Simple)”.
7. Set the “Cookie” to any decimal value between 1 and 65535.
8. Select the “Send Empty TLV (0 bytes)” checkbox.
9. Select the “Build TLV” button in the bottom right corner, and then select “Timed Write”.
10. In the Events view of the trace, verify the write response contains a “Cookie” value that matches the cookie used in step 7, and “Update Status” value of “0”.
11. Verify the accessory disconnects.
12. Select the controller and then select the “Stop” button to stop discovering accessory servers, and then select “Start” to begin again.
13. Using the IP Discovery view in the trace, verify the accessory is no longer advertising via Bonjour.
14. Using the WAC Discovery view in the trace, verify the accessory begins to advertise its WAC advertisement.
15. In the details of the WAC advertisement under Flags, check that bit 9 is set (Accessory is paired to a HomeKit controller).
16. Select the accessory, and in the “Wi-Fi Accessory Configuration” panel, select the “Join access point” button.
17. Once the accessory’s first Bonjour advertisement is recieved, select “Discover”.
18. In the “Wi-Fi Accessory Configuration” panel, enter the Wi-Fi network’s SSID and PSK for the network, and then select the “Send WAC Configuration” button.
19. Connect the Mac back to the original network.
20. Once the accessory begins advertising via Bonjour, select the “Confirm WAC Configuration” button, and then select “Discover”.
21. Verify the Pair-Verify and Discover operations complete successfully.

TCWR010 **Verify accessory already configured for Wi-Fi can be reconfigured onto a new network using the “Fail-Safe Update” procedure.**

Applies to accessories that implement the Wi-Fi Reconfiguration service.

1. Create two wireless networks that use different SSID’s and PSK’s (E.g. Network A and Network B).
2. Factory reset accessory.
3. Select the accessory, and in the “Wi-Fi Accessory Configuration” panel, select the “Join access point” button.

4. Once the accessory's first Bonjour advertisement is recieved, select "Start Pairing" and complete Pair-Setup.
5. In the "Wi-Fi Accessory Configuration" panel, enter the Wi-Fi network's SSID and PSK for Network A, and then select the "Send WAC Configuration" button.
6. Connect the Mac to Network A.
7. Once the accessory begins advertising via Bonjour on Network A, select the "Confirm WAC Configuration" button, and then select "Discover".
8. In the IP Discovery view of the trace window, select the most recent accessory advertisement and notate the "Configuration Number" value.
9. In the left sidebar, select the "Wi-Fi Configuration Control" characteristic.
10. In the "Prepare and Execute Timed Write" section, select "Build TLV".
11. In the "Wi-Fi Configuration Control" TLV builder, set the following parameters:
12. Set the "Operation" to "Update Configuration (Fail-Safe)".
13. Set the "Cookie" to any decimal value between 1 and 65535.
14. Set "SSID" to the SSID for Network B.
15. Set "Security Mode" to "WPA2-PSK".
16. Set "PSK" to the PSK for Network B.
17. Click the "Build TLV" button in the bottom right corner, and then select "Timed Write".
18. Note: Please continue with the next steps and execute all the test steps and then come back and verify the details in this step to avoid any timeout issues. In the Events view of the trace, verify the write response contains a "Cookie" value that matches the cookie used in step 13, an "updateStatus" cookie that matches the cookie used in step 13, and an "Update Status" with bit 16 (Update Pending) and bit 17 set (Session Restart Required).
19. Wait for the accessory to disconnect.
20. Connect the Mac running HAT to Network B and wait for the accessory to begin advertising again via Bonjour.
21. In the IP Discovery view of the trace window, verify the accessory begins to advertise on the new network, and that the "Configuration Number" value incremented by 1.
22. Select "Discover", and then verify Pair-Verify and the discover operation completes successfully.
23. Select the "Wi-Fi Configuration Control" characteristic and select the "Read" button.
24. Note: Please continue with the next steps and execute all the test steps and then come back and verify the details in this step to avoid any timeout issues. In the Events view of the trace, verify the read response contains a "Cookie" value that matches the cookie used in step 13, an "updateStatus" cookie that matches the cookie used in step 13, and an "Update Status" with bit 16 (Update Pending), bit 17 set (Session Restart Required), bit 21 (Link Established) and bit 22 set (Network Configured).
25. Select the "Wi-Fi Configuration Control" characteristic.

26. In the “Prepare and Execute Timed Write” section, select “Build TLV”.
27. Set the “Operation” to “Commit Configuration (Fail-Safe)”, and use the same cookie from step 13.
28. Click the “Build TLV” button in the bottom right corner, and then select “Timed Write”.
29. In the Events view of the trace, verify the write response contains a “Cookie” value that matches the cookie used in step 13, an “updateStatus” cookie that matches the cookie used in step 13, and an “Update Status” with bit 18 (Update Successful) and bit 23 set (Connection Verified). Please note that bits 21 and 22 may also be set.
30. Select the “Wi-Fi Configuration Control” characteristic.
31. In the “Prepare and Execute Timed Write” section, select “Build TLV”.
32. Set the “Operation” to “Read Configuration”.
33. Select the “Build TLV” button in the bottom right corner, and then select “Timed Write”.
34. In the Events view of the trace, select the write response and select “Details” to show the details.
35. Verify the write response contains the “Cookie” value that matches the cookie used in step 13, an “updateStatus” cookie that matches the cookie used in step 13, “Update Status” with bit 18 (Update Successful) and bit 23 set (Connection Verified), a valid “Country Code Configuration” (this TLV item is optional), a “Station Configuration” with “SSID” set to the SSID of Network B, “Security Mode” set to “WPA2-PSK”, and “PSK” with a value of “0”. Please note that bits 21 and 22 may also be set.
36. Please go back to Step 18 and step 24 above and verify the write response values as mentioned.

TCWR011 **Verify that when a “Fail-Safe” procedure fails due to an invalid SSID or PSK, on an accessory that is already configured for Wi-Fi, that the accessory returns the correct “Update Status” values and remains on the current network.**

Applies to accessories that implement the Wi-Fi Reconfiguration service.

1. Create two wireless networks that use different SSID’s and PSK’s (E.g. Network A and Network B).
2. Pair and discover accessory on Network A.
3. In the left sidebar, select the “Wi-Fi Configuration Control” characteristic.
4. In the “Prepare and Execute Timed Write” section, select “Build TLV”.
5. In the “Wi-Fi Configuration Control” TLV builder, set the following parameters:
6. Set the “Operation” to “Update Configuration (Fail-Safe)”.
7. Set the “Cookie” to any decimal value between 1 and 65535.
8. Set “SSID” to the SSID for Network B.
9. Set “Security Mode” to “WPA2-PSK”.
10. Set “PSK” to an invalid PSK for Network B. Do not use the correct PSK.
11. Click the “Build TLV” button in the bottom right corner, and then select “Timed Write”.

12. In the Events view of the trace, verify the write response contains a "Cookie" value that matches the cookie used in step 7, an "updateStatus" cookie that matches the cookie used in step 7, and an "Update Status" with bit 16 (Update Pending) and bit 17 set (Session Restart Required).
13. Wait for the accessory to disconnect, attempt to validate Network B with the provided credentials, and begin advertising again via Bonjour on Network A.
14. Select the "Wi-Fi Configuration Control" characteristic and select the "Read" button.
15. In the Events view of the trace, verify the read response contains a "Cookie" value that matches the cookie used in step 7, an "updateStatus" cookie that matches the cookie used in step 7, and an "Update Status" with bit 19 set (Update Failed).
16. In the left sidebar, select the "Wi-Fi Configuration Control" characteristic.
17. In the "Prepare and Execute Timed Write" section, select "Build TLV".
18. In the "Wi-Fi Configuration Control" TLV builder, set the following parameters:
19. Set the "Operation" to "Update Configuration (Fail-Safe)".
20. Set the "Cookie" to a new decimal value between 1 and 65535.
21. Set "SSID" to an invalid SSID. Do not use the SSID for any currenty available networks.
22. Set "Security Mode" to "WPA2-PSK".
23. Set "PSK" to the PSK for Network B.
24. Click the "Build TLV" button in the bottom right corner, and then select "Timed Write".
25. In the Events view of the trace, verify the write response contains a "Cookie" value that matches the cookie used in step 20, an "updateStatus" cookie that matches the cookie used in step 20, and an "Update Status" with bit 16 (Update Pending) and bit 17 set (Session Restart Required).
26. Wait for the accessory to disconnect, attempt to validate the network with the provided credentials, and begin advertising again via Bonjour on Network A.
27. Select the "Wi-Fi Configuration Control" characteristic and select the "Read" button.
28. In the Events view of the trace, verify the read response contains a "Cookie" value that matches the cookie used in step 20, an "updateStatus" cookie that matches the cookie used in step 20, and an "Update Status" with bit 19 set (Update Failed). Please note that bits 21 and 22 may also be set.

TCWR012 **Verify accessory responds to Update Configuration requests with HAP status code -70403 if a Fail-Safe operation is already in progress.**

Applies to accessories that implement the Wi-Fi Reconfiguration service.

1. Create two wireless networks that use different SSID's and PSK's (E.g. Network A and Network B).
2. Pair and discover accessory on Network A.
3. In the left sidebar, select the "Wi-Fi Configuration Control" characteristic.
4. In the "Prepare and Execute Timed Write" section, select "Build TLV".

5. In the “Wi-Fi Configuration Control” TLV builder, set the following parameters:
6. Set the “Operation” to “Update Configuration (Fail-Safe)”.
7. Set the “Cookie” to any decimal value between 1 and 65535.
8. Set “SSID” to the SSID for Network B.
9. Set “Security Mode” to “WPA2-PSK”.
10. Set “PSK” to the PSK for Network B.
11. Click the “Build TLV” button in the bottom right corner, and then select “Timed Write”.
12. Note: Please continue with the next steps and execute all the test steps and then come back and verify the details in this step to avoid any timeout issues. In the Events view of the trace, verify the write response contains a “Cookie” value that matches the cookie used in step 7, an “updateStatus” cookie that matches the cookie used in step 7, and an “Update Status” with bit 16 (Update Pending) and bit 17 set (Session Restart Required).
13. Wait for the accessory to disconnect.
14. Connect the Mac running HAT to Network B and wait for the accessory to begin advertising again via Bonjour.
15. In the IP Discovery view of the trace window, verify the accessory begins to advertise on the new network, and that the “Configuration Number” value incremented by 1.
16. Once the accessory begins advertising via Bonjour on Network B, select “Discover”, and then verify Pair-Verify and the discover operation completes successfully.
17. Select the “Wi-Fi Configuration Control” characteristic and select the “Read” button.
18. Note: Please continue with the next steps and execute all the test steps and then come back and verify the details in this step to avoid any timeout issues. In the Events view of the trace, verify the write response contains a “Cookie” value that matches the cookie used in step 7, an “updateStatus” cookie that matches the cookie used in step 7, and an “Update Status” with bit 16 (Update Pending), bit 17 set (Session Restart Required), bit 21 (Link Established) and bit 22 set (Network Configured).
19. In the left sidebar, select the “Wi-Fi Configuration Control” characteristic.
20. In the “Prepare and Execute Timed Write” section, select “Build TLV”.
21. In the “Wi-Fi Configuration Control” TLV builder, set the following parameters:
22. Set the “Operation” to “Update Configuration (Fail-Safe)”.
23. Set the “Cookie” to a new decimal value between 1 and 65535.
24. Set “SSID” to the SSID for Network A.
25. Set “Security Mode” to “WPA2-PSK”.
26. Set “PSK” to the PSK for Network A.
27. Click the “Build TLV” button in the bottom right corner, and then select “Timed Write”.
28. In the HTTP view of the trace window, verify the response to the write request contains HAP status code “-70403” (Resource is busy).

29. Select the “Wi-Fi Configuration Control” characteristic and select the “Read” button.
30. Note: Please continue with the next steps and execute all the test steps and then come back and verify the details in this step to avoid any timeout issues. In the Events view of the trace, verify the write response contains a “Cookie” value that matches the cookie used in step 7, an “updateStatus” cookie that matches the cookie used in step 7, and an “Update Status” with bit 16 set (Update Pending), bit 17 set (Session Restart Required), bit 21 set (Network Established), and bit 22 set (Network Configured).
31. In the left sidebar, select the “Wi-Fi Configuration Control” characteristic.
32. In the “Prepare and Execute Timed Write” section, select “Build TLV”.
33. Set the “Operation” to “Commit Configuration (Fail-Safe)”, and use the same cookie from step 7.
34. Click the “Build TLV” button in the bottom right corner, and then select “Timed Write”.
35. Select the “Wi-Fi Configuration Control” characteristic and select the “Read” button.
36. In the Events view of the trace, verify the read response contains contains only the “Cookie” value that matches the cookie used in step 7, an “updateStatus” cookie that matches the cookie used in step 7, and “Update Status” with bit 18 (Update Successful) and bit 23 set (Connection Verified). Please note that bits 21 and 22 may also be set.
37. Please go back to Step 12, step 18, step 30, and verify the write response values as mentioned.

TCWR013 **Verify “Update Status” does not persist across reboots.**

Applies to accessories that implement the Wi-Fi Reconfiguration service.

1. Create two wireless networks that use different SSID’s and PSK’s (E.g. Network A and Network B).
2. Pair and discover accessory.
3. In the left sidebar, select the “Wi-Fi Configuration Control” characteristic.
4. In the “Prepare and Execute Timed Write” section, select “Build TLV”.
5. In the “Wi-Fi Configuration Control” TLV builder, set the following parameters:
6. Set the “Operation” to “Update Configuration (Fail-Safe)”.
7. Set the “Cookie” to any decimal value between 1 and 65535.
8. Set “SSID” to the SSID for Network B.
9. Set “Security Mode” to “WPA2-PSK”.
10. Set “PSK” to the PSK for Network B.
11. Click the “Build TLV” button in the bottom right corner, and then select “Timed Write”.
12. Select the “Wi-Fi Configuration Control” characteristic and select the “Read” button.
13. Note: Please continue with the next steps and execute all the test steps and then come back and verify the details in this step to avoid any timeout issues. In the Events view of the trace, verify the write response contains a “Cookie” value that matches the cookie used in step 7, an “updateStatus” cookie that matches the cookie used in step 7, and an “Update Status” with bit 16 (Update Pending) and bit 17 set (Session Restart Required).

14. In the left sidebar, select the accessory, and then select “Disconnect”.
15. Power cycle the accessory and wait for the accessory to begin advertising again.
16. Select “Discover” and verify Pair-Verify operation completes successfully.
17. Select the “Wi-Fi Configuration Control” characteristic and select the “Read” button.
18. In the Events view of the trace, verify the write response contains a “Cookie” value that matches the cookie used in step 7 and an “Update Status” value of “0”.
19. Please go back to Step 13 above and verify the write response values as mentioned.

TCWR014 **Verify accessory rejects “Update Configuration” operations that contain invalid SSID or PSK values with HAP status code -70410.**

Applies to accessories that implement the Wi-Fi Reconfiguration service.

1. Create two wireless networks that use different SSID’s and PSK’s (E.g. Network A and Network B).
2. Pair and discover accessory on Network A.
3. In the left sidebar, select the “Wi-Fi Configuration Control” characteristic.
4. In the “Prepare and Execute Timed Write” section, select “Build TLV”.
5. In the “Wi-Fi Configuration Control” TLV builder, set the following parameters:
6. Set the “Operation” to “Update Configuration (Fail-Safe)”.
7. Set the “Cookie” to any decimal value between 1 and 65535.
8. Set “SSID” to “TestNetworkTestNetworkTestNetwork”. (SSID is longer than allowed)
9. Set “Security Mode” to “None”.
10. Click the “Build TLV” button in the bottom right corner and then select “Timed Write”.
11. In the HTTP view of the trace, select the write response, show the details, and verify the response contains HAP status code -70410.
12. In the left sidebar, select the “Wi-Fi Configuration Control” characteristic.
13. In the “Prepare and Execute Timed Write” section, select “Build TLV”.
14. In the “Wi-Fi Configuration Control” TLV builder, set the following parameters:
15. Set the “Operation” to “Update Configuration (Fail-Safe)” and enable the “Manual” checkbox.
16. Set the “Cookie” to any decimal value between 1 and 65535.
17. Set “SSID” to the SSID for Network B.
18. Set “Security Mode” to “None”.
19. Set “PSK” to “05909830cabd25439fc8b10908e660eb50fcc23d9d27ded026a8fe7c6c023b8a”. (Setting a PSK value when “Security Mode” is set to “None”.)

20. Click the “Build TLV” button in the bottom right corner and then select “Timed Write”.
21. In the HTTP view of the trace, select the write response, show the details, and verify the response contains HAP status code -70410.
22. Select the “Wi-Fi Configuration Control” characteristic.
23. In the “Prepare and Execute Timed Write” section, select “Build TLV”.
24. in the “Wi-Fi Configuration Control” TLV builder, set the following parameters:
25. Set the “Operation” to “Update Configuration (Fail-Safe)”.
26. Set the “Cookie” to any decimal value between 1 and 65535.
27. Set “SSID” to the SSID for Network B.
28. Set “Security Mode” to “WPA2-PSK”.
29. Set “PSK” to “05909830cabd25439fc8b10908e660eb50fcc23d9d27ded026a8fe7c6c023b8adfe”. (Invalid PSK credential, longer than allowed)
30. Click the “Build TLV” button in the bottom right corner and then select “Timed Write”.
31. In the HTTP view of the trace, select the write response, show the details, and verify the response contains HAP status code -70410.
32. Select the “Wi-Fi Configuration Control” characteristic.
33. In the “Prepare and Execute Timed Write” section, select “Build TLV”.
34. In the “Wi-Fi Configuration Control” TLV builder, set the following parameters:
35. Set the “Operation” to “Update Configuration (Fail-Safe)”.
36. Set the “Cookie” to any decimal value between 1 and 65535.
37. Set “SSID” to the SSID for Network B.
38. Set “Security Mode” to “WPA2-PSK”.
39. Leave the “PSK” field blank. (Invalid empty PSK credential)
40. Click the “Build TLV” button in the bottom right corner, and then select “Timed Write”.
41. In the HTTP view of the trace, select the write response, show the details, and verify the response contains HAP status code -70410.
42. Select the “Wi-Fi Configuration Control” characteristic.
43. In the “Prepare and Execute Timed Write” section, select “Build TLV”.
44. In the “Wi-Fi Configuration Control” TLV builder, set the following parameters:
45. Set the “Operation” to “Update Configuration (Fail-Safe)”.
46. Set the “Cookie” to any decimal value between 1 and 65535.
47. Set “SSID” to the SSID for Network B.
48. Set “Security Mode” to “WPA2-PSK”.

49. Set “PSK” to the PSK for Network B.
50. Click the “Build TLV” button in the bottom right corner, and then select “Timed Write”.
51. In the Events view of the trace, verify the write response contains a “Cookie” value that matches the cookie used in step 46, and an “Update Status” with bit 16 (Update Pending) and bit 17 set (Session Restart Required).
52. Verify the accessory disconnects.
53. Connect the Mac running HAT to Network B.
54. Once the accessory begins advertising via Bonjour on Network B, select “Discover”, and then verify Pair-Verify and the discover operation completes successfully.

TCWR015 **Verify accessory can be reconfigured onto an open and unsecured network.**

Applies to accessories that implement the Wi-Fi Reconfiguration service.

1. Create two wireless networks with different SSID’s (e.g., Network A and Network B), where Network A uses a PSK and Network B does not use a PSK.
2. Factory reset accessory.
3. Select the accessory, and in the “Wi-Fi Accessory Configuration” panel, select the “Join access point” button.
4. Once the accessory’s first Bonjour advertisement is recieved, select “Start Pairing” and complete Pair-Setup.
5. In the “Wi-Fi Accessory Configuration” panel, enter the Wi-Fi network’s SSID and PSK for Network A, and then select the “Send WAC Configuration” button.
6. Connect the Mac to Network A.
7. Once the accessory begins advertising via Bonjour on Network A, select the “Confirm WAC Configuration” button, and then select “Discover”.
8. In the left sidebar, select the “Wi-Fi Configuration Control” characteristic.
9. In the “Prepare and Execute Timed Write” section, select “Build TLV”.
10. In the “Wi-Fi Configuration Control” TLV builder, set the following parameters:
11. Set the “Operation” to “Update Configuration (Simple)”.
12. Set the “Cookie” to any decimal value between 1 and 65535.
13. Set “SSID” to the SSID for Network B.
14. Set “Security Mode” to “None”.
15. Click the “Build TLV” button in the bottom right corner, and then select “Timed Write”.
16. In the Events view of the trace, verify the write response contains a “Cookie” value that matches the cookie used in step 12, and “Update Status” value of “0”.

17. Verify the accessory disconnects.
18. Connect the Mac running HAT to Network B.
19. Once the accessory begins advertising via Bonjour on Network B, select "Discover", and then verify Pair-Verify and the discover operation completes successfully.

TCWR016 **If the "Operation Timeout" expires for a Fail-Safe operation, the accessory must clear the "Update Pending" flags and set the relevant error flags in the "Update Status".**

Applies to accessories that implement the Wi-Fi Reconfiguration service.

1. Create two wireless networks that use different SSID's and PSK's (e.g., Network A and Network B).
2. Pair and discover accessory on Network A.
3. In the left sidebar, select the "Wi-Fi Configuration Control" characteristic.
4. In the "Prepare and Execute Timed Write" section, select "Build TLV".
5. In the "Wi-Fi Configuration Control" TLV builder, set the following parameters:
6. Set the "Operation" to "Update Configuration (Fail Safe)".
7. Set the "Cookie" to any decimal value between 1 and 65535.
8. Set "SSID" to the SSID for Network B.
9. Set "Security Mode" to "WPA2-PSK".
10. Set "PSK" to the PSK for Network B.
11. Leave "Operation Timeout" field blank.
12. Click the "Build TLV" button in the bottom right corner and then select "Timed Write".
13. In the Events view of the trace, verify the write response contains a "Cookie" value that matches the cookie used in step 7, an "updateStatus" cookie that matches the cookie used in step 7, and an "Update Status" with bit 16 (Update Pending) and bit 17 set (Session Restart Required).
14. Verify the accessory disconnects.
15. Wait 61 seconds.
16. Select the "Wi-Fi Configuration Control" characteristic.
17. In the "Prepare and Execute Timed Write" section, select "Build TLV".
18. Set the "Operation" to "Commit Configuration (Fail-Safe)" and use the same cookie from step 7.
19. Click the "Build TLV" button in the bottom right corner, and then select "Timed Write".
20. In the Events view of the trace, verify the write response contains a "Cookie" value that matches the cookie used in step 7, an "updateStatus" cookie that matches the cookie used in step 7, and an "Update Status" with bit 19 set (Update Failed). Please note that bits 21 and 22 may also be set.
21. In the left sidebar, select the "Wi-Fi Configuration Control" characteristic.

22. In the “Prepare and Execute Timed Write” section, select “Build TLV”.
23. In the “Wi-Fi Configuration Control” TLV builder, set the following parameters:
24. Set the “Operation” to “Update Configuration (Fail Safe)”.
25. Set the “Cookie” to any decimal value between 1 and 65535.
26. Set “SSID” to the SSID for Network B.
27. Set “PSK” to the PSK for Network B.
28. Set “Security Mode” to “WPA2-PSK”.
29. Set “Operation Timeout” to “30”.
30. Click the “Build TLV” button in the bottom right corner and then select “Timed Write”.
31. In the Events view of the trace, verify the write response contains a “Cookie” value that matches the cookie used in step 24, an “updateStatus” cookie that matches the cookie used in step 24, and an “Update Status” with bit 16 (Update Pending) and bit 17 set (Session Restart Required).
32. Verify the accessory disconnects.
33. Wait 31 seconds.
34. Select the “Wi-Fi Configuration Control” characteristic.
35. In the “Prepare and Execute Timed Write” section, select “Build TLV”.
36. Set the “Operation” to “Commit Configuration (Fail-Safe)” and use the same cookie from step 24.
37. Click the “Build TLV” button in the bottom right corner and then select “Timed Write”.
38. In the Events view of the trace, verify the write response contains a “Cookie” value that matches the cookie used in step 24, an “updateStatus” cookie that matches the cookie used in step 24, and an “Update Status” with bit 19 set (Update Failed). Please note that bits 21 and 22 may also be set.

1.25 NFC Access and Pin Code Access Locks

TCL001: Any NFC Access Service must include the required characteristics.

TCL002: Verify that the accessory advertises the maximum number of supported NFC keys such as the Issuer Key, Suspended Device Credential Key and Active Device Credential Key.

TCL003: Verify that the Issuer Key is added to the accessory with HAP pairings and can be Listed with the NFC Access Issuer Key Request but not Removed with the NFC Access Issuer Key Request.

TCL004: Verify that the accessory responds with the appropriate error codes for NFC Access Issuer Key Requests (Duplicate, Not Supported, Out Of Resources, Does Not Exist).

TCL005: Verify that the accessory supports Device Credential Key Add, List, and Remove operations.

TCL006: Verify that the accessory responds with the appropriate error codes for NFC Access Device Credential Key Requests (Duplicate, Not Supported, Does Not Exist).

TCL007: Verify that the accessory supports Reader Key Add, List, and Remove operations.

TCL008: Verify that the accessory responds with the appropriate error codes for NFC Access Reader Key Requests (Duplicate, Not Supported, Out Of Resources, Does Not Exist).

TCL009: Verify that the device credential key can be suspended and resumed on the accessory.

TCL010: Verify that the NFC Issuer Key, Reader Key, and Device Credential Key persist across reboots.

TCL011: Verify that the Reader Key and Device Credential Key are cleared from the accessory when the last admin pairing is removed.

TCL012: After a Factory reset, verify that Tap to Unlock fails if the accessory has not been configured.

TCL013: Verify that the Non-Admin controllers do not have permission to write NFC Keys (Reader Key, Device Credential Key) to the accessory.

TCL014: Any Access Code Service must include the required characteristics.

TCL015: Verify that the accessory advertises the Minimum Length of Access Code, Maximum Length of Access Code, and Maximum number of Access Codes supported.

TCL016: Verify that the accessory supports the Add, List, Read, Update, and Remove Access Code operations.

TCL017: Verify that the accessory responds with the appropriate error codes for Access Code operations.

