HTTP Proxy Service (HPS)

***Bluetooth*® Test Suite**

- **Revision:** HPS.TS.p2
- **Revision Date:** 2023-06-29
- **Prepared By:** Internet Working Group
- **Published during TCRL:** TCRL.2023-1

Contents

1 Scope

This Bluetooth document contains the Test Suite Structure (TSS) and test cases to test the implementation of the Bluetooth HTTP Proxy Service (HPS), Version 1.0, with the objective to provide a high probability of air interface interoperability between the tested implementation and other manufacturers' Bluetooth devices.

2 References, definitions, and abbreviations

2.1 References

This document incorporates provisions from other publications by dated or undated reference. These references are cited at the appropriate places in the text, and the publications are listed hereinafter.

[1] Test Strategy and Terminology Overview

[2] Bluetooth Core Specification, Version 4.0 or later

[3] HTTP Proxy Service, Version 1.0

[4] HTTP Proxy Service ICS Proforma

[5] GATT Test Suite, GATT.TS

[6] IETF RFC 2616 Hypertext Transfer Protocol -- HTTP/1.1
http://www.ietf.org/rfc/rfc2616.txt

[7] IETF RFC 2818 HTTP Over TLS
http://www.ietf.org/rfc/rfc2818.txt

[8] HTTP Proxy Service IXIT Proforma

2.2 Definitions

In this Bluetooth document, the definitions from [1] and [2] apply.

2.3 Acronyms and abbreviations

In this Bluetooth document, the definitions, acronyms, and abbreviations from [1] and [2] apply.

3 Test Suite Structure (TSS)

3.1 Test Strategy

The test objectives are to verify functionality of the HTTP Proxy Service within a Bluetooth Host and enable interoperability between Bluetooth Hosts on different devices. The testing approach covers mandatory and optional requirements in the specification and matches these to the support of the IUT as described in the ICS. Any defined test herein is applicable to the IUT if the ICS logical expression defined in the Test Case Mapping Table (TCMT) evaluates to true.

The test equipment provides an implementation of the Radio Controller and the parts of the Host needed to perform the test cases defined in this Test Suite. A Lower Tester acts as the IUT's peer device and interacts with the IUT over-the-air interface. The configuration, including the IUT, needs to implement similar capabilities to communicate with the test equipment. For some test cases, it is necessary to stimulate the IUT from an Upper Tester. In practice, this could be implemented as a special test interface, a Man Machine Interface (MMI), or another interface supported by the IUT.

This Test Suite contains Valid Behavior (BV) tests complemented with Invalid Behavior (BI) tests where required. The test coverage mirrored in the Test Suite Structure is the result of a process that started with catalogued specification requirements that were logically grouped and assessed for testability enabling coverage in defined test purposes.

3.2 Test groups

The following test groups have been defined:

- Generic GATT Integrated Tests
- Characteristic Notifications
- Service Procedure
- LE Connection Procedure

4 Test cases (TC)

4.1 Introduction

4.1.1 Test case identification conventions

Test cases are assigned unique identifiers per the conventions in [1]. The convention used here is:
<spec abbreviation>/<IUT role>/<class>/**<feat>**/<func>/<subfunc>/<cap>/**<xx>-<nn>-<y>**.

Additionally, testing of this specification includes tests from the GATT Test Suite [5] referred to as Generic GATT Integrated Tests (GGIT); when used, the GGIT tests are referred to through a TCID string using the following convention:
<spec abbreviation>/<IUT role>/<GGIT test group>/< GGIT class >/<xx>-<nn>-<y>.

Identifier Abbreviation	**Spec Identifier <spec abbreviation>**
HPS	HTTP Proxy Service
Identifier Abbreviation	**Role Identifier <IUT role>**
SR	Server role
Identifier Abbreviation	**Reference Identifier <GGIT test group>**
SGGIT	Server Generic GATT Integrated Tests
Identifier Abbreviation	**Reference Identifier <GGIT class>**
CHA	Characteristic
SDP	Validate SDP Record
SER	Service
Identifier Abbreviation	**Feature Identifier <feat>**
ADV	LE Connection Procedures
CN	Characteristic Notification
CON	Configure Notification
SP	Service Procedures

Table 4.1 HPS TC feature naming conventions

4.1.2 Conformance

When conformance is claimed for a particular specification, all capabilities are to be supported in the specified manner. The mandated tests from this Test Suite depend on the capabilities to which conformance is claimed.

