Object Transfer Service (OTS)

***Bluetooth*® Test Suite**

- **Revision:** OTS.TS.p4
- **Revision Date:** 2022-06-28
- **Group Prepared By:** Sport and Fitness Working Group

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.

Contents

1 Scope

This Bluetooth document contains the Test Suite Structure (TSS) and test cases to test the implementation of the Object Transfer Service 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, by dated or undated reference, provisions from other publications. 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], [2], and [3].

[1] Test Strategy and Terminology Overview

[2] Bluetooth Core Specification, Version 4.0 or later

[3] Object Transfer Service Specification, Version 1.0

[4] ICS Proforma for Object Transfer Service, OTS.ICS

[5] GATT Test Suite, GATT.TS

[6] Characteristic and Descriptor descriptions and fixed Protocol Service Multiplexer (PSM) values are accessible via the Bluetooth SIG Assigned Numbers

[7] Object Transfer Service Implementation eXtra Information for Test, OTS.IXIT

2.2 Definitions

In this Bluetooth document, the definitions from [1], [2], and [3] apply.

2.3 Acronyms and abbreviations

In this Bluetooth document, the definitions, acronyms, and abbreviations from [1], [2], and [3] apply.

3 Test Suite Structure (TSS)

3.1 Overview

The Object Transfer Service requires the presence of L2CAP, GAP, SM (for LE), SDP (for BR/EDR), ATT and GATT. This is illustrated in Figure 3.1. “L2CAP CoC” in this diagram denotes an L2CAP connection-oriented channel, hereinafter referred to as the “Object Transfer Channel”. The transfer of objects as bulk data is achieved by instructing the Server to send or receive data through the Object Transfer Channel.

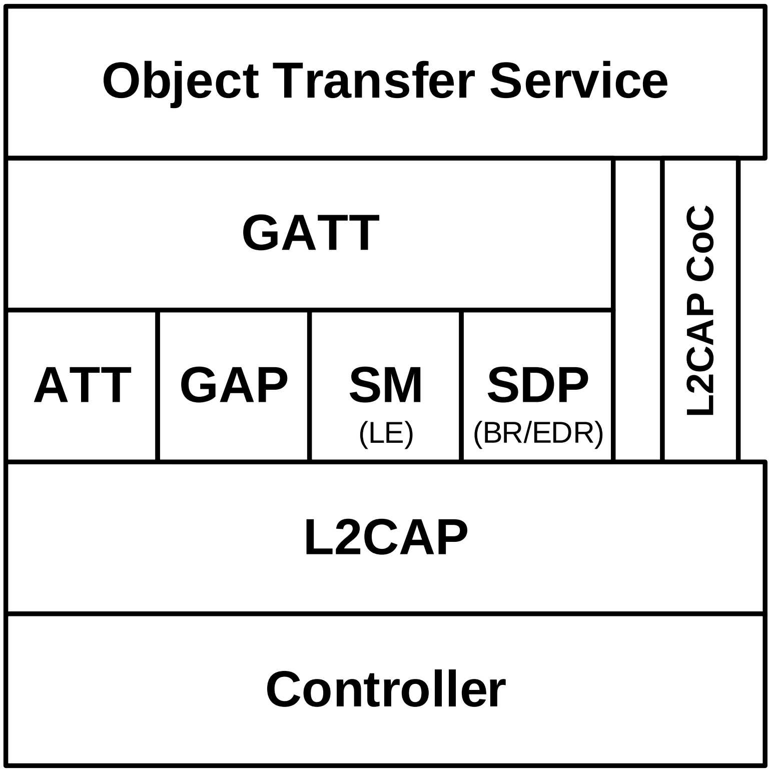

Figure 3.1: Object Transfer Service test model

The fixed Protocol Service Multiplexer (PSM) value, «PSM_OTS», used for the Object Transfer protocol is accessible via Bluetooth Assigned Numbers [6].

3.2 Test Strategy

The test objectives are to verify functionality of the Object Transfer Service within a Bluetooth Host and enable interoperability between Bluetooth Hosts on different devices. The testing approach is to cover mandatory and optional requirements in the specification and to match 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.

Note that the Execute procedure within the Object Action Control Point service procedures group is not tested in this test suite because its functionality is intentionally left by the service specification to be defined by a higher specification. It is therefore expected that profiles that use the Object Transfer Service will define incremental Server requirements covering the Execute procedure and these incremental requirements will then be tested in the associated profile test suite.

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.

Some test cases herein require two Lower Testers with different Bluetooth addresses to verify the IUT's capability to handle multiple simultaneous connections.

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:

- Generic GATT Integrated Tests
- Characteristic Declaration
- Characteristic Read
- Characteristic Read Long
- Characteristic Write
- Characteristic Write Long
- Invariant Characteristic Behavior
- Configure Indication
- Service Procedures – Object Action Control Point (OACP)
- Service Procedures – Object List Control Point (OLCP)
- Object Metadata – Error Handling
- Real Time Clock
- Object List Filter Operation
- Object Changed Operation
- Object Transfer Timeout and Disconnection
- Directory Listing Object

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>**
OTS	Object Transfer 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>**
CON	Configure Indication
CR	Characteristic Read
CRL	Characteristic Read Long
CW	Characteristic Write
CWL	Characteristic Write Long
DEC	Characteristic Declaration
DLO	Directory Listing Object
INV	Invariant Behavior
OAE	OACP Service Procedure – Error Handling
OASP	Service Procedures – Object Action Control Point (OACP)
OC	Object Changed Operation
OLE	OLCP Service Procedure – Error Handling
OLF	Object List Filter Operation
OLSP	Service Procedures – Object List Control Point (OLCP)
OME	Object Metadata – Error Handling
OTD	Object Transfer Timeout and Disconnection
RTC	Real Time Clock

Table 4.1: Object Transfer Service TC feature naming convention

4.1.2 Conformance

When conformance is claimed for a particular specification, all capabilities are to be supported in the specified manner (process-mandatory). 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 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, the outcome of the test is a Fail verdict.

4.2 Setup preambles and generic test sequences

The procedures defined in this section are provided for information, as they are used by test equipment in achieving the initial conditions in certain tests.

4.2.1 ATT Bearer

If LE is specified, the setup procedure defined in Section 4.2.1.1 is executed.

If BR/EDR is specified, the setup procedure defined in Section 4.2.1.2 is executed.

4.2.1.1 ATT Bearer on LE Transport

- Preamble Procedure

1. Establish an LE transport connection between the IUT and the Lower Tester.
2. Establish an L2CAP channel 0x0004 between the IUT and the Lower Tester over that LE transport.

4.2.1.2 ATT Bearer on BR/EDR Transport

- Preamble Procedure

1. Establish a BR/EDR transport connection between the IUT and the Lower Tester.
2. Establish several L2CAP channels (PSM 0x001F) between the IUT and the Lower Tester over that BR/EDR transport.

4.2.2 Initial Configuration

- Preamble Purpose

  Follow this preamble procedure to enable the IUT for use with the Object Action Control Point. In addition, where the Object List Control Point characteristic or Object Changed characteristic is supported by the IUT, the Client Characteristic Configuration Descriptor of each is configured to enable indications.

- Preamble Procedure

1. Establish an ATT Bearer connection between the Lower Tester and IUT as described in Section 4.2.1. If IUT permissions for the Object Action Control Point characteristic require a specific security mode or security level, establish a connection meeting those requirements.
2. The handles of the OTS characteristics have been previously discovered by the Lower Tester during the test procedures in Section 4.3 or are known to the Lower Tester by other means.
3. The handles of the Client Characteristic Configuration descriptors of the supported characteristics have been previously discovered by the Lower Tester during the test procedures in Section 4.3 or are known to the Lower Tester by other means.
4. If the IUT requires bonding, then the Lower Tester performs a bonding procedure.
5. The Lower Tester configures the Object Action Control Point characteristic for indications by writing the value 0x0002 to the Client Characteristic Configuration descriptor of the characteristic.
6. If the Object List Control Point is supported by the IUT, the Lower Tester also configures this characteristic for indications by writing the value 0x0002 to the Client Characteristic Configuration descriptor of the characteristic.
7. If the Object Changed characteristic is supported by the IUT, the Lower Tester also configures this characteristic for indications by writing the value 0x0002 to the Client Characteristic Configuration descriptor of the characteristic.

4.2.3 Object Discovery

- Preamble Purpose

  Follow this generic test sequence to enable the Lower Tester to use the OLCP to enumerate a list of the Object Names and associated object metadata for all objects presently available on the IUT.

  The handle of each supported characteristic and characteristic descriptor referenced in the preamble procedure below has already been discovered by the Lower Tester as described in Section 4.2.2 or is known to the Lower Tester by other means.

- Preamble Procedure

  1. Establish an ATT Bearer connection between the Lower Tester and IUT as described in Section 4.2.1. If IUT permissions for the Object List Control Point characteristic require a specific security mode or security level, establish a connection meeting those requirements. If the IUT requires bonding, then the Lower Tester performs a bonding procedure.
  2. The Lower Tester configures the OLCP characteristic for indications by writing the value 0x0002 to the Client Characteristic Configuration descriptor of the characteristic.
  3. If the IUT supports the Object List Filter characteristics, the Lower Tester writes the value ‘No Filter’ (0x00), with no parameter, to all three instances of the Object List Filter characteristic.
  4. The Lower Tester writes the First Op Code (0x01) to the Object List Control Point with no parameter.
  5. The IUT, after sending a Write Response to acknowledge the write to the Object List Control Point, sends an indication of the Object List Control Point characteristic.
  6. The Lower Tester receives an *ATT_Handle_Value_Indication* from the IUT containing the Object List Control Point characteristic handle and value.
  7. The Lower Tester sends an *ATT_Handle_Value_Confirmation* to the IUT.
  8. The Lower Tester reads the values of the Object Name, Object Type, Object Size and Object ID characteristics. If the value of the Object Name is long enough to require the GATT Read Long sub-procedure to be used, it reads the name in its entirety using the GATT Read Long Characteristic Values sub-procedure.
  9. The values read in step 8 are recorded as a set representing the metadata of one object.
  10. The Lower Tester writes the Next Op Code (0x04) to the Object List Control Point with no parameter.
  11. The IUT, after sending a Write Response to acknowledge the write to the Object List Control Point, sends an indication of the Object List Control Point characteristic.
  12. The Lower Tester receives an ATT_Handle_Value_Indication from the IUT containing the Object List Control Point characteristic handle and value.
  13. The Lower Tester sends an ATT_Handle_Value_Confirmation to the IUT.
  14. If the Result Code received in step 12 was ‘Success’, the Lower Tester reads and records the set of values from the Object Name, Object Type, Object Size and Object ID characteristics. The Lower Tester constructs a list of objects by adding the currently selected object’s metadata to a list which the Lower Tester is compiling internally.
  15. If the Result Code received in step 12 was ‘Success’, repeat steps 10–14 until the Result Code is ‘Out of Bounds’.
  16. The Lower Tester retains the list of object metadata that it has compiled while stepping through the available objects one by one as above so that this information about the objects may be used in a subsequent test procedure.

4.2.4 Initialization of Current Object

- Preamble Purpose

  If the Initial Condition for the test case requires one or more objects to be stored on the IUT, they may be placed there by any means supported by the implementation. Follow this preamble procedure to enable the Lower Tester to select the object that it requires to use in the test case. It uses the OLCP procedures to search for an object with a specific Object Name without requiring support for the optional OLCP Go To procedure. This preamble therefore initializes the Current Object and it is used by test procedures that require a known object to be selected for test purposes. Test cases that use this preamble will typically obtain the name of the required object from the IXIT [7].

- Preamble Procedure

  1. Establish an ATT Bearer connection between the Lower Tester and IUT as described in Section 4.2.1. If IUT permissions for the Object List Control Point characteristic require a specific security mode or security level, establish a connection meeting those requirements. If the IUT requires bonding, then the Lower Tester performs a bonding procedure.
  2. The handle of each supported characteristic and characteristic descriptor referenced in the procedure below is discovered by the Lower Tester during the test procedure in Section 4.3 or Section 4.4 or is known to the Lower Tester by other means.
  3. The Lower Tester configures the OLCP characteristic for indications by writing the value 0x0002 to the Client Characteristic Configuration descriptor of the characteristic.
  4. If the IUT supports the Object List Filter characteristics, the Lower Tester writes the value ‘No Filter’ (0x00), with no parameter, to all three instances of the Object List Filter characteristic.
  5. The Lower Tester writes the First Op Code (0x01) to the Object List Control Point with no parameter.
  6. The IUT, after sending a Write Response to acknowledge the write to the Object List Control Point, sends an indication of the Object List Control Point characteristic.
  7. The Lower Tester receives an ATT_Handle_Value_Indication from the IUT containing the Object List Control Point characteristic handle and value.
  8. The Lower Tester sends an ATT_Handle_Value_Confirmation to the IUT.
  9. If the Result Code received in step 7 was not ‘Success’, the IUT has failed to respond as expected and this procedure must terminate unsuccessfully at this point. The Lower Tester continues with steps 10–12 below only if the Result Code received in step 7 was ‘Success’.
  10. The Lower Tester reads the value of the Object Name characteristic. If the value of the Object Name is long enough to require the GATT Read Long sub-procedure to be used, it reads the name in its entirety using the GATT Read Long Characteristic Values sub-procedure.
  11. If the value of the Object Name characteristic does not exactly match the name of the required object, the Lower Tester repeats steps 6–10, after writing the Next Op Code (0x04) to the Object List Control Point with no parameter.
  12. The required object is now the Current Object.