TCL018: Verify that the accessory supports multiple access code add, multiple access code list, multiple access code read, multiple access code update, multiple access code remove within a single operation.

TCL019: Verify that Non-Admin controllers do not have permission to add Access Codes to the accessory.

TCL020: Verify that the accessory sends a Keypad disabled notification to the controller when multiple incorrect Access Codes have been entered.

TCL021: Verify that the keypad on the accessory can be disabled and enabled from the controller.

TCL022: Verify that the Access Code persists after the accessory reboot.

TCL023: Verify that the Access Codes are removed from the accessory when the last Admin pairing is removed.

TCL024: Verify that the Access Codes are removed from the accessory after performing a Factory reset.

TCL025: Verify that the Lock Current state characteristic supports Event Notification Context Permission.

TCL026: Verify that the accessory sends Context information in the Broadcast Notifications while in Disconnected Mode.

TCL027: Verify that the accessory sends Context information in Connected Mode.

TCL028: Verify that the accessory can perform NFC expedited standard and expedited fast transactions to perform Tap to Lock / Tap to Unlock operations.

TCL029: Verify that the accessory can perform NFC Step-Up transactions to perform Tap to Lock / Tap to Unlock operations.

TCL030: If the accessory implements the "Hardware Finish" characteristic, verify that the Hardware Finish characteristic is set for the proper value.

TCL031: Verify that NFC keys can be added to the accessory on the Thread network and that Tap to Unlock works correctly.

TCL032: Verify that the accessory on the Thread network supports Add, List, Read, Update, and Remove Access Code operations.

TCL033: Verify that the Lock Current state characteristic supports Event Notification Context Permission when on the Thread network.

TCL034: Verify that the accessory sends event notification context when on the Thread network.

TCL035: Verify that the NFC accessory has Interoperability with iPhone and Watch.

TCL001 **Any NFC Access Service must include the required characteristics.**

Applies to accessories that support the NFC Access service. Applies to accessories that use HAP over BLE. Applies to accessories using the HAP over Thread transport. Perform this test case with HAT using the steps below.

Required characteristics:

- NFC Access Supported Configuration (r)
- NFC Access Control Point (r/w)
- Configuration State (r/ev*)

* Notify (ev) for BLE encompasses Indicate, Indicate (disconnected), and Broadcast. Please note, characteristics with TLV8 or String formats must not support Broadcast Events.

1. Pair and discover accessory.
2. In the left sidebar of the Controllers window, see each of the accessory's services.
3. Verify the "NFC Access" service is present.
4. Verify the service includes the "NFC Access Supported Configuration" characteristic.
5. Verify the characteristic has the permission "Paired Read".
6. Verify the characteristic format is "TLV8".
7. Verify the service includes the "NFC Access Control" characteristic.

8. Verify the characteristic has the permissions “Paired Read” and “Paired Write”
9. Verify the characteristic format is “TLV8”.
10. Verify the service includes the “Configuration State” characteristic.
11. Verify the characteristic has the permissions “Paired Read”, and “Notify”.
12. Verify the characteristic format is “uint16”.

TCL002 **Verify that the accessory advertises the maximum number of supported NFC keys such as the Issuer Key, Suspended Device Credential Key and Active Device Credential Key.**

Applies to accessories that support the NFC Access service. Applies to accessories that use HAP over BLE. Applies to accessories using the HAP over Thread transport. Perform this test case with HAT using the steps below.

1. Pair and discover accessory.
2. In the left sidebar of the Controllers window, select the “NFC Access Supported Configuration” characteristic in the “NFC Access” service.
3. Perform a Paired Read on the “NFC Access Supported Configuration” characteristic.
4. In the Events view, locate the Characteristic Read Completed response, and select the Details button.
5. Verify that the TLV Type 1 value in the read response is set to “Maximum number of Issuer Keys” which is 16 or higher.
6. Verify that the TLV Type 2 value in the read response is set to “Maximum number of Suspended Device Credential Keys” which is 16 or higher.
7. Verify that the TLV Type 3 value in the read response is set to “Maximum number of Active Device Credential Keys” which is 16 or higher.

TCL003 **Verify that the Issuer Key is added to the accessory with HAP pairings and can be Listed with the NFC Access Issuer Key Request but not Removed with the NFC Access Issuer Key Request.**

Applies to accessories that support the NFC Access service. Applies to accessories that use HAP over BLE. Perform this test case with HAT using the steps below.

1. Pair and Discover the accessory.
2. For HAP over BLE accessories, enable the “Pair Resume Keep Alive” checkbox with a 27-second interval.
3. In the left sidebar of the Controllers window, select the “NFC Access Control Point” characteristic in the “NFC Access ” service.
4. Select “Write with Response” for the NFC Access Control Point characteristic.
5. In Write [tlv8] pane, select Build TLV, select Operation Type “List” from the menu, select the NFC Access Request Type as “Issuer Key Request”, then select the “Build TLV” button on the bottom of the TLV builder and write the TLV.

6. Verify in the Events view of trace, in the Write Response, accessory responds with Issuer Key Identifier on TLV Type 1 and a Status Code of 0 (SUCCESS) on TLV Type 2.
7. Copy the Issuer Key Identifier (8 bytes) to a clipboard.
8. In Write [tlv8] pane, select the Build TLV, select Operation Type "Remove" from the menu, select the NFC Access Request Type as "Issuer Key Request" from the menu, enter the identifier copied in step 7 in the "Identifier" field, then select the "Build TLV" button on the bottom of the TLV builder and write the TLV.
9. Verify in the Events view of trace, in the Write Response, accessory responds with a Status Code of 4 (NOT SUPPORTED) on TLV Type 2.
10. Add Controller 2 from left sidebar.
11. Select the Accessory name in Controller 1 and under "Add Additional Controllers", select Controller 2, then select the "Admin" and "Add Controller".
12. Disconnect the accessory under Controller 1 and navigate to Controller 2 in left sidebar and Discover the accessory under Controller 2.
13. For HAP over BLE accessories, enable the "Pair Resume Keep Alive" checkbox with a 27-second interval.
14. In the left sidebar of the Controllers window, select "NFC Access Control Point" characteristic in "NFC Access " service.
15. Select "Write with Response" for the NFC Access Control Point characteristic.
16. In Write [tlv8] pane, select Build TLV, select Operation Type "List" from the menu, select the NFC Access Request Type as "Issuer Key Request" from the menu, then select the "Build TLV" button on the bottom of the TLV builder and write the TLV.
17. Verify in the Events view of trace, in the Write Response accessory responds with two NFC Access Issuer Issuer Key Responses for two Controllers with Issuer Key Identifier and a Status Code of 0 (SUCCESS) on both NFC Access Issuer Key Response.
18. Copy the Issuer Key Identifier (8 bytes) for Controller 2 from step above to a notes or text file.
19. In Write [tlv8] pane, select Build TLV, select Operation Type "Remove" from the menu, select the NFC Access Request Type as "Issuer Key Request" from the menu, enter the identifier copied in step 18 in the "Identifier" field, then select the "Build TLV" button on the bottom of TLV builder and write the TLV.
20. Verify in the Events view of trace, in the Write Response accessory responds with a Status Code of 4 (NOT SUPPORTED) on TLV Type 2.

TCL004 **Verify that the accessory responds with the appropriate error codes for NFC Access Issuer Key Requests (Duplicate, Not Supported, Out Of Resources, Does Not Exist).**

Applies to accessories that support the NFC Access service. Applies to accessories that use HAP over BLE. Perform this test case with HAT using the steps below.

1. Pair and discover accessory.

2. For HAP over BLE accessories, enable the “Pair Resume Keep Alive” checkbox with a 27-second interval.
3. In the left sidebar under “Key Stores” select “Controller 1”.
4. Copy the “Public Key” under Key Store (Controller 1).
5. In the left sidebar of the Controllers window, select “Configuration State” characteristic in “NFC Access” service and enable Event Notifications.
6. In the left sidebar of the Controllers window, select “NFC Access Control Point” characteristic in “NFC Access” service.
7. in the Options pane select “Write with Response”.
8. In the Write[tlv8] pane, select the “Build TLV” button, on the NFC Access Control Point TLV Builder select the Operation Type as “Add”, select the NFC Access Request Type as “Issuer Key Request”, Select the Issuer Key Type as “Ed25519”, paste the “Public Key” from step 4 into “Issuer Key” field, then select “Build TLV”. Select the “Write” button to send the TLV to the accessory.
9. In the Events View of trace, on the write response, verify that the accessory returns a Status Code of 2 (DUPLICATE).
10. In the Write[tlv8] pane, select the “Build TLV” button, on the NFC Access Control Point TLV Builder select the Operation Type as “Add”, select the NFC Access Request Type as “Issuer Key Request”, select the Issuer Key Type as “NIST256”, paste the “Public Key” from step 4 into the “Issuer Key” field, then select “Build TLV”.
11. Select the “Write” button to send TLV to the accessory.
12. In the Events View of trace, on the write response, verify that the accessory returns a Status Code of 4 (NOT SUPPORTED).
13. In the left sidebar of the Controllers window, select “NFC Access Supported Configuration” characteristic in “NFC Access” service.
14. Perform a Paired Read on “NFC Access Supported Configuration” characteristic.
15. In the Events View of trace, on the read response, verify that the accessory returns the “Maximum Number of Issuer Keys” supported on TLV Type 1.
16. Keep adding the controller by selecting the Add button (+) on the bottom left until the total number of controllers is equal to the “Maximum Number of Issuer Keys”.
17. Under “Controller 1”, select the accessory name, then under “Add Additional Controllers”, select controllers one by one from the menu and select “Add Controller”.
18. Verify that the Characteristic Notification is sent for the Configuration State characteristic with incremented value on each addition of a Controller.
19. Repeat steps 17 and 18 until all of the Controllers have been added.
20. In the Write[tlv8] pane, select the “Build TLV” button, on the NFC Access Control Point TLV Builder select the Operation Type as “Add”, select the NFC Access Request Type as “Issuer Key Request”, select the Issuer Key Type as “Ed25519”, enter 32 bytes of dummy HAP Public Key “aabbccddeeffaabbccddeeffaabbccddeeffaabbccddeeffaabbccddeeffaabb” into “Issuer Key” field, then select “Build TLV”.

21. Select the "Write" button to send TLV to the accessory.
22. In the Events View of trace, on the write response, verify that the accessory returns a Status Code of 1 (OUT OF RESOURCES).
23. In the Write[tlv8] pane, select the "Build TLV" button, on the NFC Access Control Point TLV Builder select the Operation Type as "Remove", select the NFC Access Request Type as "Issuer Key Request", enter 8 bytes of dummy Identifier "aabbccddeeffaabb" into "Identifier" field, then select "Build TLV".
24. Select the "Write" button to send TLV to the accessory.
25. In the Events View of trace, on the write response, verify that the accessory returns a Status Code of 3 (DOES NOT EXIST).
26. Select the accessory name under "Controller 1" and under "Remove Additional Controllers" select controllers one by one (other than Controller 1) from the menu and select "Remove Controller".
27. Verify that the Characteristic Notification is sent for the Configuration State characteristic with incremented value on each removal of a Controller.
28. Repeat steps 26 and 27 until all of the Controllers (except Controller 1) have been removed.
29. In the Write[tlv8] pane, select the "Build TLV" button, on the NFC Access Control Point TLV Builder select the Operation Type as "List", select the NFC Access Request Type as "Issuer Key Request", then select "Build TLV".
30. Select the "Write" button to send TLV to the accessory.
31. In the Events View of trace, on the write response, verify that the accessory returns one "NFC Access Issuer Key Response" with 8 bytes of identifier and a Status Code of 0 (SUCCESS).

TCL005 **Verify that the accessory supports Device Credential Key Add, List, and Remove operations.**

Applies to accessories that support the NFC Access service. Applies to accessories that use HAP over BLE. Perform this test case with HAT using the steps below.

1. Pair and discover accessory.
2. For HAP over BLE accessories, enable the "Pair Resume Keep Alive" checkbox with a 27-second interval.
3. Launch the Companion app on iPhone, select the gear icon on the top right, enable "Keep Companion Awake", then select "Done".
4. Select the "NFC Lock" button on the Companion app on iPhone.
5. In the left sidebar select "Controller 1" and select the "Companion Browser" button under the "HomeKit Companion" pane.
6. In the pop-up Companion Browser menu, select the "Connect" button.
7. Verify that the Green checkmark is seen next to the connected phone on the Companion Browser and minimize the Companion Browser.
8. In the left sidebar of the Controllers window, select the "Configuration State" characteristic in the "NFC Access" service and enable Event Notifications.

9. In the left sidebar of the Controllers window, select the “NFC Access Control Point” characteristic in the “NFC Access” service.
10. In the Options pane, select “Write with Response”.
11. In the Write[tlv8] pane, select the “Build TLV” button, on the NFC Access Control Point TLV Builder select the Operation Type as “Add”, select the NFC Access Request Type as “Device Credential Key Request”, on the Select Companion menu select the iPhone that is running Companion app, on the Device Credential Key State select “Active”, on the Device Credential Key Type select “NIST256”.
12. Verify that the Device Credential Key and Issue Key Identifier are populated in their respective fields, then select “Build TLV”.
13. Select the “Write” button to send TLV to the accessory.
14. In the Events View of trace, on the write response, verify that the accessory returns a Status Code of 0 (SUCCESS).
15. Verify that the Characteristic Notification is sent for the Configuration State characteristic with incremented value.
16. In the Write[tlv8] pane select the “Build TLV” button, on the NFC Access Control Point TLV Builder select the Operation Type as “List”, select the NFC Access Request Type as “Device Credential Key Request”, and select the Device Credential Key State as “Active”, then select “Build TLV”.
17. Select the “Write” button to send TLV to the accessory.
18. In the Events View of trace, on the write response, verify that the accessory returns one “NFC Access Device Credential Key Response” with 8 bytes of identifier and a Status Code of 0 (SUCCESS)
19. Copy the 8 bytes of Identifier by right clicking and selecting “Copy Raw Bytes”.
20. In the Write[tlv8] pane, select the “Build TLV” button, on the NFC Access Control Point TLV Builder select the Operation Type as “Remove”, select the NFC Access Request Type as “Device Credential Key Request”.
21. Paste the 8 bytes of Identifier copied in step 19 into the Identifier field, select “Build TLV”, then select “Write” button to send TLV to the accessory.
22. In the Events View of trace, on the write response, verify that the accessory returns one “NFC Access Device Credential Key Response” with a Status Code of 0 (SUCCESS).
23. Verify that a Characteristic Notification is sent for the Configuration State characteristic with incremented value.
24. In the Write[tlv8] pane, select the “Build TLV” button, on the NFC Access Control Point TLV Builder select the Operation Type as “List”, select the NFC Access Request Type as “Device Credential Key Request”, and select the Device Credential Key State as “Active”, then select “Build TLV”.
25. Select the “Write” button to send TLV to the accessory.
26. In the Events View of trace, on the write response, verify that the accessory returns 0 bytes of response.

TCL006 **Verify that the accessory responds with the appropriate error codes for NFC Access Device Credential Key Requests (Duplicate, Not Supported, Does Not Exist).**

Applies to accessories that support the NFC Access service. Applies to accessories that use HAP over BLE. Perform this test case with HAT using the steps below.

1. Pair and discover accessory.
2. For HAP over BLE accessories, enable the "Pair Resume Keep Alive" checkbox with a 27-second interval.
3. Launch the Companion app on iPhone, select the gear icon on the top right, enable "Keep Companion Awake", then select "Done".
4. Select the "NFC Lock" button on the Companion app on iPhone.
5. In the left sidebar select "Controller 1" and select the "Companion Browser" button under the "HomeKit Companion" pane.
6. In the pop-up Companion Browser menu select the "Connect" button.
7. Verify that the Green checkmark is seen next to the connected phone on the Companion Browser and minimize the Companion Browser.
8. In the left sidebar of the Controllers window, select the "Configuration State" characteristic in the "NFC Access" service and enable Event Notifications.
9. In the left sidebar of the Controllers window, select the "NFC Access Control Point" characteristic in the "NFC Access" service.
10. In the Options pane, select "Write with Response".
11. In the Write[tlv8] pane, select the "Build TLV" button, on the NFC Access Control Point TLV Builder select the Operation Type as "Add", select the NFC Access Request Type as "Device Credential Key Request", on the Select Companion menu select the iPhone that is running Companion app, on the Device Credential Key State select "Active", on the Device Credential Key Type select "NIST256".
12. Verify that the Device Credential Key and Issue Key Identifier are populated in their respective fields, then select "Build TLV".
13. Select the "Write" button to send TLV to the accessory.
14. In the Events View of trace, on the write response, verify that the accessory returns a Status Code of 0 (SUCCESS).
15. Verify that a Characteristic Notification is sent for the Configuration State characteristic with incremented value.
16. In the Write[tlv8] pane, select the "Build TLV" button, on the NFC Access Control Point TLV Builder select the Operation Type as "Add", select the NFC Access Request Type as "Device Credential Key Request", on the Select Companion menu select the iPhone that is running Companion app, on the Device Credential Key State select "Active", on the Device Credential Key Type select "NIST256".
17. Verify that the Device Credential Key and Issue Key Identifier are populated in their respective fields, then select "Build TLV".
18. Select the "Write" button to send TLV to the accessory.
19. In the Events View of trace, on the write response, verify that the accessory returns a Status Code of 2 (DUPLICATE).

20. In the Write[tlv8] pane, select the “Build TLV” button, on the NFC Access Control Point TLV Builder select the Operation Type as “Add” and select the NFC Access Request Type as “Device Credential Key Request”, on the Select Companion menu select the iPhone that is running Companion app, on the Device Credential Key State select “Active”, on the Device Credential Key Type select “Ed25519”.
21. Verify that the Device Credential Key and Issue Key Identifier are populated in their respective fields, then select “Build TLV”.
22. Select the “Write” button to send TLV to the accessory.
23. In the Events View of trace, on the write response, verify that the accessory returns a Status Code of 4 (NOT SUPPORTED).
24. In the Write[tlv8] pane, select the “Build TLV” button, on the NFC Access Control Point TLV Builder select the Operation Type as “Remove” and select the NFC Access Request Type as “Device Credential Key Request”, enter dummy identifier “aabbccddeeffaabb” in Identifier field, then select “Build TLV”.
25. Select the “Write” button to send TLV to the accessory.
26. In the Events View of trace, on the write response, verify that the accessory returns a Status Code of 3 (DOES NOT EXIST).

TCL007 **Verify that the accessory supports Reader Key Add, List, and Remove operations.**

Applies to accessories that support the NFC Access service. Applies to accessories that use HAP over BLE. Perform this test case with HAT using the steps below.

1. Pair and discover accessory.
2. For HAP over BLE accessories, enable the “Pair Resume Keep Alive” checkbox with a 27-second interval.
3. In the left sidebar of the Controllers window, select the “Configuration State” characteristic in the “NFC Access” service and enable Event Notifications.
4. In the left sidebar of the Controllers window, select the “NFC Access Control Point” characteristic in the “NFC Access” service.
5. In the Options pane select “Write with Response”.
6. In the Write[tlv8] pane, select the “Build TLV” button, on the NFC Access Control Point TLV Builder select the Operation Type as “Add” and select the NFC Access Request Type as “Reader Key Request”, then select the Reader Key Type as “NIST256’’.
7. Verify that the Reader Private Key and Reader Identifier are populated in their respective fields, then select “Build TLV”.
8. Select the “Write” button to send this TLV to the accessory.
9. In the Events View of trace, on the write response, verify that the accessory returns a Status Code of 0 (SUCCESS).
10. Verify that a Characteristic Notification is sent for the Configuration State characteristic with incremented value.

11. In the Write[tlv8] pane, select the “Build TLV” button, on the NFC Access Control Point TLV Builder select the Operation Type as “List”, select the NFC Access Request Type as “Reader Key Request”, then select “Build TLV”.
12. Select the “Write” button to send TLV to the accessory.
13. In the Events View of trace, on the write response, verify that the accessory returns one “NFC Access Reader Key Response” with 8 bytes of identifier and a Status Code of 0 (SUCCESS).
14. Copy the 8 bytes of Identifier by right-clicking and selecting “Copy Raw Bytes”.
15. In the Write[tlv8] pane, select the “Build TLV” button.
16. On the NFC Access Control Point TLV Builder select the Operation Type as “Remove”, select the NFC Access Request Type as “Reader Key Request”.
17. Paste the 8 bytes of Identifier copied in step 14 into the Identifier field, then select “Build TLV”.
18. Select the “Write” button to send TLV to the accessory.
19. In the Events View of trace, on the write response, verify that the accessory returns one “NFC Access Reader Key Response” with a Status Code of 0 (SUCCESS).
20. Verify that a Characteristic Notification is sent for the Configuration State characteristic with incremented value.
21. In the Write[tlv8] pane, select the “Build TLV” button, on the NFC Access Control Point TLV Builder select the Operation Type as “List”, select the NFC Access Request Type as “Reader Key Request”, then select “Build TLV”.
22. Select the “Write” button to send TLV to the accessory.
23. In the Events View of trace, on the write response, verify that the accessory returns 0-byte response.

TCL008 **Verify that the accessory responds with the appropriate error codes for NFC Access Reader Key Requests (Duplicate, Not Supported, Out Of Resources, Does Not Exist).**

Applies to accessories that support the NFC Access service. Applies to accessories that use HAP over BLE. Perform this test case with HAT using the steps below.

1. Pair and discover accessory.
2. For HAP over BLE accessories, enable the “Pair Resume Keep Alive” checkbox with a 27-second interval.
3. In the left sidebar of the Controllers window, select the “Configuration State” characteristic in the “NFC Access” service and enable Event Notifications.
4. In the left sidebar of the Controllers window, select the “NFC Access Control Point” characteristic in the “NFC Access” service.
5. In the Options pane select “Write with Response”.
6. In the Write[tlv8] pane, select the “Build TLV” button, on the NFC Access Control Point TLV Builder select the Operation Type as “Add” and select the NFC Access Request Type as “Reader Key Request”, then select the Reader Key Type as “NIST256”.

7. Verify that the Reader Private Key and Reader Identifier are populated in their respective fields, then select “Build TLV”.
8. Select “Write” button to send TLV to the accessory.
9. In the Events View of trace, on the write response, verify that the accessory returns a Status Code of 0 (SUCCESS).
10. Verify that a Characteristic Notification is sent for the Configuration State characteristic with incremented value.
11. In the Write[tlv8] pane, select the “Build TLV” button, on the NFC Access Control Point TLV Builder select the Operation Type as “Add” and select the NFC Access Request Type as “Reader Key Request”, then select the Reader Key Type as “NIST256”.
12. Verify that the Reader Private Key and Reader Identifier are populated in their respective fields, then select “Build TLV”.
13. Select “Write” button to send TLV to the accessory.
14. In the Events View of trace, on the write response, verify that the accessory returns a Status Code of 2 (DUPLICATE).
15. In the Write[tlv8] pane, select the “Build TLV” button, on the NFC Access Control Point TLV Builder select the Operation Type as “Add” and select the NFC Access Request Type as “Reader Key Request”, select the Reader Key Type as “NIST256”, on the Reader Private Key enter “11223344556677889900112233445566778899001122334455667788990011 22” and keep the default Reader Identifier that is pre-populated, then select “Build TLV”.
16. Select the “Write” button to send TLV to the accessory.
17. In the Events View of trace, on the write response, verify that the accessory returns a Status Code of 1 (OUT OF RESOURCES).
18. In the Write[tlv8] pane, select the “Build TLV” button, on the NFC Access Control Point TLV Builder select the Operation Type as “Add” and select the NFC Access Request Type as “Reader Key Request”, select the Reader Key Type as “Ed25519”.
19. Verify that the Reader Private Key and Reader Identifier are populated in their respective fields, then select “Build TLV”.
20. Select the “Write” button to send TLV to the accessory.
21. In the Events View of trace, on the write response, verify that the accessory returns a Status Code of 4 (NOT SUPPORTED).
22. In the Write[tlv8] pane, select the “Build TLV” button, on the NFC Access Control Point TLV Builder select the Operation Type as “Remove” and select the NFC Access Request Type as “Reader Key Request”, enter dummy identifier “aabbccddeeffaabb” in identifier field, then select “Build TLV”.
23. Select the “Write” button to send TLV to the accessory.
24. In the Events View of trace, on the write response, verify that the accessory returns a Status Code of 3 (DOES NOT EXIST).

TCL009 **Verify that the device credential key can be suspended and resumed on the accessory.**

Applies to accessories that support the NFC Access service. Applies to accessories that use HAP over BLE. Perform this test case with HAT using the steps below.

1. Pair and discover accessory.
2. For HAP over BLE accessories, enable the "Pair Resume Keep Alive" checkbox with a 27-second interval.
3. Launch the Companion app on iPhone, select the gear icon on the top right, enable "Keep Companion Awake", then select "Done".
4. Select the "NFC Lock" button on the Companion app on iPhone.
5. In the left sidebar select "Controller 1" and select the "Companion Browser" button under the "HomeKit Companion" pane.
6. In the Companion Browser pop-up menu, select the "Connect" button.
7. Verify that the Green checkmark is seen next to the connected phone on the Companion Browser and minimize the Companion Browser.
8. In the left sidebar of the Controllers window, select the "Configuration State" characteristic in the "NFC Access" service and enable Event Notifications.
9. In the left sidebar of the Controllers window, select the "NFC Access Control Point" characteristic in the "NFC Access" service.
10. In the Options pane, select "Write with Response".
11. In the Write[tlv8] pane, select the "Build TLV" button, on the NFC Access Control Point TLV Builder select the Operation Type as "Add" and select the NFC Access Request Type as "Reader Key Request".
12. Select the Reader Key Type as "NIST256".
13. Verify that the Reader Private Key and Reader Identifier are populated in their respective fields, then select "Build TLV".
14. Select the "Write" button to send TLV to the accessory.
15. In the Events View of trace, on the write response, verify that the accessory returns a Status Code of 0 (SUCCESS).
16. Verify that the Characteristic Notification is sent for the Configuration State characteristic with incremented value.
17. In the Write[tlv8] pane, select the "Build TLV" button, on the NFC Access Control Point TLV Builder select the Operation Type as "Add" and select the NFC Access Request Type as "Device Credential Key Request", on the Select Companion menu select the iPhone that is running Companion app, on the Device Credential Key State select "Active", then select the Device Credential Key Type as "NIST256".
18. Verify that the Device Credential Key and Issue Key Identifier are populated in their respective fields, then select "Build TLV".
19. Select the "Write" button to send TLV to the accessory.