The Bluetooth Qualification Program may employ tests to verify implementation robustness. The level of implementation robustness that is verified varies from one specification to another and may be revised for cause based on interoperability issues found in the market.

Such tests may verify:

- That claimed capabilities may be used in any order and any number of repetitions not excluded by the specification
- That capabilities enabled by the implementations are sustained over durations expected by the use case
- That the implementation gracefully handles any quantity of data expected by the use case

- That in cases where more than one valid interpretation of the specification exist, the implementation complies with at least one interpretation and gracefully handles other interpretations
- That the implementation is immune to attempted security exploits

A single execution of each of the required tests is required to constitute a Pass verdict. However, it is noted that to provide a foundation for interoperability, it is necessary that a qualified implementation consistently and repeatedly pass any of the applicable tests.

In any case, where a member finds an issue with the test plan generated by Launch Studio, with the test case as described in the Test Suite, or with the test system utilized, the member is required to notify the responsible party via an erratum request such that the issue may be addressed.

4.1.3 Pass/Fail verdict conventions

Each test case has an Expected Outcome section. The IUT is granted the Pass verdict when all the detailed pass criteria conditions within the Expected Outcome section are met.

The convention in this Test Suite is that, unless there is a specific set of fail conditions outlined in the test case, the IUT fails the test case as soon as one of the pass criteria conditions cannot be met. If this occurs, then the outcome of the test is a Fail verdict.

4.2 Setup preambles

The procedures defined in this section are used to achieve specific conditions on the IUT and the test equipment within the tests defined in this document. The preambles here are commonly used to establish initial conditions.

4.2.1 ATT Bearer on BR/EDR Transport

Follow the preamble procedure described in [5] Section 4.2.1.1.

4.2.2 ATT Bearer on LE Transport

Follow the preamble procedure described in [5] Section 4.2.1.2.

4.3 Generic GATT Integrated Tests

Execute the Generic GATT Integrated Tests defined in Section 6.3, Server test procedures (SGGIT), in [5] using Table 4.2 below as input:

TCID	Service / Characteristic	Reference	Properties	Value Length (Octets)	Service Type
HPS/SR/SGGIT/SER/BV-01-C [Service GGIT – HTTP Proxy]	HTTP Proxy Service	[3] 2	-	-	Primary Service, Unique
HPS/SR/SGGIT/CHA/BV-01-C [Characteristic GGIT – URI]	URI Characteristic	[3] 3.1	0x08 (Write)	Up to 512	-
HPS/SR/SGGIT/CHA/BV-02-C [Characteristic GGIT – HTTP Headers]	HTTP Headers Characteristic	[3] 3.2	0x0A (Read, Write)	Up to 512	-
HPS/SR/SGGIT/CHA/BV-03-C [Characteristic GGIT – HTTP Entity Body]	HTTP Entity Body Characteristic	[3] 3.3	0x0A (Read, Write)	Up to 512	
HPS/SR/SGGIT/CHA/BV-04-C [Characteristic GGIT – HTTP Control Point]	HTTP Control Point Characteristic	[3] 3.4	0x08 (Write)	Skip	
HPS/SR/SGGIT/CHA/BV-05-C [Characteristic GGIT – HTTP Status Code]	HTTP Status Code Characteristic	[3] 3.5	0x10 (Notify)	Skip	
HPS/SR/SGGIT/CHA/BV-06-C [Characteristic GGIT – HTTPS Security]	HTTPS Security Characteristic	[3] 3.6	0x02 (Read)	1	
HPS/SR/SGGIT/SDP/BV-01-C [SDP Record – HTTP Proxy]	HTTP Proxy Service	[3] 2, 3.7	-	-	-

Table 4.2: Input for the GGIT Server test procedure

4.4 Configure Notification

HPS/SR/CON/BV-01-C [Configure Notification – HTTP Status Code]

- Test Purpose

  Verify that the Server can configure the HTTP Status Code characteristic for notification.