4.3 Generic GATT Integrated Tests

Execute the Generic GATT Integrated Tests defined [5] in Section 6.3 Server Test Procedures using Table 4.2 below as input:

TCID	Service / Characteristic / Descriptor	Reference	Properties	Value Length (Octets)	Service Type
OTS/SR/SGGIT/SER/BV-01-C [Service GGIT – Object Transfer]	Object Transfer Service	[3] 2	-	-	Primary or Secondary Service
OTS/SR/SGGIT/SDP/BV-01-C [SDP Record]	Object Transfer Service	[3] 2, 5	-	-	-
OTS/SR/SGGIT/CHA/BV-01-C [Characteristic GGIT – OTS Feature]	OTS Feature Characteristic	[3] 3	0x02 (Read)	Skip	-
OTS/SR/SGGIT/CHA/BV-02-C [Characteristic GGIT – Object Name]	Object Name Characteristic	[3] 3	0x02 (Read)	Skip	-
OTS/SR/SGGIT/CHA/BV-03-C [Characteristic GGIT – Object Name – Write]	Object Name Characteristic	[3] 3	0x0A (Read, Write)	Skip	-
OTS/SR/SGGIT/CHA/BV-04-C [Characteristic GGIT – Object Type]	Object Type Characteristic	[3] 3	0x02 (Read)	Skip	-
OTS/SR/SGGIT/CHA/BV-05-C [Characteristic GGIT – Object Size]	Object Size Characteristic	[3] 3	0x02 (Read)	Skip	-
OTS/SR/SGGIT/CHA/BV-06-C [Characteristic GGIT – Object First-Created]	Object First-Created Characteristic	[3] 3	0x02 (Read)	Skip	-
OTS/SR/SGGIT/CHA/BV-07-C [Characteristic GGIT – Object First-Created – Write]	Object First-Created Characteristic	[3] 3	0x0A (Read, Write)	Skip	-
OTS/SR/SGGIT/CHA/BV-08-C [Characteristic GGIT – Object Last-Modified]	Object Last-Modified Characteristic	[3] 3	0x02 (Read)	Skip	-
OTS/SR/SGGIT/CHA/BV-09-C [Characteristic GGIT – Object Last-Modified – Write]	Object Last-Modified Characteristic	[3] 3	0x0A (Read, Write)	Skip	-

TCID	Service / Characteristic / Descriptor	Reference	Properties	Value Length (Octets)	Service Type
OTS/SR/SGGIT/CHA/BV-10-C [Characteristic GGIT – Object ID]	Object ID Characteristic	[3] 3	0x02 (Read)	Skip	-
OTS/SR/SGGIT/CHA/BV-11-C [Characteristic GGIT – Object Properties]	Object Properties Characteristic	[3] 3	0x02 (Read)	Skip	-
OTS/SR/SGGIT/CHA/BV-12-C [Characteristic GGIT – Object Properties – Write]	Object Properties Characteristic	[3] 3	0x0A (Read, Write)	Skip	-
OTS/SR/SGGIT/CHA/BV-13-C [Characteristic GGIT – Object Action Control Point]	Object Action Control Point Characteristic	[3] 3	0x28 (Indicate, Write)	Skip	-
OTS/SR/SGGIT/CHA/BV-14-C [Characteristic GGIT – Object List Control Point]	Object List Control Point Characteristic	[3] 3	0x28 (Indicate, Write)	Skip	-
OTS/SR/SGGIT/CHA/BV-15-C [Characteristic GGIT – Object Changed]	Object Changed Characteristic	[3] 3	0x20 (Indicate)	Skip	-

Table 4.2: Input for the GGIT Server Test Procedure

4.4 Characteristic Declaration

- Test Purpose

  Verify that all instances of the Object List Filter characteristic property field meet the requirements of the service.

- Reference

  See Table 4.3.

- Initial Condition

  - The handle range of the service has been previously discovered by the Lower Tester in test case OTS/SR/SGGIT/SER/BV-01-C [Service GGIT – Object Transfer].
  - Establish an ATT Bearer connection between the Lower Tester and IUT as described in Section 4.2.1.

- Test Case Configuration

Test Case	Characteristic Properties Value (Requirements)
OTS/SR/DEC/BV-11-C [Characteristic Declarations – Object List Filter]	0x0A ([3] Table 3.1) Verify that the properties of **all three** instances of the characteristic meet the requirements of this test case.

Table 4.3: Characteristic Declaration test cases

- Test Procedure

  1. The Lower Tester executes the Discover All Characteristics of a Service sub-procedure.
  2. For a discovered characteristic that is listed in Table 4.3, verify that the characteristic properties field of the characteristic declaration meets the requirements of the service.

- Expected Outcome

  <u>Pass verdict</u>

  The characteristic is discovered, and the characteristic properties field of the characteristic declaration meets the requirements of the service as listed in Table 4.3.

4.5 Characteristic Read

- Test Purpose

  Read and verify that the characteristic values required by the service are compliant. The verification is done one value at a time, as enumerated in the test cases in Table 4.4, using this generic test procedure.

- Reference

  See Table 4.4.

- Initial Condition

  - The handle of each characteristic value referenced in the test cases below has been previously discovered by the Lower Tester during the test procedure in Section 4.3 or Section 4.4 or is known to the Lower Tester by other means.
  - At least one object is stored on the IUT.

- Establish an ATT Bearer connection between the Lower Tester and IUT as described in Section 4.2.1. If IUT permissions for the characteristics require a specific security mode or security level, establish a connection meeting those requirements.
- The object that is to be used in this test procedure is identified in the IXIT [7] by its Object Name. If the Object List Control Point (OLCP) is supported by the IUT (i.e., the IUT supports more than one object), run the preamble described in Section 4.2.4 to ensure that the object to be tested is selected as the Current Object.

- Test Case Configuration

Test Case	Value (Requirements)
OTS/SR/CR/BV-01-C [Characteristic Read – OTS Feature]	Defined in [6]. ([3] 3.1)
OTS/SR/CR/BV-02-C [Characteristic Read – Object Name]	Defined in [6]. ([3] 3.2.2)
OTS/SR/CR/BV-03-C [Characteristic Read – Object Type]	Defined in [6]. ([3] 3.2.3)
OTS/SR/CR/BV-04-C [Characteristic Read – Object Size]	Defined in [6]. ([3] 3.2.4)
OTS/SR/CR/BV-05-C [Characteristic Read – Object First-Created]	Defined in [6]. ([3] 3.2.5)
OTS/SR/CR/BV-06-C [Characteristic Read – Object Last-Modified]	Defined in [6]. ([3] 3.2.6)
OTS/SR/CR/BV-07-C [Characteristic Read – Object ID]	Defined in [6]. ([3] 3.2.7)
OTS/SR/CR/BV-08-C [Characteristic Read – Object Properties]	Defined in [6]. ([3] 3.2.8)
OTS/SR/CR/BV-09-C [Characteristic Read – Object List Filter]	Defined in [6]. ([3] 3.5) Verify that all three Object List Filter characteristics are read by inspecting the handles and the values.

Table 4.4: Characteristic Read Value test cases

- Test Procedure

1. The Lower Tester sends an ATT_Read_Request to the IUT to read the characteristic value.
2. The IUT sends an ATT_Read_Response to the Lower Tester.
3. Verify that the characteristic value meets the requirements of the service.

- Expected Outcome

Pass verdict

The characteristic is successfully read, and the characteristic value meets the requirements of the service.

When the OTS Feature characteristic is read, the features that are reported as supported or not supported according to the OTS Feature characteristic value agree with the feature declaration stated in the ICS [4].

4.6 Characteristic Read Long

Read and verify that the characteristic values required by the service are compliant when the GATT Read Long sub-procedure is used.

OTS/SR/CRL/BV-01-C [Characteristic Read Long – Object Name]

- Reference

  [3] 1.4, 3.2.2

- Initial Condition

  - The handle of the Object Name characteristic 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.
  - At least one object is stored on the IUT.
  - Establish an ATT Bearer connection between the Lower Tester and IUT as described in Section 4.2.1. If IUT permissions for the characteristic require a specific security mode or security level, establish a connection meeting those requirements. If the IUT requires bonding, then the Lower Tester performs a bonding procedure.
  - The object that is to be used in this test procedure is identified in the IXIT [7] by its Object Name. If the Object List Control Point (OLCP) is supported by the IUT (i.e., the IUT supports more than one object), run the preamble described in Section 4.2.4 to ensure that the object to be tested is selected as the Current Object.
  - For the purposes of this test case, the Lower Tester does not permit a larger ATT_MTU size than the default ATT_MTU size to be negotiated.
  - The length of the value of the Object Name characteristic associated with the Current Object is sufficiently long that it exceeds the capacity of the default ATT_MTU size so that it is necessary to use the GATT Read Long sub-procedure in order to read the characteristic value in its entirety. The string value to be used for the test is provided in the IXIT [7].

- Test Procedure

  1. The Lower Tester executes the GATT Read Long Characteristic Values sub-procedure for the Object Name characteristic.
  2. Verify that the characteristic value meets the requirements of the service.

- Expected Outcome

  Pass verdict

  The characteristic value is successfully read in its entirety and the value meets the requirements of the service.

OTS/SR/CRL/BV-02-C [Characteristic Read Long – Object List Filter]

- Reference

  [3] 1.4, 3.5

- Initial Condition

  - The handles of the Object List Filter characteristics have been previously discovered by the Lower Tester during the test procedure in Section 4.4 or are known to the Lower Tester by other means.

  - At least one object is stored on the IUT.
  - Establish an ATT Bearer connection between the Lower Tester and IUT as described in Section 4.2.1. If IUT permissions for the characteristic require a specific security mode or security level, establish a connection meeting those requirements. If the IUT requires bonding, then the Lower Tester performs a bonding procedure.
  - For the purposes of this test case, the Lower Tester does not permit a larger ATT_MTU size than the default ATT_MTU size to be negotiated.
  - The length of the value of at least one Object List Filter characteristic to be used in the test procedure is sufficiently long that it exceeds the capacity of the default ATT_MTU size so that it is necessary to use the GATT Read Long sub-procedure in order to read the characteristic value in its entirety.
  - The filter condition used in the Object List Filter characteristic that contains the long attribute value are ‘Name Starts With’, ‘Name Ends With’, ‘Name Contains’ or ‘Name is Exactly’. The string value to be used in the parameter field is provided in the IXIT [7].

- Test Procedure

  1. The Lower Tester executes the GATT Read Long Characteristic Values sub-procedure for the Object List Filter characteristics.
  2. Verify that each characteristic value meets the requirements of the service.

- Expected Outcome

  Pass verdict

  The characteristic value is successfully read in its entirety and the value meets the requirements of the service.

  Note: Although three instances of the Object List Filter are supported, verifying that a long attribute value has been read from at least one instance of the characteristic is sufficient for the purposes of this test case.

4.7 Characteristic Write

- Test Purpose

  Write and verify that the characteristic values required by the service are compliant. It is also verified that the value of the Object Last-Modified characteristic is not updated by the IUT in response to values being written to other characteristics, including object metadata characteristics.

  The verification is done one characteristic at a time, as enumerated in the test cases in Table 4.5, using this generic test procedure.

- Reference

  See Table 4.5.

- Initial Condition

  - The handle of each characteristic value referenced in the test cases below has been previously discovered by the Lower Tester during the test procedure in Section 4.3 or Section 4.4 or is known to the Lower Tester by other means.
  - At least one object is stored on the IUT.

- Establish an ATT Bearer connection between the Lower Tester and IUT as described in Section 4.2.1. If IUT permissions for the characteristic require a specific security mode or security level, establish a connection meeting those requirements.
- The object that is to be used in this test procedure is identified in the IXIT [7] by its Object Name. If the Object List Control Point (OLCP) is supported by the IUT (i.e., the IUT supports more than one object), run the preamble described in Section 4.2.4 to ensure that the object to be tested is selected as the Current Object.
- If the Object Last-Modified characteristic is supported by the IUT, the Lower Tester has read and stored the value of the Object Last-Modified characteristic.

- Test Case Configuration

Test Case	Value (Requirements)
OTS/SR/CW/BV-01-C [Characteristic Write – Object Name]	Defined in [6]. ([3] 3.2.2)
OTS/SR/CW/BV-02-C [Characteristic Write – Object First-Created]	Defined in [6]. ([3] 3.2.5)
OTS/SR/CW/BV-03-C [Characteristic Write – Object Properties]	Defined in [6]. ([3] 3.2.8)
OTS/SR/CW/BV-04-C [Characteristic Write – Object List Filter]	Defined in [6]. ([3] 3.5.1)
OTS/SR/CW/BV-05-C [Characteristic Write – Object Last-Modified]	Defined in [6]. ([3] 3.2.6)

Table 4.5: Characteristic Write Value test cases

- Test Procedure

1. The Lower Tester sends an ATT_Write_Request to the IUT to write the characteristic value.
2. The IUT sends an ATT_Write_Response to the Lower Tester.
3. The Lower Tester sends an ATT_Read_Request to the IUT to read the characteristic value written in step 1.
4. The IUT sends an ATT_Read_Response to the Lower Tester.
5. If the Object Last-Modified characteristic is supported by the IUT, the Lower Tester additionally reads the value of the Object Last-Modified characteristic, unless this characteristic was already read in step 3.
6. Verify that the characteristic value meets the requirements of the service.

Note that, if OTS/SR/CW/BV-05-C [Characteristic Write – Object Last-Modified] is selected, this test case is run last in order to avoid changing the value of the Object Last-Modified characteristic until after the other test cases in the test group have been performed.

- Expected Outcome

Pass verdict

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

If the characteristic written in step 1 is not the Object Last-Modified characteristic, the value of the Object Last-Modified characteristic read in step 5 is identical to the value found during the Initial Condition.

4.8 Characteristic Write Long

- Test Purpose

  Write and verify that the characteristic values required by the service are compliant. The verification is done one value at a time, as enumerated in the test cases in Table 4.6, using this generic test procedure.

- Reference

  [3] 1.4. See also Table 4.6.

- Initial Condition

  - The handle of each characteristic value referenced in the test cases below has been previously discovered by the Lower Tester during the test procedure in Section 4.3 or Section 4.4 or is known to the Lower Tester by other means.
  - At least one object is stored on the IUT.
  - Establish an ATT Bearer connection between the Lower Tester and IUT as described in Section 4.2.1. If IUT permissions for the characteristic require a specific security mode or security level, establish a connection meeting those requirements. If the IUT requires bonding, then the Lower Tester performs a bonding procedure.
  - The object that is to be used in this test procedure is identified in the IXIT [7] by its original Object Name. If the Object List Control Point (OLCP) is supported by the IUT (i.e., the IUT supports more than one object), run the preamble described in Section 4.2.4 to ensure that the object to be tested is selected as the Current Object.
  - For the purposes of this test case, the Lower Tester does not permit a larger ATT_MTU size than the default ATT_MTU size to be negotiated.
  - The length of the value to be written to the characteristic is sufficiently long such that it exceeds the capacity of the default ATT_MTU size so that it is necessary to use the GATT Write Long sub-procedure to write the characteristic value in its entirety and the GATT Read Long sub-procedure in order to read the characteristic value back in its entirety. Therefore, the test value used is more than (ATT_MTU-1) octets in length. The string value to be used for the test is provided in the IXIT [7].