20. In the Events View of trace, on the write response, verify the write TLV is a success (0x00).
21. Verify that the Characteristic Notification is sent for the Configuration State characteristic with incremented value.
22. On the Companion app on iPhone verify the status on top shows “Tap to Unlock Ready”.
23. Tap the NFC-sensitive area of the accessory with an iPhone running the Companion app.
24. In the Companion App Transaction details, verify that the “Select”, “Auth 0”, “Auth 1”, and “Control Flow” commands are received, and that the companion app shows the transaction as “Succeeded” on the bottom.
25. Verify that the accessory performed the Lock or Unlock operation after the Standard transaction succeeded.
26. If the transaction above succeeds, go to the next step. If the transaction fails, perform steps 23-25 one more time and verify that the transaction succeeds on the Companion app.
27. In the Write[tlv8] pane, select the “Build TLV” button, on the NFC Access Control Point TLV Builder select the Operation Type as “Add” and select the NFC Access Request Type as “Device Credential Key Request”, on the Select Companion menu select the iPhone that is running Companion app, on the Device Credential Key State select “Suspended”, then select the Device Credential Key Type as “NIST256”.
28. Verify that the Device Credential Key and Issue Key Identifier are populated in their respective fields, then select “Build TLV”.
29. Select the “Write” button to send TLV to the accessory.
30. In the Events View of trace, on the write response, verify the write TLV is a success (0x00).
31. Verify Characteristic Notification is sent for the Configuration State characteristic with incremented value.
32. In the Write[tlv8] pane, select the “Build TLV” button, on the NFC Access Control Point TLV Builder select the Operation Type as “List” and select the NFC Access Request Type as “Device Credential Key Request”, on the Device Credential Key state select “Suspended”, then select “Build TLV”.
33. Select the “Write” button to send TLV to the accessory.
34. In the Events View of trace, on the write response, verify the TLV Type 1 is set for an 8-byte identifier which is suspended and TLV Type 3 is set for value 0 (SUCCESS).
35. Tap the NFC-sensitive area of the accessory with an iPhone running the Companion app.
36. In the Companion App Transaction details, verify that the “Select”, “Auth 0”, “Auth 1”, and “Control Flow” commands are received, and that the companion app shows the transaction as “Succeeded” on the bottom.
37. Verify that the transaction succeeds on the Companion app but the accessory must not lock/ unlock since the device credential key is suspended.
38. In the Write[tlv8] pane, select the “Build TLV” button, on the NFC Access Control Point TLV Builder select the Operation Type as “Add” and select the NFC Access Request Type as “Device Credential Key Request”, on the Select Companion menu select the iPhone that is running Companion app, on the Device Credential Key State select “Active”, then select the Device Credential Key Type as “NIST256”.

39. Verify that the Device Credential Key and Issue Key Identifier are populated in their respective fields, then select "Build TLV".
40. Select the "Write" button to send TLV to the accessory.
41. In the Events View of trace, on the write response, verify the write TLV is a success (0x00).
42. Verify that a Characteristic Notification is sent for the Configuration State characteristic with incremented value.
43. Tap the NFC-sensitive area of the accessory with an iPhone running the Companion app.
44. In the Companion App Transaction details, verify that the "Select", "Auth 0", "Auth 1", and "Control Flow" commands are received, and that the companion app shows the transaction as "Succeeded" on the bottom.
45. Verify that the transaction succeeds and the accessory must lock/ unlock since the device credential key is active.

TCL010 **Verify that the NFC Issuer Key, Reader Key, and Device Credential Key persist across reboots.**

Applies to accessories that support the NFC Access service. Applies to accessories that use HAP over BLE. Perform this test case with HAT using the steps below.

1. Pair and discover accessory.
2. For HAP over BLE accessories, enable the "Pair Resume Keep Alive" checkbox with a 27-second interval.
3. Launch the Companion app on iPhone, select the gear icon on the top right, enable "Keep Companion Awake", then select "Done".
4. Select the "NFC Lock" button on the Companion app on iPhone.
5. In the left sidebar select "Controller 1" and select the "Companion Browser" button under the "HomeKit Companion" pane.
6. In the Companion Browser pop-up menu, select the "Connect" button.
7. Verify that the Green checkmark is seen next to the connected phone on the Companion Browser and minimize the Companion Browser.
8. In the left sidebar of the Controllers window, select the "Configuration State" characteristic in the "NFC Access" service and enable Event Notifications.
9. In the left sidebar of the Controllers window, select the "NFC Access Control Point" characteristic in the "NFC Access" service.
10. In the Options pane, select "Write with Response".
11. In the Write[tlv8] pane, select the "Build TLV" button, on the NFC Access Control Point TLV Builder select the Operation Type as "Add" and select the NFC Access Request Type as "Reader Key Request", then select the Reader Key Type as "NIST256".
12. Verify that the Reader Private Key and Reader Identifier are populated in their respective fields, then select "Build TLV".

13. Select the “Write” button to send TLV to the accessory.
14. In the Events View of trace, on the write response, verify that the accessory returns a Status Code of 0 (SUCCESS).
15. Verify that a Characteristic Notification is sent for the Configuration State characteristic with incremented value.
16. In the Write[tlv8] pane, select the “Build TLV” button, on the NFC Access Control Point TLV Builder select the Operation Type as “Add” and select the NFC Access Request Type as “Device Credential Key Request”, on the Select Companion menu select the iPhone that is running Companion app, on the Device Credential Key State select “Active”, then select the Device Credential Key Type as “NIST256”.
17. Verify that the Device Credential Key and Issue Key Identifier are populated in their respective fields, then select “Build TLV”.
18. Select the “Write” button to send TLV to the accessory.
19. In the Events View of trace, on the write response, verify that the write TLV is a success (0x00).
20. Verify that a Characteristic Notification is sent for the Configuration State characteristic with incremented value.
21. In the Write[tlv8] pane, select the “Build TLV” button, on the NFC Access Control Point TLV Builder select the Operation Type as “List” and select the NFC Access Request Type as “Issuer Key Request”, then select “Build TLV”.
22. Select the “Write” button to send TLV to the accessory.
23. In the Events View of trace, on the write response verify that the accessory returns TLV Type 1 with 8 bytes of Issuer Key Identifier and TLV Type 2 with value 0 (SUCCESS). Right-click the Issuer Key Identifer and copy the raw bytes into a notes or text file.
24. In the Write[tlv8] pane, select the “Build TLV” button, on the NFC Access Control Point TLV Builder select the Operation Type as “List” and select the NFC Access Request Type as “Reader Key Request”, then select “Build TLV”.
25. Select the “Write” button to send TLV to the accessory.
26. In the Events View of trace, on the write response, verify that the accessory returns TLV Type 1 with 8 bytes of the Reader Key Identifier and TLV Type 2 with value 0 (SUCCESS). Right-click the Reader Key Identifer and copy the raw bytes into a notes or text file.
27. In the Write[tlv8] pane, select the “Build TLV” button, on the NFC Access Control Point TLV Builder select the Operation Type as “List” and select the NFC Access Request Type as “Device Credential Key Request”, on the Device Credential Key State select “Active”, then select “Build TLV”.
28. Select the “Write” button to send TLV to the accessory.
29. In the Events View of trace, on the write response, verify that the accessory returns TLV Type 1 with 8 bytes of the Device Key Identifier and TLV Type 3 with value 0 (SUCCESS). Right-click the Device Credential Key Identifer and copy the raw bytes into a notes or text file.
30. Reboot the accessory, wait for it to show up in BLE discovery, then perform “Discover” under the Summary pane.

31. For HAP over BLE accessories, deselect and select the "Pair Resume Keep Alive" checkbox with a 27-second interval.
32. In the Options pane, deselect and select "Write with Response".
33. In the Write[tlv8] pane, select the "Build TLV" button, on the NFC Access Control Point TLV Builder select the Operation Type as "List" and select the NFC Access Request Type as "Issuer Key Request", then select "Build TLV".
34. Select the "Write" button to send TLV to the accessory.
35. In the Events View of trace, on the write response, verify that the accessory returns TLV Type 1 with 8 bytes of the Issuer Key Identifier same as that returned on step 23 and TLV Type 2 with value 0 (SUCCESS).
36. In the Write[tlv8] pane, select the "Build TLV" button, on the NFC Access Control Point TLV Builder select the Operation Type as "List" and select the NFC Access Request Type as "Reader Key Request", select "Build TLV".
37. Select the "Write" button to send TLV to the accessory.
38. In the Events View of trace, on the write response, verify that the accessory returns TLV Type 1 with 8 bytes of the Reader Key Identifier same as that returned on step 26 and TLV Type 2 with value 0 (SUCCESS).
39. In the Write[tlv8] pane, select the "Build TLV" button, on the NFC Access Control Point TLV Builder select the Operation Type as "List" and select the NFC Access Request Type as "Device Credential Key Request", on the Device Credential Key State select "Active", then select "Build TLV".
40. Select the "Write" button to send TLV to the accessory.
41. In the Events View of trace, on the write response, verify that the accessory returns TLV Type 1 with 8 bytes of the Device Key Identifier same as that returned on step 29 and TLV Type 3 with value 0 (SUCCESS).

TCL011 **Verify that the Reader Key and Device Credential Key are cleared from the accessory when the last admin pairing is removed.**

Applies to accessories that support the NFC Access service. Applies to accessories that use HAP over BLE. Applies to accessories using the HAP over Thread transport. Perform this test case with HAT using the steps below.

1. Pair and discover accessory.
2. For HAP over BLE accessories, enable the "Pair Resume Keep Alive" checkbox with a 27-second interval.
3. Launch the Companion app on iPhone, select the gear icon on the top right, enable "Keep Companion Awake", then select "Done".
4. Select the "NFC Lock" button on the Companion app on iPhone.
5. In the left sidebar select "Controller 1" and select the "Companion Browser" button under the "HomeKit Companion" pane.

6. In the Companion Browser pop-up menu, select the “Connect” button.
7. Verify that the Green checkmark is seen next to the connected phone on the Companion Browser and minimize the Companion Browser.
8. In the left sidebar of the Controllers window, select the “Configuration State” characteristic in the “NFC Access” service and enable Event Notifications.
9. In the left sidebar of the Controllers window, select the “NFC Access Control Point” characteristic in the “NFC Access” service.
10. In the Options pane, select “Write with Response”.
11. In the Write[tlv8] pane, select the “Build TLV” button, on the NFC Access Control Point TLV Builder select the Operation Type as “Add” and select the NFC Access Request Type as “Reader Key Request”, then select the Reader Key Type as “NIST256”.
12. Verify that the Reader Private Key and Reader Identifier are populated in respective fields, then select “Build TLV”.
13. Select the “Write” button to send TLV to the accessory.
14. For HAP over BLE accessories, in the Events View of trace, on the write response verify that the accessory returns a Status Code of 0 (SUCCESS).
15. For HAP over Thread accessories, in the HAP Traffic View of trace under Thread, on the write response verify that the accessory returns a Status Code of 0 (SUCCESS).
16. Verify that the Characteristic Notification is sent for the Configuration State characteristic with incremented value.
17. In the Write[tlv8] pane, select the “Build TLV” button, on the NFC Access Control Point TLV Builder select the Operation Type as “Add” and select the NFC Access Request Type as “Device Credential Key Request”, on the Select Companion menu select the iPhone that is running Companion app, on the Device Credential Key State select “Active”, then select the Device Credential Key Type as “NIST256”.
18. Verify that the Device Credential Key and Issue Key Identifier are populated in their respective fields, then select “Build TLV”.
19. Select the “Write” button to send TLV to the accessory.
20. For HAP over BLE accessories, in the Events View of trace, on the write response verify that the accessory returns a Status Code of 0 (SUCCESS).
21. For HAP over Thread accessories, in the HAP Traffic View of trace under Thread, on the write response verify that the accessory returns a Status Code of 0 (SUCCESS).
22. Verify that a Characteristic Notification is sent for the Configuration State characteristic with incremented value.
23. In the Write[tlv8] pane, select the “Build TLV” button, on the NFC Access Control Point TLV Builder select the Operation Type as “List” and select the NFC Access Request Type as “Issuer Key Request”, then select “Build TLV”.
24. Select the “Write” button to send TLV to the accessory.

25. For HAP over BLE accessories, in the Events View of trace, on the write response verify that the accessory returns TLV Type 1 with 8 bytes of the Issuer Key Identifier and TLV Type 2 with value 0 (SUCCESS).

26. For HAP over Thread accessories, in the HAP Traffic View of trace under Thread, on the write response verify that the accessory returns TLV Type 1 with 8 bytes of the Issuer Key Identifier and TLV Type 2 with value 0 (SUCCESS).

27. In the Write[tlv8] pane, select the "Build TLV" button, on the NFC Access Control Point TLV Builder select the Operation Type as "List" and select the NFC Access Request Type as "Reader Key Request", then select "Build TLV".

28. Select the "Write" button to send TLV to the accessory.

29. For HAP over BLE accessories, in the Events View of trace, on the write response verify that the accessory returns TLV Type 1 with 8 bytes of the Reader Key Identifier and TLV Type 2 with value 0 (SUCCESS).

30. For HAP over Thread accessories, in the HAP Traffic View of trace under Thread, on the write response verify that the accessory returns TLV Type 1 with 8 bytes of the Reader Key Identifier and TLV Type 2 with value 0 (SUCCESS).

31. In the Write[tlv8] pane, select the "Build TLV" button, on the NFC Access Control Point TLV Builder select the Operation Type as "List" and select the NFC Access Request Type as "Device Credential Key Request", on the Device Credential Key State select "Active", then select "Build TLV".

32. Select the "Write" button to send TLV to the accessory.

33. For HAP over BLE accessories, in the Events View of trace, on the write response verify that the accessory returns TLV Type 1 with 8 bytes of the Device Key Identifier and TLV Type 3 with value 0 (SUCCESS).

34. For HAP over Thread accessories, in the HAP Traffic View of trace under Thread, on the write response verify that the accessory returns TLV Type 1 with 8 bytes of the Device Key Identifier and TLV Type 3 with value 0 (SUCCESS).

35. Remove pairing with the accessory.

36. Put the accessory in pairing mode and pair and discover the accessory.

37. For HAP over BLE accessories, deselect and select the "Pair Resume Keep Alive" checkbox with a 27-second interval.

38. In the Options pane, deselect and select "Write with Response".

39. In the Write[tlv8] pane, select the "Build TLV" button, on the NFC Access Control Point TLV Builder select the Operation Type as "List" and select the NFC Access Request Type as "Reader Key Request", then select "Build TLV".

40. Select the "Write" button to send TLV to the accessory.

41. For HAP over BLE accessories, in the Events View of trace, on the write response, verify that the accessory returns 0 bytes value.

42. For HAP over Thread accessories, in the HAP Traffic View of trace under Thread, on the write response, verify that the accessory returns 0 bytes value.

43. In the Write[tlv8] pane, select the “Build TLV” button, on the NFC Access Control Point TLV Builder select the Operation Type as “List” and select the NFC Access Request Type as “Device Credential Key Request”, select the Device Credential Key State as “Active”, then select “Build TLV”.
44. Select the “Write” button to send TLV to the accessory.
45. For HAP over BLE accessories, in the Events View of trace, on the write response verify that the accessory returns 0 bytes value.
46. For HAP over Thread accessories, in the HAP Traffic View of trace under Thread, on the write response verify that the accessory returns 0 bytes value.

TCL012 **After a Factory reset, verify that Tap to Unlock fails if the accessory has not been configured.**

Applies to accessories that support the NFC Access service. Applies to accessories that use HAP over BLE. Perform this test case with HAT using the steps below.

1. Pair and discover accessory.
2. For HAP over BLE accessories, enable the “Pair Resume Keep Alive” checkbox with a 27-second interval.
3. Launch the Companion app on iPhone, select the gear icon on the top right, enable “Keep Companion Awake”, then select “Done”.
4. Select the “NFC Lock” button on the Companion app on iPhone.
5. In the left sidebar select “Controller 1” and select the “Companion Browser” button under the “HomeKit Companion” pane.
6. In the Companion Browser pop-up menu, select the “Connect” button.
7. Verify that the Green checkmark is seen next to the connected phone on the Companion Browser and minimize the Companion Browser.
8. In the left sidebar of the Controllers window, select the “Configuration State” characteristic in the “NFC Access” service and enable Event Notifications.
9. In the left sidebar of the Controllers window, select the “NFC Access Control Point” characteristic in the “NFC Access” service.
10. In the Options pane, select “Write with Response”.
11. In the Write[tlv8] pane, select the “Build TLV” button, on the NFC Access Control Point TLV Builder select the Operation Type as “Add” and select the NFC Access Request Type as “Reader Key Request”, then select the Reader Key Type as “NIST256”.
12. Verify that the Reader Private Key and Reader Identifier are populated in their respective fields, then select “Build TLV”.
13. Select the “Write” button to send TLV to the accessory.
14. In the Events View of trace, on the write response, verify that the accessory returns a Status Code of 0 (SUCCESS).

15. Verify that a Characteristic Notification is sent for the Configuration State characteristic with incremented value.
16. In the Write[tlv8] pane, select the "Build TLV" button, on the NFC Access Control Point TLV Builder select the Operation Type as "Add" and select the NFC Access Request Type as "Device Credential Key Request", on the Select Companion menu select the iPhone that is running Companion app, on the Device Credential Key State select "Active", then select the Device Credential Key Type as "NIST256".
17. Verify that the Device Credential Key and Issue Key Identifier are populated in their respective fields, then select "Build TLV".
18. Select the "Write" button to send TLV to the accessory.
19. In the Events View of trace, on the write response, verify that the write TLV is a success (0x00).
20. Verify that a Characteristic Notification is sent for the Configuration State characteristic with incremented value.
21. On the Companion app on iPhone verify that the status on top shows "Tap to Unlock Ready".
22. Tap the NFC-sensitive area of the accessory with an iPhone running the Companion app.
23. In the Companion App Transaction details, verify that the "Select", "Auth 0", "Auth 1", and "Control Flow" commands are received, and that the companion app shows the transaction as "Succeeded" on the bottom.
24. Verify that the accessory performed the Lock or Unlock operation after the transaction succeeded.
25. If the transaction above succeeds, go to the next step. If it fails, perform steps 22-24 one more time and verify that the transaction succeeds on the Companion app.
26. Factory reset the accessory as per the manufacturer's instructions.
27. Tap the NFC-sensitive area of the accessory with an iPhone running the Companion app.
28. Verify Companion app shows "Transaction Failed".
29. Tap the NFC-sensitive area of the accessory again with an iPhone running the Companion app and verify that the Companion app shows "Transaction Failed".

TCL013 **Verify that the Non-Admin controllers do not have permission to write NFC Keys (Reader Key, Device Credential Key) to the accessory.**

Applies to accessories that support the NFC Access service. Applies to accessories that use HAP over BLE. Applies to accessories using the HAP over Thread transport. Perform this test case with HAT using the steps below.

1. Pair and discover accessory.
2. Add an additional Controller, Controller 2.
3. In Controller 1, select the Accessory name, under "Add Additional Controllers", select Controller 2, do not select the "Admin" checkbox, then select "Add Controller".
4. Under Controller 1, disconnect the accessory, navigate to Controller 2 in the left sidebar, and Discover the accessory.

5. For HAP over BLE accessories, enable the "Pair Resume Keep Alive" checkbox with a 27-second interval.
6. Launch the Companion app on iPhone, select the gear icon on the top right, enable "Keep Companion Awake", then select "Done".
7. Select the "NFC Lock" button on the Companion app on iPhone.
8. In the left sidebar select "Controller 2" and select the "Companion Browser" button under the "HomeKit Companion" pane.
9. In the Companion Browser pop-up menu, select the "Connect" button.
10. Verify that the Green checkmark is seen next to the connected phone on the Companion Browser and minimize the Companion Browser.
11. In the left sidebar of the Controllers window, select the "Configuration State" characteristic in the "NFC Access" service and enable Event Notifications.
12. In the left sidebar of the Controllers window, select the "NFC Access Control Point" characteristic in the "NFC Access" service.
13. In the Options pane, select "Write with Response".
14. In the Write[tlv8] pane, select the "Build TLV" button, on the NFC Access Control Point TLV Builder select the Operation Type as "Add" and select the NFC Access Request Type as "Reader Key Request", then select the Reader Key Type as "NIST256".
15. Verify that the Reader Private Key and Reader Identifier are populated in their respective fields, then select "Build TLV".
16. Select the "Write" button to send TLV to the accessory.
17. For HAP over BLE accessories, in the HAP Transaction View of trace, on the write response verify that the accessory returns a Status Code of 0x06 (INVALID REQUEST).
18. For HAP over Thread accessories, in the HAP Traffic View of trace under Thread, on the write response verify that the accessory returns a Status Code of 0x06 (INVALID REQUEST).
19. Verify that no Characteristic Notification is sent for the Configuration State characteristic with incremented value.
20. In the Write[tlv8] pane, select the "Build TLV" button, on the NFC Access Control Point TLV Builder select the Operation Type as "Add", select the NFC Access Request Type as "Device Credential Key Request", on the Select Companion menu select the iPhone that is running Companion app, on the Device Credential Key State select "Active", then select the Device Credential Key Type as "NIST256".
21. Verify that the Device Credential Key and Issue Key Identifier are populated in their respective fields, then select "Build TLV".
22. Select the "Write" button to send TLV to the accessory.
23. For HAP over BLE accessories, in the HAP Transaction View of trace, on the write response verify that the accessory returns a Status Code of 0x06 (INVALID REQUEST).
24. For HAP over Thread accessories, in the HAP Traffic View of trace under Thread, on the write response verify that the accessory returns a Status Code of 0x06 (INVALID REQUEST).

25. Verify that no Characteristic Notification is sent for the Configuration State characteristic with incremented value.
26. On the Companion app on iPhone, verify that the status on top shows "Tap to Unlock Ready".
27. Tap the NFC-sensitive area of the accessory with an iPhone running the Companion app.
28. On the Companion app, verify that the Transaction failed and the accessory does not perform the Lock or Unlock operation.
29. Tap the NFC-sensitive area of the accessory with an iPhone running the Companion app one more time and verify that the app shows "Transaction Failed".

TCL014 **Any Access Code Service must include the required characteristics.**

Applies to accessories that support the Access Code service. Applies to accessories that use HAP over BLE. Applies to accessories using the HAP over Thread transport. Perform this test case with HAT using the steps below.

Required characteristics:

- Access Code Supported Configuration (r)
- Access Code Control Point (r/w)
- Configuration State (r/ev*)
- Active (r/w/ev*)

* Notify (ev) for BLE encompasses Indicate, indicate (disconnected), and Broadcast. Please note, characteristics with TLV8 or String formats must not support Broadcast Events.

1. Pair and discover accessory.
2. In the left sidebar of the Controllers window, see each of the accessory's services.
3. Verify that the "Access Code" service is present.
4. Verify that the service includes the "Access Code Supported Configuration" characteristic.
5. Verify that the characteristic has the permission "Paired Read".
6. Verify that the characteristic format is "TLV8".
7. Verify that the service includes the "Access Code Control" characteristic.
8. Verify that the characteristic has the permission "Paired Read" and "Paired Write".
9. Verify that the characteristic format is "TLV8".
10. Verify that the service includes the "Configuration State" characteristic.
11. Verify that the characteristic has the permission "Paired Read", and "Notify".
12. Verify that the characteristic format is "uint16".
13. Verify that the service includes the "Active" characteristic.

14. Verify that the characteristic has the permission “Paired Read”, “Paired Write” and “Notify”.
15. Verify that the characteristic format is “uint8”.

TCL015 **Verify that the accessory advertises the Minimum Length of Access Code, Maximum Length of Access Code, and Maximum number of Access Codes supported.**

Applies to accessories that support the Access Code service. Applies to accessories that use HAP over BLE. Applies to accessories using the HAP over Thread transport. Perform this test case with HAT using the steps below.

1. Pair and discover accessory.
2. For HAP over BLE accessories, enable the “Pair Resume Keep Alive” checkbox with a 27-second interval.
3. From the left column, select the “Access Code” service and then select the “Access Code Supported Configuration” characteristic and verify the permission is “Paired Read” and the format is TLV8.
4. From the left column, select the Access Code Service and perform a “Paired Read” on the “Access Code Supported Configuration” characteristic.
5. Select the read response from the Event view and verify that the response contains TLV Type 1 with value 1 (Arabic Numerals), TLV Type 2 with a Minimum Access Code length, TLV Type 3 with Maximum Access Code length and TLV Type 4 with Maximum supported Access Codes.
6. Verify that the value of TLV Type 3 is less than or equal to 8 and the value of TLV Type 2 is greater than or equal to 4.
7. Verify that the value of TLV Type 3 for Maximum Length of Access Code is greater than or equal to the value of TLV Type 2 for Minimum Length of Access Code.
8. Perform a Paired Write on the Access Code Supported Configuration characteristics by entering “0101010201040301080402640O” under Write [tlv8] and selecting “Write”.
9. For HAP over BLE accessories, in the HAP Transaction View of trace, on the write response verify that the accessory returns a Status Code of 0x01 (Unsupported PDU).
10. For HAP over Thread accessories, in the HAP Traffic View of trace, on the write response verify that the accessory returns a Status Code of 0x01 (Unsupported PDU).

TCL016 **Verify that the accessory supports the Add, List, Read, Update, and Remove Access Code operations.**

Applies to accessories that support the Access Code service. Applies to accessories that use HAP over BLE. Perform this test case with HAT using the steps below.

1. Pair and discover accessory.
2. For HAP over BLE accessories, enable the “Pair Resume Keep Alive” checkbox with a 27-second interval.
3. In the left sidebar of the Controllers window, select the “Configuration State” characteristic in the “Access Code” service and enable Event Notifications.