- Reference

  [3] 3.5.1

- Initial Condition

  - The handle of the HTTP Status Code characteristic and the HTTP Status Code client characteristic configuration descriptor has been previously discovered by the Lower Tester during the test procedure in Section 4.3 or is known to the Lower Tester by other means.
  - Establish an ATT Bearer connection between the Lower Tester and IUT as described in Section 4.2.1 if using a BR/EDR transport or Section 4.2.2 if using an LE transport.

- Test Procedure

  1. The Lower Tester sends an ATT_Write_Request, to disable notification by writing value 0x0000 to the client characteristic configuration descriptor of the characteristic.
  2. The Lower Tester sends an ATT_Write_Request, to enable notification by writing value 0x0001 to the client characteristic configuration descriptor of the HTTP Status Code characteristic.
  3. The Lower Tester reads the value of the client characteristic configuration descriptor.

- Expected Outcome

  Pass verdict

  The characteristic descriptor is successfully written and the value returned when read is consistent with the value written.

4.5 Characteristic Notification

HPS/SR/CN/BV-01-C [Characteristic Notification – HTTP Status Code]

- Test Purpose

  Verify that the Server can notify the HTTP Status Code characteristic.

- Reference

  [3] 3.5

- Initial Condition

  - The HTTP Status Code characteristic is configured for notification.
  - Establish an ATT Bearer connection between the Lower Tester and IUT as described in Section 4.2.1 if using a BR/EDR transport or Section 4.2.2 if using an LE transport.

- Test Procedure

  1. A connection is established between the Lower Tester and IUT.
  2. Configure the HTTP Proxy data characteristics: URI, HTTP Header and HTTP Entity Body, conditioned by the IXIT [8], and write a value to the HTTP Control Point characteristic, so that the IUT will send one notification of the HTTP Status characteristic.

3. The Lower Tester receives one *ATT_Handle_Value_Notification* from the IUT containing the HTTP Status code value (uint16), received HTTP header size in octets, and received HTTP entity body size (1 octet) in Data Status field.
4. Verify that the characteristic value meets the requirements of the service.

- Expected Outcome

  Pass verdict

  The IUT indicates a value in HTTP Status Code characteristic appropriate to the requirements in the Service.

4.6 Service Procedures

4.6.1 Supported Control Point Procedure Request

- Test Purpose

  Verify that when the HTTP configuration characteristics are configured, and one of the valid op-codes defined for the HTTP control point in [3] Table 3.2 is requested by writing to the control point, a notification on HTTP Status Code characteristic is sent to the HPS client, marking the end of the requested procedure.

Test Case	URI, [8] HTTP Header, [8] HTTP Body: [8] HTTP Control Point [3]	HTTP Security: read	Status Code Expected decimal
HPS/SR/SP/BV-01-C [HTTP GET]	Write: URI Write: Header Write: Body Write: HTTP GET Read: Header Read: Body	N/A	200
HPS/SR/SP/BV-02-C [HTTP HEAD]	Write: URI Write: Header Write: Body Write: HTTP HEAD Read: Header	N/A	200
HPS/SR/SP/BV-03-C [HTTP POST]	Write: URI Write: Header Write: Body Write: HTTP POST	N/A	200
HPS/SR/SP/BV-04-C [HTTP PUT]	Write: URI Write: Header Write: Body Write: HTTP PUT	N/A	200
HPS/SR/SP/BV-05-C [HTTP DELETE]	Write: URI Write: Header Write: Body Write: HTTP DELETE	N/A	200
HPS/SR/SP/BV-06-C [HTTPS GET]	Write: URI Write: Header Write: Body Write: HTTPS GET Read: Header Read: Body	TRUE	200