- Test Case Configuration

Test Case	Value (Requirements)
OTS/SR/CWL/BV-01-C [Characteristic Write Long – Object Name]	Defined in [6]. ([3] 3.2.2)
OTS/SR/CWL/BV-02-C [Characteristic Write Long – Object List Filter]	Defined in [6]. ([3] 3.5) The filter condition is 'Name Starts With', 'Name Ends With', 'Name Contains' or 'Name is Exactly'. Verifying that a long attribute value has been written to one instance of the Object List Filter characteristic is sufficient for the purposes of this test case.

Table 4.6: Characteristic Write Long Value test cases

- Test Procedure

  1. The Lower Tester executes the GATT Read Long Characteristic Values sub-procedure.

2. The Lower Tester executes the GATT Write Long Characteristic Values sub-procedure for the characteristic in step 1.
3. The Lower Tester executes the GATT Read Long Characteristic Values sub-procedure for the characteristic in step 2.
4. Verify that the characteristic value meets the requirements of the service.
5. Write the original value that was read in step 1 back to the characteristic. This step is included to ensure there is no impact on a subsequent test case.

- Expected Outcome

  Pass verdict

  The characteristic value is successfully written in its entirety and the characteristic value returned when read back in step 3 is consistent with the value written in step 2.

4.9 Invariant Characteristic Behavior

Verify that the value of each invariant characteristic supported by the IUT is reset to the value required by the service upon each disconnection/reconnection.

OTS/SR/INV/BV-01-C [Invariant – Object List Filter]

- Reference

  [3] 3.5.1

- Initial Condition

  - The following test procedure does not require an object to be selected.
  - The handles of the three Object List Filter characteristics have been previously discovered by the Lower Tester during the test procedure in OTS/SR/CR/BV-09-C [Characteristic Read – Object List Filter] or are 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 IUT permissions for the characteristic require a specific security mode or security level, establish a connection meeting those requirements.
  - If the IUT allows bonding, the Lower Tester performs a bonding procedure; bonding is performed to check that the IUT does not treat the characteristic values under test as persistent values.

- Test Procedure

  1. The Lower Tester writes values to each of the three Object List Filter characteristics as described in OTS/SR/CW/BV-04-C [Characteristic Write – Object List Filter]. The values written are greater than zero and are not RFU values, and a valid parameter value is included to satisfy the requirements of the service where required.
  2. After receiving the third Write Response from the IUT, the Lower Tester disconnects.
  3. Following disconnection, the Lower Tester reconnects to the IUT as described in Section 4.2.1.
  4. The Lower Tester reads the value of each of the three Object List Filter characteristics as described in OTS/SR/CR/BV-09-C [Characteristic Read – Object List Filter].

- Expected Outcome

  Pass verdict

  Write Responses are received during steps 1 and 2, confirming that all three Object List Filter characteristics have been written successfully.

  The value of each Object List Filter characteristic read in step 4 is 0x00 (No Filter) with no parameter.

4.10 Configure Indication

- Test Purpose

  Verify compliant operation in response to disabling and enabling characteristic indication. The verification is done one value at a time, as enumerated in the test cases in Table 4.7, using this generic test procedure.

- Reference

  See Table 4.7.

- Initial Condition

  - The handle of each characteristic value referenced in the test cases below 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.
  - The handle of the client characteristic configuration descriptor of each characteristic referenced in the test cases below 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.
  - If IUT permissions for the characteristic descriptor require a specific security mode or security level, establish a connection meeting those requirements.
  - Establish an ATT Bearer connection between the Lower Tester and IUT as described in Section 4.2.1.

- Test Case Configuration

Test Case	Value (Requirements)
OTS/SR/CON/BV-01-C [Configure Indication – OACP]	0x0002 ([3] 3, 3.3)
OTS/SR/CON/BV-02-C [Configure Indication – OLCP]	0x0002 ([3] 3, 3.4)
OTS/SR/CON/BV-03-C [Configure Indication – Object Changed]	0x0002 ([3] 3, 3.6)

Table 4.7: Configure Indication test cases

- Test Procedure

  1. Disable indication by writing value 0x0000 to the Client Characteristic Configuration descriptor of the characteristic.
  2. The Lower Tester reads the value of the Client Characteristic Configuration descriptor.
  3. Enable indication by writing value 0x0002 to the Client Characteristic Configuration descriptor of the characteristic.
  4. 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 in steps 2 and 4 is consistent with the value written.

4.11 Service Procedures – Object Action Control Point

Verify compliant operation of the Object Action Control Point procedures.

OTS/SR/OASP/BV-01-C [OACP - Create]

- Test Purpose

  Verify that the IUT can perform the OACP Create procedure

- Reference

  [3] 3.3.2.1

- Initial Condition

  - Perform the preamble described in Section 4.2.2.
  - The object types and the minimum and maximum sizes of objects supported by the IUT are known from the IXIT [7].

- Test Procedure

  1. The Lower Tester writes the Create Op Code (0x01) to the Object Action Control Point with a Parameter Value representing an object size and object type that is supported by the IUT e.g., the Size field could contain 0x00000400 (representing an allocated object size of 1024 octets) – the range of acceptable parameter values will depend on the implementation and is known from the IXIT – and the Type field could contain the UUID «Firmware».
  2. The IUT, after sending a Write Response to acknowledge the write to the Object Action Control Point, sends an indication of the Object Action Control Point characteristic.
  3. The Lower Tester receives an ATT_Handle_Value_Indication from the IUT containing the Object Action Control Point characteristic handle and value.
  4. The Lower Tester sends an ATT_Handle_Value_Confirmation to the IUT.
  5. The Lower Tester reads all of the supported Object Metadata characteristic values associated with the newly created object, using the procedure described in Section 4.5. The values reported are noted.

- Expected Outcome

  <u>Pass verdict</u>

  The IUT sends an indication of the Object Action Control Point characteristic containing the Response Code Op Code (0x60), the Request Op Code (0x01) followed by the Result Code for 'Success' (0x01) with no Response Parameter.

  The value reported in the Object Type characteristic is a UUID equal to the value of the Type parameter included with the Create Op Code.

  The value reported in the Allocated Size field of the Object Size characteristic is greater than or equal to the value of the Size parameter included with the Create Op Code.

  The value reported in the Current Size field of the Object Size characteristic is 0.

  The value reported in the Object ID characteristic, if supported, is a unique Object ID (i.e., locally unique within the service) that is not a RFU value and meets the requirements of the service.

  The value reported in the Object Name characteristic is a zero length string.

  The value of each field of the Object First-Created characteristic, if supported, is 0.

  The value of each field of the Object Last-Modified characteristic, if supported, is 0.

The value of the Object Properties characteristic, if supported, includes at least bit 3 set to 1 (i.e., Write is True).

OTS/SR/OASP/BV-02-C [OACP - Delete]

- Test Purpose

  Verify that the IUT can perform the Delete procedure.

- Reference

  [3] 3.3.2.2, 1.9

- Initial Condition

  - At least one test object is stored on the IUT.
  - Perform the preamble described in Section 4.2.2.
  - The object that is to be used in this test procedure is identified in the IXIT [7] by its Object Name. If the Object List Control Point (OLCP) is supported by the IUT (i.e., the IUT supports more than one object), run the preamble described in Section 4.2.4 to ensure that the object to be tested is selected as the Current Object.
  - The object properties permit deletion.

- Test Procedure

  1. The Lower Tester writes the Delete Op Code (0x02) to the Object Action Control Point with no Parameter.
  2. The IUT, after sending a Write Response to acknowledge the write to the Object Action Control Point, sends an indication of the Object Action Control Point characteristic.
  3. The Lower Tester receives an ATT_Handle_Value_Indication from the IUT containing the Object Action Control Point characteristic handle and value.
  4. The Lower Tester sends an ATT_Handle_Value_Confirmation to the IUT.
  5. The Lower Tester writes the Delete Op Code (0x02) to the Object Action Control Point, with no Parameter, a second time.
  6. The IUT, after sending a Write Response to acknowledge the write to the Object Action Control Point, sends an indication of the Object Action Control Point characteristic.
  7. The Lower Tester receives an ATT_Handle_Value_Indication from the IUT containing the Object Action Control Point characteristic handle and value.
  8. The Lower Tester sends an ATT_Handle_Value_Confirmation to the IUT.

- Expected Outcome

  Pass verdict

  In step 2, the IUT sends an indication of the Object Action Control Point characteristic containing the Response Code Op Code (0x60), the Request Op Code (0x02) followed by the Result Code for ‘Success’ (0x01) with no Response Parameter.

  In step 6, the IUT sends an indication of the Object Action Control Point characteristic containing the Response Code Op Code (0x60), the Request Op Code (0x02) followed by the Result Code for ‘Invalid Object’ (0x05) with no Response Parameter.

OTS/SR/OASP/BV-03-C [OACP - Calculate Checksum]

- Test Purpose

  Verify that the IUT can perform the Calculate Checksum procedure.

- Reference

  [3] 3.3.2.3

- Initial Condition

  - At least one object is stored on the IUT.
  - Perform the preamble described in Section 4.2.2.
  - The object that is to be used in this test procedure is identified in the IXIT [7] by its Object Name.
  - If the Object List Control Point (OLCP) is supported by the IUT (i.e., the IUT supports more than one object), run the preamble described in Section 4.2.4 to ensure that the object to be tested is selected as the Current Object.

- Test Procedure

  1. The Lower Tester reads the value of the Object Size characteristic.
  2. The Lower Tester writes the Calculate Checksum Op Code (0x03) to the Object Action Control Point with a Length parameter value equal to the Current Size field of the Object Size characteristic that was read in step 1. The Offset parameter is set to 0.
  3. The IUT, after sending a Write Response to acknowledge the write to the Object Action Control Point, sends an indication of the Object Action Control Point characteristic.
  4. The Lower Tester receives an ATT_Handle_Value_Indication from the IUT containing the Object Action Control Point characteristic handle and value.
  5. The Lower Tester sends an ATT_Handle_Value_Confirmation to the IUT.
  6. If the IUT supports the OACP Read procedure, the Lower Tester reads the object and calculates its checksum independently from the IUT by using an ISO/IEC 3309 compliant, 32-bit CRC algorithm.

- Expected Outcome

  Pass verdict

  The IUT sends an indication of the Object Action Control Point characteristic containing the Response Code Op Code (0x60), the Request Op Code (0x03) followed by the Result Code for ‘Success’ (0x01) with a 32-bit Checksum included in the Response Parameter.

  If the IUT supports the OACP Read procedure, the checksum value reported in the Response parameter is identical to the checksum value that was calculated independently by the Lower Tester.

OTS/SR/OASP/BV-04-C [OACP – Read – Entire Object]

- Test Purpose

  Verify that the IUT can perform the Read procedure to read the entire object.

- Reference

  [3] 3.3.2.5

- Initial Condition

  - At least one object is stored on the IUT.
  - Perform the preamble described in Section 4.2.2.
  - The object that is to be used in this test procedure is identified in the IXIT [7] by its Object Name. If the Object List Control Point (OLCP) is supported by the IUT (i.e., the IUT supports more than

one object), run the preamble described in Section 4.2.4 to ensure that the object to be tested is selected as the Current Object.

- The object properties permit reading.
- The Lower Tester knows whether the IUT supports the OACP Calculate Checksum procedure from having read the OTS Feature Characteristic previously, or by other means.

- Test Procedure

1. The Lower Tester reads the value of the Object Size characteristic.
2. The Lower Tester opens an Object Transfer Channel.
3. The Lower Tester writes the Read Op Code (0x05) to the Object Action Control Point with a Length parameter value equal to the Current Size field of the Object Size characteristic value read in step 1. The Offset parameter is set to 0.
4. The IUT, after sending a Write Response to acknowledge the write to the Object Action Control Point, sends an indication of the Object Action Control Point characteristic.
5. The Lower Tester receives an ATT_Handle_Value_Indication from the IUT containing the Object Action Control Point characteristic handle and value.
6. The Lower Tester sends an ATT_Handle_Value_Confirmation to the IUT.
7. The IUT sends data to the Lower Tester via the Object Transfer Channel opened in step 2.
8. The Lower Tester waits for the object transfer operation to complete.
9. The Lower Tester continues with the following additional steps only if the IUT supports the OACP Calculate Checksum procedure. Otherwise, the test case concludes at this point.
10. The Lower Tester writes the Calculate Checksum Op Code (0x03) to the Object Action Control Point with a Length parameter value equal to the value used for the Length parameter in step 3. The Offset parameter is set to 0.
11. The IUT, after sending a Write Response to acknowledge the write to the Object Action Control Point, sends an indication of the Object Action Control Point characteristic.
12. The Lower Tester receives an ATT_Handle_Value_Indication from the IUT containing the Object Action Control Point characteristic handle and value.
13. The Lower Tester sends an ATT_Handle_Value_Confirmation to the IUT.
14. The Lower Tester calculates a checksum of the object data that it read from the IUT independently, using an ISO/IEC 3309 compliant, 32-bit CRC algorithm.

- Expected Outcome

Pass verdict

In step 4, the IUT sends an indication of the Object Action Control Point characteristic containing the Response Code Op Code (0x60), the Request Op Code (0x05) followed by the Result Code for ‘Success’ (0x01) with no Response Parameter.

Object data is received by the Lower Tester via the Object Transfer Channel.

The expected number of octets is received.

The following criteria apply only if the IUT supports the OACP Calculate Checksum procedure:

- In step 11, the IUT sends an indication of the Object Action Control Point characteristic containing the Response Code Op Code (0x60), the Request Op Code (0x03) followed by the Result Code for ‘Success’ (0x01) with a 32-bit Checksum included in the Response Parameter.
- The checksum value reported in the Response Parameter matches the checksum value calculated independently by the Lower Tester in step 14.

OTS/SR/OASP/BV-05-C [OACP – Read – Part of Object]

- Test Purpose

  Verify that the IUT can perform the Read procedure to read part of an object from an Offset position.