4. In the left sidebar of the Controllers window, select the “Access Code Control Point” characteristic in the “Access Code” service.
5. In the Options pane, select “Write with Response”.
6. In the Write[tlv8] pane, select the “Build TLV” button, on the Access Code Control Point TLV Builder select the Operation Type as “Add”, enter an Access Code “11111111” in the field, select the “Add” button, then select “Build TLV”.
7. Select the “Write” button to send TLV to the accessory.
8. In the Events View of trace, on the write response, verify that the write TLV is a success (0x00).
9. Verify that a Characteristic Notification is sent for the Configuration State characteristic with incremented value.
10. Enter the Access Code “11111111” on the keypad of the accessory and verify the accessory locks or unlocks.
11. In the Write[tlv8] pane, select the “Build TLV” button, on the Access Code Control Point TLV Builder select the Operation Type as “List”, then select “Build TLV”.
12. Select the “Write” button to send TLV to the accessory.
13. In the Events View of trace, on the write response, verify that the accessory returns an “Access Code Control Response” with an Access Code Identifier value of “0”.
14. In the Write[tlv8] pane, select the “Build TLV” button, on the Access Code Control Point TLV Builder select the Operation Type as “Read”, enter an Access Code Identifier “0” in the field, select the “Add” button, then select “Build TLV”.
15. Select the “Write” button to send TLV to the accessory.
16. In the Events View of trace, on the write response, verify that the accessory returns TLV Type 2 with Access Code value “11111111” and TLV Type 4 with Status Code value 0 (SUCCESS).
17. In the Write[tlv8] pane, select the “Build TLV” button, on the Access Code Control Point TLV Builder select the Operation Type as “Update”, enter an Access Code Identifier “0” in the field, enter Access Code as “22222222”, select the “Add” button, then select “Build TLV”.
18. Select the “Write” button to send TLV to the accessory.
19. In the Events View of trace, on the write response verify that the write TLV is a success (0x00)
20. Verify that the Characteristic Notification is sent for the Configuration State characteristic with incremented value.
21. Enter the Access Code “11111111” on the keypad of the accessory and verify that the accessory does not lock or unlock.
22. Enter the Access Code “22222222” on the keypad of the accessory and verify that the accessory locks or unlocks.
23. In the Write[tlv8] pane, select the “Build TLV” button, on the Access Code Control Point TLV Builder select the Operation Type as “Remove”, enter an Access Code Identifier “0” in the field, select the “Add” button, then select “Build TLV”.

24. Select the “Write” button to send TLV to the accessory.
25. In the Events View of trace, on the write response verify that the write TLV is a success (0x00)
26. Enter the Access Code “22222222” on the keypad of the accessory and verify that the accessory does not lock or unlock.
27. Verify that a Characteristic Notification is sent for the Configuration State characteristic with incremented value.

TCL017 **Verify that the accessory responds with the appropriate error codes for Access Code operations.**

Applies to accessories that support the Access Code service. Applies to accessories that use HAP over BLE. Perform this test case with HAT using the steps below.

1. Pair and discover accessory.
2. For HAP over BLE accessories, enable the “Pair Resume Keep Alive” checkbox with a 27-second interval.
3. In the left sidebar of the Controllers window, select the “Configuration State” characteristic in the “Access Code“ service and enable Event Notifications.
4. In the left sidebar of the Controllers window, select the “Access Code Control Point” characteristic in the “Access Code” service.
5. In the Options pane, select “Write with Response”.
6. In the Write[tlv8] pane, select the “Build TLV” button, on the Access Code Control Point TLV Builder select the Operation Type as “Add”, enter an Access Code “11111111” in the field, select the “Add” button, then select “Build TLV”.
7. Select the “Write” button to send TLV to the accessory.
8. In the Events View of trace, on the write response, verify that the write TLV is a success (0x00)
9. Verify that a Characteristic Notification is sent for the Configuration State characteristic with incremented value.
10. Enter the Access Code “11111111” on the keypad of the accessory and verify that the accessory locks or unlocks.
11. In the Write[tlv8] pane, select“Write” button to send the same TLV to the accessory.
12. In the Events View of trace, on the write response verify that the accessory responds with an Access Code Control Response with a status code value of 4 (Duplicate).
13. In the Write[tlv8] pane, select the “Build TLV” button, on the Access Code Control Point TLV Builder select the Operation Type as “Add” and enter an Access Code “111” in the field, select the “Add” button, then select “Build TLV”.
14. Select the “Write” button to send TLV to the accessory.
15. In the Events View of trace, on the write response, verify that the accessory responds with the Access Code Control Response with a status code value of 5 (Error. Smaller than min length).

16. In the Write[tlv8] pane, select the “Build TLV” button, on the Access Code Control Point TLV Builder select the Operation Type as “Add”, enter an Access Code “111111111” in the field, select the “Add” button, then select “Build TLV”.
17. Select the “Write” button to send TLV to the accessory.
18. In the Events View of trace, on the write response, verify that the accessory responds with the Access Code Control Response with a status code value of 6 (Error. Larger than max length).
19. In the Write[tlv8] pane, select the “Build TLV” button, on the Access Code Control Point TLV Builder select the Operation Type as “Add”, enter an Access Code “aaaa” in the field, select the “Add” button, then select “Build TLV”.
20. Select the “Write” button to send TLV to the accessory.
21. In the Events View of trace, on the write response, verify the accessory responds with an Access Code Control Response with a status code value of 7 (Error. Invalid Character).
22. In the Write[tlv8] pane, select the “Build TLV” button, on the Access Code Control Point TLV Builder select the Operation Type as “Update”, enter an Access Code Identifier “0” in the field, enter Access Code as “222” and select the “Add” button, then select “Build TLV”.
23. Select the “Write” button to send TLV to the accessory.
24. In the Events View of trace, on the write response, verify that the accessory responds with an Access Code Control Response with a status code value of 5 (Error. Smaller than min length).
25. In the Write[tlv8] pane, select the “Build TLV” button, on the Access Code Control Point TLV Builder select the Operation Type as “Update”, enter an Access Code Identifier “0” in the field, enter Access Code as “222222222” and select the “Add” button, then select “Build TLV”.
26. Select the “Write” button to send TLV to the accessory.
27. In the Events View of trace, on the write response, verify that the accessory responds with an Access Code Control Response with a status code value of 6 (Error. Larger than max length).
28. In the Write[tlv8] pane, select the “Build TLV” button, on the Access Code Control Point TLV Builder select the Operation Type as “Update”, enter an Access Code Identifier “0” in the field, enter Access Code as “aaaa” and select the “Add” button, then select “Build TLV”.
29. Select the “Write” button to send TLV to the accessory.
30. In the Events View of trace, on the write response, verify that the accessory responds with an Access Code Control Response with a status code value of 7 (Error. Invalid Character).
31. In the Write[tlv8] pane, select the “Build TLV” button, on the Access Code Control Point TLV Builder select the Operation Type as “Update”, enter an Access Code Identifier “10” in the field, enter Access Code as “3333” and select the “Add” button, then select “Build TLV”.
32. Select the “Write” button to send TLV to the accessory.
33. In the Events View of trace, on the write response, verify that the accessory responds with an Access Code Control Response with a status code value of 9 (Error. Does Not Exist).
34. In the Write[tlv8] pane, select the “Build TLV” button, on the Access Code Control Point TLV Builder select the Operation Type as “Read”, enter an Access Code Identifier “10” in the field, select the “Add” button, then select “Build TLV”.

35. Select the “Write” button to send TLV to the accessory.
36. In the Events View of trace, on the write response, verify that the accessory responds with an Access Code Control Response with a status code value of 9 (Error. Does Not Exist).
37. In the Write[tlv8] pane, select the “Build TLV” button, on the Access Code Control Point TLV Builder select the Operation Type as “Remove”, enter an Access Code Identifier “10” in the field, select the “Add” button, then select “Build TLV”.
38. Select the “Write” button to send TLV to the accessory.
39. In the Events View of trace, on the write response, verify that the accessory responds with an Access Code Control Response with a status code value of 9 (Error. Does Not Exist).
40. From the left column, select the Access Code Service and perform a “Paired Read” on the “Access Code Supported Configuration” characteristic.
41. Select the read response from the Event view and verify that the response contains TLV Type 1 with value 1 (Arabic Numerals), TLV Type 2 with a Minimum Access Code length, TLV Type 3 with Maximum Access Code length and TLV Type 4 with Maximum supported Access Codes.
42. In the left sidebar of the Controllers window, select the “Access Code Control Point” characteristic in the “Access Code” service.
43. In the Options pane, select “Write with Response”.
44. In the Write[tlv8] pane, select the “Build TLV” button, on the Access Code Control Point TLV Builder select the Operation Type as “Add”, enter a random Access Code that is within the supported length in the field, select “Add” button, then select “Build TLV”.
45. Select the “Write” button to send TLV to the accessory.
46. In the Events View of trace, on the write response, verify that the write TLV is a success (0x00)
47. Keep count of the Access Codes added and repeat steps 42-46 until the number of Access Codes added is equal to the Maximum supported Access Codes (TLV Type 4) obtained on step 41.
48. Add one more supported length Access Code to the accessory by repeating steps 42-46.
49. In the Events View of trace, on the write response, verify that the accessory responds with an Access Code Control Response with a status code value of 2 (Error. Exceeded maximum allowed access codes).

TCL018 **Verify that the accessory supports multiple access code add, multiple access code list, multiple access code read, multiple access code update, multiple access code remove within a single operation.**

Applies to accessories that support the Access Code service. Applies to accessories that use HAP over BLE. Perform this test case with HAT using the steps below.

1. Pair and discover accessory.
2. For HAP over BLE accessories, enable the “Pair Resume Keep Alive” checkbox with a 27-second interval.

3. In the left sidebar of the Controllers window, select the "Configuration State" characteristic in the "Access Code" service and enable Event Notifications.

4. In the left sidebar of the Controllers window, select the "Access Code Control Point" characteristic in the "Access Code" service.

5. In the Options pane, select "Write with Response".

6. In the Write[tlv8] pane, select the "Build TLV" button, on the Access Code Control Point TLV Builder select the Operation Type as "Add", enter an Access Code "0000" in the field and select the "Add" button, enter an Access Code "1111" in the field and select "Add", enter an Access Code "2222" in the field and select "Add", enter an Access Code "3333" in the field and select "Add", enter an Access Code "4444" in the field and select "Add", then select "Build TLV".

7. Select the "Write" button to send TLV to the accessory.

8. In the Events View of trace, on the write response, verify that the accessory responds with 5 Access Code Control Response TLVs with Status Code of 0 (Success).

9. Verify that a Characteristic Notification is sent for the Configuration State characteristic with incremented value.

10. Enter the Access Codes 0000, 1111, 2222, 3333, 4444 on the keypad one by one and verify that the accessory locks or unlocks with these Access Codes.

11. In the Write[tlv8] pane, select the "Build TLV" button, on the Access Code Control Point TLV Builder select the Operation Type as "Read" and enter an Access Code Identifier "0" in the field and select the "Add" button, enter an Access Code Identifier "1" in the field and select "Add", enter an Access Code Identifier "2" in the field and select "Add", enter an Access Code Identifier "3" in the field and select "Add", enter an Access Code Identifier "4" in the field and select "Add", then select "Build TLV".

12. Select the "Write" button to send TLV to the accessory.

13. In the Events View of trace, on the write response, verify that the accessory responds with 5 Access Code Control Response TLVs with Access Code values 0000, 1111, 2222, 3333, 4444 returned in each Access Code Control Response TLV and a Status Code of value of 0 (Success).

14. In the Write[tlv8] pane, select the "Build TLV" button, on the Access Code Control Point TLV Builder select the Operation Type as "Update" and enter an Access Code Identifier of "0", Access Code of "000000" and select the "Add" button, enter an Access Code Identifier of "1", Access Code of "111111" and select the "Add" button, enter an Access Code Identifier of "2", Access Code of "222222" and select the "Add" button, enter an Access Code Identifier of "3", Access Code of "333333" and select the "Add" button, enter an Access Code Identifier of "4", Access Code of "444444" and select the "Add" button, then select "Build TLV".

15. Select the "Write" button to send this TLV to the accessory.

16. In the Events View of trace, on the write response, verify that the accessory responds with 5 Access Code Control Response TLVs with Status Code value of 0 (Success).

17. Verify that a Characteristic Notification is sent for the Configuration State characteristic with incremented value.

18. Enter the Access Codes 000000, 111111, 222222, 333333 and 444444 on the keypad one by one and verify that the accessory locks or unlocks with these Access Codes.

19. In the Write[tlv8] pane, select the "Build TLV" button, on the Access Code Control Point TLV Builder select the Operation Type as "List", then select "Build TLV".
20. Select the "Write" button to send TLV to the accessory.
21. In the Events View of trace, on the write response verify the accessory responds with 5 Access Code Control Response TLVs with Access Code Identifiers 0, 1, 2, 3, and 4.
22. In the Write[tlv8] pane, select the "Build TLV" button, on the Access Code Control Point TLV Builder select the Operation Type as "Remove", enter an Access Code Identifier "0" in the field and select "Add" button, enter an Access Code Identifier "1" in the field and select "Add", enter an Access Code Identifier "2" in the field and select "Add", enter an Access Code Identifier "3" in the field and select "Add", enter an Access Code Identifier "4" in the field and select "Add", then select "Build TLV".
23. Select the "Write" button to send TLV to the accessory.
24. In the Events View of trace, on the write response, verify that the accessory responds with 5 Access Code Control Response TLVs with Status Code value of 0 (Success).
25. Verify that a Characteristic Notification is sent for the Configuration State characteristic with incremented value.
26. Enter the Access Codes 000000, 111111, 222222, 333333, 444444 on keypad one by one and verify that the accessory must not lock or unlock with these Access Codes.
27. In the Write[tlv8] pane, select the "Build TLV" button.
28. On the Access Code Control Point TLV Builder select the Operation Type as "List".
29. Select "Build TLV" and select the "Write" button to send this TLV to the accessory.
30. In the Events View of trace, on the write response, verify that the accessory does not respond with any Access Code Control Response TLVs.

TCL019 **Verify that Non-Admin controllers do not have permission to add Access Codes to the accessory.**

Applies to accessories that support the Access Code service. Applies to accessories that use HAP over BLE. Applies to accessories using the HAP over Thread transport. Perform this test case with HAT using the steps below.

1. Pair and discover accessory.
2. Add an additional Controller, Controller 2.
3. Select the Accessory name in Controller 1. Under "Add Additional Controllers", select Controller 2. Do not select the "Admin", and "Add Controller".
4. Disconnect the accessory under Controller 1 and navigate to Controller 2 in the left sidebar and Discover the accessory under Controller 2.
5. For HAP over BLE accessories, enable the "Pair Resume Keep Alive" checkbox with a 27-second interval.
6. In the left sidebar of the Controller 2 window, select the "Configuration State" characteristic in the "Access Code" service and enable Event Notifications.

7. In the left sidebar of the Controller 2 window, select the “Access Code Control Point” characteristic in the “Access Code” service.
8. In the Options pane, select “Write with Response”.
9. In the Write[tlv8] pane, select the “Build TLV” button, on the Access Code Control Point TLV Builder select the Operation Type as “Add”, enter an Access Code “11111111” in the field, select the “Add” button, then select “Build TLV”.
10. Select the “Write” button to send TLV to the accessory.
11. For HAP over BLE accessories, in the HAP Transaction View of trace, on the write response verify that the accessory returns a Status Code of 0x06 (INVALID REQUEST).
12. For HAP over Thread accessories, in the HAP Traffic View of trace under Thread, on the write response verify that the accessory returns a Status Code of 0x06 (INVALID REQUEST).
13. Verify that no Characteristic Notification is sent for the Configuration State characteristic with incremented value.

TCL020 **Verify that the accessory sends a Keypad disabled notification to the controller when multiple incorrect Access Codes have been entered.**

Applies to accessories that support the Access Code service. Applies to accessories that use HAP over BLE. Perform this test case with HAT using the steps below.

1. Pair and discover accessory.
2. For HAP over BLE accessories, enable the “Pair Resume Keep Alive” checkbox with a 27-second interval.
3. In the left sidebar of the Controllers window, select the “Active” characteristic in the “Access Code” service and enable Event Notifications.
4. In the left sidebar of the Controllers window, select the “Access Code Control Point” characteristic in the “Access Code” service.
5. In the Write[tlv8] pane, select the “Build TLV” button, on the Access Code Control Point TLV Builder select the Operation Type as “Add”, enter an Access Code “1111” in the field, select “Add”, then select “Build TLV”.
6. Select the “Write” button to send TLV to the accessory.
7. In the Events View of trace, on the write response, verify that the accessory responds with a Status Code of 0 (Success).
8. Start a stopwatch.
9. Keep a count of invalid attempts and enter an invalid Access Code on the accessory’s keypad, keep noting invalid attempts and keep entering invalid Access Codes until a Characteristic Notification for Active characteristic is sent with value 0 (keypad inactive).
10. Verify that the keypad is actually disabled by pressing the keys on the accessory’s keypad to confirm that it does not respond.

11. Wait until a Characteristic Notification is sent for Active characteristics with value 1 (keypad active) and keep a count of invalid attempts, enter invalid Access Codes on the keypad until a Characteristic Notification for Active characteristic is sent with value 0 (keypad inactive).
12. Repeat step 11 until the time elapsed is 10 minutes and count the number of invalid attempts, and verify this is less than or equal to 10 attempts.
13. Continue to wait until a Characteristic Notification is sent for Active characteristics with value 1 (keypad active). Keep a count of invalid attempts and enter invalid Access Codes until a Characteristic Notification for Active characteristic is sent with value 0 (keypad inactive).
14. Repeat step 13 until the time elapsed is 60 minutes and count the number of invalid attempts, and verify this is less than or equal to 20 attempts.
15. Manually change the accessory to lock or unlock state.
16. Verify that the keypad is enabled by entering the correct Access Code and verifying that the accessory locks or unlocks.
17. Enter the permissible attempts of invalid Access Code entries inside the 10-minute window before keypad lockout as supported by the manufacturer and verify the keypad does not yet lockout.
18. Enter the correct Access Code on the keypad and verify that the accessory locks or unlocks.

TCL021 **Verify that the keypad on the accessory can be disabled and enabled from the controller.**

Applies to accessories that support the Access Code service. Applies to accessories that use HAP over BLE. Perform this test case with HAT using the steps below.

1. Pair and discover accessory.
2. For HAP over BLE accessories, enable the "Pair Resume Keep Alive" checkbox with a 27-second interval.
3. In the left sidebar of the Controllers window, select the "Access Code Control Point" characteristic in the "Access Code" service.
4. In the Options pane, select "Write with Response".
5. In the Write[tlv8] pane, select the "Build TLV" button, on the Access Code Control Point TLV Builder select the Operation Type as "Add", enter an Access Code "1111" in the field, select "Add" button, then select "Build TLV".
6. Select the "Write" button to send TLV to the accessory.
7. In the Events View of trace, on the write response, verify that the accessory responds with a Status Code of 0 (Success).
8. Enter the Access Code "1111" on the accessory's keypad and verify that it locks or unlocks.
9. In the left sidebar of the Controllers window, select the "Active" characteristic in the "Access Code" service.
10. In the Options pane, select "Write with Response".

11. In the Write[Unsigned Integer] pane, select the “Write 0” button (Inactive).
12. In the Events View of trace, on the write response, verify that the accessory returns a Response value of “0” (Inactive).
13. Enter the Access Code “1111” on keypad and verify that the keypad is disabled and the accessory does not lock or unlock.
14. In the Write[Unsigned Integer] pane, select the “Write 1” button (Active).
15. In the Events View of trace, on the write response, verify that the accessory returns a Response value of “1” (Active).
16. Enter the Access Code “1111” on the accessory’s keypad and verify that it locks or unlocks.

TCL022 **Verify that the Access Code persists after the accessory reboot.**

Applies to accessories that support the Access Code service. Applies to accessories that use HAP over BLE. Perform this test case with HAT using the steps below.

1. Pair and discover accessory.
2. For HAP over BLE accessories, enable the “Pair Resume Keep Alive” checkbox with a 27-second interval.
3. In the left sidebar of the Controllers window, select the “Configuration State” characteristic in the “Access Code” service and enable Event Notifications.
4. In the left sidebar of the Controllers window, select the “Access Code Control Point” characteristic in the “Access Code” service.
5. In the Options pane, select “Write with Response”.
6. In the Write[tlv8] pane, select the “Build TLV” button, on the Access Code Control Point TLV Builder select the Operation Type as “Add”, enter an Access Code “0000” in the field, select the “Add” button, then select “Build TLV”.
7. Select the “Write” button to send TLV to the accessory.
8. In the Events View of trace, on the write response, verify that the accessory responds with Access Code Control Response TLV with a Status Code of 0 (Success).
9. Verify that a Characteristic Notification is sent for the Configuration State characteristic with incremented value.
10. Enter the Access Code “0000” on the keypad and verify that the accessory locks or unlocks.
11. Reboot the accessory and wait for the accessory to show up in BLE discovery and then perform “Discover” under the Summary pane.
12. For HAP over BLE accessories, deselect and select the “Pair Resume Keep Alive” checkbox with a 27-second interval.
13. In the left sidebar of the Controllers window, select the “Access Code Control Point” characteristic in the “Access Code” service.

14. In the Options pane deselect and select “Write with Response”.
15. In the Write[tlv8] pane, select the “Build TLV” button, on the Access Code Control Point TLV Builder select the Operation Type as “Read”, enter an Access Code Identifier “0” in the field, select the “Add” button, then select “Build TLV”.
16. Select the “Write” button to send TLV to the accessory.
17. In the Events View of trace, on the write response, verify that the accessory responds with Access Code Control Response TLVs with Access Code value “0000” returned in Access Code Control Response TLV and a Status Code of value of 0 (Success).
18. Enter the Access Code “0000” on the keypad and verify that the accessory locks or unlocks.
19. In the left sidebar of the Controllers window, select the “Configuration State” characteristic in the “Access Code” service and perform a Paired Read.
20. Verify that the value of the “Configuration State” is the same as the value in step 9.

TCL023 **Verify that the Access Codes are removed from the accessory when the last Admin pairing is removed.**

Applies to accessories that support the Access Code service. Applies to accessories that use HAP over BLE. Applies to accessories using the HAP over Thread transport. Perform this test case with HAT using the steps below.

1. Pair and discover accessory.
2. For HAP over BLE accessories, enable the “Pair Resume Keep Alive” checkbox with a 27-second interval.
3. In the left sidebar of the Controllers window, select the “Configuration State” characteristic in the “Access Code” service and enable Event Notifications.
4. In the left sidebar of the Controllers window, select the “Configuration State” characteristic in the “Access Code” service and perform a Paired Read.
5. Verify that the accessory responds with a Configuration State value that is 0 or 1.
6. In the left sidebar of the Controllers window, select the “Access Code Control Point” characteristic in the “Access Code” service.
7. In the Options pane, select “Write with Response”.
8. In the Write[tlv8] pane, select the “Build TLV” button, on the Access Code Control Point TLV Builder select the Operation Type as “Add”, enter an Access Code “0000” in the field, select the “Add” button, then select “Build TLV”.
9. Select the “Write” button to send TLV to the accessory.
10. For HAP over BLE accessories, in the Events View of trace, on the write response verify that the accessory responds with Access Code Control Response TLV with a Status Code of 0 (Success).
11. For HAP over Thread accessories, in the HAP Traffic View of trace under Thread, on the write response verify that the accessory responds with Access Code Control Response TLV with a Status Code of 0 (Success).

12. Verify that a Characteristic Notification is sent for the Configuration State characteristic with incremented value.
13. Enter the Access Code "0000" on keypad and verify that the accessory locks or unlocks.
14. In the left sidebar of the Controllers window, select the accessory and select "Remove Pairing" to remove pairing with accessory. Verify that the accessory goes through the normal Remove Pairing process.
15. Enter the Access Codes "0000" on keypad and verify that the accessory does not lock or unlock.
16. Put the accessory in pairing mode and pair and discover accessory.
17. For HAP over BLE accessories, deselect and select the "Pair Resume Keep Alive" checkbox with a 27-second interval.
18. In the left sidebar of the Controllers window, select the "Configuration State" characteristic in the "Access Code" service and perform a Paired Read.
19. Verify that the accessory responds with a Configuration State value that is 0 or 1.
20. In the left sidebar of the Controllers window, select the "Access Code Control Point" characteristic in the "Access Code" service.
21. In the Options pane, deselect and select "Write with Response".
22. In the Write[tlv8] pane, select the "Build TLV" button, on the Access Code Control Point TLV Builder select the Operation Type as "Read", enter an Access Code Identifier "0" in the field, select the "Add" button, then select "Build TLV".
23. Select the "Write" button to send TLV to the accessory.
24. For HAP over BLE accessories, in the Events View of trace, on the write response, verify that the accessory responds with Access Code Control Response TLV with a Status Code of 9 (Error. Does Not Exist).
25. For HAP over Thread accessories, in the HAP Traffic View of trace under Thread, on the write response, verify that the accessory responds with Access Code Control Response TLV with a Status Code of 9 (Error. Does Not Exist).

TCL024 **Verify that the Access Codes are removed from the accessory after performing a Factory reset.**

Applies to accessories that support the Access Code service. Applies to accessories that use HAP over BLE. Perform this test case with HAT using the steps below.

1. Pair and discover accessory.
2. For HAP over BLE accessories, enable the "Pair Resume Keep Alive" checkbox with a 27-second interval.
3. In the left sidebar of the Controllers window, select the "Configuration State" characteristic in the "Access Code" service and enable Event Notifications.
4. In the left sidebar of the Controllers window, select the "Configuration State" characteristic in the "Access Code" service and perform a Paired Read.

