

Out-of-Band Pairing Uniform Resource Identifier (OPU)

Bluetooth® Test Suite

- **Revision:** OPU.TS.p0
- **Revision Date:** 2026-09-29
- **Prepared By:** Core Specification Working Group
- **Published during TCRL:** TCRL.pkg104-addition



This document, regardless of its title or content, is not a Bluetooth Specification as defined in the Bluetooth Patent/Copyright License Agreement (“PCLA”) and Bluetooth Trademark License Agreement. Use of this document by members of Bluetooth SIG is governed by the membership and other related agreements between Bluetooth SIG Inc. (“Bluetooth SIG”) and its members, including the PCLA and other agreements posted on Bluetooth SIG’s website located at www.bluetooth.com.

THIS DOCUMENT IS PROVIDED “AS IS” AND BLUETOOTH SIG, ITS MEMBERS, AND THEIR AFFILIATES MAKE NO REPRESENTATIONS OR WARRANTIES AND DISCLAIM ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING ANY WARRANTY OF MERCHANTABILITY, TITLE, NON-INFRINGEMENT, FITNESS FOR ANY PARTICULAR PURPOSE, THAT THE CONTENT OF THIS DOCUMENT IS FREE OF ERRORS.

TO THE EXTENT NOT PROHIBITED BY LAW, BLUETOOTH SIG, ITS MEMBERS, AND THEIR AFFILIATES DISCLAIM ALL LIABILITY ARISING OUT OF OR RELATING TO USE OF THIS DOCUMENT AND ANY INFORMATION CONTAINED IN THIS DOCUMENT, INCLUDING LOST REVENUE, PROFITS, DATA OR PROGRAMS, OR BUSINESS INTERRUPTION, OR FOR SPECIAL, INDIRECT, CONSEQUENTIAL, INCIDENTAL OR PUNITIVE DAMAGES, HOWEVER CAUSED AND REGARDLESS OF THE THEORY OF LIABILITY, AND EVEN IF BLUETOOTH SIG, ITS MEMBERS, OR THEIR AFFILIATES HAVE BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

This document is proprietary to Bluetooth SIG. This document may contain or cover subject matter that is intellectual property of Bluetooth SIG and its members. The furnishing of this document does not grant any license to any intellectual property of Bluetooth SIG or its members.

This document is subject to change without notice.

Copyright ©2026 by Bluetooth SIG, Inc. The Bluetooth word mark and logos are owned by Bluetooth SIG, Inc. Other third-party brands and names are the property of their respective owners.



Contents

1	Scope	4
2	References, definitions, and abbreviations	5
2.1	References	5
2.2	Definitions	5
2.3	Acronyms and abbreviations	5
3	Test Suite Structure (TSS)	6
3.1	Overview	6
3.2	Test Strategy	6
3.3	Test groups	7
4	Test cases (TC)	8
4.1	Introduction	8
4.1.1	Test case identification conventions	8
4.1.2	Conformance	8
4.1.3	Pass/Fail verdict conventions	9
4.2	Decoding	9
4.2.1	Decode OOB Pairing URI, Pair With Device	9
	OPU/OD/DEC/BV-01-C [Decode OOB Pairing URI, Pair With Device, BR/EDR]	9
	OPU/OD/DEC/BV-02-C [Decode OOB Pairing URI, Pair With Device, LE]	9
4.2.2	Invalid URIs, Invalid or Malformed OPU, Pairing not Initiated	10
	OPU/OD/ERR/BI-01-C [Invalid URIs, Invalid or Malformed OPU, Pairing not Initiated, BR/EDR]	10
	OPU/OD/ERR/BI-02-C [Invalid URIs, Invalid or Malformed OPU, Pairing not Initiated, LE]	10
4.2.3	Verify all OPU parsing rules, Initiate Pairing	11
	OPU/OD/ERR/BI-03-C [Verify all OPU parsing rules, Initiate Pairing, BR/EDR]	11
	OPU/OD/ERR/BI-04-C [Verify all OPU parsing rules, Initiate Pairing, LE]	11
4.3	Encoding	12
4.3.1	Encode OOB Pairing URI	12
	OPU/OE/ENC/BV-01-C [Encode OOB Pairing URI, BR/EDR]	12
	OPU/OE/ENC/BV-02-C [Encode OOB Pairing URI, LE]	12
	OPU/OE/ENC/BV-04-C [Verify Vendor-Specific Components Format]	13
	OPU/OE/ERR/BI-02-C [Reject Pairing Requests with insufficient security requirements for OPU]	13
4.3.2	Reject Pairing Request, with the OOB data flag not set, received before the OOB-only pairing timer expires	14
	OPU/OE/ERR/BI-03-C [Reject Pairing Request with the OOB data flag not set received before the OOB-only pairing timer expires, BR/EDR]	14
	OPU/OE/ERR/BI-04-C [Reject Pairing Request with the OOB data flag not set received before the OOB-only pairing timer expires, LE]	14
	OPU/OE/ERR/BI-05-C [Reject Secure Simple Pairing with P-192, BR/EDR]	15
5	Test case mapping	16
6	Revision history and acknowledgments	17