- Reference

  [3] 3.3.2.5

- Initial Condition

  - At least one object is stored on the IUT.
  - Perform the preamble described in Section 4.2.2.
  - The object that is to be used in this test procedure is identified in the IXIT [7] by its Object Name. If the Object List Control Point (OLCP) is supported by the IUT (i.e., the IUT supports more than one object), run the preamble described in Section 4.2.4 to ensure that the object to be tested is selected as the Current Object.
  - The object properties permit reading.
  - The Lower Tester knows the test value (non-zero) to use for the Offset parameter in this test case from the IXIT [7].

- Test Procedure

  1. The Lower Tester reads the value of the Object Size characteristic.
  2. The Lower Tester opens an Object Transfer Channel.
  3. The Lower Tester obtains the value to use for the Offset parameter of the Read Op Code from the IXIT [7].
  4. The Lower Tester writes the Read Op Code (0x05) to the Object Action Control Point with a Length parameter value chosen so that the sum of the Length parameter value and the Offset value does not exceed the value of the Current Size field of the Object Size characteristic value read in step 1.
  5. The IUT, after sending a Write Response to acknowledge the write to the Object Action Control Point, sends an indication of the Object Action Control Point characteristic.
  6. The Lower Tester receives an ATT_Handle_Value_Indication from the IUT containing the Object Action Control Point characteristic handle and value.
  7. The Lower Tester sends an ATT_Handle_Value_Confirmation to the IUT.
  8. The IUT sends data to the Lower Tester via the Object Transfer Channel opened in step 2.

- Expected Outcome

  Pass verdict

  The IUT sends an indication of the Object Action Control Point characteristic containing the Response Code Op Code (0x60), the Request Op Code (0x05) followed by the Result Code for ‘Success’ (0x01) with no Response Parameter.

  Object data is received by the Lower Tester via the Object Transfer Channel.

  The expected number of octets is received as specified in the Length parameter value.

OTS/SR/OASP/BV-06-C [OACP – Write – Entire Object]

- Test Purpose

  Verify that the IUT can perform the Write procedure to write or overwrite the entire object contents.

- Reference

  [3] 3.3.2.6

- Initial Condition

  - At least one object is stored on the IUT.
  - Perform the preamble described in Section 4.2.2.
  - The object that is to be used in this test procedure is identified in the IXIT [7] by its Object Name. If the Object List Control Point (OLCP) is supported by the IUT (i.e., the IUT supports more than one object), run the preamble described in Section 4.2.4 to ensure that the object to be tested is selected as the Current Object.
  - The object properties permit writing.

- Test Procedure

  1. The Lower Tester reads the value of the Object Size characteristic.
  2. The Lower Tester opens an Object Transfer Channel.
  3. The Lower Tester writes the Write Op Code (0x06) to the Object Action Control Point with a Length parameter value equal to the Allocated Size field of the Object Size characteristic value read in step 1. The Offset parameter is set to 0. The Mode parameter is set to 0x00.
  4. The IUT, after sending a Write Response to acknowledge the write to the Object Action Control Point, sends an indication of the Object Action Control Point characteristic.
  5. The Lower Tester receives an ATT_Handle_Value_Indication from the IUT containing the Object Action Control Point characteristic handle and value.
  6. The Lower Tester sends an ATT_Handle_Value_Confirmation to the IUT.
  7. The Lower Tester sends data consisting of the number of octets specified in the Length parameter (see step 3) to the IUT via the Object Transfer Channel opened in step 2.
  8. If the IUT supports the OACP Read procedure, the Lower Tester reads back the entire object.

- Expected Outcome

  Pass verdict

  The IUT sends an indication of the Object Action Control Point characteristic containing the Response Code Op Code (0x60), the Request Op Code (0x06) followed by the Result Code for ‘Success’ (0x01) with no Response Parameter.

  Object data is accepted by the IUT via the Object Transfer Channel.

  The quantity of octets specified in the Length parameter is transferred to the IUT.

  If the IUT supports the OACP Read procedure, the object read after executing the procedure contains the data that was written to it.

OTS/SR/OASP/BV-07-C [OACP – Write – Patching an Object]

- Test Purpose

  Verify that the IUT can perform the Write procedure to write or overwrite part of an existing object.

- Reference

  [3] 3.3.2.6

- Initial Condition

  - At least one object is stored on the IUT.

  - Perform the preamble described in Section 4.2.2.
  - The object that is to be used in this test procedure is identified in the IXIT [7] by its Object Name. If the Object List Control Point (OLCP) is supported by the IUT (i.e., the IUT supports more than one object), run the preamble described in Section 4.2.4 to ensure that the object to be tested is selected as the Current Object.
  - The object properties permit writing and patching.
  - The Lower Tester knows non-zero test values to use for the Offset and Length parameters in this test case from the IXIT [7]. These define the portion of the object that is to be patched.

- Test Procedure

  1. The Lower Tester writes the First Op Code (0x01) to the Object List Control Point with no parameter. This ensures that an object has been selected.
  2. The Lower Tester opens an Object Transfer Channel.
  3. The Lower Tester writes the Write Op Code (0x06) to the Object Action Control Point with the Offset and Length parameter values provided in the IXIT [7]. The Mode parameter is set to 0x00.
  4. The IUT, after sending a Write Response to acknowledge the write to the Object Action Control Point, sends an indication of the Object Action Control Point characteristic.
  5. The Lower Tester receives an ATT_Handle_Value_Indication from the IUT containing the Object Action Control Point characteristic handle and value.
  6. The Lower Tester sends an ATT_Handle_Value_Confirmation to the IUT.
  7. The Lower Tester sends data consisting of the number of octets specified in the Length parameter (see step 2) to the IUT via the Object Transfer Channel opened in step 1.
  8. If the IUT supports the OACP Read procedure, the Lower Tester reads back the entire object.

- Expected Outcome

  Pass verdict

  The IUT sends an indication of the Object Action Control Point characteristic containing the Response Code Op Code (0x60), the Request Op Code (0x06) followed by the Result Code for ‘Success’ (0x01) with no Response Parameter.

  Object data is accepted by the IUT via the Object Transfer Channel.

  The quantity of octets specified in the Length parameter is transferred to the IUT.

  If the IUT supports the OACP Read procedure, the object read after executing the procedure contains the data that was written to it, at the correct location.

OTS/SR/OASP/BV-08-C [OACP – Write – Increased Allocated Size]

- Test Purpose

  Verify that the IUT can perform the Write procedure when the parameters necessitate an increase in the allocated size of the object. Verify that the IUT can append octets to the end of an existing object. Verify that the IUT automatically increases the value of the Allocated Size to accommodate the extended object.

- Reference

  [3] 3.3.2.6.1

- Initial Condition

  - At least one object is stored on the IUT.

  - The Lower Tester knows an appropriate number of octets to be written for the purposes of this test case from the IXIT [7].
  - Perform the preamble described in Section 4.2.2.
  - The object that is to be used in this test procedure is identified in the IXIT [7] by its Object Name. If the Object List Control Point (OLCP) is supported by the IUT (i.e., the IUT supports more than one object), run the preamble described in Section 4.2.4 to ensure that the object to be tested is selected as the Current Object.
  - The object to be used in this test procedure initially has a Current Size and Allocated Size that are equal to each other and are greater than 0. The object properties permit appending.
  - In the following test procedure, the octets written are appended to the end of the existing object contents.

- Test Procedure

  1. The Lower Tester reads the value of the Object Size characteristic. The values of the Current Size and Allocated Size fields are noted.
  2. The Lower Tester opens an Object Transfer Channel.
  3. The Lower Tester writes the Write Op Code (0x06) to the Object Action Control Point with the Mode parameter set to 0x00, the Offset parameter set equal to the value of the Current Size field of the Object Size characteristic that was read in step 1 and a Length parameter representing the number of octets to be appended as specified in the IXIT [7].
  4. The IUT, after sending a Write Response to acknowledge the write to the Object Action Control Point, sends an indication of the Object Action Control Point characteristic.
  5. The Lower Tester receives an ATT_Handle_Value_Indication from the IUT containing the Object Action Control Point characteristic handle and value.
  6. The Lower Tester sends an ATT_Handle_Value_Confirmation to the IUT.
  7. The Lower Tester sends the expected number of octets to be appended to the object to the IUT via the Object Transfer Channel opened in step 2.
  8. The Lower Tester reads the value of the Object Size characteristic again. The values of the Current Size and Allocated Size fields are noted.

- Expected Outcome

  Pass verdict

  The IUT sends an indication of the Object Action Control Point characteristic containing the Response Code Op Code (0x60), the Request Op Code (0x06) followed by the Result Code for ‘Success’ (0x01) with no Response Parameter.

  Object data is accepted by the IUT via the Object Transfer Channel.

  The number of octets that were specified in the Length parameter are transferred to the IUT.

  The value of the Current Size field noted in step 8 is larger than the value noted in step 1 and the difference in the values equals the number of octets that were transferred to the IUT.

  The value of the Allocated Size field noted in step 8 is larger than the value noted in step 1 and is greater than or equal to the new Current Size.

OTS/SR/OASP/BV-09-C [OACP – Write – Truncate Object]

- Test Purpose

  Verify that the IUT can perform the Write procedure when the parameters specify that the object is to be truncated to a smaller size.

- Reference

  [3] 3.3.2.6

- Initial Condition

  - At least one object is stored on the IUT.
  - The Lower Tester knows an appropriate number of octets to be written for the purposes of this test case from the IXIT [7].
  - The object properties permit truncation, and the current size of the object is larger than the number of octets that are to be written to it in this test case.
  - Perform the preamble described in Section 4.2.2.
  - The object that is to be used in this test procedure is identified in the IXIT [7] by its Object Name. If the Object List Control Point (OLCP) is supported by the IUT (i.e., the IUT supports more than one object), run the preamble described in Section 4.2.4 to ensure that the object to be tested is selected as the Current Object.

- Test Procedure

  1. The Lower Tester reads the value of the Object Size characteristic. The values of the Current Size and Allocated Size fields are noted.
  2. The Lower Tester opens an Object Transfer Channel.
  3. The Lower Tester writes the Write Op Code (0x06) to the Object Action Control Point with the Mode parameter set to 0x02 (i.e., the Truncate bit = True), the Offset parameter set to 0 and a Length parameter representing the number of octets to be written as specified in the IXIT [7].
  4. The IUT, after sending a Write Response to acknowledge the write to the Object Action Control Point, sends an indication of the Object Action Control Point characteristic.
  5. The Lower Tester receives an ATT_Handle_Value_Indication from the IUT containing the Object Action Control Point characteristic handle and value.
  6. The Lower Tester sends an ATT_Handle_Value_Confirmation to the IUT.
  7. The Lower Tester sends the expected number of octets to be written to the object to the IUT via the Object Transfer Channel opened in step 2.
  8. The Lower Tester reads the value of the Object Size characteristic again. The values of the Current Size and Allocated Size fields are noted.

- Expected Outcome

  Pass verdict

  The IUT sends an indication of the Object Action Control Point characteristic containing the Response Code Op Code (0x60), the Request Op Code (0x06) followed by the Result Code for ‘Success’ (0x01) with no Response Parameter.

  Object data is accepted by the IUT via the Object Transfer Channel.

  The number of octets that were specified in the Length parameter are transferred to the IUT.

  The value of the Current Size field noted in step 8 is smaller than the value noted in step 1 and the new Current Size value equals the number of octets that were transferred to the IUT.

  The value of the Allocated Size field noted in step 8 is the same as the value noted in step 1.

OTS/SR/OASP/BV-10-C [OACP – Abort]

- Test Purpose

  Verify that the IUT can perform the Abort procedure.

- Reference

  [3] 3.3.2.7

- Initial Condition

  - At least one object is stored on the IUT.
  - The object used is large enough so that the object transfer is still in progress when the OACP Abort Op Code is sent during the test procedure. It is necessary for the purposes of this test case that the object transfer does not finish too quickly, as there would be nothing for the Lower Tester to abort. The Lower Tester may also control the flow rate via the Object Transfer Channel to ensure that the object transfer does not complete too quickly.
  - Perform the preamble described in Section 4.2.2.
  - The object that is to be used in this test procedure is identified in the IXIT [7] by its Object Name. If the Object List Control Point (OLCP) is supported by the IUT (i.e., the IUT supports more than one object), run the preamble described in Section 4.2.4 to ensure that the object to be tested is selected as the Current Object.
  - The object properties permit reading.

- Test Procedure

  1. The Lower Tester reads the value of the Object Size characteristic.
  2. The Lower Tester opens an Object Transfer Channel.
  3. The Lower Tester writes the Read Op Code (0x05) to the Object Action Control Point with a Length parameter value equal to the Current Size field of the Object Size characteristic value read in step 1. The Offset parameter is set to 0.
  4. The IUT, after sending a Write Response to acknowledge the write to the Object Action Control Point, sends an indication of the Object Action Control Point characteristic.
  5. The Lower Tester receives an ATT_Handle_Value_Indication from the IUT containing the Object Action Control Point characteristic handle and value.
  6. The Lower Tester sends an ATT_Handle_Value_Confirmation to the IUT.
  7. The Lower Tester begins to receive data from the IUT via the Object Transfer Channel opened in step 2.
  8. The Lower Tester writes the Abort Op Code (0x07) to the Object Action Control Point with no parameter.
  9. The IUT, after sending a Write Response to acknowledge the write to the Object Action Control Point, sends an indication of the Object Action Control Point characteristic.
  10. The Lower Tester receives an ATT_Handle_Value_Indication from the IUT containing the Object Action Control Point characteristic handle and value.
  11. The Lower Tester sends an ATT_Handle_Value_Confirmation to the IUT.
  12. The IUT ceases sending data to the Lower Tester via the Object Transfer Channel.

- Expected Outcome

  Pass verdict

  In step 4, the IUT sends an indication of the Object Action Control Point characteristic containing the Response Code Op Code (0x60), the Request Op Code (0x05) followed by the Result Code for ‘Success’ (0x01) with no Response Parameter.

  Some object data is received by the Lower Tester via the Object Transfer Channel.

  In step 9, the IUT sends an indication of the Object Action Control Point characteristic containing the Response Code Op Code (0x60), the Request Op Code (0x09) followed by the Result Code for ‘Success’ (0x01) with no Response Parameter.

  The flow of data from the IUT via the Object Transfer Channel ceases following reception of the Abort procedure Response indication and no more octets are received through this channel.

4.12 Service Procedures – Object Action Control Point Error Handling

Verify compliant operation when the Lower Tester uses Object Action Control Point procedure and an error results.