5. Verify that the accessory responds with a Configuration State value that is 0 or 1.
6. In the left sidebar of the Controllers window, select the “Access Code Control Point” characteristic in the “Access Code” service.
7. In the Options pane select “Write with Response”.
8. In the Write[tlv8] pane, select the “Build TLV” button, on the Access Code Control Point TLV Builder select the Operation Type as “Add”, enter an Access Code “0000” in the field, select the “Add” button, then select “Build TLV”.
9. Select the “Write” button to send TLV to the accessory.
10. In the Events View of trace, on the write response, verify that the accessory responds with Access Code Control Response TLV with a Status Code of 0 (Success).
11. Verify that a Characteristic Notification is sent for the Configuration State characteristic with incremented value.
12. Enter the Access Code “0000” on the keypad and verify that the accessory locks or unlocks.
13. Factory Reset the accessory per the manufacturer’s instructions.
14. Enter the Access Code “0000” on the keypad and verify that the accessory does not lock or unlock.
15. Pair and discover the accessory again.
16. For HAP over BLE accessories, deselect and select the “Pair Resume Keep Alive” checkbox with a 27-second interval.
17. In the left sidebar of the Controllers window, select the “Configuration State” characteristic in the “Access Code” service and perform a Paired Read.
18. Verify that the accessory responds with a Configuration State value that is 0 or 1.
19. In the left sidebar of the Controllers window, select the “Access Code Control Point” characteristic in the “Access Code” service.
20. In the Options pane, deselect and select “Write with Response”.
21. In the Write[tlv8] pane, select the “Build TLV” button, on the Access Code Control Point TLV Builder select the Operation Type as “Read”, enter an Access Code Identifier “0” in the field, select the “Add” button, then select “Build TLV”.
22. Select the “Write” button to send TLV to the accessory.
23. In the Events View of trace, on the write response, verify that the accessory responds with Access Code Control Response TLV with a Status Code of 9 (Error. Does Not Exist).
24. Enter the Access Code “0000” on the keypad and verify that the accessory does not lock or unlock.

TCL025 **Verify that the Lock Current state characteristic supports Event Notification Context Permission.**

Applies to accessories that support the NFC Access service. Applies to accessories that support the Access Code service. Applies to accessories that use HAP over BLE. Perform this test case with HAT using the steps below.

1. Pair and discover accessory.
2. In the left sidebar of the Controllers window, select the “Lock Current State” characteristic in the “Lock Mechanism” service.
3. Verify that the characteristic permissions include Event Notification Context.

TCL026 **Verify that the accessory sends Context information in the Broadcast Notifications while in Disconnected Mode.**

Applies to accessories that support the NFC Access service. Applies to accessories that support the Access Code service. Applies to accessories that use HAP over BLE. Perform this test case with HAT using the steps below.

1. Pair and discover accessory.
2. In the left sidebar of the Controllers window, select the “Service Signature” characteristic in the “HAP Protocol Information” service. Next, in the Protocol Configuration panel, select “generate broadcast key” and “Get all params”, then select Send.
3. In the Events view of trace, verify that Protocol Configuration Completed is seen without errors.
4. In the left sidebar of the Controllers window, select the “Lock Current State” characteristic in the “Lock Mechanism” service and in the Characteristic Configuration panel, enter “1” for Set Broadcast Interval, select “Bit 0 Enable/Disable Broadcast Notification”, then select Send Request.
5. In the Events view of trace, verify that Characteristic Configuration Completed is seen without errors.
6. Launch the Companion app on iPhone, select the gear icon on the top right, enable “Keep Companion Awake”, then select “Done”.
7. Select the “NFC Lock” button on the Companion app on iPhone.
8. In the left sidebar of the HAT window, select “Controller 1” and select the “Companion Browser” button under the “HomeKit Companion” pane.
9. In the Companion Browser pop-up menu, select the “Connect” button.
10. Verify that the Green checkmark is seen next to the connected phone on the Companion Browser and minimize the Companion Browser.
11. In the left sidebar of the Controllers window, select the “NFC Access Control Point” characteristic in the “NFC Access” service.
12. In the Options pane select “Write with Response”.
13. In the Write[tlv8] pane, select the “Build TLV” button, on the NFC Access Control Point TLV Builder select the Operation Type as “Add” and select the NFC Access Request Type as “Reader Key Request”, then select the Reader Key Type as “NIST256”.
14. Verify that the Reader Private Key and Reader Identifier are populated in their respective fields, then select “Build TLV”.
15. Select the “Write” button to send TLV to the accessory.

16. In the Events View of trace, on the write response, verify that the accessory returns a Status Code of 0 (Success).
17. In the Options pane deselect and select “Write with Response”.
18. In the Write[tlv8] pane, select the the “Build TLV” button, on the NFC Access Control Point TLV Builder select the Operation Type as “Add” and select the NFC Access Request Type as “Device Credential Key Request”, on the Select Companion menu, select the iPhone that is running Companion app, on the Device Credential Key State select “Active”, then select the Device Credential Key Type as “NIST256”.
19. Verify that the Device Credential Key and Issue Key Identifier are populated in their respective fields, then select “Build TLV”.
20. Select the “Write” button to send TLV to the accessory.
21. In the Events View of trace, on the write response, verify that the accessory returns a Status Code of 0 (Success).
22. Verify that the Companion app shows “Tap to Unlock Ready”.
23. In the Controllers window select accessory name, in the Summary Panel, select “Disconnect”.
24. Tap the NFC-sensitive area of the accessory with an iPhone running the Companion app and verify the Transaction succeeds.
25. Navigate to BLE Discovery on the HAT tool and verify that the accessory sends an Encrypted Broadcasted Notification with a context identifier that is 4 bytes of Issuer Key Identifier and verify that the source is set as “NFC”. Copy the 4 bytes of Issuer Key Identifier to a notes/text file.
26. Verify that the state number is incremented on Encrypted Broadcast Notifications.
27. In the left sidebar of the Controllers window, select the “NFC Access Control Point” characteristic in the “NFC Access” service.
28. In the Options pane deslect and select “Write with Response”.
29. In the Write[tlv8] pane, select the “Build TLV” button, on the NFC Access Control Point TLV Builder select the Operation Type as “List”, select the NFC Access Request Type as “Issuer Key Request”, then select “Build TLV”.
30. Select the “Write” button to send TLV to the accessory.
31. In the Events View of trace, on the write response, verify that the accessory returns 8 bytes of Issuer Key Identifier and a Status Code of 0 (Success).
32. Verify that the first 4 bytes of Issuer Key Identifier matches the 4 bytes of Issuer Key returned on the Encrypted broadcast notification copied to the notes/text file in step 25.
33. If the accessory supports the Access Code Service, proceed with next steps. If it does not support the Access Code Service skip to step 43.
34. In the left sidebar of the Controllers window, select the “Access Code Control Point” characteristic in the “Access Code” service.
35. In the Options pane, select “Write with Response”.

36. In the Write[tlv8] pane, select the "Build TLV" button, on the Access Code Control Point TLV Builder select the Operation Type as "Add", enter an Access Code "1111" in the field, select "Add" button, then select "Build TLV".
37. Select the "Write" button to send TLV to the accessory.
38. In the Events View of trace, on the write response, verify that the accessory responds with a Status Code of 0 (Success) and that the response also contains 4 bytes of Access Code Identifier 00000000.
39. In the Controllers window select accessory name, in the Summary Panel, select "Disconnect".
40. Enter the Access Code on the keypad and verify that the accessory locks or unlocks.
41. Navigate to BLE Discovery on the HAT tool and verify that the accessory sends an Encrypted Broadcasted notification with a context identifier that is 4 bytes of Access Code Identifier 00000000 and verify that the source is set as "Keypad".
42. Verify that the state number is incremented on Encrypted Broadcast Notifications.
43. In the Controllers window select accessory name, in the Summary Panel, select "Disconnect".
44. Perform a manual Lock/Unlock operation on the accessory and verify that an Encrypted Broadcast Notification is sent by the accessory with no context identifier and no context information such as Keypad/NFC is seen.
45. In the left sidebar of the Controllers window, select the "Lock Current State" characteristic in the "Lock Mechanism" service. In the Characteristic Configuration panel, verify that "Bit 0 Enable/ Disable Broadcast Notification" is deselected, then Send Request.
46. In the Events view of trace, verify that Characteristic Configuration Completed is seen without errors.
47. Perform a Manual Lock/Unlock on the accessory, and perform Tap to unlock with the Companion app and verify that no Encrypted Broadcast Notification is seen on BLE Discovery.

TCL027 **Verify that the accessory sends Context information in Connected Mode.**

Applies to accessories that support the NFC Access service. Applies to accessories that support the Access Code service. Applies to accessories that use HAP over BLE. Perform this test case with HAT using the steps below.

1. Pair and discover accessory.
2. For HAP over BLE accessories, enable the "Pair Resume Keep Alive" checkbox with a 27-second interval.
3. In the left sidebar of the Controllers window, select the "Lock Current State" characteristic in the "Lock Mechanism" service and "Enable" Events Notifications (Connected Events).
4. Launch the Companion app on iPhone, select the gear icon on the top right, enable "Keep Companion Awake", then select "Done".
5. Select the "NFC Lock" button on the Companion app on iPhone. In the left sidebar select "Controller 1" and select the "Companion Browser" button under the "HomeKit Companion" pane.
6. In the Companion Browser pop-up menu, select the "Connect" button.

7. Verify that the Green checkmark is seen next to the connected phone on the Companion Browser and minimize the Companion Browser.
8. In the left sidebar of the Controllers window, select the “NFC Access Control Point” characteristic in the “NFC Access” service.
9. In the Options pane select “Write with Response”.
10. In the Write[tlv8] pane, select the “Build TLV” button, on the NFC Access Control Point TLV Builder select the Operation Type as “Add” and select the NFC Access Request Type as “Reader Key Request”, then select the Reader Key Type as “NIST256”.
11. Verify that the Reader Private Key and Reader Identifier are populated in their respective fields, then select “Build TLV”.
12. Select the “Write” button to send TLV to the accessory.
13. In the Events View of trace, on the write response, verify that the write TLV is a success (0x00).
14. In the Write[tlv8] pane, select the “Build TLV” button, on the NFC Access Control Point TLV Builder select the Operation Type as “Add” and select the NFC Access Request Type as “Device Credential Key Request”, on the Select Companion menu select the iPhone that is running Companion app, on the Device Credential Key State select “Active”, then select the Device Credential Key Type as “NIST256”.
15. Verify that the Device Credential Key and Issuer Key Identifier are populated in their respective fields, then select “Build TLV”.
16. Select the “Write” button to send TLV to the accessory.
17. In the Events View of trace, on the write response, verify that the write TLV is a success (0x00).
18. In the Write[tlv8] pane, select the “Build TLV” button, on the NFC Access Control Point TLV Builder select the Operation Type as “List” and select the NFC Access Request Type as “Issuer Key Request”, then select “Build TLV”.
19. Select the “Write” button to send TLV to the accessory.
20. In the Events View of trace, on the write response, verify that TLV Type 1 has 8 bytes of Issuer Key Identifier, and TLV Type 2 has a value of 0 (SUCCESS). Copy the 8 bytes of Issuer Key Identifier to a notes/text file.
21. Tap the NFC-sensitive area of the accessory with an iPhone running the Companion app.
22. Verify that the Companion app shows the transaction as “Succeeded” at the bottom.
23. Verify that the accessory performed the Lock or Unlock operation after the transaction succeeded.
24. In the Events View of trace, verify that a “Characteristic Notification” is sent by the accessory. If you don’t see a “Characteristic Notification” on first NFC Tap, repeat steps 21-24 one more time.
25. In the HAP Procedures view, select the “Read Response” for the Lock Current State characteristic and verify that the TLV Type 1 has 4-byte value which is same as first 4 bytes of Issuer Key Identifier copied to a notes/text file on step 20.
26. Verify that TLV Type 2 has a value of 2 (NFC).
27. If the accessory supports the “Access Code” Service, continue the below steps. If it does not support the “Access Code” Service, stop the tests.

28. In the left sidebar of the Controllers window, select the “Access Code Control Point” characteristic in the “Access Code” service.

29. In the Options pane select “Write with Response”.

30. In the Write[tlv8] pane, select the “Build TLV” button, on the Access Code Control Point TLV Builder select the Operation Type as “Add”, enter Access Code string as “1111”, select the “Add” button, then select “Build TLV”.

31. Select the “Write” button to send this TLV to the accessory.

32. In the Events View of trace, on the write response, verify that the write TLV is a success (0x00).

33. In the Events View of trace, verify that TLV Type 1 has Access Code Identifier and that TLV Type 2 has Access Code. Copy the 4 bytes of Access Code Identifier to notes/text file.

34. Enter the Access Code “1111” on the keypad of the accessory and verify that a “Characteristic Notification” is seen on the Events View of trace.

35. In the HAP Procedures view, select “Read Response” for the Lock Current State characteristic and verify that TLV Type 1 has 4-byte value which is the same as 4 bytes of the Access Code Identifier copied on step 33.

36. Verify that TLV Type 2 has a value of 1 (Keypad).

TCL028 **Verify that the accessory can perform NFC expedited standard and expedited fast transactions to perform Tap to Lock / Tap to Unlock operations.**

Applies to accessories that support the NFC Access service. Applies to accessories that use HAP over BLE. Perform this test case with HAT using the steps below.

1. Pair and discover accessory.

2. For HAP over BLE accessories, enable the “Pair Resume Keep Alive” checkbox with a 27-second interval.

3. Launch the Companion app on iPhone, select the gear icon on the top right, enable “Keep Companion Awake”, then select “Done”.

4. Select the “NFC Lock” button on the Companion app on iPhone.

5. In the left sidebar select “Controller 1” and select “Companion Browser” button under the“HomeKit Companion” pane.

6. In the Companion Browser pop-up menu, select the “Connect” button.

7. Verify that the Green checkmark is seen next to the connected phone on the Companion Browser and minimize the Companion Browser.

8. In the left sidebar of the Controllers window, select the “NFC Access Control Point” characteristic in the “NFC Access” service.

9. In the Options pane select “Write with Response”.

10. In the Write[tlv8] pane, select the "Build TLV" button, on the NFC Access Control Point TLV Builder select the Operation Type as "Add" and select the NFC Access Request Type as "Reader Key Request", then select the Reader Key Type as"NIST256".
11. Verify that the Reader Private Key and Reader Identifier are populated in their respective fields, then select "Build TLV".
12. Select the "Write" button to send TLV to the accessory.
13. In the Events View of trace, on the write response, verify that the write TLV is a success (0x00).
14. In the Write[tlv8] pane, select the "Build TLV" button, on the NFC Access Control Point TLV Builder select the Operation Type as "Add" and select the NFC Access Request Type as "Device Credential Key Request", on the Select Companion menu select the iPhone that is running Companion app, on the Device Credential Key State select "Active", then select the Device Credential Key Type as "NIST256".
15. Verify that the Device Credential Key and Issue Key Identifier are populated in their respective fields, then select "Build TLV".
16. Select the "Write" button to send TLV to the accessory.
17. In the Events View of trace, on the write response, verify that the write TLV is a success (0x00).
18. On the Companion app on iPhone, verify that the status on top shows "Tap to Unlock Ready".
19. Tap the NFC-sensitive area of the accessory with an iPhone running the Companion app.
20. In the Companion App Transaction details, verify that the "Select", "Auth 0", "Auth 1", and "Control Flow" commands are received, and that the companion app shows the transaction as "Succeeded" on the bottom.
21. Verify that the accessory performed the Lock or Unlock operation after the Standard transaction succeeded.
22. If the transaction above succeeds, go to the next step. If it fails, perform steps 19-21 one more time and verify that the transaction succeeds.
23. Tap the NFC-sensitive area of the accessory with an iPhone running the Companion app.
24. In the Companion App Transaction details, verify that the "Select", "Auth 0", and "Control Flow" commands are received, and that the companion app shows the transaction as "Succeeded" on the bottom.
25. Verify that the accessory performed the Lock or Unlock operation after the Fast transaction succeeded.
26. If the transaction above succeeds, stop the test. If it fails, perform steps 23-25 one more time and verify that the transaction succeeds.

TCL029 **Verify that the accessory can perform NFC Step-Up transactions to perform Tap to Lock / Tap to Unlock operations.**

Applies to accessories that support the NFC Access service. Applies to accessories that use HAP over BLE. Perform this test case with HAT using the steps below.

1. Pair and discover accessory.

2. For HAP over BLE accessories, enable the "Pair Resume Keep Alive" checkbox with a 27-second interval.
3. Launch the Companion app on iPhone, select the gear icon on the top right, enable "Keep Companion Awake", then select "Done".
4. Select the "NFC Lock" button on the Companion app on iPhone.
5. In the left sidebar, select "Controller 1" then select the "Companion Browser" button under the "HomeKit Companion" pane.
6. In the Companion Browser pop-up menu, select the "Connect" button.
7. Verify that the Green checkmark is seen next to the connected phone on the Companion Browser and minimize the Companion Browser.
8. In the left sidebar of the Controllers window, select the "NFC Access Control Point" characteristic in the "NFC Access" service.
9. In the Options pane select "Write with Response".
10. In the Write[tlv8] pane, select the "Build TLV" button, on the NFC Access Control Point TLV Builder select the Operation Type as "Add" and select the NFC Access Request Type as "Reader Key Request", then select the Reader Key Type as "NIST256".
11. Verify that the Reader Private Key and Reader Identifier are populated in their respective fields, then select "Build TLV".
12. Select the "Write" button to send TLV to the accessory.
13. In the Events View of trace, on the write response, verify that the write TLV is a success (0x00).
14. In the Write[tlv8] pane, select the "Build TLV" button, on the NFC Access Control Point TLV Builder select the Operation Type as "Add" and select the NFC Access Request Type as "Device Credential Key Request", on the Select Companion menu select the iPhone that is running Companion app, select "Cancel" on NFC Access Control Point TLV builder.
15. On the Companion app on iPhone, verify that the status on top shows "Tap to Unlock Ready".
16. Tap the NFC-sensitive area of the accessory with an iPhone running the Companion app until the transaction completes.
17. In the Companion app Transaction details verify that the "Select", "Auth 0", "Auth 1", and "Control Flow" commands are received, "Step-Up Started" message is seen and verify the Companion app shows the Step-Up transaction as "Succeeded" at the end of transaction.
18. Verify that the accessory performed the Lock or Unlock operation after the Step-Up transaction succeeded.
19. If the transaction above succeeds, stop the test. If it fails, perform steps 16-18 one more time and verify that the Step-Up transaction succeeds.

TCL030 **If the accessory implements the "Hardware Finish" characteristic, verify that the Hardware Finish characteristic is set for the proper value.**

Applies to accessories that support the NFC Access service. Applies to accessories that use HAP over BLE. Applies to accessories using the HAP over Thread transport. Perform this test case with HAT using the steps below.

1. Pair and discover accessory.
2. In left sidebar of Controllers window, select "Accessory Information Service".
3. Verify that the permission on the "Hardware Finish" characteristic is "Paired Read" and format is "TLV8".
4. Select the "Hardware Finish" characteristic from the list and perform a paired read on the characteristic and verify from the Events view of Trace that TLV Type 1 value is set correctly and the value closely matches the actual Finish of the accessory hardware (matteBlack =0x000000, satinChrome = 0xE3E3E3, satinNickel =0xDAD5CE, polishedBrass =0xECD6AA).

TCL031 **Verify that NFC keys can be added to the accessory on the Thread network and that Tap to Unlock works correctly.**

Applies to accessories that support the NFC Access service. Applies to accessories using the HAP over Thread transport. Perform this test case with HAT using the steps below.

1. Select the "+" at the bottom of the left sidebar and select "Create Shared Key Store".
2. Select the "+" at the bottom of the left sidebar and select "Create BLE Controller" to make a new virtual BLE controller. In the Summary panel of the controller, assign the Shared Key Store created in Step 1, and click on the "Start'' button to begin discovering BLE accessories.
3. Select the "+" at the bottom of the left sidebar and select "Create Thread Controller" to make a new virtual Thread controller. In the Summary panel of the controller, assign the Shared Key Store created in Step 1, and click on the "Start'' button to begin discovering Thread accessories.
4. Pair and discover the accessory using the BLE Controller.
5. With the Thread network present, navigate to the "Thread Transport" service, select the "Thread Control Point" characteristic, click "Build TLV" within the "Write [tlv8]" panel, select "Set Thread Parameters" from the menu, enter the details of the thread border router network, enter "0" in the "Forming Allowed" field, then click "Build TLV".
6. Select "Write" to send the TLV to the accessory and check the Thread Discovery view of HAT Trace. The accessory must connect to the Border Router and show its Bonjour advertisement in the Thread Discovery view of HAT trace.
7. Discover the accessory using the Thread Controller and verify the Events View in HAT Trace shows the "Discovered Accessories" response. Click the Details button to verify that the response contains Services and Characteristics of the thread accessory.
8. Launch the Companion app on iPhone, select the gear icon on the top right, enable "Keep Companion Awake", then select "Done".
9. Select the "NFC Lock" button on the Companion app on iPhone.
10. In the left sidebar, select "Controller 2" (Thread controller) then select the "Companion Browser" button under the "HomeKit Companion" pane.

11. In the Companion Browser pop-up menu, select the “Connect” button.
12. Verify that the Green checkmark is seen next to the connected phone on the Companion Browser and minimize the Companion Browser.
13. In the left sidebar under Controllers 2, select the “Configuration State” characteristic in the “NFC Access” service and enable event notifications.
14. In the left sidebar of the Controllers 2, select the “NFC Access Control Point” characteristic in the “NFC Access” service.
15. In the Options pane select “Write with Response”.
16. In the Write[tlv8] pane, select the “Build TLV” button, on the NFC Access Control Point TLV Builder select the Operation Type as “Add” and select the NFC Access Request Type as “Reader Key Request”, then select the Reader Key Type as “NIST256”.
17. Verify that the Reader Private Key and Reader Identifier are populated in their respective fields, then select “Build TLV”.
18. Select the “Write” button to send TLV to the accessory.
19. In the HAP Traffic view under Thread, on the Thread Response verify that the accessory returns a status code of 0 (Success).
20. In the HAP Traffic view under Thread, verify Thread Event notification is sent for the Configuration State characteristic with incremented value.
21. In the Write[tlv8] pane, select the “Build TLV” button, on the NFC Access Control Point TLV Builder select the Operation Type as “Add” and select the NFC Access Request Type as “Device Credential Key Request”, on the Select Companion menu select the iPhone that is running Companion app, on the Device Credential Key State select “Active”, then select the Device Credential Key Type as “NIST256”.
22. Verify that the Device Credential Key and Issue Key Identifier are populated in their respective fields, then select “Build TLV”.
23. Select the “Write” button to send TLV to the accessory.
24. In the HAP Traffic view under Thread, on the Thread Response verify that the accessory returns a status code of 0 (Success).
25. In the HAP Traffic view under Thread, verify that a Thread Event notification is sent for the Configuration State characteristic with incremented value.
26. On the Companion app on iPhone, verify that the status on top shows “Tap to Unlock Ready”.
27. Tap the NFC-sensitive area of the accessory with an iPhone running the Companion app.
28. In the Companion App Transaction details, verify that the “Select”, “Auth 0”, “Auth 1”, and “Control Flow” commands are received, and that the companion app shows the transaction as “Succeeded” on the bottom.
29. Verify that the accessory performed the Lock or Unlock operation after the Standard transaction succeeded.
30. If the transaction above succeeds, go to the next step. If it fails, perform steps 27-29 one more time and verify that the transaction succeeds.

31. Tap the NFC-sensitive area of the accessory with an iPhone running the Companion app.
32. In the Companion App Transaction details, verify that the "Select", "Auth 0", and "Control Flow" commands are received, and that the companion app shows the transaction as "Succeeded" on the bottom.
33. Verify that the accessory performed the Lock or Unlock operation after the Fast transaction succeeded.
34. If the transaction above succeeds, stop the test. If it fails, perform steps 31-33 one more time and verify that the transaction succeeds.
35. Verify that the accessory performed the Lock or Unlock operation after the Fast transaction succeeded.

TCL032 **Verify that the accessory on the Thread network supports Add, List, Read, Update, and Remove Access Code operations.**

Applies to accessories that support the Access Code service. Applies to accessories using the HAP over Thread transport. Perform this test case with HAT using the steps below.

1. Select the "+" at the bottom of the left sidebar and select "Create Shared Key Store".
2. Select the "+" at the bottom of the left sidebar and select "Create BLE Controller" to make a new virtual BLE controller. In the Summary panel of the controller, assign the Shared Key Store created in Step 1, and click on the "Start'' button to begin discovering BLE accessories.
3. Select the "+" at the bottom of the left sidebar and select "Create Thread Controller" to make a new virtual Thread controller. In the Summary panel of the controller, assign the Shared Key Store created in Step 1, and click on the "Start'' button to begin discovering Thread accessories.
4. Pair and discover the accessory using the BLE Controller.
5. With the Thread network present, navigate to the "Thread Transport" service, select the "Thread Control Point" characteristic, click "Build TLV" within the "Write [tlv8]" panel, select "Set Thread Parameters" from the menu, enter the details of the thread border router network, enter "0" in the "Forming Allowed" field, then click "Build TLV".
6. Select "Write" to send the TLV to the accessory and check Thread Discovery view of HAT Trace. The accessory must connect to the Border Router and show its Bonjour advertisement in the Thread Discovery view of HAT trace.
7. Discover the accessory using the Thread Controller and verify that the Events View in HAT Trace shows the "Discovered Accessories" response. Click on the Details button to verify that the response contains the Services and Characteristics of the thread accessory.
8. In the left sidebar under Controller 2 (Thread), select the "Configuration State" characteristic in the "Access Code" service and enable event notifications.
9. In the left sidebar under Controller 2 (Thread), select the "Access Code Control Point" characteristic in the "Access Code" service.
10. In the Options pane, select "Write with Response".
11. In the Write[tlv8] pane, select the "Build TLV" button, on the Access Code Control Point TLV Builder select the Operation Type as "Add", enter an Access Code "1111" in the field, select the "Add" button, then select "Build TLV".