Test Case	URI, [8] HTTP Header, [8] HTTP Body: [8] HTTP Control Point [3]	HTTP Security: read	Status Code Expected decimal
HPS/SR/SP/BV-07-C [HTTPS HEAD]	Write: URI Write: Header Write: Body Write: HTTPS HEAD Read: Body	TRUE	200
HPS/SR/SP/BV-08-C [HTTPS POST]	Write: URI Write: Header Write: Body Write: HTTPS POST	TRUE	200
HPS/SR/SP/BV-09-C [HTTPS PUT]	Write: URI Write: Header Write: Body Write HTTPS PUT	TRUE	200
HPS/SR/SP/BV-10-C [HTTPS DELETE]	Write: URI Write: Header Write: Body Write: HTTPS DELETE	TRUE	200

Table 4.3 HTTPS valid command test

- Reference

  [3] 3.5, 3.6, Appendix B

  [6], [7], [8]

- Initial Condition

  - Establish an ATT Bearer connection between the Lower Tester and IUT as described in Section 4.2.1 if using a BR/EDR transport or Section 4.2.2 if using an LE transport.
  - The HTTP Status Code characteristic is configured for notification.
  - The Lower Tester configures the URI, HTTP Headers and HTTP Entity Body Characteristics with valid values, conditioned by information in the IXIT [8]. If no entity body will be sent, the Lower Tester configures the Entity Body Characteristic by writing a zero length value; GET, HEAD and DELETE.
  - The IUT is connected to an Upper Tester which emulates an HTTP Server.

- Test Procedure

  1. The Lower Tester performs a Write Characteristic Value to the HTTP Control Point characteristic, with a corresponding command chosen from [3] Table 3.2.
  2. The IUT conducts HTTP operations with the ‘Upper Tester’.
  3. The ‘Upper Tester’ responds with an HTTP response code.
  4. The IUT captures that HTTP response code, and the sizes of the received header and message body, if any.
  5. The IUT notifies the Lower Tester with the HTTP Status Code characteristic, from the previous step.
  6. If an HTTPS command was written to the HTTP Control Point characteristic, the IUT (HPS Client) will read the HTTP Security characteristic.

- Expected Outcome

  Pass verdict

  An appropriate HTTP or HTTPS response code is notified to the Lower Tester. If an HTTPS command was written to HTTP Control Point, the result of reading the HTTP Security Characteristic will be TRUE.

HPS/SR/SP/BV-11-C [HTTP Request Cancel]

- Test Purpose

  Verify that an HTTP Cancel request message received from the HPS Client is processed properly by the IUT.

- Reference

  [3] 3.4, 3.5, 3.6, Appendix B, [6], [7], [8]

- Initial Condition

  - Establish an ATT Bearer connection between the Lower Tester and IUT as described in Section 4.2.1 if using a BR/EDR transport or Section 4.2.2 if using an LE transport.
  - The HTTP Status Code characteristic is configured for notification.
  - The Lower Tester configures the URI, HTTP Headers and HTTP Entity Body Characteristics with valid values, conditioned by information in the IXIT [8].
  - The IUT is connected to an Upper Tester which emulates an HTTP Server.

- Test Procedure

  1. The Lower Tester performs a Write Characteristic Value to the HTTP Control Point characteristic, with a corresponding command chosen from [3] Table 3.2, except HTTP Cancel.
  2. The IUT accepts the control point command., enters the ER state, and begins conducting HTTP operations with the ‘Upper Tester’.
  3. While the IUT is in ER state, the Lower Tester performs a Write Characteristic Value to the HTTP Control Point with a HTTP Request Cancel code.

- Expected Outcome

  Pass verdict

  The IUT accepts the second write with the HTTP Cancel command.

HPS/SR/SP/BV-12-C [Unsupported Control Point Procedure Request]

- Test Purpose

  Verify that when a control point op-code that is not supported by the server is requested, a notification on HTTP Status characteristic with status ‘Invalid PDU’ is sent to client.

- Reference

  [3] 3.5, Appendix B

- Initial Condition

  - Establish an ATT Bearer connection between the Lower Tester and IUT as described in Section 4.2.1 if using a BR/EDR transport or Section 4.2.2 if using an LE transport.