OTS/SR/OAE/BI-01-C [OACP - Op Code Not Supported]

- Test Purpose

  Verify that the IUT responds appropriately when a Client writes an unsupported Op Code to the Object Action Control Point.

- Reference

  [3] 3.3.2.8

- Initial Condition

  Perform the preamble described in Section 4.2.2.

- Test Procedure

  1. The Lower Tester writes an Op Code Value of 0x00 to the Object Action Control Point without a Parameter Value.
  2. The IUT, after sending a Write Response to acknowledge the write to the Object Action Control Point, sends an indication of the Object Action Control Point characteristic.
  3. The Lower Tester receives an ATT_Handle_Value_Indication from the IUT containing the Object Action Control Point characteristic handle and value.
  4. The Lower Tester sends an ATT_Handle_Value_Confirmation to the IUT.
  5. Verify that the IUT response meets the requirements of the service.
  6. The Lower Tester writes an Op Code value from the Reserved for Future Use (RFU) range other than 0x00 to the Object Action Control Point without a Parameter Value.
  7. The IUT, after sending a Write Response to acknowledge the write to the Object Action Control Point, sends an indication of the Object Action Control Point characteristic.
  8. The Lower Tester receives an ATT_Handle_Value_Indication from the IUT containing the Object Action Control Point characteristic handle and value.
  9. The Lower Tester sends an ATT_Handle_Value_Confirmation to the IUT.
  10. Verify that the IUT response meets the requirements of the service.

- Expected Outcome

  Pass verdict

  In step 2, the IUT sends an indication of the Object Action Control Point characteristic with the Response Code Op Code (0x60), the Request Op Code (0x00), followed by the Response Value for ‘Op Code not supported’ (0x02) and without a Response Parameter.

  In step 7, the IUT sends an indication of the Object Action Control Point characteristic with the Response Code Op Code (0x60), the Request Op Code (i.e., the RFU value written in step 6), followed by the Response Value for ‘Op Code not supported’ (0x02) and without a Response Parameter.

  The IUT returns to a stable state.

OTS/SR/OAE/BI-02-C [OACP - Invalid Object – Excluded by Filter Conditions]

- Test Purpose

  Verify that the IUT responds appropriately when the Client attempts to perform an OACP operation on an Invalid Object.

- Reference

  [3] 3.3.2.8, 1.9

- Initial Condition

  - At least one object is stored on the IUT.
  - Perform the preamble described in Section 4.2.2.
  - The object that is to be used in this test procedure is identified in the IXIT [7] by its Object Name. Run the preamble described in Section 4.2.4 to ensure that the object to be tested is selected as the Current Object.
  - The object properties permit the OACP procedure that is used in this test case.

  Note: It is not necessary in this test case that an Object Transfer Channel is available because the “Invalid Object” error response has a higher priority than “Channel Unavailable” and is required to take precedence.

- Test Procedure

  1. The Lower Tester reads all of the supported Object Metadata characteristic values.
  2. The Lower Tester writes a value to one of the Object List Filter characteristics chosen such that the object does not conform to the filter conditions specified. This causes the object to be excluded by the filter conditions and hence the Current Object becomes an Invalid Object.
  3. The Lower Tester writes one of the following Op Codes to the Object Action Control Point, choosing one that is supported by the IUT: Delete, Calculate Checksum, Execute, Read, or Write. If the Op Code requires a parameter, a valid parameter value is included.
  4. The IUT, after sending a Write Response to acknowledge the write to the Object Action Control Point, sends an indication of the Object Action Control Point characteristic.
  5. The Lower Tester receives an ATT_Handle_Value_Indication from the IUT containing the Object Action Control Point characteristic handle and value.
  6. The Lower Tester sends an ATT_Handle_Value_Confirmation to the IUT.

- Expected Outcome

  Pass verdict

  In step 4, the IUT sends an indication of the Object Action Control Point characteristic containing the Response Code Op Code (0x60) and the Request Op Code followed by the Result Code for ‘Invalid Object’ (0x05) with no Response Parameter.

  The IUT returns to a stable state.

OTS/SR/OAE/BI-03-C [OACP - Channel Unavailable - Read]

- Test Purpose

  Verify that the IUT responds appropriately when the Client attempts to read object data from the IUT without an Object Transfer Channel being available for object transfer.

- Reference

  [3] 3.3.2.8

- Initial Condition

  - At least one object is stored on the IUT.
  - The object properties permit reading.
  - Perform the preamble described in Section 4.2.2.
  - The object that is to be used in this test procedure is identified in the IXIT [7] by its Object Name. If the Object List Control Point (OLCP) is supported by the IUT (i.e., the IUT supports more than one object), run the preamble described in Section 4.2.4 to ensure that the object to be tested is selected as the Current Object.
  - No Object Transfer Channel is open; any Object Transfer Channel that was open has been closed.

- Test Procedure

  1. The Lower Tester reads the value of the Object Size characteristic.
  2. The Lower Tester writes the Read Op Code (0x05) to the Object Action Control Point with a Length parameter value equal to the value of the Current Size field of the Object Size characteristic read in step 1. The Offset parameter is set to 0.
  3. The IUT, after sending a Write Response to acknowledge the write to the Object Action Control Point, sends an indication of the Object Action Control Point characteristic.
  4. The Lower Tester receives an ATT_Handle_Value_Indication from the IUT containing the Object Action Control Point characteristic handle and value.
  5. The Lower Tester sends an ATT_Handle_Value_Confirmation to the IUT.

- Expected Outcome

  Pass verdict

  The IUT sends an indication of the Object Action Control Point characteristic containing the Response Code Op Code (0x60), the Request Op Code (0x05) followed by the Result Code for ‘Channel Unavailable’ (0x06) with no Response Parameter.

  The IUT does not attempt to send any object data to the Lower Tester.

  The IUT returns to a stable state.

OTS/SR/OAE/BI-04-C [OACP - Channel Unavailable - Write]

- Test Purpose

  Verify that the IUT responds appropriately when the Client attempts to write object data to the IUT without an Object Transfer Channel being available for object transfer.

- Reference

  [3] 3.3.2.8

- Initial Condition

  - At least one object is stored on the IUT.
  - Perform the preamble described in Section 4.2.2.
  - The object that is to be used in this test procedure is identified in the IXIT [7] by its Object Name. If the Object List Control Point (OLCP) is supported by the IUT (i.e., the IUT supports more than one object), run the preamble described in Section 4.2.4 to ensure that the object to be tested is selected as the Current Object.
  - The properties of the Current Object permit writing.
  - No Object Transfer Channel is open; any Object Transfer Channel that was open has been closed.

- Test Procedure

  1. The Lower Tester reads the value of the Object Size characteristic.
  2. The Lower Tester writes the Write Op Code (0x06) to the Object Action Control Point with a Length parameter value equal to the Allocated Size field of the Object Size characteristic value read in step 1. The Offset parameter is set to 0. The Mode parameter is set to 0x00.
  3. The IUT, after sending a Write Response to acknowledge the write to the Object Action Control Point, sends an indication of the Object Action Control Point characteristic.
  4. The Lower Tester receives an ATT_Handle_Value_Indication from the IUT containing the Object Action Control Point characteristic handle and value.
  5. The Lower Tester sends an ATT_Handle_Value_Confirmation to the IUT.

- Expected Outcome

  Pass verdict

  The IUT sends an indication of the Object Action Control Point characteristic containing the Response Code Op Code (0x60), the Request Op Code (0x06) followed by the Result Code for ‘Channel Unavailable’ (0x06) with no Response Parameter.

  The IUT returns to a stable state.

OTS/SR/OAE/BI-05-C [OACP - Unsupported Type]

- Test Purpose

  Verify that the IUT responds appropriately when a Client attempts to create a new object with an object type that is not permitted by the IUT.

- Reference

  [3] 3.3.2.8

- Initial Condition

  - Perform the preamble described in Section 4.2.2.
  - The object types and object sizes supported by the IUT are known from the IXIT [7].

- Test Procedure

  1. The Lower Tester writes the Create Op Code (0x01) to the Object Action Control Point with a Parameter Value. The Type field of the Parameter contains the UUID of an object type that is not supported by the IUT; e.g., the UUID for «Object Transfer Service» may be used here for test purposes since this UUID has already been assigned to identify a service and therefore will never be used to identify a valid object type. The Size field contains any acceptable value (e.g., 0x00000200, representing an allocated object size of 512 octets).
  2. The IUT, after sending a Write Response to acknowledge the write to the Object Action Control Point, sends an indication of the Object Action Control Point characteristic.
  3. The Lower Tester receives an ATT_Handle_Value_Indication from the IUT containing the Object Action Control Point characteristic handle and value.
  4. The Lower Tester sends an ATT_Handle_Value_Confirmation to the IUT.
  5. The Lower Tester reads the value of the Object Type characteristic, using the procedures described in Section 4.5. The value reported is noted.

- Expected Outcome

  Pass verdict

  The IUT sends an indication of the Object Action Control Point characteristic containing the Response Code Op Code (0x60), the Request Op Code (0x01) followed by the Result Code for ‘Unsupported Type’ (0x07) with no Response Parameter.

  The value reported in the Object Type characteristic is not equal to the value of the Type parameter included with the Create Op Code.

  The IUT returns to a stable state.

4.12.1 OACP - Procedure Not Permitted

- Test Purpose

  Verify that the IUT responds appropriately when the Client attempts to use a supported OACP procedure when the object properties do not permit that operation to be performed on the object. The verification is done one value at a time, as enumerated in the test cases in Table 4.8, using this generic test procedure.

- Reference

  [3] 3.3.2.8

- Initial Condition

  - At least one object is stored on the IUT.
  - If the Object Properties characteristic is not writable, the object provided for the purposes of the test case has the object property as specified in Table 4.8 already set to ‘False’, indicating that the relevant procedure is not permitted for that object, except where the value ‘False’ for the specific object property is not allowed by the implementation.

- The handle of each characteristic value referenced in the test cases below has been previously discovered by the Lower Tester during the test procedure in Section 4.3 or Section 4.4 or is known to the Lower Tester by other means.
- Perform the preamble described in Section 4.2.2.
- The object that is to be used in this test procedure is identified in the IXIT [7] by its Object Name. If the Object List Control Point (OLCP) is supported by the IUT (i.e., the IUT supports more than one object), run the preamble described in Section 4.2.4 to ensure that the object to be tested is selected as the Current Object.

Note: It does not matter in this test case whether an Object Transfer Channel is available because the “Procedure Not Permitted” error response has a higher priority than “Channel Unavailable” and is required to take precedence.

- Test Case Configuration

Test Case	OACP Op Code	Property	Object Properties characteristic
OTS/SR/OAE/BI-06-C [OACP - Delete Not Permitted]	0x02	Delete	Bit 0
OTS/SR/OAE/BI-07-C [OACP - Execute Not Permitted]	0x04	Execute	Bit 1
OTS/SR/OAE/BI-08-C [OACP - Read Not Permitted]	0x05	Read	Bit 2
OTS/SR/OAE/BI-09-C [OACP - Write Not Permitted]	0x06	Write	Bit 3

Table 4.8: OACP Procedure Not Permitted test cases

- Test Procedure

1. The Lower Tester reads the value of the Object Properties characteristic.
2. If the property bit that is specified in Table 4.8 is set to 1 (True) and the Object Properties characteristic is writable, the Lower Tester writes a value to this characteristic to attempt to change the value of this individual bit to zero (False).
3. The IUT either accepts the write request by sending an ATT_Write_Response or rejects it by sending an Error Response with an Attribute Protocol Application Error Code set to “Write Request Rejected” (0x80).
4. The Lower Tester receives the ATT_Write_Response or ATT_Error_Response from the IUT.
5. The Lower Tester reads the value of the Object Properties characteristic again to confirm the value of the relevant property bit shown in Table 4.8.
6. The Lower Tester writes the OACP Op Code specified in the test case to the Object Action Control Point. If the Op Code requires a parameter, a valid parameter value is included. The Offset parameter, where required, is set to 0.
7. The IUT, after sending a Write Response to acknowledge the write to the Object Action Control Point, sends an indication of the Object Action Control Point characteristic.
8. The Lower Tester receives an ATT_Handle_Value_Indication from the IUT containing the Object Action Control Point characteristic handle and value.
9. The Lower Tester sends an ATT_Handle_Value_Confirmation to the IUT.

- Expected Outcome

Pass verdict

The IUT sends an indication of the Object Action Control Point characteristic containing the Response Code Op Code (0x60) and the Request Op Code (as specified in the test case) followed by the Result Code with no Response Parameter.

If the relevant bit shown in Table 4.8 of the Object Properties characteristic when read in step 5 had the value 'False', the Result Code is 'Procedure Not Permitted' (0x08).

The IUT returns to a stable state.

Alternate outcome if the object property bit is 'True':

If the relevant bit shown in Table 4.8 of the Object Properties characteristic when read in step 5 had the value 'True', the Result Code is any code that is not equal to 'Procedure Not Permitted' (0x08).

The IUT returns to a stable state.

OTS/SR/OAE/BI-10-C [OACP – Object Locked]

- Test Purpose

  Verify that the IUT responds appropriately when the Client attempts to write object data to the IUT when the Current Object has been locked.

- Reference

  [3] 3.3.2.6, 3.3.2.8

- Initial Condition

  - At least one object is stored on the IUT.
  - Perform the preamble described in Section 4.2.2.
  - The object that is to be used in this test procedure is identified in the IXIT [7] by its Object Name. If the Object List Control Point (OLCP) is supported by the IUT (i.e., the IUT supports more than one object), run the preamble described in Section 4.2.4 to ensure that the object to be tested is selected as the Current Object.
  - The object properties of the Current Object permit writing.

- Test Procedure

  1. The Lower Tester reads the Object Size characteristic.
  2. The Lower Tester opens an Object Transfer Channel.
  3. Induce the IUT to lock the object (e.g., by starting an OACP Write operation to the same object using an additional, concurrently connected, Lower Tester).
  4. The Lower Tester writes the Write Op Code (0x06) to the Object Action Control Point with a Length parameter value equal to the Allocated Size field of the Object Size characteristic value read in step 1. The Offset parameter is set to 0. The Mode parameter is set to 0x00.
  5. The IUT, after sending a Write Response to acknowledge the write to the Object Action Control Point, sends an indication of the Object Action Control Point characteristic.
  6. The Lower Tester receives an ATT_Handle_Value_Indication from the IUT containing the Object Action Control Point characteristic handle and value.
  7. The Lower Tester sends an ATT_Handle_Value_Confirmation to the IUT.