12. Select the “Write” button to send TLV to the accessory.
13. In the HAP Traffic view under Thread, on the Thread Response verify that the accessory returns a status code of 0 (Success).
14. In the HAP Traffic view under Thread, verify that a Thread Event notification is sent for the Configuration State characteristic with incremented value.
15. Enter the Access Code “1111” on the accessory’s keypad and verify that the accessory locks or unlocks.
16. In the Write[tlv8] pane, select the “Build TLV” button, on the Access Code Control Point TLV Builder select the Operation Type as “List”, then select “Build TLV”.
17. Select the “Write” button to send TLV to the accessory.
18. In the HAP Traffic view under Thread, on the Thread Response verify that the accessory returns an Access Code Identifier with a value of 0.
19. In the Write[tlv8] pane select the “Build TLV” button, on the Access Code Control Point TLV Builder select the Operation Type as “Read”, enter an Access Code Identifier “0” in the field, select “Add” button, then select “Build TLV”.
20. Select the “Write” button to send TLV to the accessory.
21. In the HAP Traffic view under Thread, on the Thread Response verify that the accessory returns TLV Type 2 with an Access Code value “1111” and TLV Type 4 with a Status Code value of 0 (SUCCESS).
22. In the Write[tlv8] pane, select the “Build TLV” button, on the Access Code Control Point TLV Builder select the Operation Type as “Update”, enter an Access Code Identifier “0” in the field, enter Access Code as “2222”, select “Add”, then select “Build TLV”.
23. Select the “Write” button to send TLV to the accessory.
24. In the HAP Traffic view under Thread, on the Thread Response verify that the accessory returns a status code of 0 (Success).
25. In the HAP Traffic view under Thread, verify that a Thread Event notification is sent for the Configuration State characteristic with incremented value.
26. Enter the Access Code “1111” on the accessory’s keypad and verify that the accessory does not lock or unlock.
27. Enter the Access Code “2222” on the accessory’s keypad and verify that the accessory locks or unlocks.
28. In the Write[tlv8] pane, select the “Build TLV” button, on the Access Code Control Point TLV Builder select the Operation Type as “Remove”, enter an Access Code Identifier “0” in the field, select “Add”, then select “Build TLV”.
29. Select the “Write” button to send TLV to the accessory.
30. In the HAP Traffic view under Thread, on the Thread Response verify that the accessory returns a status code of 0 (Success).
31. In the HAP Traffic view under Thread, verify that a Thread Event notification is sent for the Configuration State characteristic with incremented value.

32. Enter the Access Code “2222” on the accessory’s keypad and verify that the accessory does not lock or unlock.

TCL033 **Verify that the Lock Current state characteristic supports Event Notification Context Permission when on the Thread network.**

Applies to accessories that support the NFC Access service. Applies to accessories that support the Access Code service. Applies to accessories using the HAP over Thread transport. Perform this test case with HAT using the steps below.

1. Select the “+” at the bottom of the left sidebar and select “Create Shared Key Store”.
2. Select the “+” at the bottom of the left sidebar and select “Create BLE Controller” to make a new virtual BLE controller. In the Summary panel of the controller, assign the Shared Key Store created in Step 1, and click on the “Start’’ button to begin discovering BLE accessories.
3. Select the “+” at the bottom of the left sidebar and select “Create Thread Controller” to make a new virtual Thread controller. In the Summary panel of the controller, assign the Shared Key Store created in Step 1, and click on the “Start’’ button to begin discovering Thread accessories.
4. Pair and discover the accessory using the BLE Controller.
5. With the Thread network present, navigate to the “Thread Transport” service, select the “Thread Control Point” characteristic, click “Build TLV” within “Write [tlv8]” panel, select “Set Thread Parameters” from the menu, enter the details of the thread border router network, enter “0” in the “Forming Allowed” field, then click “Build TLV”.
6. Select “Write” to send the TLV to the accessory and check the Thread Discovery view of HAT Trace. The accessory must connect to the Border Router and show its Bonjour advertisement in the Thread Discovery view of HAT trace.
7. Discover the accessory using the Thread Controller and verify Events View in HAT Trace shows the “Discovered Accessories” response. Click on the Details button to verify that the response contains the Services and Characteristics of the thread accessory.
8. In the left sidebar of Controller 2 (Thread), select the “Lock Current State” characteristic in the “Lock Mechanism” service.
9. Verify that the characteristic permissions include “Event Notification Context”.

TCL034 **Verify that the accessory sends event notification context when on the Thread network.**

Applies to accessories that support the NFC Access service. Applies to accessories that support the Access Code service. Applies to accessories using the HAP over Thread transport. Perform this test case with HAT using the steps below.

1. Select the “+” at the bottom of the left sidebar and select “Create Shared Key Store”.
2. Select the “+” at the bottom of the left sidebar and select “Create BLE Controller” to make a new virtual BLE controller. In the Summary panel of the controller, assign the Shared Key Store created in Step 1, and click on the “Start’’ button to begin discovering BLE accessories.

3. Select the "+" at the bottom of the left sidebar and select "Create Thread Controller" to make a new virtual Thread controller. In the Summary panel of the controller, assign the Shared Key Store created in Step 1, and click on the "Start'' button to begin discovering Thread accessories.
4. Pair and discover the accessory using BLE Controller.
5. With the Thread network present, navigate to "Thread Transport" service, select the "Thread Control Point" characteristic, click "Build TLV" within the "Write [tlv8]" panel, select "Set Thread Parameters" from the menu, enter the details of the thread border router network, enter "0" in the "Forming Allowed" field, then click "Build TLV".
6. Select "Write" to send the TLV to the accessory and check Thread Discovery view of HAT Trace. The accessory must connect to the Border Router and show its Bonjour advertisement in the Thread Discovery view of HAT trace.
7. Discover the accessory using the Thread Controller and verify that the Events View in HAT Trace shows the "Discovered Accessories" response. Click on the Details button to verify that the response contains the Services and Characteristics of the thread accessory.
8. Launch the Companion app on iPhone, select the gear icon on the top right, enable "Keep Companion Awake", then select "Done".
9. Select the "NFC Lock" button on the Companion app on iPhone.
10. In the left sidebar, select "Controller 2" (Thread controller), then select the "Companion Browser" button under the "HomeKit Companion" pane.
11. In the Companion Browser pop-up menu, select the "Connect" button.
12. Verify that the Green checkmark is seen next to the connected phone on the Companion Browser and minimize the Companion Browser.
13. In the left sidebar under Controller 2, select the "Configuration State" characteristic in the "NFC Access" service and enable event notifications.
14. In the left sidebar under Controller 2, select the "Lock Current State" characteristic in the "Lock Mechanism" service and enable event notifications.
15. In the left sidebar of Controller 2, select the "NFC Access Control Point" characteristic in the "NFC Access" service.
16. In the Options pane select "Write with Response".
17. In the Write[tlv8] pane, select the "Build TLV" button, on the NFC Access Control Point TLV Builder select the Operation Type as "List" and select the NFC Access Request Type as "Issuer Key Request", then select "Build TLV".
18. Select the "Write" button to send TLV to the accessory.
19. In the Events View of trace, on the write response, verify that TLV Type 1 has 8 bytes of Issuer Key Identifier, and TLV Type 2 has a value of 0 (SUCCESS).
20. Copy the 8 bytes of Issuer Key Identifier to a notes or text file.
21. In the Write[tlv8] pane, select the "Build TLV" button, on the NFC Access Control Point TLV Builder select the Operation Type as "Add" and select the NFC Access Request Type as "Reader Key Request", then select the Reader Key Type as "NIST256".

22. Verify that the Reader Private Key and Reader Identifier are populated in their respective fields, then select "Build TLV".

23. Select the "Write" button to send TLV to the accessory.

24. In the HAP Traffic view under Thread, on the Thread Response verify that the accessory returns a status code of 0 (Success).

25. In the HAP Traffic view under Thread, verify that a Thread Event notification is sent for the Configuration State characteristic with incremented value.

26. In the Write[tlv8] pane, select the "Build TLV" button, on the NFC Access Control Point TLV Builder select the Operation Type as "Add" and select the NFC Access Request Type as "Device Credential Key Request", on the Select Companion menu select the iPhone that is running Companion app, on the Device Credential Key State select "Active", then select the Device Credential Key Type as "NIST256".

27. Verify that the Device Credential Key and Issue Key Identifier are populated in their respective fields, then select "Build TLV".

28. Select the "Write" button to send TLV to the accessory.

29. In the HAP Traffic view under Thread, on the Thread Response verify that the accessory returns a status code of 0 (Success).

30. In the HAP Traffic view under Thread, verify that a Thread Event notification is sent for the Configuration State characteristic with incremented value.

31. On the Companion app on iPhone verify that the status on top shows "Tap to Unlock Ready".

32. Tap the NFC-sensitive area of the accessory with an iPhone running the Companion app.

33. In the Companion App Transaction details, verify that the "Select", "Auth 0", "Auth 1", and "Control Flow" commands are received, and that the companion app shows the transaction as "Succeeded" on the bottom.

34. Verify that the accessory performed the Lock or Unlock operation after the Standard transaction succeeded.

35. In the HAP Traffic view under Thread, verify that the accessory sends a notification with a context identifier that is the same as the first 4 bytes of the Issuer Key Identifier copied on step 20 and that the source "NFC" is set by the accessory.

36. If the transaction above succeeds, go to the next step. If it fails, perform steps 32-35 one more time and verify that the transaction succeeds.

37. Tap the NFC-sensitive area of the accessory with an iPhone running the Companion app.

38. In the Companion App Transaction details, verify that the "Select", "Auth 0", and "Control Flow" commands are received, and that the companion app shows the transaction as "Succeeded" on the bottom.

39. Verify that the accessory performed the Lock or Unlock operation after the Fast transaction succeeded.

40. In the HAP Traffic view under Thread, verify that the accessory sends a notification with a context identifier that is the same as the first 4 bytes of the Issuer Key Identifier copied on step 20 and that the source "NFC" is set by the accessory.

41. If the transaction above succeeds, stop the test. If it fails, perform steps 37-40 one more time and verify that the transaction succeeds.

42. If the accessory supports the "Access Code" Service, continue the steps below. If it does not support the "Access Code" Service, stop the tests.

43. In the left sidebar under Controller 2, select the "Access Code Control Point" characteristic in the "Access Code" service.

44. In the Options pane, select "Write with Response".

45. In the Write[tlv8] pane, select the "Build TLV" button, on the Access Code Control Point TLV Builder select the Operation Type as "Add", enter an Access Code "1111" in the field, select "Add" button, then select "Build TLV".

46. Select the "Write" button to send TLV to the accessory.

47. In the HAP Traffic view under Thread, on the Thread Response verify that the accessory returns a status code of 0 (Success) and 4 bytes of Access Code Identifier.

48. Copy the Access Code Identifier to a notes or text file.

49. Enter the Access Code "1111" on the accessory's keypad and verify that the accessory locks or unlocks.

50. In the HAP Traffic view under Thread, verify that the accessory sends a notification with a context identifier that is the same as the 4 bytes of Access Code Identifier copied on step 48, and that the source "Keypad" is set by the accessory.

TCL035 **Verify that the NFC accessory has Interoperability with iPhone and Watch.**

Applies to accessories that support the NFC Access service. Perform this test case using the Home app on iOS.

1. Run the tests on an NFC accessory per the NFC Interoperability Test Plan on the MFi Portal.

2. Verify that the NFC accessory meets all the criteria on the NFC Interoperability Test Plan.

1.26 Accessory Firmware Updates

TCAFU001: Accessories that update firmware directly in the Home App must have "Firmware Update" service that includes the required characteristics.

TCAFU002: Accessories that support the Firmware Update service must also include the Data Stream Transport Management service.

TCAFU003: If the accessory implements the Firmware Update service, verify that the Accessory Information Service's Firmware Revision characteristic has the Notify permission.

TCAFU004: Accessories must communicate their readiness for staging the firmware update by setting the appropriate value for the Staging Not Ready Reasons field of the Firmware Update Status for Low Battery.

TCAFU005: Accessories must communicate their readiness for staging the firmware update by setting the appropriate value for the Staging Not Ready Reasons field of the Firmware Update Status for Low Connectivity.

TCAFU006: Accessories must communicate their readiness for staging the firmware update by setting the appropriate value for the Staging Not Ready Reasons field of the Firmware Update Status for all other reasons.

TCAFU007: Once the staging process has started, accessories must reflect this by changing the Firmware Update State in the Firmware Update Status to Staging In-Progress.

TCAFU008: If the transfer is paused and partially transferred update is retained, the Firmware Update State must change to Staging Paused.

TCAFU009: If a newer version of the SuperBinary is being offered by the secondary Controller while the Accessory is in the middle of staging a SuperBinary from the primary Controller, the Accessory should begin to transfer the newer SuperBinary.

TCAFU010: If a lower version of the SuperBinary is being offered by the secondary Controller while the Accessory is in the middle of staging a SuperBinary from the primary Controller, the Platform Accessory should deny the secondary offer.

TCAFU011: If the the Staging process is paused before it has completed and the SuperBinary cannot be retained, the "Firmware Update State" characteristic must be set to "Idle".

TCAFU012: Once the firmware update is staged successfully on the accessory, the Firmware Update State must change to Staging Succeeded.

TCAFU013: Accessories must begin to transfer the newer version of SuperBinary even if they already have another lower version completely staged.

TCAFU014: Accessories must communicate their readiness for applying the firmware update by setting the appropriate value for the Update Not Ready Reasons field of the Firmware Update Status for Low Battery.

TCAFU015: Accessories must communicate their readiness for applying the firmware update by setting the appropriate value for the Update Not Ready Reasons field of the Firmware Update Status for Staged Firmware Update Unavailable.

TCAFU016: Accessories must communicate their readiness for applying the firmware update by setting the appropriate value for the Update Not Ready Reasons field of the Firmware Update Status for Critical Operation In-Progress.

TCAFU017: Accessories must communicate their readiness for applying the firmware update by setting the appropriate value for the Update Not Ready Reasons field of the Firmware Update Status for all other reasons.

TCAFU018: Once the apply update process has started on the accessory, it must reflect this by changing the Firmware Update State in the Firmware Update Status to Update In-Progress.

TCAFU019: After applying the firmware update, the Firmware Update State will be reset back to Idle regardless of whether it was completed successfully or resulted in a failure.

TCAFU020: Accessories must not allow a firmware image to be downgraded after a successful firmware update.

TCAFU001 **Accessories that update firmware directly in the Home App must have "Firmware Update" service that includes the required characteristics.**

Applies to accessories that support Firmware Updates over UARP. Perform this test case with HAT using the steps below.

Required characteristics:

- Firmware Update Readiness (pr,ev*)
- Firmware Update Status (pr,ev*)

*Notify for BLE encompasses Indicate, Indicate (disconnected), and Broadcast. Please note, characteristics with TLV8 or String formats must not support Broadcast Events.

1. Pair and discover accessory.
2. In the left sidebar of the Controllers window, see each of the accessory's services.
3. Verify that the "Firmware Update" service is present.
4. Verify that the service includes the "Firmware Update Readiness" characteristic.
5. Verify that the characteristic has the permissions "Paired Read", and "Notify".
6. Verify that the service includes the "Firmware Update Status" characteristic.
7. Verify that the characteristic has the permissions "Paired Read", and "Notify".

TCAFU002 **Accessories that support the Firmware Update service must also include the Data Stream Transport Management service.**

Applies to accessories that support Firmware Updates over UARP. Perform this test case with HAT using the steps below.

1. Pair and discover accessory.
2. In the left sidebar of the Controllers window, see each of the accessory's services.
3. Verify required characteristics are included in Data Stream Transport Management service.

TCAFU003 **If the accessory implements the Firmware Update service, verify that the Accessory Information Service's Firmware Revision characteristic has the Notify permission.**

Applies to accessories that support Firmware Updates over UARP. Perform this test case with HAT using the steps below.

1. Pair and discover accessory.
2. In the left sidebar of the Controllers window, see each of the accessory's services.
3. Verify that the Firmware Revision characteristic contains the Notify permission.

TCAFU004 **Accessories must communicate their readiness for staging the firmware update by setting the appropriate value for the Staging Not Ready Reasons field of the Firmware Update Status for Low Battery.**

Applies to accessories that support Firmware Updates over UARP. Applies to accessories that can operate on battery power. Perform this test case with HAT using the steps below.

1. Pair and discover accessory.
2. Enable event notifications on the Firmware Update Readiness characteristic under the Firmware Update service.
3. Read the characteristic and verify that there is no value set for the Staging Not Ready Reasons field in the Events traffic view.
4. Change the accessory state to Low Battery mode.
5. Verify that the accessory sends a notification for the Firmware Update Readiness characteristic with the correct value set for the Staging Not Ready Reasons field: Bit 1 Low Battery.
6. Change the accessory state back to normal.
7. Verify that the accessory sends a notification for the Firmware Update Readiness characteristic with no value set for the Staging Not Ready Reasons field.
8. Verify that the accessory clears the Staging Not Ready Reasons field if power cycle is needed for Step 6.

TCAFU005 **Accessories must communicate their readiness for staging the firmware update by setting the appropriate value for the Staging Not Ready Reasons field of the Firmware Update Status for Low Connectivity.**

Applies to accessories that support Firmware Updates over UARP. Perform this test case with HAT using the steps below.

1. Pair and discover accessory.
2. Enable event notifications on the Firmware Update Readiness characteristic under the Firmware Update service.
3. Read the characteristic and verify that there is no value set for the Staging Not Ready Reasons field in the Events traffic view.
4. Change the accessory state into Low Connectivity.
5. Verify that the accessory sends a notification for the Firmware Update Readiness characteristic with the correct value set for the Staging Not Ready Reasons field: Bit 2 Connectivity.
6. Change the accessory state back to normal.

7. Verify that the accessory sends a notification for the Firmware Update Readiness characteristic with no value set for the Staging Not Ready Reasons field.

TCAFU006 **Accessories must communicate their readiness for staging the firmware update by setting the appropriate value for the Staging Not Ready Reasons field of the Firmware Update Status for all other reasons.**

Applies to accessories that support Firmware Updates over UARP. Perform this test case with HAT using the steps below.

1. Pair and discover accessory.
2. Enable event notifications on the Firmware Update Readiness characteristic under the Firmware Update service.
3. Read the characteristic and verify that there is no value set for the Staging Not Ready Reasons field in the Events traffic view.
4. Change the accessory state to any other state when the accessory is not ready for staging firmware update.
5. Verify that the accessory sends a notification for the Firmware Update Readiness characteristic with the correct value set for the Staging Not Ready Reasons field: Bit 0 Other.
6. Change the accessory state back to normal.
7. Verify that the accessory sends a notification for the Firmware Update Readiness characteristic with no value set for the Staging Not Ready Reasons field.

TCAFU007 **Once the staging process has started, accessories must reflect this by changing the Firmware Update State in the Firmware Update Status to Staging In-Progress.**

Applies to accessories that support Firmware Updates over UARP. Perform this test case with HAT using the steps below.

1. Pair and discover accessory.
2. For HAP over BLE, enable the "Pair Resume Keep Alive Enabled'' checkbox in the Connection panel.
3. Enable event notifications on the Firmware Update Status characteristic under the Firmware Update service.
4. Read the characteristic and verify that the Firmware Update State field is set to "Idle'' in the Events traffic view.
5. Verify that the Staged Firmware Version field is an empty string since there is no staged firmware.
6. In the left sidebar of the Controllers window, select the "Data Stream Transport Management" service.
7. Under the "HomeKit Data Stream" panel, select "Send Start Command".
8. For HAP over BLE and Thread, select "Send Hello after session starts", then select "Start Session" within 10 seconds of sending the Start Command to set up the HDS connection.

9. For IP accessories, select “Send Hello after session starts”, then select “Connect’’ within 10 seconds of sending the Start Command to set up the HDS connection.
10. Select “Establish” to establish the asset transfer session.
11. Under the “Firmware Update Asset Transfer” panel, select “Choose File...”.
12. Select the SuperBinary file to be used for performing the Firmware Update.
13. Select “Start” to begin the staging process.
14. Verify that the staging process has started.
15. Verify that the accessory sends a notification for the Firmware Update Status characteristic with the correct value set for the Firmware Update State field: 1 - Staging In-Progress.
16. Select “Stop” to abort the staging process.
17. Select “Reset” to reset the firmware update process.

TCAFU008 **If the transfer is paused and partially transferred update is retained, the Firmware Update State must change to Staging Paused.**

Applies to accessories that support Firmware Updates over UARP. Perform this test case with HAT using the steps below.

1. Pair and discover accessory.
2. For HAP over BLE, enable the “Pair Resume Keep Alive Enabled’’ checkbox in the Connection panel.
3. Enable event notifications on the Firmware Update Status characteristic under the Firmware Update service.
4. Read the characteristic and verify that the Firmware Update State field is set to “Idle’’ in the Events traffic view.
5. In the left sidebar of the Controllers window, select the “Data Stream Transport Management” service.
6. Under the “HomeKit Data Stream” panel, select “Send Start Command”.
7. For HAP over BLE and Thread, select “Send Hello after session starts”, then select “Start Session” within 10 seconds of sending the Start Command to set up the HDS connection.
8. For IP accessories, select “Send Hello after session starts”, then select “Connect’’ within 10 seconds of sending the Start Command to set up the HDS connection.
9. Select “Establish” to establish the asset transfer session.
10. Under the “Firmware Update Asset Transfer” panel, select “Choose File...”.
11. Select the SuperBinary file to be used for performing the Firmware Update.
12. Select “Start” to begin the staging process.
13. Select “Pause” to halt the staging process.
14. Verify that the staging process has paused.

15. Verify that the accessory sends a notification for the Firmware Update Status characteristic with the correct value set for the Firmware Update State field: 2 - Staging Paused.
16. In the left sidebar of the Controllers window, select the "Data Stream Transport Management" service.
17. Under the "Firmware Update Asset Transfer" panel, select "Resume Staging".
18. Verify that the staging process can be resumed again.
19. Verify that the accessory sends a notification for the Firmware Update Status characteristic with the correct value set for the Firmware Update State field: 1 - Staging In-Progress.
20. Wait for the staging process to complete so the Status changes to Staging completed.
21. Verify that the accessory sends a notification for the Firmware Update Status characteristic with the correct value set for the Firmware Update State field: 3 - Staging Succeeded.
22. In the left sidebar of the Controllers window, select the "Data Stream Transport Management" service.
23. Select "Revoke" to rescind the firmware update offer.
24. Select "Reset" to reset the firmware update process.

TCAFU009 **If a newer version of the SuperBinary is being offered by the secondary Controller while the Accessory is in the middle of staging a SuperBinary from the primary Controller, the Accessory should begin to transfer the newer SuperBinary.**

Applies to accessories that support Firmware Updates over UARP. Perform this test case with HAT using the steps below.

1. Pair and discover accessory (Controller 1).
2. For HAP over BLE, enable the "Pair Resume Keep Alive Enabled'' checkbox in the Connection panel.
3. Select the "+" at the bottom of the left sidebar and create a new Controller 2.
4. In the left sidebar, select Controller 1, then select the accessory.
5. Under the "Add Additional Controllers" panel, select "Controller 2" as the Controller, enable the "Admin'' checkbox, then select the "Add Controller" button.
6. Under Controller 1, enable event notifications on the Firmware Update Status characteristic under the Firmware Update service.
7. Read the characteristic and verify that the Firmware Update State field is set to "Idle'' in the Events traffic view.
8. In the left sidebar, select Controller 1, then select the "Data Stream Transport Management" service.
9. Under the "HomeKit Data Stream" panel, select "Send Start Command".
10. For HAP over BLE and Thread, select "Send Hello after session starts", then select "Start Session" within 10 seconds of sending the Start Command to set up the HDS connection.
11. For IP accessories, select "Send Hello after session starts", then select "Connect'' within 10 seconds of sending the Start Command to set up the HDS connection.

12. Select “Establish” to establish the asset transfer session.
13. Under the “Firmware Update Asset Transfer” panel, select “Choose File...”.
14. Select the SuperBinary file with the lower Firmware version (N+1) for performing the Firmware Update.
15. Select “Start” to begin the staging process.
16. Select “Pause” to halt the staging process.
17. Verify that the staging process has paused.
18. Verify that the accessory sends a notification for the Firmware Update Status characteristic with the correct value set for the Firmware Update State field: 2 - Staging Paused.
19. For HAP over BLE, select Controller 1 in the left sidebar, select the accessory, then select “Disconnect”.
20. In the left sidebar, select Controller 2, select the accessory, then select “Discover”.
21. For HAP over BLE, enable the “Pair Resume Keep Alive Enabled’’ checkbox in the Connection panel.
22. Enable event notifications on the Firmware Update Status characteristic under the Firmware Update service.
23. Repeat steps 8-13 to set up the HDS connection and begin the firmware update with Controller 2.
24. Select the SuperBinary file with a higher firmware version (N+2) than Step 14.
25. Select “Start” to begin the staging process.
26. Verify that the accessory is able to begin to transfer the newer SuperBinary.
27. Verify that the accessory sends a notification for the Firmware Update Status characteristic with the correct value set for the Firmware Update State field: 1 - Staging In-Progress.
28. Wait for the staging process to complete so the Status changes to Staging completed.
29. Verify that the accessory sends a notification for the Firmware Update Status characteristic with the correct value set for the Firmware Update State field: 3 - Staging Succeeded.
30. Verify that the Staged Firmware Version field reflects the firmware version of the higher-version SuperBinary file used in Step 24.
31. In the left sidebar of the Controllers window, select the “Data Stream Transport Management” service.
32. Select “Revoke” to rescind the firmware update offer.
33. Select “Reset” to reset the firmware update process.

TCAFU010 **If a lower version of the SuperBinary is being offered by the secondary Controller while the Accessory is in the middle of staging a SuperBinary from the primary Controller, the Platform Accessory should deny the secondary offer.**

Applies to accessories that support Firmware Updates over UARP. Perform this test case with HAT using the steps below.

1. Pair and discover accessory (Controller 1).

2. For HAP over BLE, enable the "Pair Resume Keep Alive Enabled'' checkbox in the Connection panel.
3. Select the "+" at the bottom of the left sidebar and create a new Controller 2.
4. In the left sidebar, select Controller 1, then select the accessory.
5. Under the "Add Additional Controllers" panel, select "Controller 2" as the Controller, enable the "Admin'' checkbox, then select the "Add Controller" button.
6. Under Controller 1, enable event notifications on the Firmware Update Status characteristic under the Firmware Update service.
7. Read the characteristic and verify that the Firmware Update State field is set to "Idle'' in the Events traffic view
8. In the left sidebar, select Controller 1, then select the "Data Stream Transport Management" service.
9. Under the "HomeKit Data Stream" panel, select "Send Start Command".
10. For HAP over BLE and Thread, select "Send Hello after session starts", then select "Start Session" within 10 seconds of sending the Start Command to set up the HDS connection.
11. For IP accessories, select "Send Hello after session starts", then select "Connect'' within 10 seconds of sending the Start Command to set up the HDS connection.
12. Select "Establish" to establish the asset transfer session.
13. Under the "Firmware Update Asset Transfer" panel, select "Choose File...".
14. Select the SuperBinary file with a higher firmware version (N+2) for performing the Firmware Update.
15. Select "Start" to begin the staging process.
16. Select "Pause" to halt the staging process.
17. Verify that the staging process has paused.
18. Verify that the accessory sends a notification for the Firmware Update Status characteristic with the correct value set for the Firmware Update State field: 2 - Staging Paused.
19. For HAP over BLE, select Controller 1 in the left sidebar, select the accessory, then select "Disconnect''.
20. In the left sidebar, select Controller 2, select the accessory, then select "Discover''.
21. For HAP over BLE, enable the "Pair Resume Keep Alive Enabled'' checkbox in the Connection panel.
22. Enable event notifications on the Firmware Update Status characteristic under the Firmware Update service.
23. Repeat steps 8-13 to set up the HDS connection and begin the firmware update with Controller 2.
24. Select the SuperBinary file with a lower firmware version (N+1) than Step 14.
25. Select "Start" to begin the staging process.
26. Verify that the staging process fails and the accessory denies the secondary offer of a lower-version SuperBinary file.
27. For HAP over BLE, select Controller 2 in the left sidebar, select the accessory, then select "Disconnect''.