- The HTTP Status Code characteristic is configured for notification.
- The Lower Tester configures the URI, HTTP Header and HTTP Entity Body Characteristics with valid values.
- The IUT is connected to an Upper Tester which emulates an HTTP Server.

- Test Procedure

  1. The Lower Tester performs a Write Characteristic Value to the HTTP Control Point characteristic, with a command not defined in [3] Table 3.2.
  2. The IUT determines that the command is not supported.

- Expected Outcome

  Pass verdict

  The IUT sends an “Invalid Request” error response to the Lower Tester.

HPS/SR/SP/BI-01-C [Control Point Request without HTTP Status configuration]

- Test Purpose

  If a control point procedure is requested without configuring the HTTP Status Characteristic for notification, the request is answered with an error code ‘Client Characteristic Configuration Descriptor Improperly Configured’.

- Reference

  [3] 3.5, 3.6, Appendix B

  [6], [7], [8]

- Initial Condition

  - Establish an ATT Bearer connection between the Lower Tester and IUT as described in Section 4.2.1 if using a BR/EDR transport or Section 4.2.2 if using an LE transport.
  - The HTTP Status Code characteristic is configured to inhibit notification; 0x0000 written to HTTP Proxy characteristic client characteristic configuration descriptor.
  - The Lower Tester configures the URI, HTTP Headers and HTTP Entity Body Characteristics with valid values, conditioned by information in the IXIT [8]; see 4.6.1.
  - The IUT is connected to an Upper Tester which emulates an HTTP Server.

- Test Procedure

  1. The Lower Tester performs a Write with Response to the HTTP Control Point characteristics, with a command chosen from [3] Table 3.2.

- Expected Outcome

  Pass verdict

  The HPS will respond with the error code “Client Characteristic Configuration Descriptor Improperly Configured”.

HPS/SR/SP/BI-02-C [No network available]

- Test Purpose

  Verify that, if an HPS is disconnected from the internet, a write to the HTTP Control Point is answered with an error code.

- Reference

  [3] 1.6, 3.4, [8]

- Initial Condition

  - The Lower Tester configures the URI, HTTP Header and HTTP Entity Body Characteristics with valid values, conditioned by information in the IXIT [8].
  - Establish an ATT Bearer connection between the Lower Tester and IUT as described in Section 4.2.1 if using a BR/EDR transport or Section 4.2.2 if using an LE transport.
  - The IUT is logically disconnected to an Upper Tester which emulates an HTTP Server. Example conditions: network disconnected, connection to a router not established.

- Test Procedure

  1. The Lower Tester performs a Write Characteristic Value to the HTTP Control Point characteristics, with a command chosen from [3] Table 3.2.
  2. The IUT determines that the network connection is not available.

- Expected Outcome

  Pass verdict

  The IUT sends an “ATT Network Not Available” error code to the Lower Tester.

HPS/SR/SP/BI-03-C [Received HTTP Header is too large]

- Test Purpose

  Verify that the IUT can detect an incoming HTTP header size which is too large for the HTTP Headers characteristic, and indicate this size in the HTTP Status Code notification.

- Reference

  [3] 3.5, Appendix B

  [6], [7], [8].

- Initial Condition

  - Establish an ATT Bearer connection between the Lower Tester and IUT as described in Section 4.2.1 if using a BR/EDR transport or Section 4.2.2 if using an LE transport.
  - The HTTP Status Code characteristic is configured for notification.
  - The Lower Tester configures the URI, HTTP Headers and HTTP Entity Body Characteristics with valid values, conditioned by information in the IXIT for an HTTP HEAD Request.
  - The IUT is connected to an Upper Tester which emulates an HTTP Server.
  - The ‘Upper Tester’ is conditioned to return an HTTP Header larger than 512 octets.

- Test Procedure

  1. The Lower Tester performs a Write Characteristic Value to the HTTP Control Point characteristic, with an HTTP Head request from [3] Table 3.2.
  2. The IUT conducts HTTP Head Request with the 'Upper Tester'.
  3. The 'Upper Tester' responds with an HTTP response code
  4. The IUT captures that HTTP response code and the size of the received header.
  5. An appropriate HTTP response code is notified to the Lower Tester in the HTTP Status Code.