- Expected Outcome

  Pass verdict

  The IUT sends an indication of the Object Action Control Point characteristic containing the Response Code Op Code (0x60), the Request Op Code (0x06) followed by the Result Code for 'Object Locked' (0x09) with no Response Parameter.

  The IUT returns to a stable state.

OTS/SR/OAE/BI-11-C [OACP - Operation Failed]

- Test Purpose

  Verify that the IUT responds appropriately when the Client attempts to use the OACP Abort procedure when there is no object transfer in progress.

- Reference

  [3] 3.3.2.7, 3.3.2.8

- Initial Condition

  - At least one object is stored on the IUT.
  - Perform the preamble described in Section 4.2.2.
  - There is no object transfer in progress.

- Test Procedure

  1. The Lower Tester writes the Abort Op Code (0x07) to the Object Action Control Point with no parameter.
  2. The IUT, after sending a Write Response to acknowledge the write to the Object Action Control Point, sends an indication of the Object Action Control Point characteristic.
  3. The Lower Tester receives an ATT_Handle_Value_Indication from the IUT containing the Object Action Control Point characteristic handle and value.
  4. The Lower Tester sends an ATT_Handle_Value_Confirmation to the IUT.

- Expected Outcome

  Pass verdict

  The IUT sends an indication of the Object Action Control Point characteristic containing the Response Code Op Code (0x60), the Request Op Code (0x09) followed by the Result Code for ‘Operation Failed’ (0x0A) with no Response Parameter.

  The IUT returns to a stable state.

OTS/SR/OAE/BI-12-C [OACP - Client Characteristic Configuration Descriptor Improperly Configured]

- Test Purpose

  Verify that the IUT responds appropriately when a Client attempts to perform an Object Action Control Point procedure with a Client Characteristic Configuration descriptor that is improperly configured.

- Reference

  [3] 3.3.2.8

- Initial Condition

  - At least one object is stored on the IUT.
  - Perform the preamble described in Section 4.2.2.

- Test Procedure

  1. The Lower Tester resets the value of the Client Characteristic Configuration descriptor of the Object Action Control Point characteristic to 0x0000.

2. The Lower Tester writes a valid Op Code to the Object Action Control Point that is supported by the IUT.
3. Verify that the IUT response meets the requirements of the service.

- Expected Outcome

  Pass verdict

  The IUT rejects the Write Request by sending an Error Response with an Attribute Protocol Error Code set to "Client Characteristic Configuration Descriptor Improperly Configured" (0xFD).

  The IUT returns to a stable state.

OTS/SR/OAE/BI-13-C [OACP – Invalid Parameter]

- Test Purpose

  Verify that the IUT responds appropriately when the parameter value is invalid when a Client requests an OACP procedure with an Op Code that requires a parameter.

- Reference

  [3] 3.3.2.8

- Initial Condition

  - At least one object is stored on the IUT.
  - Perform the preamble described in Section 4.2.2.
  - The object that is to be used in this test procedure is identified in the IXIT [7] by its Object Name. If the Object List Control Point (OLCP) is supported by the IUT (i.e., the IUT supports more than one object), run the preamble described in Section 4.2.4 to ensure that the object to be tested is selected as the Current Object.
  - The object's properties permit the OACP operation used in the test procedure.
  - If the OACP procedure to be used in the test procedure requires an Object Transfer Channel to be available, the Lower Tester opens the required channel before running the test procedure. This is done to ensure that the "Channel Unavailable" error response is not triggered during the test.

- Test Procedure

  1. The Lower Tester reads the value of the Object Size characteristic.
  2. The Lower Tester writes one of the following Op Codes to the Object Action Control Point with the Offset parameter set to a value that exceeds the value of the Current Size field of the Object Size characteristic read in step 1: Calculate Checksum (0x03), Read (0x05) or Write (0x06), choosing one Op Code from these that is supported by the IUT. The Length parameter is set to a non-zero value such that the sum of the Offset and Length parameter values does not exceed the value of the Allocated Size field of the Object Size characteristic. The Mode parameter, if required, is set to 0x00.
  3. The IUT, after sending a Write Response to acknowledge the write to the Object Action Control Point, sends an indication of the Object Action Control Point characteristic.
  4. The Lower Tester receives an ATT_Handle_Value_Indication from the IUT containing the Object Action Control Point characteristic handle and value.
  5. The Lower Tester sends an ATT_Handle_Value_Confirmation to the IUT.

- Expected Outcome

  <u>Pass verdict</u>

  The IUT sends an indication of the Object Action Control Point characteristic containing the Response Code Op Code (0x60), the Request Op Code (as selected in step 2) followed by the Result Code for 'Invalid Parameter' (0x03) with no Response Parameter.

4.13 Service Procedures – Object List Control Point

Verify compliant operation when the Lower Tester uses Object List Control Point procedures.

OTS/SR/OLSP/BV-01-C [OLCP - First, Last, Previous and Next]

- Test Purpose

  Verify that the IUT can perform the OLCP First, Last, Previous and Next procedures.

- Reference

  [3] 1.8, 3.4.2.1, 3.4.2.2, 3.4.2.3, 3.4.2.4, 3.4.2.9

- Initial Condition

  - At least two objects are stored on the IUT.
  - The handle of each characteristic and characteristic descriptor referenced in the test procedure below has been previously discovered by the Lower Tester during the test procedure in Section 4.3 or Section 4.4 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 IUT permissions for the Object List Control Point characteristic require a specific security mode or security level, establish a connection meeting those requirements. If the IUT requires bonding, then the Lower Tester performs a bonding procedure.
  - The object list sort order being used by the IUT may be any order (such as ordered in ascending order by object name) provided that the order remains constant during the execution of this test case.

- Test Procedure

  1. The Lower Tester configures the OLCP characteristic for indications as described in the test procedure in Section 4.3.
  2. If the IUT supports the Object List Filter characteristics, the Lower Tester writes the value 'No Filter' (0x00) with no parameter to all three instances of the Object List Filter characteristic.
  3. The Lower Tester writes the First Op Code (0x01) to the Object List Control Point with no parameter.
  4. The IUT, after sending a Write Response to acknowledge the write to the Object List Control Point, sends an indication of the Object List Control Point characteristic.
  5. The Lower Tester receives an ATT_Handle_Value_Indication from the IUT containing the Object List Control Point characteristic handle and value.
  6. The Lower Tester sends an ATT_Handle_Value_Confirmation to the IUT.
  7. The Lower Tester reads the value of the Object Name. If the value of the Object Name is long enough to require the GATT Read Long sub-procedure to be used, it reads the name in its entirety using the GATT Read Long Characteristic Values sub-procedure.
  8. The name read in step 7 is recorded.
  9. The Lower Tester writes the Next Op Code (0x04) to the Object List Control Point with no parameter.

10. The IUT, after sending a Write Response to acknowledge the write to the Object List Control Point, sends an indication of the Object List Control Point characteristic.
11. The Lower Tester receives an ATT_Handle_Value_Indication from the IUT containing the Object List Control Point characteristic handle and value.
12. The Lower Tester sends an ATT_Handle_Value_Confirmation to the IUT.
13. If the Result Code received in step 11 was ‘Success’, repeat steps 9–12 until the Result Code is ‘Out Of Bounds’.
14. The Lower Tester reads the value of the Object Name. If the value of the Object Name is long enough to require the GATT Read Long sub-procedure to be used, it reads the name in its entirety using the GATT Read Long Characteristic Values sub-procedure.
15. The name read in step 14 is recorded.
16. The Lower Tester writes the Previous Op Code (0x03) to the Object List Control Point with no parameter.
17. The IUT, after sending a Write Response to acknowledge the write to the Object List Control Point, sends an indication of the Object List Control Point characteristic.
18. The Lower Tester receives an ATT_Handle_Value_Indication from the IUT containing the Object List Control Point characteristic handle and value.
19. The Lower Tester sends an ATT_Handle_Value_Confirmation to the IUT.
20. If the Result Code received in step 11 was ‘Success’, repeat steps 16–19 until the Result Code is ‘Out Of Bounds’.
21. The Lower Tester reads the value of the Object Name. If the value of the Object Name is long enough to require the GATT Read Long sub-procedure to be used, it reads the name in its entirety using the GATT Read Long Characteristic Values sub-procedure.
22. The name read in step 21 is recorded.
23. The Lower Tester writes the Last Op Code (0x02) to the Object List Control Point with no parameter.
24. The IUT, after sending a Write Response to acknowledge the write to the Object List Control Point, sends an indication of the Object List Control Point characteristic.
25. The Lower Tester receives an ATT_Handle_Value_Indication from the IUT containing the Object List Control Point characteristic handle and value.
26. The Lower Tester sends an ATT_Handle_Value_Confirmation to the IUT.
27. The Lower Tester reads the value of the Object Name. If the value of the Object Name is long enough to require the GATT Read Long sub-procedure to be used, it reads the name in its entirety using the GATT Read Long Characteristic Values sub-procedure.
28. The name read in step 27 is recorded.

- Expected Outcome

  Pass verdict

  In steps 4, 10, 17, and 24, the IUT sends an indication of the Object List Control Point characteristic containing the Response Code Op Code (0x70) followed by the Request Op Code (0x01, 0x02, 0x03 or 0x04 as specified in the test procedure) and the Result Code for ‘Success’ (0x01) with no Response Parameter.

  The value of the Object Name characteristic read in step 21 exactly matches the value of the Object Name characteristic read in step 7. This corresponds to the name of the first object in the list with the present list sort order.

  The value of the Object Name characteristic read in step 14 exactly matches the value of the Object Name characteristic read in step 27. This corresponds to the name of the last object in the list with the present list sort order.

OTS/SR/OLSP/BV-02-C [OLCP - Go To]

- Test Purpose

  Verify that the IUT can perform the Go To procedure.

- Reference

  [3] 1.8, 3.4.2.5

- Initial Condition

  - At least two objects are stored on the IUT.
  - The handle of each characteristic and characteristic descriptor referenced in the test procedure below has been previously discovered by the Lower Tester during the test procedure in Section 4.3 or Section 4.4 or is known to the Lower Tester by other means.
  - The Lower Tester discovers the object name and object ID of at least two objects presently available on the IUT by running the preamble described in Section 4.2.3 or by other means. The Lower Tester selects two of the discovered objects for use in the following test procedure. These two objects will be referred to in the test procedure as “Object A” and “Object B”.

- Test Procedure

  1. If the IUT supports the Object List Filter characteristics, the Lower Tester writes the value ‘No Filter’ (0x00) with no parameter to all three instances of the Object List Filter characteristic.
  2. The Lower Tester writes the Go To Op Code (0x05) to the Object List Control Point with the parameter value set to the Object ID value of Object A.
  3. The IUT, after sending a Write Response to acknowledge the write to the Object List Control Point, sends an indication of the Object List Control Point characteristic.
  4. The Lower Tester receives an ATT_Handle_Value_Indication from the IUT containing the Object List Control Point characteristic handle and value.
  5. The Lower Tester sends an ATT_Handle_Value_Confirmation to the IUT.
  6. The Lower Tester reads the value of the Object Name characteristic. If the value of the Object Name is long enough to require the GATT Read Long sub-procedure to be used, it reads the name in its entirety using the GATT Read Long Characteristic Values sub-procedure.
  7. Verify that the value of the Object Name characteristic matches the name of Object A.
  8. The Lower Tester writes the Go To Op Code (0x06) to the Object List Control Point with the parameter value set to the Object ID value of Object B.
  9. The IUT, after sending a Write Response to acknowledge the write to the Object List Control Point, sends an indication of the Object List Control Point characteristic.
  10. The Lower Tester receives an ATT_Handle_Value_Indication from the IUT containing the Object List Control Point characteristic handle and value.
  11. The Lower Tester sends an ATT_Handle_Value_Confirmation to the IUT.
  12. The Lower Tester reads the value of the Object Name characteristic.
  13. Verify that the value of the Object Name characteristic matches the name of Object B.
  14. Repeat steps 2–6.
  15. Verify that the value of the Object Name characteristic matches the name of Object A.

- Expected Outcome

  Pass verdict

  In steps 3 and 9, the IUT sends an indication of the Object List Control Point characteristic containing the Response Code Op Code (0x70) followed by the Request Op Code(0x05) and the Result Code 'Success' (0x01) with no Response Parameter.

  In step 7, the value of the Object Name characteristic matches the name of Object A.

  In step 13, the value of the Object Name characteristic matches the name of Object B.

  In step 15, the value of the Object Name characteristic matches the name of Object A.

OTS/SR/OLSP/BV-03-C [OLCP - Order]

- Test Purpose

  Verify that the IUT can perform the Order procedure.

- Reference

  [3] 3.4.2.6

- Initial Condition
  - At least two objects are stored on the IUT.
  - The handle of each characteristic and characteristic descriptor referenced in the test procedure below has been previously discovered by the Lower Tester during the test procedure in Section 4.3 or Section 4.4 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 IUT permissions for the Object List Control Point characteristic require a specific security mode or security level, establish a connection meeting those requirements. If the IUT requires bonding, then the Lower Tester performs a bonding procedure.
  - The Lower Tester discovers the object name and object ID of all objects presently available on the IUT by running the preamble described in Section 4.2.3.

- Test Procedure
  1. If the IUT supports the Object List Filter characteristics, the Lower Tester writes the value 'No Filter' (0x00) with no parameter to all three instances of the Object List Filter characteristic.
  2. The Lower Tester writes the Order Op Code (0x06) to the Object List Control Point with the List Sort Order parameter set to 'Order the list by Object Name, ascending' (0x01).
  3. The IUT, after sending a Write Response to acknowledge the write to the Object List Control Point, sends an indication of the Object List Control Point characteristic.
  4. The Lower Tester receives an ATT_Handle_Value_Indication from the IUT containing the Object List Control Point characteristic handle and value.
  5. The Lower Tester sends an ATT_Handle_Value_Confirmation to the IUT.
  6. The Lower Tester writes the First Op Code (0x01) to the Object List Control Point with no parameter.
  7. The IUT, after sending a Write Response to acknowledge the write to the Object List Control Point, sends an indication of the Object List Control Point characteristic.
  8. The Lower Tester receives an ATT_Handle_Value_Indication from the IUT containing the Object List Control Point characteristic handle and value.
  9. The Lower Tester sends an ATT_Handle_Value_Confirmation to the IUT.