28. In the left sidebar, select the accessory under Controller 1.
29. For HAP over BLE, select "Pair Resume" in the Connection panel and then select "Discover".
30. For HAP over BLE, enable event notifications on the Firmware Update Status characteristic under the Firmware Update service.
31. In the left sidebar, select Controller 1, then select the "Data Stream Transport Management" service.
32. Under the "Firmware Update Asset Transfer" panel, select "Resume Staging".
33. Verify that the staging process can be resumed again.
34. Verify that the accessory sends a notification for the Firmware Update Status characteristic with the correct value set for the Firmware Update State field: 1 - Staging In-Progress.
35. Wait for the staging process to complete so the Status changes to Staging completed.
36. Verify that the accessory sends a notification for the Firmware Update Status characteristic with the correct value set for the Firmware Update State field: 3 - Staging Succeeded.
37. Verify that the Staged Firmware Version field reflects the firmware version of the higher-version SuperBinary file used in Step 14.
38. In the left sidebar of the Controllers window, select the "Data Stream Transport Management" service.
39. Select "Revoke" to rescind the firmware update offer.
40. Select "Reset" to reset the firmware update process.

TCAFU011 **If the the Staging process is paused before it has completed and the SuperBinary cannot be retained, the "Firmware Update State" characteristic must be set to "Idle".**

Applies to accessories that support Firmware Updates over UARP. Perform this test case with HAT using the steps below.

1. Pair and discover accessory.
2. For HAP over BLE, enable the "Pair Resume Keep Alive Enabled'' checkbox in the Connection panel.
3. Enable event notifications on the Firmware Update Status characteristic under the Firmware Update service.
4. Read the characteristic and verify that the Firmware Update State field is set to "Idle'' in the Events traffic view.
5. In the left sidebar of the Controllers window, select the "Data Stream Transport Management" service.
6. Under the "HomeKit Data Stream" panel, select "Send Start Command".
7. For HAP over BLE and Thread, select "Send Hello after session starts", then select "Start Session" within 10 seconds of sending the Start Command to set up the HDS connection.
8. For IP accessories, select "Send Hello after session starts", then select "Connect'' within 10 seconds of sending the Start Command to set up the HDS connection.
9. Select "Establish" to establish the asset transfer session.

10. Under the “Firmware Update Asset Transfer” panel, select “Choose File...”.
11. Select the SuperBinary file to be used for performing the Firmware Update.
12. Select “Start” to begin the staging process.
13. Select “Pause” to halt the staging process.
14. Verify that the accessory sends a notification for the Firmware Update Status characteristic with the correct value set for the Firmware Update State field: 0 - Idle.
15. In the left sidebar of the Controllers window, select the “Data Stream Transport Management” service.
16. Select “Reset” to reset the firmware update process.

TCAFU012 **Once the firmware update is staged successfully on the accessory, the Firmware Update State must change to Staging Succeeded.**

Applies to accessories that support Firmware Updates over UARP. Perform this test case with HAT using the steps below.

1. Pair and discover accessory.
2. For HAP over BLE, enable the “Pair Resume Keep Alive Enabled’’ checkbox in the Connection panel.
3. Enable event notifications on the Firmware Update Status characteristic under the Firmware Update service.
4. Read the characteristic and verify that the Firmware Update State field is set to “Idle’’ in the Events traffic view.
5. Verify that the Staged Firmware Version field is an empty string since there is no staged firmware.
6. In the left sidebar of the Controllers window, select the “Data Stream Transport Management” service.
7. Under the “HomeKit Data Stream” panel, select “Send Start Command”.
8. For HAP over BLE and Thread, select “Send Hello after session starts”, then select “Start Session” within 10 seconds of sending the Start Command to set up the HDS connection.
9. For IP accessories, select “Send Hello after session starts”, then select “Connect’’ within 10 seconds of sending the Start Command to set up the HDS connection.
10. Select “Establish” to establish the asset transfer session.
11. Under the “Firmware Update Asset Transfer” panel, select “Choose File...”.
12. Select the SuperBinary file to be used for performing the Firmware Update.
13. Select “Start” to begin the staging process.
14. Wait for the staging process to complete so the Status changes to Staging completed.
15. Verify that the accessory sends a notification for the Firmware Update Status characteristic with the correct value set for the Firmware Update State field: 3 - Staging Succeeded.
16. Verify that the Staged Firmware Version field reflects the version of the currently staged firmware.

17. Verify that the string conforms to the same format that is required by the Firmware Revision characteristic.
18. In the left sidebar of the Controllers window, select the "Data Stream Transport Management" service.
19. Select "Revoke" to rescind the firmware update offer.
20. Select "Reset" to reset the firmware update process.

TCAFU013 **Accessories must begin to transfer the newer version of SuperBinary even if they already have another lower version completely staged.**

Applies to accessories that support Firmware Updates over UARP. Perform this test case with HAT using the steps below.

1. Pair and discover accessory.
2. For HAP over BLE, enable the "Pair Resume Keep Alive Enabled'' checkbox in the Connection panel.
3. Enable event notifications on the Firmware Update Status characteristic under the Firmware Update service.
4. Read the characteristic and verify that the Firmware Update State field is set to "Idle'' in the Events traffic view.
5. Verify that the Staged Firmware Version field is an empty string since there is no staged firmware.
6. In the left sidebar of the Controllers window, select the "Data Stream Transport Management" service.
7. Under the "HomeKit Data Stream" panel, select "Send Start Command".
8. For HAP over BLE and Thread, select "Send Hello after session starts", then select "Start Session" within 10 seconds of sending the Start Command to set up the HDS connection.
9. For IP accessories, select "Send Hello after session starts", then select "Connect'' within 10 seconds of sending the Start Command to set up the HDS connection.
10. Select "Establish" to establish the asset transfer session.
11. Under the "Firmware Update Asset Transfer" panel, select "Choose File...".
12. Select the lower version of SuperBinary file (N+1) to be used for performing the Firmware Update.
13. Select "Start" to begin the staging process.
14. Wait for the staging process to complete so the Status changes to Staging completed.
15. Verify that the accessory sends a notification for the Firmware Update Status characteristic with the correct value set for the Firmware Update State field: 3 - Staging Succeeded.
16. Select "Reset" to reset the firmware update process.
17. Select "Establish" to establish the asset transfer session.
18. Verify that the Staged Firmware Version field reflects the version of the currently staged firmware.
19. In the left sidebar of the Controllers window, select the "Data Stream Transport Management" service.

20. Under the “Firmware Update Asset Transfer” panel, select “Choose File...”.
21. Select the SuperBinary file with a higher firmware version (N+2) than Step 12 to be used for performing the Firmware Update.
22. Select “Start” to begin the staging process.
23. Verify that the accessory is able to begin to transfer the newer SuperBinary.
24. Wait for the staging process to complete so the Status changes to Staging completed.
25. Verify that the accessory sends a notification for the Firmware Update Status characteristic with the correct value set for the Firmware Update State field: 3 - Staging Succeeded.
26. Verify that the Staged Firmware Version field reflects the firmware version of the higher-version SuperBinary file used in Step 21.
27. In the left sidebar of the Controllers window, select the “Data Stream Transport Management” service.
28. Select “Revoke” to rescind the firmware update offer.
29. Select “Reset” to reset the firmware update process.

TCAFU014 **Accessories must communicate their readiness for applying the firmware update by setting the appropriate value for the Update Not Ready Reasons field of the Firmware Update Status for Low Battery.**

Applies to accessories that support Firmware Updates over UARP. Applies to accessories that can operate on battery power. Perform this test case with HAT using the steps below.

1. Pair and discover accessory.
2. For HAP over BLE, enable the “Pair Resume Keep Alive Enabled’’ checkbox in the Connection panel.
3. Enable event notifications on the Firmware Update Status characteristic under the Firmware Update service.
4. Read the characteristic and verify that the Firmware Update State field is set to “Idle’’ in the Events traffic view.
5. In the left sidebar of the Controllers window, select the “Data Stream Transport Management” service.
6. Under the “HomeKit Data Stream” panel, select “Send Start Command”.
7. For HAP over BLE and Thread, select “Send Hello after session starts”, then select “Start Session” within 10 seconds of sending the Start Command to set up the HDS connection.
8. For IP accessories, select “Send Hello after session starts”, then select “Connect’’ within 10 seconds of sending the Start Command to set up the HDS connection.
9. Select “Establish” to establish the asset transfer session.
10. Under the “Firmware Update Asset Transfer” panel, select “Choose File...”.
11. Select the SuperBinary file to be used for performing the Firmware Update.
12. Select “Start” to begin the staging process.

13. Wait for the staging process to complete so the Status changes to Staging completed.
14. Enable event notifications on the Firmware Update Readiness characteristic under the Firmware Update service.
15. Read the characteristic and verify that there is no value set for the Update Not Ready Reasons field.
16. Change the accessory state to Low Battery mode.
17. Verify that the accessory sends a notification for the Firmware Update Readiness characteristic with the correct value set for the Update Not Ready Reasons field: Bit 1 Low Battery.
18. Change the accessory state back to normal.
19. Verify that the accessory sends a notification for the Firmware Update Readiness characteristic with no value set for the Update Not Ready Reasons field.
20. Verify that the accessory clears the Update Not Ready Reasons field if power cycle is needed for Step 18.
21. In the left sidebar of the Controllers window, select the "Data Stream Transport Management" service.
22. Select "Revoke" to rescind the firmware update offer.
23. Select "Reset" to reset the firmware update process.

TCAFU015 **Accessories must communicate their readiness for applying the firmware update by setting the appropriate value for the Update Not Ready Reasons field of the Firmware Update Status for Staged Firmware Update Unavailable.**

Applies to accessories that support Firmware Updates over UARP. Perform this test case with HAT using the steps below.

1. Pair and discover accessory.
2. For HAP over BLE, enable the "Pair Resume Keep Alive Enabled'' checkbox in the Connection panel.
3. Enable event notifications on the Firmware Update Status characteristic under the Firmware Update service.
4. Read the characteristic and verify that the Firmware Update State field is set to "Idle'' in the Events traffic view.
5. In the left sidebar of the Controllers window, select the "Data Stream Transport Management" service.
6. Under the "HomeKit Data Stream" panel, select "Send Start Command".
7. For HAP over BLE and Thread, select "Send Hello after session starts", then select "Start Session" within 10 seconds of sending the Start Command to set up the HDS connection.
8. For IP accessories, select "Send Hello after session starts", then select "Connect'' within 10 seconds of sending the Start Command to set up the HDS connection.
9. Select "Establish" to establish the asset transfer session.
10. Under the "Firmware Update Asset Transfer" panel, select "Choose File...".

11. Select the SuperBinary file to be used for performing the Firmware Update.
12. Select "Start" to begin the staging process.
13. Wait for the staging process to complete so the Status changes to Staging completed.
14. Enable event notifications on the Firmware Update Readiness characteristic under the Firmware Update service.
15. Read the characteristic and verify that there is no value set for the Update Not Ready Reasons field.
16. Change the accessory state to Staged Firmware Update Unavailable.
17. Verify that the accessory sends a notification for the Firmware Update Readiness characteristic with the correct value set for the Update Not Ready Reasons field: Bit 2 Staged Firmware Update Unavailable.
18. Change the accessory state back to normal.
19. Verify that the accessory sends a notification for the Firmware Update Readiness characteristic with no value set for the Update Not Ready Reasons field.
20. In the left sidebar of the Controllers window, select the "Data Stream Transport Management" service.
21. Select "Revoke" to rescind the firmware update offer.
22. Select "Reset" to reset the firmware update process.

TCAFU016 **Accessories must communicate their readiness for applying the firmware update by setting the appropriate value for the Update Not Ready Reasons field of the Firmware Update Status for Critical Operation In-Progress.**

Applies to accessories that support Firmware Updates over UARP. Perform this test case with HAT using the steps below.

1. Pair and discover accessory.
2. For HAP over BLE, enable the "Pair Resume Keep Alive Enabled'' checkbox in the Connection panel.
3. Enable event notifications on the Firmware Update Status characteristic under the Firmware Update service.
4. Read the characteristic and verify that the Firmware Update State field is set to "Idle'' in the Events traffic view.
5. In the left sidebar of the Controllers window, select the "Data Stream Transport Management" service.
6. Under the "HomeKit Data Stream" panel, select "Send Start Command".
7. For HAP over BLE and Thread, select "Send Hello after session starts", then select "Start Session" within 10 seconds of sending the Start Command to set up the HDS connection.
8. For IP accessories, select "Send Hello after session starts", then select "Connect'' within 10 seconds of sending the Start Command to set up the HDS connection.
9. Select "Establish" to establish the asset transfer session.

10. Under the “Firmware Update Asset Transfer” panel, select “Choose File...”.
11. Select the SuperBinary file to be used for performing the Firmware Update.
12. Select “Start” to begin the staging process.
13. Wait for the staging process to complete so the Status changes to Staging completed.
14. Enable event notifications on the Firmware Update Readiness characteristic under the Firmware Update service.
15. Read the characteristic and verify that there is no value set for the Update Not Ready Reasons field.
16. Change the accessory state to Critical Operation In-Progress.
17. Verify that the accessory sends a notification for the Firmware Update Readiness characteristic with the correct value set for the Update Not Ready Reasons field: Bit 3 Critical Operation In-Progress.
18. Change the accessory state back to normal.
19. Verify that the accessory sends a notification for the Firmware Update Readiness characteristic with no value set for the Update Not Ready Reasons field.
20. In the left sidebar of the Controllers window, select the “Data Stream Transport Management” service.
21. Select “Revoke” to rescind the firmware update offer.
22. Select “Reset” to reset the firmware update process.

TCAFU017 **Accessories must communicate their readiness for applying the firmware update by setting the appropriate value for the Update Not Ready Reasons field of the Firmware Update Status for all other reasons.**

Applies to accessories that support Firmware Updates over UARP. Perform this test case with HAT using the steps below.

1. Pair and discover accessory.
2. For HAP over BLE, enable the “Pair Resume Keep Alive Enabled’’ checkbox in the Connection panel.
3. Enable event notifications on the Firmware Update Status characteristic under the Firmware Update service.
4. Read the characteristic and verify that the Firmware Update State field is set to “Idle’’ in the Events traffic view.
5. In the left sidebar of the Controllers window, select the “Data Stream Transport Management” service.
6. Under the “HomeKit Data Stream” panel, select “Send Start Command”.
7. For HAP over BLE and Thread, select “Send Hello after session starts”, then select “Start Session” within 10 seconds of sending the Start Command to set up the HDS connection.
8. For IP accessories, select “Send Hello after session starts”, then select “Connect’’ within 10 seconds of sending the Start Command to set up the HDS connection.
9. Select “Establish” to establish the asset transfer session.

10. Under the “Firmware Update Asset Transfer” panel, select “Choose File...”.
11. Select the SuperBinary file to be used for performing the Firmware Update.
12. Select “Start” to begin the staging process.
13. Wait for the staging process to complete so the Status changes to Staging completed.
14. Enable event notifications on the Firmware Update Readiness characteristic under the Firmware Update service.
15. Read the characteristic and verify that there is no value set for the Update Not Ready Reasons field.
16. Change the accessory state to any other state when the accessory is not ready for applying the firmware update.
17. Verify that the accessory sends a notification for the Firmware Update Readiness characteristic with the correct value set for the Update Not Ready Reasons field: Bit 0 Other.
18. Change the accessory state back to normal.
19. Verify that the accessory sends a notification for the Firmware Update Readiness characteristic with no value set for the Update Not Ready Reasons field.
20. In the left sidebar of the Controllers window, select the “Data Stream Transport Management” service.
21. Select “Revoke” to rescind the firmware update offer.
22. Select “Reset” to reset the firmware update process.

TCAFU018 **Once the apply update process has started on the accessory, it must reflect this by changing the Firmware Update State in the Firmware Update Status to Update In-Progress.**

Applies to accessories that support Firmware Updates over UARP. Perform this test case with HAT using the steps below.

1. Pair and discover accessory.
2. For HAP over BLE, enable the “Pair Resume Keep Alive Enabled’’ checkbox in the Connection panel.
3. Enable event notifications on the Firmware Update Status characteristic under the Firmware Update service.
4. Read the characteristic and verify that the Firmware Update State field is set to “Idle’’ in the Events traffic view.
5. In the left sidebar of the Controllers window, select the “Data Stream Transport Management” service.
6. Under the “HomeKit Data Stream” panel, select “Send Start Command”.
7. For HAP over BLE and Thread, select “Send Hello after session starts”, then select “Start Session” within 10 seconds of sending the Start Command to set up the HDS connection.
8. For IP accessories, select “Send Hello after session starts”, then select “Connect’’ within 10 seconds of sending the Start Command to set up the HDS connection.
9. Select “Establish” to establish the asset transfer session.

10. Under the “Firmware Update Asset Transfer” panel, select “Choose File...”.
11. Select the SuperBinary file with a lower Firmware version (N+1) for performing the Firmware Update.
12. Select “Start” to begin the staging process.
13. Wait for the staging process to complete so the Status changes to Staging completed.
14. Verify that the accessory sends a notification for the Firmware Update Status characteristic with the correct value set for the Firmware Update State field: 3 - Staging Succeeded.
15. Under the “Firmware Update Asset Transfer” panel, select “Apply”.
16. Verify that the accessory sends a notification for the Firmware Update Status characteristic with the correct value set for the Firmware Update State field: 4 - Update In-Progress.

TCAFU019 **After applying the firmware update, the Firmware Update State will be reset back to Idle regardless of whether it was completed successfully or resulted in a failure.**

Applies to accessories that support Firmware Updates over UARP. Perform this test case with HAT using the steps below.

1. Pair and discover accessory.
2. For HAP over BLE, enable the ‘‘Pair Resume Keep Alive Enabled’’ checkbox in the Connection panel.
3. Enable event notifications on the Firmware Revision characteristic under the Accessory Information service.
4. Read the characteristic and Verify that the value must be set to the current firmware version of the accessory.
5. Enable event notifications on the Firmware Update Status characteristic under the Firmware Update service.
6. Read the characteristic and verify that the Firmware Update State field is set to ‘‘Idle’’ in the Events traffic view.
7. Verify that the Staged Firmware Version field is an empty string since there is no staged firmware.
8. In the left sidebar of the Controllers window, select the “Data Stream Transport Management” service.
9. Under the “HomeKit Data Stream” panel, select “Send Start Command”.
10. For HAP over BLE and Thread, select “Send Hello after session starts”, then select “Start Session” within 10 seconds of sending the Start Command to set up the HDS connection.
11. For IP accessories, select “Send Hello after session starts”, then select “Connect’’ within 10 seconds of sending the Start Command to set up the HDS connection.
12. Select “Establish” to establish the asset transfer session.
13. Under the “Firmware Update Asset Transfer” panel, select “Choose File...”.
14. Select the SuperBinary file with a higher firmware version (N+2) for performing the Firmware Update.
15. Select “Start” to begin the staging process.

16. Wait for the staging process to complete so the Status changes to Staging completed.
17. Verify that the accessory sends a notification for the Firmware Update Status characteristic with the correct value set for the Firmware Update State field: 3 - Staging Succeeded.
18. Under the "Firmware Update Asset Transfer" panel, select "Apply".
19. Start a timer from the point in time when the accessory is no longer reachable.
20. Stop the timer to the point where the accessory is available for communication again.
21. Verify that the accessory sends a notification for the Firmware Update Status characteristic with the correct value set for the Firmware Update State field: 0 - Idle.
22. Verify that the time in seconds that the accessory was unresponsive while the firmware update was being applied is less than the Update Duration field.
23. Verify that the Staged Firmware Version field resets to an empty string after the staged firmware is applied.
24. Verify that the accessory sends a notification for the Firmware Revision characteristic with the value set to the recently applied firmware version.

TCAFU020 **Accessories must not allow a firmware image to be downgraded after a successful firmware update.**

Applies to accessories that support Firmware Updates over UARP. Perform this test case with HAT using the steps below.

1. Pair and discover accessory.
2. For HAP over BLE, enable the "Pair Resume Keep Alive Enabled'' checkbox in the Connection panel.
3. Enable event notifications on the Firmware Update Status characteristic under the Firmware Update service.
4. Read the characteristic and verify that the Firmware Update State field is set to "Idle'' in the Events traffic view.
5. In the left sidebar of the Controllers window, select the "Data Stream Transport Management" service.
6. Under the "HomeKit Data Stream" panel, select "Send Start Command".
7. For HAP over BLE and Thread, select "Send Hello after session starts", then select "Start Session" within 10 seconds of sending the Start Command to set up the HDS connection.
8. For IP accessories, select "Send Hello after session starts", then select "Connect'' within 10 seconds of sending the Start Command to set up the HDS connection.
9. Select "Establish" to establish the asset transfer session.
10. Under the "Firmware Update Asset Transfer" panel, select "Choose File...".
11. Select the SuperBinary file (N+2) to be used for performing the Firmware Update.
12. Select "Start" to begin the staging process.
13. Wait for the staging process to complete so the Status changes to Staging completed.

14. Verify that the accessory sends a notification for the Firmware Update Status characteristic with the correct value set for the Firmware Update State field: 3 - Staging Succeeded.
15. Under the "Firmware Update Asset Transfer" panel, select "Apply".
16. Wait for the apply process to complete so the Status changes to Apply firmware update succeeded.
17. Under the "Firmware Update" panel, select "Reset" to reset the firmware update process.
18. Discover the accessory under the Controller.
19. Under the "HomeKit Data Stream" panel, select "Send Start Command".
20. For HAP over BLE and Thread, select "Send Hello after session starts", then select "Start Session" within 10 seconds of sending the Start Command to set up the HDS connection.
21. For IP accessories, select "Send Hello after session starts", then select "Connect'' within 10 seconds of sending the Start Command to set up the HDS connection.
22. Select "Establish" to establish the asset transfer session.
23. Under the "Firmware Update Asset Transfer" panel, select "Choose File...".
24. Select the SuperBinary file with a lower firmware version (N+1) than Step 11.
25. Select "Start" to begin the staging process.
26. Verify that the staging process fails and the accessory denies the firmware image to be downgraded.

Chapter 2

Reliability Test Cases

2.1 Stress

TCS001: Discovery + Pair Setup + Accessory Deletion (IP and BLE).

TCS002: Discovery + Pair Setup/Add Pairings + Accessory Deletion/Remove Pairings.

TCS003: Pair Verify + Read/Write Reliability.

TCS004: Pair Verify + Multiple Characteristic Write Reliability (IP Only).

TCS005: Pair Verify + Read Reliability.

TCS006: Pair Verify + Multiple Characteristic Read Reliability (IP Only).

TCS007: Cold Characteristic Write Reliability (BLE Only).

TCS008: Cold Multiple Characteristic Write Reliability (BLE Only).

TCS009: Cold Characteristic Read Reliability (BLE Only).

TCS010: Cold Multiple Characteristic Read Reliability (BLE Only).

TCS011: Pair Verify + Add Pairing, Remove Pairing Reliability (IP and BLE).

TCS012: For IP Camera accessories - Audio and Video stream initiate.

TCS013: For IP Camera accessories - Audio and Video stream initiate.

TCS014: For IP Camera accessories - Audio and Video stream initiate.

TCS015: For IP Camera accessories - Audio and Video stream initiate.

TCS016: For IP Camera accessories - Audio and video are synced after 1 hour.

TCS017: Verify accessory is still functional after sitting idle for 24 hours.

TCS018: Associate the maximum number of supported bridged programmable switch accessories to the bridge, configure bridged programmable switches, and verify successful automation execution.

TCS019: Negotiate, start, and then stop IP camera stream 2,000 times.

TCS020: Accessory must be able to boot and work with HomeKit without Internet access. If Internet access is blocked for an accessory it can rely on the presence of a local NTP server advertised via DHCP.

TCS001 **Discovery + Pair Setup + Accessory Deletion (IP and BLE).**

Applies to all accessories. Applies to accessories using the HAP over Thread transport. For HAP over Wi-Fi or Ethernet accessories, perform this test case automatically with HCA. For HAP over BLE and Thread accessories, perform this test case manually with HAT using the steps below.

1. Perform Pair Setup with accessory.
2. Ensure that the initial Pair Verify and the parsing of accessory's attribute database are successful.
3. Remove the admin pairing using the controller.
4. This procedure should be repeated for 100 iterations, with each iteration starting as soon as accessory is discovered broadcasting as unpaired.

TCS002 **Discovery + Pair Setup/Add Pairings + Accessory Deletion/Remove Pairings.**

Applies to all accessories. Applies to accessories using the HAP over Thread transport. For HAP over Wi-Fi or Ethernet accessories, perform this test case automatically with HCA. For HAP over BLE and Thread accessories, perform this test case manually with HAT using the steps below.

1. Pair and discover accessory with Controller 1.
2. In Controllers window, select "+" to create a new IP/BLE/Thread Controller 2.
3. Repeat step 2 to create 3 controllers.
4. Using Controller 1, select the accessory name. In the "Add Additional Controllers" pane, select Controller 2 as Controller, ensure the checkbox for Admin is disabled, and select the "Add Controller" button.
5. Repeat step 4 to add pairings for Controller 3.
6. On the left pane of the Controllers window, select the accessory name under Controller 2, select the "Start" button, and select the "Discover" button.
7. Verify that the Pair-Verify and Discover operations complete successfully.
8. On the left pane of the Controllers window, select the accessory name under Controller 3, select the "Start" button, and select the "Discover" button.
9. Verify that the Pair-Verify and Discover operations complete successfully.
10. Using Controller 1, select the "Remove Pairing" button.
11. Repeat steps 1-10 99 times.