1 Scope

This Bluetooth document contains the Test Suite Structure (TSS) and test cases to test the implementation of the Bluetooth Out-of-Band Pairing Uniform Resource Identifier (URI) specification 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. Additional definitions and abbreviations can be found in [1] and [2].

- [1] Bluetooth Core Specification, Version 5.0 or later
- [2] Test Strategy and Terminology Overview
- [3] Out-of-Band Pairing Uniform Resource Identifier Specification, Version 1.0
- [4] ICS Proforma for Out-of-Band Pairing Uniform Resource Identifier (OPU)
- [5] Characteristic and Descriptor descriptions are accessible via the [Bluetooth SIG Assigned Numbers](#)

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 Overview

3.2 Test Strategy

The test objectives are to verify the functionality of the Out-of-Band Pairing URI 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 Out-of-Band Pairing URI specification defines how Bluetooth pairing information is encoded/decoded. An Encoder generates a QR code with Bluetooth Pairing information and a Decoder uses the Bluetooth Pairing information to Pair with the Encoder.

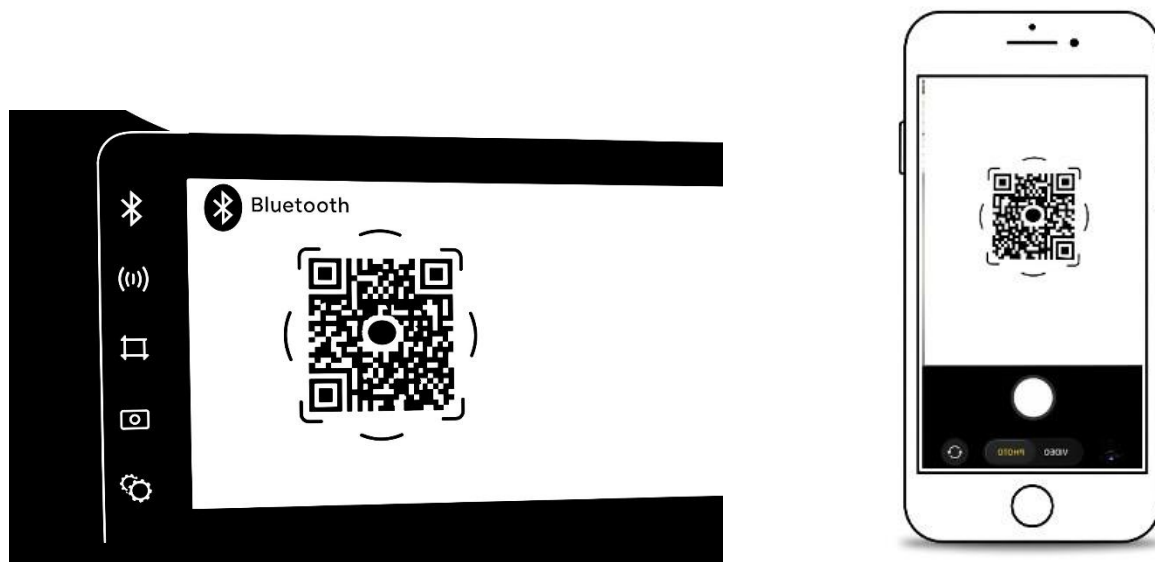


Figure 3.1: A phone pairing with the car kit by scanning an OPU displayed on the car display.