- Expected Outcome

  Pass verdict

  The HTTP Headers size notified in the HTTP Status Code is larger than 512 octets by setting bit 1 in the Data Status octet.

  The first 512 octets of the HTTP header are present in the HTTP Headers characteristic.

HPS/SR/SP/BI-04-C [Received HTTP Message Body is too large]

- Test Purpose

  Verify that the IUT can detect an incoming HTTP message body size which is too large for the HTTP Message Body characteristic, and indicate this size in the HTTP Status Code notification.

- Reference

  [3] 3.5, Appendix B

  [6], [7], [8].

- Initial Condition

  - Establish an ATT Bearer connection between the Lower Tester and IUT as described in Section 4.2.1 if using a BR/EDR transport or Section 4.2.2 if using an LE transport.
  - The HTTP Status Code characteristic is configured for notification.
  - The Lower Tester configures the URI, HTTP Headers and HTTP Entity Body Characteristics with valid values, conditioned by information in the IXIT for an HTTP GET Request.
  - The IUT is connected to an Upper Tester which emulates an Upper Tester.
  - The 'Upper Tester' is conditioned to return an HTTP Message Body larger than 512 octets.

- Test Procedure

  1. The Lower Tester performs a Write Characteristic Value to the HTTP Control Point characteristic, with an HTTP Head request from [3] Table 3.2.
  2. The IUT conducts HTTP GET Request with the 'Upper Tester'.
  3. The 'Upper Tester' responds with an HTTP response code
  4. The IUT captures that HTTP response code and the size of the received message body.
  5. An appropriate HTTP response code is notified to the Lower Tester in the HTTP Status Code

- Expected Outcome

  Pass verdict

  The HTTP Message Body size notified in the HTTP Status Code is larger than 512 octets, by setting bit 3 in the Data Status octet.

  The first 512 octets of the HTTP message body are present in the HTTP Entity Body Characteristic.

HPS/SR/SP/BI-05-C [Request already in progress]

- Test Purpose

  Verify that, if an HPS client writes a command other than 'HTTP Request Cancel' to an HPS server while the HPS server is in ER state, the HPS server response with a 'Request already in Progress' error code.

- Reference

  [3] 1.6, 3.4, [8]

- Initial Condition

  - The Lower Tester configures the URI, HTTP Headers and HTTP Entity Body Characteristics with valid values, conditioned by information in the IXIT [8]. The command can be any valid value except HTTP CANCEL.
  - Establish an ATT Bearer connection between the Lower Tester and IUT as described in Section 4.2.1 if using a BR/EDR transport or Section 4.2.2 if using an LE transport.
  - The IUT is logically connected to an Upper Tester which emulates an HTTP Server.

- Test Procedure

  1. The Lower Tester performs a Write Characteristic Value to the HTTP Control Point characteristics, with a command chosen from [3] Table 3.2, except for HTTP Cancel Request.
  2. The Lower Tester waits for a Write Response from the IUT.
  3. The Lower tester performs another Write Characteristic Value to the HTTP Control Point characteristic, with a valid new command chosen from [3] Table 3.2, except for HTTP Cancel Request.

- Expected Outcome

  Pass verdict

  The IUT sends an "ATT Request Already in Progress" error code to the Lower Tester.

HPS/SR/SP/BI-06-C [Invalid Request]

- Test Purpose

  Verify that, if an HPS client writes a command to an HPS server while the HPS server but omits one of the essential characteristics (URI, HTTP Headers, HTTP Message Body), the HPS replies with an Invalid Request error code.

- Reference

  [3] 1.6, 3.4, [8]

- Initial Condition

  - Establish an ATT Bearer connection between the Lower Tester and IUT as described in Section 4.2.1 if using a BR/EDR transport or Section 4.2.2 if using an LE transport.
  - The IUT is logically connected to an Upper Tester which emulates an Upper Tester.