TCS003 **Pair Verify + Read/Write Reliability.**

Applies to all accessories. Perform this test case using HCA.

1. Pair verify with accessory.
2. Perform 50,000 read/write operations to each writable characteristic containing the "is user interactive" property.
3. Verify the number of successful read/write operations are above 99.9%.

TCS004 **Pair Verify + Multiple Characteristic Write Reliability (IP Only).**

Applies to accessories that use HAP over Ethernet or Wi-Fi.

1. Iterate 100 times performing a Pair Verify followed by multiple characteristic writes to accessory.
2. Utilize all writable characteristics that provide "user interactive" functionality.
3. Writes should be exercised one right after the other.
4. Between each iteration, the IP connection should be cleanly torn down by the controller.

TCS005 **Pair Verify + Read Reliability.**

Applies to all accessories. For HAP over Wi-Fi or Ethernet accessories, perform this test case automatically with HCA. For HAP over BLE and Thread accessories, perform this test case manually with HAT using the steps below.

1. Pair Verify with accessory. Perform 100 read operations on each readable characteristic that provide "user interactive" functionality.

TCS006 **Pair Verify + Multiple Characteristic Read Reliability (IP Only).**

Applies to accessories that use HAP over Ethernet or Wi-Fi.

1. Iterate 100 times performing a Pair Verify followed by multiple characteristic reads to accessory.
2. Utilize all readable characteristics that provide "user interactive" functionality.
3. Reads should be exercised one right after the other.
4. Between each iteration, the IP connection should be cleanly torn down by the controller.

TCS007 **Cold Characteristic Write Reliability (BLE Only).**

Applies to accessories that use HAP over BLE.

1. Beginning without an active Bluetooth session, perform 100 write operations on each writable characteristic that provide "user interactive" functionality.

2. Between each iteration, the Bluetooth LE connection should be torn down by the controller.

TCS008 **Cold Multiple Characteristic Write Reliability (BLE Only).**

Applies to accessories that use HAP over BLE.

1. Iterate 100 times performing Bluetooth LE multiple characteristic writes to accessory.
2. Begin each test without an active Bluetooth session.
3. Utilize all writable characteristics that provide "user interactive" functionality.
4. Writes should be exercised one right after the other once Pair Verify establishes a connection.
5. Between each iteration, the Bluetooth LE connection should be torn down by the controller.

TCS009 **Cold Characteristic Read Reliability (BLE Only).**

Applies to accessories that use HAP over BLE.

1. Beginning without an active Bluetooth session, perform 100 read operations on each readable characteristic that provide "user interactive" functionality.
2. Between each iteration, the Bluetooth LE connection should be torn down by the controller.

TCS010 **Cold Multiple Characteristic Read Reliability (BLE Only).**

Applies to accessories that use HAP over BLE.

1. Iterate 100 times performing Bluetooth LE multiple characteristic reads to accessory.
2. Begin each test without an active Bluetooth session.
3. Utilize all readable characteristics containing the "is user interactive" property.
4. Reads should be exercised one right after the other once Pair Verify establishes a connection.
5. Between each iteration, the Bluetooth LE connection should be torn down by the controller.

TCS011 **Pair Verify + Add Pairing, Remove Pairing Reliability (IP and BLE).**

Applies to all accessories.

1. Iterate 100 times performing a Pair Verify followed by an add pairing operation, followed by a remove pairing operation.
2. Between each iteration, the IP or Bluetooth LE connection should be cleanly torn down by the controller.

TCS012 **For IP Camera accessories - Audio and Video stream initiate.**

Applies to IP camera accessories.

1. Set the encryption level to AES-128.
2. Initiate an IP-Camera stream.
3. Initiate 2-way audio.
4. Verify that the IP camera stream and 2-way audio stream are setup successfully and sustained for several minutes.

TCS013 **For IP Camera accessories - Audio and Video stream initiate.**

Applies to IP camera accessories.

1. Set the encryption level to AES-256.
2. Initiate an IP-Camera stream.
3. Initiate 2-way audio.
4. Verify that the IP camera stream and 2-way audio stream are setup successfully and sustained for several minutes.

TCS014 **For IP Camera accessories - Audio and Video stream initiate.**

Applies to IP camera accessories.

1. Connect the IP-Camera an HAT to a router that provides an IPv4 address.
2. Initiate an IP-Camera stream.
3. Initiate 2-way audio.
4. Verify that the IP camera stream and 2-way audio stream are setup successfully and sustained for several minutes.

TCS015 **For IP Camera accessories - Audio and Video stream initiate.**

Applies to IP camera accessories.

1. Connect the IP-Camera an HAT to a router that provides an IPv6 address only.
2. Initiate an IP-Camera stream.
3. Initiate 2-way audio.
4. Verify that the IP camera stream and 2-way audio stream are setup successfully and sustained for several minutes.

TCS016 **For IP Camera accessories - Audio and video are synced after 1 hour.**

Applies to IP camera accessories.

1. Initiate an IP-Camera stream.
2. Initiate 2-way audio.
3. Let the stream run for 1 hour.
4. Audio and Video quality should be consistent, Audio and Video must be in sync and the streams shouldn't stop throughout the duration of this test.

TCS017 **Verify accessory is still functional after sitting idle for 24 hours.**

Applies to all accessories.

1. Pair and discover the accessory.
2. Read and write to accessory's characteristics.
3. Let the accessory sit idle for 24 hours.
4. Verify reads and writes to accessory's characteristics complete successfully after sitting idle for 24 hours.

For IP cameras: Verify IP camera can stream and take snapshots after sitting idle for 24 hours.

TCS018 **Associate the maximum number of supported bridged programmable switch accessories to the bridge, configure bridged programmable switches, and verify successful automation execution.**

Applies to bridge accessories. Perform this test case with HAT using the steps below.

1. Pair and discover accessory.
2. Associate the maximum number of supported bridged programmable switch accessories to the bridge.
3. Configure each bridged programmable switch.
4. On each bridged accessory, execute each button press type and verify that the correct corresponding event characteristic is sent from the accessory.

TCS019 **Negotiate, start, and then stop IP camera stream 2,000 times.**

Applies to IP camera accessories. Perform this test case using HCA.

1. Pair and discover IP camera.
2. Negotiate and start a stream.
3. Stop the stream.
4. Read the streaming status characteristic, verify it is Available and not Busy.

5. Repeat steps 2-4 2,000 times.

TCS020 **Accessory must be able to boot and work with HomeKit without Internet access. If Internet access is blocked for an accessory it can rely on the presence of a local NTP server advertised via DHCP.**

Applies to accessories that use HAP over Ethernet or Wi-Fi. Perform this test case with HAT using the steps below.

1. Pair and discover accessory with HAT.
2. Perform a read to the accessory's primary functional characteristics and verify the accessory responds.
3. Perform a write to the accessory's primary functional characteristics and verify accessory responds.
4. In the Summary panel, select the "Disconnect" button.
5. Remove the ethernet cable connected to the router's WAN port, disconnect modem, or remove coaxial cable from modem/router combination to disable internet access.
6. After the accessory begins advertising again via Bonjour, select "Discover" button in the Pairing panel.
7. Verify pair-verify completes successfully.
8. Perform a read to the accessory's primary functional characteristics and verify the accessory responds.
9. Perform a write to the accessory's primary functional characteristics and verify accessory responds.
10. Do not restore the internet to the router for at least 24 hours.
11. After 24hrs, with the internet still disabled on the router, perform a read to the accessory's primary functional characteristics and verify the accessory responds.
12. Perform a write to the accessory's primary functional characteristics and verify accessory responds.
13. Restore internet access to the router.
14. Perform a read to the accessory's primary functional characteristics and verify the accessory responds.
15. Perform a write to the accessory's primary functional characteristics and verify accessory responds.

Chapter 3

User Test Cases

3.1 Home app

TCHAA001: User must be able to pair accessory to the home using the Home app.

TCHAA004: If the accessory requires additional authorization and/or additional setup, the Home app must be able to add the accessory to the home after auth data and/or additional setup is complete.

TCHAA001 **User must be able to pair accessory to the home using the Home app.**

Applies to all accessories. Perform this test case using the Home app on iOS.

1. Verify Pair Setup with the accessory is successful using the Home app.

TCHAA004 **If the accessory requires additional authorization and/or additional setup, the Home app must be able to add the accessory to the home after auth data and/or additional setup is complete.**

Applies to all accessories. Applies to accessories that support HomeKit Accessory Protocol specification R15 or earlier. Perform this test case using the Home app on iOS.

1. Perform Pair Setup using the Home app.
2. Use the accessory app to complete Additional Setup.
3. Verify after authorization data has been processed, the Home app is able to read/write to the accessory.

3.2 App for In-Field Provisioning through Software Authentication

TCSWAA001: Accessory app must support updating the accessory to a HomeKit software token-based authentication supported firmware.

TCSWAA002: The accessory app must support retrieving the setup code and passing the setup payload to the Home app to initiate pairing by using the addAndSetupAccessoriesWithPayload API.

TCSWAA003: The accessory app must indicate to the user that iOS 11.3 or newer is required to use a software token-based authentication accessory.

TCSWAA001 **Accessory app must support updating the accessory to a HomeKit software token-based authentication supported firmware.**

Applies to accessories that use software token authentication. Perform this test case using an iOS device running the accessory app.

1. Perform pair-setup with the accessory using the accessory app.
2. Using the accessory app, perform a firmware update.
3. After update process completes, verify that the accessory has the option to be added to HomeKit.

TCSWAA002 **The accessory app must support retrieving the setup code and passing the setup payload to the Home app to initiate pairing by using the addAndSetupAccessoriesWithPayload API.**

Applies to accessories that support in-field provisioning for software token authentication. Perform this test case using an iOS device running the accessory app.

1. Perform pair-setup with the accessory using the accessory app.
2. Perform an accessory firmware update that enables HomeKit integration.
3. After the firmware update is complete, perform pair-setup with the Home app.
4. Verify pair-setup completes successfully and the accessory is present within the Home app.

TCSWAA003 **The accessory app must indicate to the user that iOS 11.3 or newer is required to use a software token-based authentication accessory.**

Applies to accessories that use software token authentication. Perform this test case using an iOS device running the accessory app.

1. Verify the accessory app warns the user that a minimum iOS version of 11.3 is required.

3.3 App with full HomeKit API Support

TCFA001: The accessory must either receive firmware updates automatically from a manufacturer hosted server without user interaction (i.e., pushed by a server), or use the accessory app to receive firmware updates.

TCFA002: The accessory app must be able to add, delete, and re-add the manufacturer's accessories.

TCFA003: After Pair Setup, user must be prompted to add the accessory to an existing room or to create a new room.

TCFA004: After Pair Setup, the user must be prompted to rename any accessory services which the user can interact with.

TCFA005: The accessory app must be able to read/write to the manufacturer's accessory.

TCFA006: The accessory app must handle first launch when their accessory has been paired with a HomeKit application other than their own.

TCFA007: Accessory app must not initiate connections to accessories at a frequency of less than once every 2 minutes, unless the connection is user initiated. Where supported, notifications should be used instead of periodic reads.

TCFA008: Accessory app must support the creating, listing, renaming, and modifying of homes.

TCFA009: Accessory app must support the creating, listing, renaming, and modifying of rooms.

TCFA010: If supported, accessory app must support the creating, listing, renaming, and modifying of service groups.

TCFA011: If supported, accessory apps must allow creation, deletion, and modification of scenes.

TCFA012: Accessory Apps must not create or delete HomeKit objects, such as rooms, zones, action sets, service groups, scenes, etc., without user consent.

TCFA013: Accessory app must support the renaming of all services which the user may interact that contain Apple-defined characteristics.

TCFA014: Accessory apps must not allow shared users to edit home configurations.

TCFA015: Accessory app must remain stable when a user's home configuration utilizes the full HomeKit feature set.

TCFA016: If an accessory supports additional authorization data, confirm the accessory app properly sets additional authorization data (e.g for apps that interact with Security Class characteristics, such as lock-related apps).

TCFA017: Accessory app must not allow users to delete homes.

TCFA018: Accessory apps that ask users to create an account must provide an option for users to defer this step until HomeKit setup is complete and accessory is usable. Accessory apps must not require an account to receive firmware updates.

TCFA019: Accessory app must not change values (e.g., turn off a light bulb) without user intent.

TCFA020: A accessory app that only supports iOS 10 or later must not use the External Accessory framework to browse for and/or configure unpaired accessories. Instead, the addAndSetupAccessories API must be used.

TCFA021: Accessory app must support identifying a HomeKit accessory that is part of a home using the identify API.

TCFA022: If the accessory accessory requires additional setup prior to being paired with the Home app, the Home app must be able to pair with that accessory after the additional setup is complete.

TCFA023: Accessory app must allow user configuration of "Associated Service Type" for outlets and switches using the updateAssociated - ServiceType API.

TCFA024: If applicable, the accessory app must support adding endpoints to a bridge that does not have a physical interface for adding endpoints.

TCFA001 **The accessory must either receive firmware updates automatically from a manufacturer hosted server without user interaction (i.e., pushed by a server), or use the accessory app to receive firmware updates.**

Applies to all accessories. Perform this test case using an iOS device running the accessory app.

1. Verify that firmware updates are either (1) pushed over-the-air from the manufacturer and do not require user interaction or (2) accessory app provides a way to update the firmware of the accessory.

TCFA002 **The accessory app must be able to add, delete, and re-add the manufacturer's accessories.**

Applies to all accessories. Perform this test case using an iOS device running the accessory app.

1. Perform Pair Setup with the accessory using the accessory app.
2. Open the Home app and verify the accessory is displayed.
3. Remove the accessory from the home using the accessory app.
4. Open the Home app and verify the accessory was removed from the home.
5. Pair with the accessory again using the accessory app.
6. Open the Home app and verify the accessory has been successfully added to the home.

TCFA003 **After Pair Setup, user must be prompted to add the accessory to an existing room or to create a new room.**

Applies to all accessories. Perform this test case using an iOS device running the accessory app.

1. After Pair Setup completes, verify user is presented with options to add the accessory to an existing room or to create a new room.

TCFA004 **After Pair Setup, the user must be prompted to rename any accessory services which the user can interact with.**

Applies to all accessories. Perform this test case using an iOS device running the accessory app.

1. Verify after Pair Setup completes, user is presented with the option to rename any accessory services that the user can interact with.
2. There should also be customer-facing information indicating that this name will be recognized by Siri.
3. Ensure that the user is prompted to rename any service the user interacts with.

TCFA005 **The accessory app must be able to read/write to the manufacturer's accessory.**

Applies to all accessories. Perform this test case using an iOS device running the accessory app.

1. Perform Pair Setup with the accessory using the accessory app.
2. Verify that the accessory app can control the accessory using all of the Apple Defined Services the accessory supports (switches, toggles, sliders, and read only fields as appropriate).

TCFA006 **The accessory app must handle first launch when their accessory has been paired with a HomeKit application other than their own.**

Applies to all accessories. Perform this test case using an iOS device running the accessory app.

1. Perform Pair Setup using the Home app.
2. Launch the accessory app (first launch).
3. Ensure there are no issues or crashes experienced while using the accessory app and accessory.

TCFA007 **Accessory app must not initiate connections to accessories at a frequency of less than once every 2 minutes, unless the connection is user initiated. Where supported, notifications should be used instead of periodic reads.**

Applies to all accessories. Perform this test case using an iOS device running the accessory app.

1. On the iOS device, navigate to Settings, Display & Brightness.
2. Tap "Auto-Lock".
3. Choose "3 Minutes" from the "Auto-Lock" page.
4. Download the "Home App/HomeKit for iOS" profile from https://developer.apple.com/bug-reporting/profiles-and-logs/.
5. Install the profile using the provided instructions from the website.
6. Open Console and select the device that you installed the profile on.
7. Filter by "homed".
8. Launch accessory app and navigate to the accessory control screen e.g. temperature control for thermostat.
9. Verify you do not see the following more than once every 2 minutes: "Answering incoming message kCharacteristicReadRequestKey (UUID) from client '«your app»' that does expect a response" or "Answering incoming message kMultipleCharacteristicReadRequestKey (UUID) from client '«your app»' that does expect a response".

TCFA008 **Accessory app must support the creating, listing, renaming, and modifying of homes.**

Applies to all accessories. Perform this test case using an iOS device running the accessory app.

1. Verify the accessory app allows creation, listing, renaming and modifying of homes.
2. Verify any changes to the home made in the accessory app are reflected in the Home app.
3. Verify any changes to the home made in the Home app are reflected in the accessory app.

TCFA009 **Accessory app must support the creating, listing, renaming, and modifying of rooms.**

Applies to all accessories. Perform this test case using an iOS device running the accessory app.

1. Verify the accessory app allows creation, listing, renaming and modifying of rooms.
2. Verify any changes to rooms made in the accessory app are reflected in the Home app.
3. Verify any changes to rooms made in the Home app are reflected in the accessory app.

TCFA010 **If supported, accessory app must support the creating, listing, renaming, and modifying of service groups.**

Applies to all accessories. Perform this test case using an iOS device running the accessory app.

1. Verify accessory app supports for the creating, listing, renaming, and modifying of service groups.
2. Verify modifications to service groups from within the accessory app are reflected in the Home app.
3. Verify modifications to service groups from within the Home app are reflected in the accessory app.

TCFA011 **If supported, accessory apps must allow creation, deletion, and modification of scenes.**

Applies to all accessories. Perform this test case using an iOS device running the accessory app.

1. Verify that scenes can be added, modified, and removed.
2. Verify the existence of pre-defined Good Morning, Good Night, Arrive Home, and Leave Home scenes.
3. Verify modifications to scenes from within the accessory app are reflected in the Home app.
4. Verify modifications to scenes from within the Home app are reflected in the accessory app.

TCFA012 **Accessory Apps must not create or delete HomeKit objects, such as rooms, zones, action sets, service groups, scenes, etc., without user consent.**

Applies to all accessories. Perform this test case using an iOS device running the accessory app.

1. Verify that HomeKit objects, such as rooms, zones, action sets, service groups, triggers, scenes, etc., are not created or deleted by the accessory app without explicit user consent.

TCFA013 **Accessory app must support the renaming of all services which the user may interact that contain Apple-defined characteristics.**

Applies to all accessories. Perform this test case using an iOS device running the accessory app.

1. Verify that all services in which the user may interact with can be renamed.
2. Verify accessory control via Siri is successful when utilizing the accessory's service name provided during the setup flow or subsequent renaming.
3. Ensure user can rename any service the user can interact with.

TCFA014 **Accessory apps must not allow shared users to edit home configurations.**

Applies to all accessories. Perform this test case using an iOS device running the accessory app.

1. Add a shared user to your home.
2. Ensure user is not an admin. (Allow Editing is disabled).
3. Verify the accessory app does not show or allow any edit options to the home or accessories.

TCFA015 **Accessory app must remain stable when a user's home configuration utilizes the full HomeKit feature set.**

Applies to all accessories. Perform this test case using an iOS device running the accessory app.

1. Add all manufacturer-supported features to a home.
2. In the Home app, create a home that includes: all trigger types, HAP over Wi-Fi or Ethernet (standalone and bridged) accessories, HAP over BLE accessories, and at least one of each Apple-defined service and characteristic.
3. Ensure accessory app remains stable during usage.

TCFA016 **If an accessory supports additional authorization data, confirm the accessory app properly sets additional authorization data (e.g for apps that interact with Security Class characteristics, such as lock-related apps).**

Applies to accessories that support HomeKit Accessory Protocol specification R15 or earlier. Perform this test case using an iOS device running the accessory app.

1. If an accessory supports additional authorization data, confirm the accessory app properly sets additional authorization data.
2. Verify that after additional authorization is complete, that reads/writes performed in the Home app are successful.

TCFA017 **Accessory app must not allow users to delete homes.**

Applies to all accessories. Perform this test case using an iOS device running the accessory app.

1. Open the accessory app.
2. Ensure the option to delete the home is not exposed or an error is given preventing deletion.

TCFA018 **Accessory apps that ask users to create an account must provide an option for users to defer this step until HomeKit setup is complete and accessory is usable. Accessory apps must not require an account to receive firmware updates.**

Applies to all accessories. Perform this test case using an iOS device running the accessory app.

1. Open the accessory app.
2. Verify the accessory app does not require an account for setup, or provides a way to skip until HomeKit setup is complete.
3. Verify the accessory can receive firmware updates without an account.

TCFA019 **Accessory app must not change values (e.g., turn off a light bulb) without user intent.**

Applies to all accessories. Perform this test case using an iOS device running the accessory app.

1. Perform Pair Setup with the accessory using the accessory app.
2. Verify the accessory app does not make any changes to the accessory without user intent.

TCFA020 **A accessory app that only supports iOS 10 or later must not use the External Accessory framework to browse for and/or configure unpaired accessories. Instead, the addAndSetupAccessories API must be used.**

Applies to all accessories. Perform this test case using an iOS device running the accessory app.

1. Verify the accessory app uses the addAndSetupAccessories API if it only supports iOS 10 or later.

TCFA021 **Accessory app must support identifying a HomeKit accessory that is part of a home using the identify API.**

Applies to all accessories. Perform this test case using an iOS device running the accessory app.

1. Perform Pair Setup with the accessory using the accessory app.
2. Verify that the accessory app supports identifying the accessory.

TCFA022 **If the accessory accessory requires additional setup prior to being paired with the Home app, the Home app must be able to pair with that accessory after the additional setup is complete.**

Applies to all accessories. Perform this test case using an iOS device running the accessory app.

1. Complete additional setup steps in the accessory app.
2. Verify that the Home app can pair with the accessory.
3. Factory reset the accessory.
4. Pair with the accessory using the Home app.
5. Complete additional setup steps in accessory app.

TCFA023 **Accessory app must allow user configuration of "Associated Service Type" for outlets and switches using the updateAssociated - ServiceType API.**

Applies to all accessories. Perform this test case using an iOS device running the accessory app.

1. Verify the accessory app allows user configuration of "Associated Service Type" for outlets and switches.

TCFA024 **If applicable, the accessory app must support adding endpoints to a bridge that does not have a physical interface for adding endpoints.**

Applies to all accessories. Perform this test case using an iOS device running the accessory app.

1. Verify user is able to use accessory app to add endpoints.

3.4 App with limited HomeKit API Support

TCLA001: Accessory app must provide firmware updates for the accessory.

TCLA002: The accessory app must direct users to the Home app for accessory control and configuration.

TCLA003: If applicable, the accessory app must support adding endpoints to a bridge that does not have a physical interface for adding endpoints.

TCLA001 **Accessory app must provide firmware updates for the accessory.**

Applies to all accessories. Perform this test case using an iOS device running the accessory app.

1. Verify the accessory app provides a way to update the firmware of the accessory.

TCLA002 **The accessory app must direct users to the Home app for accessory control and configuration.**

Applies to all accessories. Perform this test case using an iOS device running the accessory app.

1. Verify the accessory app directs the user to set up and control the accessory through the Home app.

TCLA003 **If applicable, the accessory app must support adding endpoints to a bridge that does not have a physical interface for adding endpoints.**

Applies to all accessories. Perform this test case using an iOS device running the accessory app.

1. Verify user is able to use accessory app to add endpoints.

3.5 App Not Required

TCNA001: The accessory must receive firmware updates automatically from a manufacturer hosted server without user interaction, i.e., pushed by a server.

TCNA002: The accessory must not have any user-interactive characteristics i.e., a custom characteristic, that would need configuration by the user for desired operation.

TCNA003: If the bridge can support adding endpoints through a physical, non-app based process, i.e., pairing or link button, verify endpoints can be added without using the accessory app.

TCNA001 **The accessory must receive firmware updates automatically from a manufacturer hosted server without user interaction, i.e., pushed by a server.**

Applies to all accessories.

1. Verify that firmware updates are pushed over-the-air from the manufacturer and do not require an app or user interaction to receive the update.

TCNA002 **The accessory must not have any user-interactive characteristics i.e., a custom characteristic, that would need configuration by the user for desired operation.**

Applies to all accessories.

1. Verify the accessory does not require configuration of custom characteristics that need to be configured through the accessory app for desired operation of the accessory.

TCNA003 **If the bridge can support adding endpoints through a physical, non-app based process, i.e., pairing or link button, verify endpoints can be added without using the accessory app.**

Applies to all accessories.

1. Verify the bridge can add endpoints manually without requiring use of the accessory app.

Chapter 4

Revision History

- Added: TCI050, TCPP003
- Revised: TCADX006, TCAFU004, TCAFU014, TCBW001, TCBW002, TCBW003, TCBW004, TCBW005, TCB012, TCB016, TCB052, TCF002, TCF033, TCF041, TCH004, TCH010, TCH014, TCH058, TCH059, TCH083, TCHDS006, TCICAV069, TCL001, TCL002, TCL003, TCL004, TCL005, TCL006, TCL007, TCL008, TCL009, TCL010, TCL011, TCL012, TCL013, TCL015, TCL016, TCL017, TCL018, TCL019, TCL020, TCL021, TCL022, TCL023, TCL024, TCL026, TCL027, TCL028, TCL029, TCL030, TCL031, TCL032, TCL033, TCL034, TCL035, TCLS001, TCLS002, TCLS003, TCLS004, TCLS005, TCLS006, TCLS007, TCLS008, TCLS009, TCLS010, TCLS011, TCLS012, TCLS013, TCLS014, TCLS015, TCLS017, TCLS018, TCLS019, TCLS021, TCLS022, TCLS023, TCLS024, TCLS025, TCLS026, TCLS027, TCLS028, TCLS029, TCLS030, TCLS031, TCLS032, TCLS033, TCLS034, TCLS035, TCLS036, TCLS037, TCLS038, TCLS039, TCLS040, TCLS041, TCLS042, TCLS043, TCLS044, TCLS045, TCLS046, TCLS047, TCLS048, TCLS049, TCR002, TCR003, TCR004, TCR005, TCR008, TCR009, TCR016, TCR019, TCR020, TCR021, TCR022, TCR026, TCR032, TCR034, TCR035, TCR046, TCR059, TCR060, TCR061, TCR076, TCR097, TCR098, TCT004, TCT008, TCT021, TCT026, TCT033, TCT034, TCT036, TCT095, TCT100, TCWR003, TCWR004
- Removed: TCH013