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.

3.3 Test groups

The following test groups have been defined:

- Encode
- Decode
- Error Handling

4 Test cases (TC)

4.1 Introduction

4.1.1 Test case identification conventions

Test cases are assigned unique identifiers per the conventions in [2]. The convention used here is: **<spec abbreviation>/<IUT role>/<class>/<feat>/<func>/<subfunc>/<cap>/<xx>-<nn>-<y>**.

Identifier Abbreviation	Spec Identifier <spec abbreviation>
OPU	Out-of-Band Pairing URI
Identifier Abbreviation	Role Identifier <IUT role>
OD	OPU Decoder
OE	OPU Encoder
Identifier Abbreviation	Features and Behaviors Identifier <feat>
DEC	Decode
ENC	Encode
ERR	Error Handling

Table 4.1: OPU 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 exists, 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 the Bluetooth SIG qualification tool, 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 Decoding

4.2.1 Decode OOB Pairing URI, Pair With Device

- Test Purpose
Verify that an IUT can decode an OPU and use Secure Connections OOB Pairing.
- Reference
[\[3\]](#) 3.1.2, 5.6
- Initial Condition
 - The Lower Tester can generate an OOB Pairing URI.
- Test Case Configuration

Test Case	Transport
OPU/OD/DEC/BV-01-C [Decode OOB Pairing URI, Pair With Device, BR/EDR]	BR/EDR
OPU/OD/DEC/BV-02-C [Decode OOB Pairing URI, Pair With Device, LE]	LE

Table 4.2: Decode OOB Pairing URI, Pairing With Device test cases

- Test Procedure
Repeat the test procedure at least 5 times using a random selection of supported data elements, and values if appropriate, eventually targeting every supported data element in the ICS [\[4\]](#) and one or more unknown data elements.
 1. The Lower Tester generates an OOB Pairing URI using the targeted parameters for this round.
 2. The IUT receives the OOB Pairing URI and parses the data elements.
 3. The IUT initiates a connection with the Lower Tester.
 4. The IUT initiates the pairing procedure with the Lower Tester using the data parsed in Step 2. If the Transport is BR/EDR, the OOB_Auth_Data flag included in the LMP_IO_CAPABILITY_REQ PDU is set to OOB Authentication Data received (0x01). If the Transport is LE, the OOB_Auth_Data flag is set to 0x01 in the SMP Pairing Request.
 5. The Lower Tester disconnects the connection with the IUT.
- Expected Outcome
Pass verdict
In Step 2, the IUT correctly parses all the data elements indicated as supported in the ICS [\[4\]](#).
In Step 4, the IUT uses the decoded Random Number and Confirm value and sets the OOB_Auth_data flag to complete Secure Connections OOB pairing with the Lower Tester.



4.2.2 Invalid URIs, Invalid or Malformed OPU, Pairing not Initiated

- Test Purpose

Verify that an IUT ignores invalid OOB Pairing URI's that have invalid or malformed OPUs. The IUT will not initiate pairing.

- Reference

[3] 5.5

- Initial Condition

- The Lower Tester can generate invalid or malformed OOB Pairing URIs.

- Test Case Configuration

Test Case	Transport
OPU/OD/ERR/BI-01-C [Invalid URIs, Invalid or Malformed OPU, Pairing not Initiated, BR/EDR]	BR/EDR
OPU/OD/ERR/BI-02-C [Invalid URIs, Invalid or Malformed OPU, Pairing not Initiated, LE]	LE

Table 4.3: Invalid URIs, Invalid or Malformed OPU, Pairing not Initiated, test cases

- Test Procedure

Repeat the test procedure for each round in Table 4.4.