- Test Procedure

  1. The Lower Tester configures some of the URI, HTTP Headers and HTTP Entity Body Characteristics with valid values, conditioned by information in the IXIT [8] for an HTTP PUT command. The Lower Tester must omit one or more of these characteristics.
  2. The Lower Tester performs a Write Characteristic Value to the HTTP Control Point characteristics, with an HTTP PUT command.

- Expected Outcome

  Pass verdict

  The IUT sends an “Invalid Request” error code to the Lower Tester.

4.7 LE Connection Procedure

HPS/SR/ADV/BV-01-C [Advertise HPS UUID in AD type field]

- Test Purpose

  Verify that an HPS Server IUT includes the HPS UUID in the AD type field.

- Reference

  [2] GAP 11.1.1

- Initial Condition

  - The HPS Server IUT is in Link Layer State “standby”.

- Test Procedure

  The Upper Tester orders the IUT to start advertising; the IUT enters a discoverable mode.

- Test Condition

  It must be possible to order IUT to start advertising.

- Expected Outcome

  Pass Verdict

  The advertising type field contains the HPS Service UUID.

HPS/SR/ADV/BV-02-C [Connect with HPS Clients that include the HPS UUID in the AD type field]

- Test Purpose

  Verify that an HPS Server IUT will detect and connect with lower tester which includes the HPS UUID in a Service Solicitation AD type field.

- Reference

  [2] GAP 11.1.9

- Initial Condition

  - The HPS Server IUT is in Link Layer State “standby”.

- Test Procedure

  The IUT enters a discoverable mode.

  The Lower Tester starts advertising with a Service Solicitation data type. The AD type includes the HPS UUID.

- Expected Outcome

  Pass Verdict

  The IUT sends a connect request packet to the Lower Tester.

5 Test case mapping

The Test Case Mapping Table (TCMT) maps test cases to specific capabilities in the ICS. The IUT is tested in all roles for which support is declared in the ICS document.

The columns for the TCMT are defined as follows:

Item: Contains a logical expression based on specific entries from the associated ICS document. Contains a logical expression (using the operators AND, OR, NOT as needed) based on specific entries from the applicable ICS document(s). The entries are in the form of y/x references, where y corresponds to the table number and x corresponds to the feature number as defined in the ICS document for the HTTP Proxy Service (HPS) [4].

Feature: A brief, informal description of the feature being tested.

Test Case(s): The applicable test case identifiers are required for Bluetooth Qualification if the corresponding y/x references defined in the Item column are supported.

For the purpose and structure of the ICS/IXIT, refer to [1].

Item	Feature	Test Case(s)
HPS 1/1	Service Supported over BR/EDR	HPS/SR/SGGIT/SDP/BV-01-C
HPS 1/2	Service Supported over LE	HPS/SR/SGGIT/SER/BV-01-C
HPS 2/2	HPS URI Characteristic	HPS/SR/SGGIT/CHA/BV-01-C
HPS 2/3	HTTP Headers Characteristic	HPS/SR/SGGIT/CHA/BV-02-C
HPS 2/4	HTTP Entity Body Characteristic	HPS/SR/SGGIT/CHA/BV-03-C
HPS 2/5	HTTP Control Point Characteristic	HPS/SR/SGGIT/CHA/BV-04-C HPS/SR/SP/BV-12-C HPS/SR/SP/BI-02-C
HPS 2/6	HTTP Status Code Characteristic	HPS/SR/SGGIT/CHA/BV-05-C HPS/SR/CON/BV-01-C HPS/SR/CN/BV-01-C HPS/SR/SP/BI-01-C HPS/SR/SP/BI-05-C
HPS 2/7	HTTPS Security Characteristic	HPS/SR/SGGIT/CHA/BV-06-C
HPS 2/8	HTTP GET	HPS/SR/SP/BV-01-C HPS/SR/SP/BI-04-C
HPS 2/9	HTTP HEAD	HPS/SR/SP/BV-02-C HPS/SR/SP/BI-03-C
HPS 2/10	HTTP POST	HPS/SR/SP/BV-03-C
HPS 2/11	HTTP PUT	HPS/SR/SP/BV-04-C HPS/SR/SP/BI-06-C
HPS 2/12	HTTP DELETE	HPS/SR/SP/BV-05-C
HPS 2/13	HTTPS GET	HPS/SR/SP/BV-06-C
HPS 2/14	HTTPS HEAD	HPS/SR/SP/BV-07-C
HPS 2/15	HTTPS POST	HPS/SR/SP/BV-08-C
HPS 2/16	HTTPS PUT	HPS/SR/SP/BV-09-C
HPS 2/17	HTTPS DELETE	HPS/SR/SP/BV-10-C
HPS 2/18	HTTP CANCEL	HPS/SR/SP/BV-11-C