10. The Lower Tester reads the value of the Object Name characteristic. If the value of the Object Name is long enough to require the GATT Read Long sub-procedure to be used, it reads the name in its entirety using the GATT Read Long Characteristic Values sub-procedure.
11. Verify that the value of the Object Name characteristic matches the name of the object which comes first in the list when the names are listed in ascending alphabetical order.
12. The Lower Tester writes the Last Op Code (0x02) to the Object List Control Point with no parameter.
13. The IUT, after sending a Write Response to acknowledge the write to the Object List Control Point, sends an indication of the Object List Control Point characteristic.
14. The Lower Tester receives an ATT_Handle_Value_Indication from the IUT containing the Object List Control Point characteristic handle and value.
15. The Lower Tester sends an ATT_Handle_Value_Confirmation to the IUT.
16. The Lower Tester reads the value of the Object Name characteristic.
17. Verify that the value of the Object Name characteristic matches the name of the object which comes last in the list when the names are listed in ascending alphabetical order.
18. The Lower Tester writes the Order Op Code (0x07) to the Object List Control Point with the List Sort Order parameter set to ‘Order the list by Object Name, descending’ (0x11).
19. Repeat steps 3–10.
20. Verify that the value of the Object Name characteristic matches the name of the object which comes first in the list when the names are listed in descending alphabetical order.
21. Repeat steps 12–16.
22. Verify that the value of the Object Name characteristic matches the name of the object which comes last in the list when the names are listed in descending alphabetical order.

- Expected Outcome

  Pass verdict

  In steps 3, 7, and 13, the IUT sends an indication of the Object List Control Point characteristic containing the Response Code Op Code (0x70) followed by the Request Op Code (0x06, 0x01 or 0x02) and the Result Code for ‘Success’ (0x01) with no Response Parameter.

  In step 11, the value of the Object Name characteristic matches the name of the object which comes first in the list when the names are listed in ascending alphabetical order as defined in the service.

  In step 17, the value of the Object Name characteristic matches the name of the object which comes last in the list when the names are listed in ascending alphabetical order as defined in the service.

  In step 20, the value of the Object Name characteristic is the same as in step 17 (i.e., the previously last object has become the first object due to reversing the list sort order in step 18).

  In step 22, the value of the Object Name characteristic is the same as in step 11 (i.e., the previously first object has become the last object due to reversing the list sort order in step 18).

OTS/SR/OLSP/BV-04-C [OLCP - Request Number of Objects]

- Test Purpose

  Verify that the IUT can perform the Request Number of Objects procedure.

- Reference

  [3] 3.4.2.7

- Initial Condition
  - At least two objects are stored on the IUT. The Lower Tester has discovered how many objects are stored on the IUT indirectly during the procedure described in OTS/SR/OLSP/BV-01-C [OLCP - First, Last, Previous and Next] or Section 4.2.3 or by other means.
  - The handle of each characteristic and characteristic descriptor referenced in the test procedure below has been previously discovered by the Lower Tester during the test procedure in Section 4.3 or Section 4.4 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 IUT permissions for the Object List Control Point characteristic require a specific security mode or security level, establish a connection meeting those requirements. If the IUT requires bonding, then the Lower Tester performs a bonding procedure.
  - The Lower Tester has configured the OLCP characteristic for indications as described in the test procedure in Section 4.3.
- Test Procedure
  1. If the IUT supports the Object List Filter characteristics, the Lower Tester writes the value 'No Filter' (0x00) with no parameter to all three instances of the Object List Filter characteristic.
  2. The Lower Tester writes the Request Number of Objects Op Code (0x07) to the Object List Control Point with no parameter.
  3. The IUT, after sending a Write Response to acknowledge the write to the Object List Control Point, sends an indication of the Object List Control Point characteristic.
  4. The Lower Tester receives an ATT_Handle_Value_Indication from the IUT containing the Object List Control Point characteristic handle and value.
  5. The Lower Tester sends an ATT_Handle_Value_Confirmation to the IUT.
  6. Verify that the number of objects reported in the value of the Response Parameter matches the number of objects stored on the IUT.
- Expected Outcome

  Pass verdict

  In step 3, the IUT sends an indication of the Object List Control Point characteristic containing the Response Code Op Code (0x70) followed by the Request Op Code (0x07) and the Result Code for 'Success' (0x01) with a Response Parameter in UINT32 format.

  The number of objects reported in the value of the Response Parameter matches the number of objects stored on the IUT.

OTS/SR/OLSP/BV-05-C [OLCP - Clear Marking]

- Test Purpose

  Verify that the IUT can perform the Clear Marking procedure.
- Reference

  [3] 3.4.2.8
- Initial Condition
  - At least two objects are stored on the IUT. At least one object is a marked object; i.e., the Mark property bit for at least one object is set to 1 (True).

- The handle of each characteristic and characteristic descriptor referenced in the test procedure below has been previously discovered by the Lower Tester during the test procedure in Section 4.3 or Section 4.4 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 IUT permissions for the Object List Control Point characteristic require a specific security mode or security level, establish a connection meeting those requirements. If the IUT requires bonding, then the Lower Tester performs a bonding procedure.
- The Lower Tester has configured the OLCP characteristic for indications as described in the test procedure in Section 4.3.

- Test Procedure

1. If the IUT supports the Object List Filter characteristics, the Lower Tester writes the value ‘No Filter’ (0x00) with no parameter to all three instances of the Object List Filter characteristic.
2. The Lower Tester writes the First Op Code (0x01) to the Object List Control Point with no parameter.
3. The IUT, after sending a Write Response to acknowledge the write to the Object List Control Point, sends an indication of the Object List Control Point characteristic.
4. The Lower Tester receives an ATT_Handle_Value_Indication from the IUT containing the Object List Control Point characteristic handle and value.
5. The Lower Tester sends an ATT_Handle_Value_Confirmation to the IUT.
6. The Lower Tester writes the Clear Marking Op Code (0x08) to the Object List Control Point with no parameter.
7. The IUT, after sending a Write Response to acknowledge the write to the Object List Control Point, sends an indication of the Object List Control Point characteristic.
8. The Lower Tester receives an ATT_Handle_Value_Indication from the IUT containing the Object List Control Point characteristic handle and value.
9. The Lower Tester sends an ATT_Handle_Value_Confirmation to the IUT.
10. The Lower Tester reads the value of the Object ID and Object Properties characteristics.
11. The values read in step 10 are recorded.
12. The Lower Tester writes the Next Op Code (0x04) to the Object List Control Point with no parameter.
13. The IUT, after sending a Write Response to acknowledge the write to the Object List Control Point, sends an indication of the Object List Control Point characteristic.
14. The Lower Tester receives an ATT_Handle_Value_Indication from the IUT containing the Object List Control Point characteristic handle and value.
15. The Lower Tester sends an ATT_Handle_Value_Confirmation to the IUT.
16. If the Result Code received in step 14 was ‘Success’, the Lower Tester reads and records the values of the Object ID and Object Properties characteristics.
17. If the Result Code received in step 14 was ‘Success’, repeat steps 12–16 until the Result Code is ‘Out of Bounds’.
18. Inspect the properties of all the objects that were recorded.

- Expected Outcome

Pass verdict

In step 3, the IUT sends an indication of the Object List Control Point characteristic containing the Response Code Op Code (0x70) followed by the Request Op Code (0x01) and the Result Code for ‘Success’ (0x01) with no Response Parameter.

In step 7, the IUT sends an indication of the Object List Control Point characteristic containing the Response Code Op Code (0x70) followed by the Request Op Code (0x08) and the Result Code for ‘Success’ (0x01) with no Response Parameter.

In step 18, the object properties of all the objects available on the IUT are confirmed to have the Mark bit (bit 7) set to 0 (False).

4.14 Service Procedures – Object List Control Point Error Handling

Verify compliant operation when the Lower Tester uses Object List Control Point procedure and an error results.

OTS/SR/OLE/BI-01-C [OLCP - Op Code Not Supported]

- Test Purpose

  Verify that the IUT responds appropriately when a Client writes an unsupported Op Code to the Object List Control Point.

- Reference

  [3] 3.4.2.9

- Initial Condition

  - Establish an ATT Bearer connection between the Lower Tester and IUT as described in Section 4.2.1. If IUT permissions for the characteristic require a specific security mode or security level, establish a connection meeting those requirements. If the IUT requires bonding, then the Lower Tester performs a bonding procedure.

  - The Lower Tester has configured the OLCP characteristic for indications as described in the test procedure in Section 4.3.

- Test Procedure

  1. The Lower Tester writes an Op Code Value of 0x00 to the Object List Control Point without a Parameter Value.
  2. The IUT, after sending a Write Response to acknowledge the write to the Object List Control Point, sends an indication of the Object List Control Point characteristic.
  3. The Lower Tester receives an ATT_Handle_Value_Indication from the IUT containing the Object Action List Point characteristic handle and value.
  4. The Lower Tester sends an ATT_Handle_Value_Confirmation to the IUT.
  5. Verify that the IUT response meets the requirements of the service.
  6. The Lower Tester writes an Op Code value from the Reserved for Future Use (RFU) range other than 0x00 to the Object List Control Point without a Parameter Value.
  7. The IUT, after sending a Write Response to acknowledge the write to the Object List Control Point, sends an indication of the Object List Control Point characteristic.
  8. The Lower Tester receives an ATT_Handle_Value_Indication from the IUT containing the Object List Control Point characteristic handle and value.
  9. The Lower Tester sends an ATT_Handle_Value_Confirmation to the IUT.
  10. Verify that the IUT response meets the requirements of the service.

- Expected Outcome

  Pass verdict

  In step 2, the IUT sends an indication of the Object List Control Point characteristic with the Response Code Op Code (0x70), the Request Op Code (0x00), followed by the Response Value for ‘Op Code not supported’ (0x02) and without a Response Parameter.

  In step 7, the IUT sends an indication of the Object List Control Point characteristic with the Response Code Op Code (0x70), the Request Op Code (i.e., the RFU value written in step 6), followed by the Response Value for ‘Op Code not supported’ (0x02) and without a Response Parameter.

  The IUT returns to a stable state.

OTS/SR/OLE/BI-02-C [OLCP - Invalid Parameter]

- Test Purpose

  Verify that the IUT responds appropriately when a Client writes a supported Op Code followed by an invalid Parameter Value to the Object List Control Point.

- Reference

  [3] 3.4.1, 3.4.2.9

- Initial Condition

  - Two or more objects are stored on the IUT. This is done to ensure that the “No Object” error response is not triggered during the test procedure. Care should also be taken not to cause the “Too Many Objects” error response to be triggered.

  - Establish an ATT Bearer connection between the Lower Tester and IUT as described in Section 4.2.1. If IUT permissions for the characteristic require a specific security mode or security level, establish a connection meeting those requirements. If the IUT requires bonding, then the Lower Tester performs a bonding procedure.

  - The Lower Tester has configured the OLCP characteristic for indications as described in the procedure in Section 4.3.

- Test Procedure

  1. A connection is established between the Lower Tester and IUT.
  2. The Lower Tester writes the Order Op Code (0x06) to the Object List Control Point with an invalid Parameter Value (i.e., the List Sort Order parameter is set to a RFU value).
  3. Verify that the IUT response meets the requirements of the service.

- Expected Outcome

  Pass verdict

  The IUT sends a Write Response followed by an indication of the Object List Control Point characteristic with the Response Code Op Code (0x70), the Request Op Code (0x06) followed by the Response Value for ‘Invalid Parameter’ (0x03) and without a Response Parameter.

OTS/SR/OLE/BI-03-C [OLCP - No Object]

- Test Purpose

  Verify that the IUT responds appropriately when a Client requests an OLCP procedure that cannot be performed because there is no object available on the IUT to which the procedure can be applied.

- Reference

  [3] 3.4.2.1, 3.4.2.9

- Initial Condition

  - No object is available on the IUT. Note that it may require the IUT to be reset or other implementation-specific actions to be taken to achieve this state prior to running the test procedure.
  - The handle of each characteristic and characteristic descriptor referenced in the test procedure below has been previously discovered by the Lower Tester during the test procedure in Section 4.3 or Section 4.4 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 IUT permissions for the Object List Control Point characteristic require a specific security mode or security level, establish a connection meeting those requirements. If the IUT requires bonding, then the Lower Tester performs a bonding procedure.
  - The Lower Tester has configured the OLCP characteristic for indications as described in the test procedure in Section 4.3.

- Test Procedure

  1. The Lower Tester writes the First Op Code (0x01) to the Object List Control Point with no parameter.
  2. The IUT, after sending a Write Response to acknowledge the write to the Object List Control Point, sends an indication of the Object List Control Point characteristic.
  3. The Lower Tester receives an ATT_Handle_Value_Indication from the IUT containing the Object List Control Point characteristic handle and value.
  4. The Lower Tester sends an ATT_Handle_Value_Confirmation to the IUT.
  5. Verify that response meets the requirements of the service.

- Expected Outcome

  Pass verdict

  The IUT sends an indication of the Object List Control Point characteristic containing the Response Code Op Code (0x70) followed by the Request Op Code (0x01) and the Result Code for ‘No Object’ (0x07) with no Response Parameter.

OTS/SR/OLE/BI-04-C [OLCP - Object ID Not Found]

- Test Purpose

  Verify that the IUT responds appropriately when an Object ID cannot be found on the Server.

- Reference

  [3] 3.4.2.5, 3.4.2.9

- Initial Condition

  - At least one object is stored on the IUT.

- The handle of each characteristic and characteristic descriptor referenced in the test procedure below has been previously discovered by the Lower Tester during the test procedure in Section 4.3 or Section 4.4 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 IUT permissions for the Object List Control Point characteristic require a specific security mode or security level, establish a connection meeting those requirements. If the IUT requires bonding, then the Lower Tester performs a bonding procedure.
- The Lower Tester discovers the object name and object ID of every object presently available on the IUT by running the preamble described in Section 4.2.3. The Lower Tester generates a value for an object ID to be used in the following test procedure by ensuring that it does not match the value of the object ID of any object that is presently available on the IUT.