1. The Lower Tester generates an OPU using the invalid parameter in Table 4.4 and all other valid parameters.
2. The IUT receives the OPU and parses the data elements.
3. The IUT notifies the Upper Tester of an invalid OOB Pairing URI and does not initiate pairing.

Round	Parameter
1	Empty Secure Connections Random Number
2	Empty Secure Connections Confirm Value
3	BR/EDR if Transport Type is LE or LE if Transport Type is BR/EDR
4	Invalid/Malformed Address
5	Invalid Secure Connections Random Number
6	Invalid Confirm Value
7	Missing ';' between identifier:data pairs
8	Extra colon '::' between identifier:data pairs
9	Invalid BLUETOOTH prefix
10	Missing BLUETOOTH: prefix
11	Invalid CORE protocol value
12	Missing CORE:OPU pair
13	Lower Case HEX character in Device address
14	Two instances of the same ID (other than Vendor Specific ID)
15	URL that is not Base64 encoded
16	Manufacturer Name that is not Base64 encoded



Round	Parameter
17	Model Number that is not Base64 encoded
18	Vendor ID that is not Base64 encoded
19	Product ID that contains invalid characters
20	LE Advertising Data that is not Base64 encoded

Table 4.4: Error Handling test cases

- Expected Outcome

Pass verdict

The IUT detects the invalid or malformed OPU, aborts parsing, and does not initiate connection establishment nor pairing.

The IUT notifies the Upper Tester of the invalid OOB Pairing URI.

4.2.3 Verify all OPU parsing rules, Initiate Pairing

- Test Purpose

Verify that an IUT that receives an OOB Pairing URI successfully parses all OPU components and still completes OOB pairing while ignoring the unknown components. Also verify that the IUT ignores all the characters appearing after the end of the OPU (i.e., after the consecutive semi-colons) and space characters outside of Base-64-encoded strings. When leading zeros are omitted, the IUT restores the parsed value back to the original size.

- Reference

[3] 5.3, 5.5, 5.7

- Initial Condition

- The Lower Tester can generate an OOB Pairing URI.

- Test Case Configuration

Test Case	Transport
OPU/OD/ERR/BI-03-C [Verify all OPU parsing rules, Initiate Pairing, BR/EDR]	BR/EDR
OPU/OD/ERR/BI-04-C [Verify all OPU parsing rules, Initiate Pairing, LE]	LE

Table 4.5: Verify all OPU parsing rules, Initiate Pairing test cases

- Test Procedure

Repeat the test procedure for each round in Table 4.6.

- The Lower Tester generates an OPU using the invalid parameter in Table 4.6 and all other valid parameters.
- The IUT receives the OPU and parses the data elements.
- The IUT initiates a connection with the Lower Tester.
- The IUT initiates the pairing procedure with the Lower Tester using the data parsed in Step 2.
- The Lower Tester disconnects the connection with the IUT.



Round	Parameter
1	Add “;;DN:ZHVwbGljYXRI” to the end of the OPU
2	Add the identifier:value pair “UNKNOWN:abcd;” in the middle of the OPU
3	Add 3 spaces before each of the Base-64 encoded strings
4	Create 3 random unknown identifier:value pairs.

Table 4.6: Error Handling test cases

- Expected Outcome

Pass verdict

The IUT ignores the unknown or malformed Extended components and completes OOB pairing.

4.3 Encoding

4.3.1 Encode OOB Pairing URI

- Test Purpose

Verify that an IUT can generate an OOB Pairing URI and accept a pairing request.

- Reference

[3] 3.1.1

- Initial Condition

- The IUT can generate an OOB Pairing URI.
- The Lower Tester can receive an OOB URI.

- Test Case Configuration

Test Case	Transport
OPU/OE/ENC/BV-01-C [Encode OOB Pairing URI, BR/EDR]	BR/EDR
OPU/OE/ENC/BV-02-C [Encode OOB Pairing URI, LE]	LE

Table 4.7: Encode OOB Pairing URI test cases

- Test Procedure

Repeat the test procedure 5 times.