Item	Feature	Test Case(s)
HPS 2/19	HPS Advertising includes HPS UUID in AD type field	HPS/SR/ADV/BV-01-C
HPS 2/10	HPS Server connects with HPS client including HPS UUID in AD type field	HPS/SR/ADV/BV-02-C

Table 5.1: Test case mapping

6 Revision history and acknowledgments

Revision History

Publication Number	Revision Number	Date	Comments
0	1.0.0	2015-10-06	Prepared for publication
	1.0.1r00	2016-03-01	TSE 6950: Updated pass verdict for test case TP/SP/BI-04-C (now HPS/SR/SP/BI-04-C after ID conversion).
	1.0.1r01	2016-03-10	TSE 6954: Updated test procedure for HTTPS Service Procedures (HPS/SR/SP test group in Section 4.10).
	1.0.1r02	2016-04-13	TSE 6951: Corrected step 3 of test procedure for TP/CN/BV-01-C (now HPS/SR/CN/BV-01-C after ID conversion).
	1.0.1r03	2016-04-13	Converted to new Test Case ID conventions as defined in TSTO v4.1.
	1.0.1r04	2016-04-22	Reviewed by Alicia Courtney. Editorial changes, including corrections to new Test Case ID conventions.
	1.0.1r05	2016-05-12	Removal of section 4.1.3 Other General Information; the information there was not essential to the document
1	1.0.1	2016-07-13	Prepared for TCRL 2016-1 publication.
	1.0.1 edition 2r00	2018-11-29	Editorial changes only. Template updated. Revision History and contributors moved to the end of the document.
	1.0.1 edition 2	2019-12-13	Updated copyright page and confidentiality markings to support new Documentation Marking Requirements, performed minor formatting updates, and accepted all tracked changes to prepare for edition 2 publication.
	p2r00–r02	2023-04-12 – 2023-05-26	TSE 22823 (rating 2): Converted the following test cases to GGIT: HPS/SR/SD/BV-01-C, HPS/SR/SDP/BV-01-C, HPS/SR/DEC/BV-01-C – -06-C, HPS/SR/DES/BV-01-C, HPS/SR/CR/BV-01-C – -03-C, and HPS/SR/CW/BV-01-C – -07-C. The new GGIT converted TCIDs are: HPS/SR/SGGIT/CHA/BV-01-C – -06-C, HPS/SR/SGGIT/SDP/BV-01-C, and HPS/SR/SGGIT/SER/BV-01-C. Updated the TCMT accordingly. Updated the test groups section. Added a Publication Number column to the Revision History. Revised the document numbering convention, setting the last release publication of 1.0.1 as p1. Performed other editorials to align the document with the latest TS template and updated the scope, references, Test Strategy, test case identification conventions, conformance, Pass/Fail verdict conventions, and TCMT introductory text. Changed section titles for single test cases to Heading 8 or 9 per TS template. Replaced the Bluetooth logo in the footer and updated the copyright page to align with v2

Publication Number	Revision Number	Date	Comments
			of the DNMD. Deleted draft revision history comments prior to p0.
2	p2	2023-06-29	Approved by BTI on 2023-05-28. Prepared for TCRL 2023-1 publication.

Acknowledgments

Name	Company
Meagan Schuver	Bluetooth SIG, Inc.
Victor Zhodzishsky	Broadcom
Joe Decuir	CSR
Krishna Shingala	Nordic Semiconductor