- The Lower Tester formulates the targeting data elements and values.
- The Upper Tester commands the IUT to generate an OPU using all supported parameters indicated as supported in the ICS [4].
- The Upper Tester submits the generated OPU to the Lower Tester.
- The Lower Tester initiates an Out-of-Band pairing request with the IUT using the OPU parameters in Step 3.
- The IUT and the Lower Tester have successfully completed Out-of-Band pairing.
- The IUT and the Lower Tester disconnect the connection.

- Expected Outcome

Pass verdict

In Step 3, the Lower Tester verifies that the IUT correctly encoded all the data elements indicated as supported in the ICS [4].



In Step 3, the IUT generates unique Secure Connections Random numbers and Confirm values in each repeat.

In Step 4 for the LE transport, the IUT generates unique Public Keys in each repeat.

OPU/OE/ENC/BV-04-C [Verify Vendor-Specific Components Format]

- Test Purpose

Verify that an IUT that supports vendor-specific components uses the correct format.

- Reference

[3] 5.7.1

- Initial Condition

- The IUT can generate an OOB Pairing URI.
- The Lower Tester can receive an OOB URI.

- Test Procedure

1. The Lower Tester formulates the targeting data elements and values.
2. The Upper Tester commands the IUT to generate an OPU using all supported parameters indicated as supported in the ICS [4], including a Vendor Specific component.
3. The Upper Tester submits the generated OPU to the Lower Tester.
4. The Lower Tester verifies that the decoded “VS” identifier starts with either a “CID” with 4 hex characters or “OID” with 2 hex characters followed by data of any length.

- Expected Outcome

Pass verdict

In Step 4, the data in the VS component is encoded with Base64 and has the correct CID or OID identifiers followed by data of any length.

OPU/OE/ERR/BI-02-C [Reject Pairing Requests with insufficient security requirements for OPU]

- Test Purpose

Verify that an IUT rejects a Pairing Request that does not meet the security requirements of OPU.

- Reference

[3] 3.1.1

- Initial Condition

- The IUT can generate an OOB Pairing URI.
- The Lower Tester can receive an OOB URI.

- Test Procedure

Repeat the test procedure for each round in Table 4.8.

1. The Lower Tester formulates the targeting data elements and values.
2. The Upper Tester commands the IUT to generate an OPU using all supported parameters indicated as supported in the ICS [4].
3. The Upper Tester submits the generated OPU to the Lower Tester.



4. The Lower Tester initiates a connection and a pairing procedure with the Invalid Pairing Procedure specified in [Table 4.8](#).
5. The IUT aborts the pairing procedure initiated by the Lower Tester.

Round	Invalid Pairing Procedure
1	LE Legacy Pairing Request
2	64-bit LTK

Table 4.8: Error Handling test cases

- Expected Outcome

Pass verdict

In Step 5, the IUT aborts the Pairing request.

4.3.2 Reject Pairing Request, with the OOB data flag not set, received before the OOB-only pairing timer expires

- Test Purpose

Verify that an IUT that implements the OPU Encoder role that encodes an OPU for the purpose of OOB pairing correctly applies a timer. If the pairing timer has a non-zero timeout, the IUT rejects a Pairing Request that is received before the timer expires with the OOB data flag not set. If the pairing timer has a zero timeout, the pairing request is always rejected regardless of when the request is initiated.

- Reference

[3] 3.1.1, 6.2, 6.3

- Initial Condition

- The IUT can generate an OOB Pairing URI.
- The Lower Tester can initiate a pairing request.
- The OPU pairing timeout is defined by the TSPX_OPU_Pairing_TO IXIT value.

- Test Case Configuration

Test Case	Transport	OOB
OPU/OE/ERR/BI-03-C [Reject Pairing Request with the OOB data flag not set received before the OOB-only pairing timer expires, BR/EDR]	BR/EDR	OOB_Auth_Data set to 0x00
OPU/OE/ERR/BI-04-C [Reject Pairing Request with the OOB data flag not set received before the OOB-only pairing timer expires, LE]	LE	OOB Data flag set to 0 in the Pairing Feature Exchange

Table 4.9: Reject Pairing Request with OPU Pairing Timeout, Peer does not support OOB test cases

- Test Procedure

1. The Lower Tester formulates the targeting data elements and values.
2. The Upper Tester commands the IUT to generate an OPU using all supported parameters indicated as supported in the ICS [4].
3. The Upper Tester submits the generated OPU to the Lower Tester.



4. Before the TSPX_OPU_Pairing_TO, the Lower Tester initiates a pairing request with the IUT with OOB as specified in [Table 4.9](#). If TSPX_OPU_Pairing_TO is 0, the pairing request can be initiated at any time.
5. Perform either alternative 5A or 5B depending on the Transport.

Alternative 5A (LE):

5A.1 The IUT rejects the pairing request from Step 4.

Alternative 5B (BR/EDR):

5B.1 The IUT rejects the IO Capability request with error 0x0E (Rejected due to Security Reasons).

- Expected Outcome

Pass verdict

In Step 5A.1, the IUT rejects the pairing request.

In Step 5B.1, the IUT rejects the authentication procedure with error 0x0E.

OPU/OE/ERR/BI-05-C [Reject Secure Simple Pairing with P-192, BR/EDR]

- Test Purpose

Verify that an IUT terminates a connection when a Secure Simple Pairing request with P-192 elliptic curve is received from the Lower Tester.

- Reference

[\[3\]](#) 6.2

- Initial Condition

- The IUT can generate an OOB Pairing URI.
- The Lower Tester can receive an OOB URI.

- Test Procedure

1. The Lower Tester formulates the targeting data elements and values.
2. The Upper Tester commands the IUT to generate an OPU using all supported parameters indicated as supported in the ICS [\[4\]](#).
3. The Upper Tester submits the generated OPU to the Lower Tester.
4. The Lower Tester initiates a Secure Simple Pairing procedure with the IUT with OOB_Auth_Data set to 1.
5. The IUT rejects the IO Capability Response with no OOB Data present with 0x0E (Rejected due to Security Reasons).

- Expected Outcome

Pass verdict

In Step 5, the IUT rejects the authentication procedure with error 0x0E.



5 Test case mapping

The Test Case Mapping Table (TCMT) maps test cases to specific requirements 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 Out-of-Band Pairing Uniform Resource Identifier [4].

If a test case is mandatory within the respective layer, then the y/x reference is omitted.

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. Further details about the function of the TCMT are elaborated in [2].

For the purpose and structure of the ICS/IXIT, refer to [2].

Item	Feature	Test Case(s)
OPU 3/2 AND OPU 2/1	OPU Decoder, BR/EDR	OPU/OD/DEC/BV-01-C OPU/OD/ERR/BI-01-C OPU/OD/ERR/BI-03-C
OPU 3/2 AND OPU 2/2	OPU Decoder, LE	OPU/OD/DEC/BV-02-C OPU/OD/ERR/BI-02-C OPU/OD/ERR/BI-04-C
OPU 3/1 AND OPU 2/1	OPU Encoder, BR/EDR	OPU/OE/ENC/BV-01-C OPU/OE/ERR/BI-03-C OPU/OE/ERR/BI-05-C
OPU 3/1 AND OPU 2/2	OPU Encoder, LE	OPU/OE/ENC/BV-02-C OPU/OE/ERR/BI-02-C OPU/OE/ERR/BI-04-C
OPU 10/7	OPU Encoder, Vendor-Specific component	OPU/OE/ENC/BV-04-C

Table 5.1: Test case mapping

6 Revision history and acknowledgments

Revision History

Publication Number	Revision Number	Date	Comments
0	p0	2026-09-29	Approved by BTI on 2026-09-09. OPU v1.0 adopted by the BoD on 2026-09-28. Prepared for initial publication.

Acknowledgments

Name	Company
Gene Chang	Bluetooth SIG, Inc.