- Test Procedure

1. The Lower Tester writes the Go To Op Code (0x06) to the Object List Control Point with the parameter value set to an Object ID value that does not match the ID of any object present on the IUT.
2. The IUT, after sending a Write Response to acknowledge the write to the Object List Control Point, sends an indication of the Object List Control Point characteristic.
3. The Lower Tester receives an ATT_Handle_Value_Indication from the IUT containing the Object List Control Point characteristic handle and value.
4. The Lower Tester sends an ATT_Handle_Value_Confirmation to the IUT.

- Expected Outcome

Pass verdict

In step 2, the IUT sends an indication of the Object List Control Point characteristic containing the Response Code Op Code (0x70) followed by the Request Op Code(0x05) and the Result Code ‘Object ID Not Found’ (0x08) with no Response Parameter.

OTS/SR/OLE/BI-05-C [OLCP - Client Characteristic Configuration Descriptor Improperly Configured]

- Test Purpose

Verify that the IUT responds appropriately when a Client attempts to perform an Object List Control Point procedure with a Client Characteristic Configuration descriptor that is improperly configured.

- Reference

[3] 3.4.2.9.1

- Initial Condition

- At least one object is stored on the IUT.
- Establish an ATT Bearer connection between the Lower Tester and IUT as described in Section 4.2.1. If IUT permissions for the characteristic require a specific security mode or security level, establish a connection meeting those requirements. If the IUT requires bonding, then the Lower Tester performs a bonding procedure.
- The Lower Tester has configured the OLCP characteristic for indications as described in the test procedure in Section 4.3.

- Test Procedure

  1. The Lower Tester resets the value of the Client Characteristic Configuration descriptor of the Object List Control Point characteristic to 0x0000.
  2. The Lower Tester writes a valid Op Code to the Object List Control Point that is supported by the IUT.
  3. Verify that the IUT response meets the requirements of the service.

- Expected Outcome

  Pass verdict

  The IUT rejects the Write Request by sending an Error Response with an Attribute Protocol Error Code set to “Client Characteristic Configuration Descriptor Improperly Configured” (0xFD).

  The IUT returns to a stable state.

4.15 Service Procedure – Object Metadata Error Handling

Verify compliant operation when the Lower Tester performs operations on object metadata an error results.

OTS/SR/OME/BI-01-C [Object Metadata - Object Name Already Exists]

- Test Purpose

  Verify that the IUT responds appropriately when a Client attempts to write a name to the Object Name characteristic that is identical to the name of another object stored on the IUT.

- Reference

  [3] 1.10, 3.3.2.8.1

- Initial Condition

  - At least two objects are stored on the IUT.
  - The object names are short enough to fit within the payload of a GATT Write sub-procedure when the default ATT_MTU size is used.
  - Perform the preamble described in Section 4.2.2.

- Test Procedure

  1. The Lower Tester writes the First Op Code (0x01) to the Object List Control Point.
  2. The Lower Tester reads the Object Name characteristic and stores the result.
  3. The Lower Tester writes the Next Op Code (0x04) to the Object List Control Point.
  4. The IUT, after sending a Write Response to acknowledge the write to the Object List Control Point, sends an indication of the Object List Control Point characteristic.
  5. The Lower Tester receives an ATT_Handle_Value_Indication from the IUT containing the Object List Control Point characteristic handle and value.
  6. The Lower Tester sends an ATT_Handle_Value_Confirmation to the IUT.
  7. The Lower Tester attempts to write an identical name to that which it read in step 2 to the Object Name characteristic.
  8. Verify that the IUT response meets the requirements of the service.
  9. The Lower Tester reads the Object Name characteristic.

- Expected Outcome

  Pass verdict

  In step 4, the IUT sends an indication of the Object List Control Point characteristic containing the Response Code Op Code (0x70), the Request Op Code (0x04) followed by the Result Code for ‘Success’ (0x01) with no Response Parameter.

  The IUT rejects the write request in step 7 by sending an Error Response with an Attribute Protocol Application Error Code set to “Object Name Already Exists” (0x83).

  The value of the Object Name characteristic read in step 9 does not equal the value that was attempted to be written in step 7 as the write request was rejected. (See Note 1.)

  The IUT returns to a stable state.

  Note 1: if the IUT fails this test case, ensure the original value is restored to the affected object name. This is done to ensure that there is no impact on subsequent test cases.

OTS/SR/OME/BI-02-C [Object Metadata – Object Not Selected]

- Test Purpose

  Verify that the IUT responds appropriately when a Client attempts to read or write an Object Metadata characteristic while the Current Object is an Invalid Object.

  Verify that when an object ceases to be an Invalid Object following a change in filter conditions, the “Object Not Selected” error response is no longer triggered.

- Reference

  [3] 1.9, 1.10, 3.3.2.8.1

- Initial Condition

  - At least one object is stored on the IUT.
  - Establish an ATT Bearer connection between the Lower Tester and IUT as described in Section 4.2.1. If IUT permissions for the characteristic require a specific security mode or security level, establish a connection meeting those requirements.
  - The object that is to be used in this test procedure is identified in the IXIT [7] by its Object Name. If the Object List Control Point (OLCP) is supported by the IUT (i.e., the IUT supports more than one object), run the preamble described in Section 4.2.2 to ensure that the object to be tested is selected as the Current Object.

- Test Procedure

  1. The Lower Tester reads one or more of the supported Object Metadata characteristics to prepare values for the Object List Filter characteristic and Object Metadata characteristic that are written in the following steps.
  2. The Lower Tester writes a value to one of the Object List Filter characteristics. The value is chosen such that the Current Object does not pass the filter conditions specified. This causes the object to be excluded by the filter conditions and hence the Current Object becomes an Invalid Object.
  3. The Lower Tester attempts to write a new value to one of the supported Object Metadata characteristics that is writable.
  4. Verify that the IUT response(s) meet the requirements of the service.

5. The Lower Tester sets all the Object List Filter characteristics to “No Filter” (0x00).
6. The Lower Tester reads the value of the same Object Metadata characteristic that was used in step 3.

- Expected Outcome

  Pass verdict

  The IUT rejects the write request in step 3 by sending an Error Response with an Attribute Protocol Application Error Code set to “Object Not Selected” (0x81).

  In step 6, the value of the characteristic is read successfully. The value of the Object Metadata characteristic read in step 6 is not equal to the value that was attempted to be written in step 3.

  The IUT returns to a stable state.

OTS/SR/OME/BI-03-C [Object Metadata – Object Properties – Write Request Rejected]

- Test Purpose

  Verify that the IUT responds appropriately when a Client attempts to write a RFU value to the Object Properties characteristic.

- Reference

  [3] 1.10, 3.2.8.1

- Initial Condition

  - At least one object is stored on the IUT.
  - Establish an ATT Bearer connection between the Lower Tester and IUT as described in Section 4.2.1. If IUT permissions for the characteristic require a specific security mode or security level, establish a connection meeting those requirements. If the IUT requires bonding, then the Lower Tester performs a bonding procedure.
  - The object that is to be used in this test procedure is identified in the IXIT [7] by its Object Name. If the Object List Control Point (OLCP) is supported by the IUT (i.e., the IUT supports more than one object), run the preamble described in Section 4.2.4 to ensure that the object to be tested is selected as the Current Object.

- Test Procedure

  1. The Lower Tester reads the Object Properties characteristic and stores the value it has read.
  2. The Lower Tester attempts to write a RFU value to the Object Properties characteristic.
  3. Verify that the IUT response meets the requirements of the service.
  4. The Lower Tester reads the value of the Object Properties characteristic.
  5. The Lower Tester writes the value that it read in step 1 back to the Object Properties characteristic. This step is performed to ensure that there is no impact on a subsequent test case.

- Expected Outcome

  Pass verdict

  In step 2, the IUT rejects the Write Request by sending an Error Response with an Attribute Protocol Application Error Code set to “Write Request Rejected” (0x80).

  The value of the Object Properties characteristic read in step 4 does not equal the RFU value that was attempted to be written in step 2 as the Write Request was rejected.

  The IUT returns to a stable state.

4.16 Real Time Clock Operations

Verify behavior where the expected outcome depends on whether the IUT has access to a Real Time Clock (RTC) or not.

OTS/SR/RTC/BV-01-C [RTC - Object Last-Modified - With RTC]

- Test Purpose

  Verify that an IUT that has access to a RTC updates the value of the Object Last-Modified characteristic in response to a change to the object.

  Verify that the properties of this characteristic do not permit writing to the characteristic over the air.

- Reference

  [3] 3.2.6

- Initial Condition

  - The IUT has access to a RTC for the purposes of updating the object last-modified metadata. Ensure that the RTC is running and has been set to a valid time.
  - The handle of each characteristic value referenced in the test cases below has been previously discovered by the Lower Tester during the test procedure in Section 4.3 or Section 4.4 or is known to the Lower Tester by other means.
  - The properties of the Object Last-Modified characteristic are known from running the procedure described in Section 4.3.
  - At least one object is stored on the IUT.
  - Establish an ATT Bearer connection between the Lower Tester and IUT as described in Section 4.2.1. If IUT permissions require a specific security mode or security level, establish a connection meeting those requirements.
  - The object that is to be used in this test procedure is identified in the IXIT [7] by its Object Name. If the Object List Control Point (OLCP) is supported by the IUT (i.e., the IUT supports more than one object), run the preamble described in Section 4.2.4 to ensure that the object to be tested is selected as the Current Object.

- Test Procedure

  1. The Lower Tester reads the Object Last-Modified characteristic value.
  2. The Lower Tester records the date/time value read.
  3. The Lower Tester performs an action that causes the contents of the Current Object to be modified by writing some data to the object contents as described in OTS/SR/OASP/BV-06-C [OACP – Write – Entire Object], or OTS/SR/OASP/BV-07-C [OACP – Write – Patching an Object], or OTS/SR/OASP/BV-08-C [OACP – Write – Increased Allocated Size] or, if this is not supported by the IUT, the Upper Tester carries out an implementation-specific procedure to modify the object (e.g., directly at the Server IUT via the UI).
  4. The IUT updates the object last-modified metadata of the object.
  5. The Lower Tester reads the Object Last-Modified characteristic value again.

- Expected Outcome

  Pass verdict

  The properties of the Object Last-Modified characteristic report that writing to this characteristic is not permitted.

The value read in step 5 is not equal to the value read in step 1 (this is accepted as evidence that the timestamp value has been updated by the IUT).

The value read in step 5 is consistent with the time at which step 3 was performed.

OTS/SR/RTC/BV-02-C [RTC - Object Last-Modified - Without RTC]

- Test Purpose

  Verify that the IUT handles the object last-modified metadata appropriately when an object is modified and the IUT does not have access to a Real Time Clock (RTC).

  Verify that the characteristic properties support writing a value to the Object Last Modified characteristic.

- Reference

  [3] 3.2.6.1

- Initial Condition

  - The IUT does not have access to an RTC.
  - The handle of each characteristic value referenced in the test cases below has been previously discovered by the Lower Tester during the test procedure in Section 4.3 or Section 4.4 or is known to the Lower Tester by other means.
  - The properties of the Object Last-Modified characteristic are known from running the procedure described in Section 4.3.
  - At least one object is stored on the IUT.
  - Establish an ATT Bearer connection between the Lower Tester and IUT as described in Section 4.2.1. If IUT permissions require a specific security mode or security level, establish a connection meeting those requirements.
  - The object that is to be used in this test procedure is identified in the IXIT [7] by its Object Name. If the Object List Control Point (OLCP) is supported by the IUT (i.e., the IUT supports more than one object), run the preamble described in Section 4.2.4 to ensure that the object to be tested is selected as the Current Object.

- Test Procedure

  1. The Lower Tester writes a valid date/time value to the Object Last-Modified characteristic. The value written to each field of the characteristic is not equal to zero.
  2. The Lower Tester performs an action that causes the contents of the Current Object to be modified. e.g., by writing some data to the object contents as described in OTS/SR/OASP/BV-06-C [OACP – Write – Entire Object], or OTS/SR/OASP/BV-07-C [OACP – Write – Patching an Object], or OTS/SR/OASP/BV-08-C [OACP – Write – Increased Allocated Size] or, if this is not supported by the IUT, the Upper Tester may carry out an implementation-specific procedure to modify the object (e.g., directly at the Server IUT via the UI).
  3. The IUT updates the object last-modified metadata of the object with the special value that indicates that a valid date/time is not available.
  4. The Lower Tester reads the Object Last-Modified characteristic value.

- Expected Outcome

  Pass verdict

  A valid date/time value is successfully written to the Object Last-Modified characteristic in step 1.

The value read in step 4 has all fields set to 0, the special value that indicates that a valid date/time is not available.

4.17 Object List Filter Operations

Verify compliant operation when the Lower Tester writes filter conditions to the Object List Filter characteristics. The verification is done for a selection of filter conditions.

OTS/SR/OLF/BV-01-C [Object List Filter – Object Name]

- Test Purpose

  Verify that the IUT behaves appropriately when a Client writes any of the name-based filter values to an Object List Filter characteristic.

- Reference

  [3] 3.5.1.2, 3.5.1.3, 3.5.1.4, 3.5.1.5

- Initial Condition

  - At least two objects are stored on the IUT.
  - The handle of each characteristic and characteristic descriptor referenced in the test procedure below has been previously discovered by the Lower Tester during the test procedure in Section 4.3 or Section 4.4 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 IUT permissions for the Object Action Control Point characteristic require a specific security mode or security level, establish a connection meeting those requirements. If the IUT requires bonding, then the Lower Tester performs a bonding procedure.
  - Since the IUT supports three instances of the Object List Filter characteristic, these will be referred to in the following test procedure as ‘Object List Filter1’, ‘Object List Filter2’ and ‘Object List Filter3’. It does not matter in which order these labels are assigned to the Object List Filter characteristics.

- Test Procedure

  1. The Lower Tester writes the value ‘No Filter’ (0x00) to Object List Filter1 with no parameter.
  2. The Lower Tester writes the value ‘No Filter’ (0x00) to Object List Filter2 with no parameter.
  3. The Lower Tester writes the value ‘No Filter’ (0x00) to Object List Filter3 with no parameter.
  4. The Lower Tester discovers the object name of all objects presently available on the IUT by running the procedure described in Section 4.2.3.
  5. The Lower Tester selects the name of one of the objects discovered in step 4 to use in the remainder of this test procedure, picking an object name that is at least three characters long and that does not exceed the capacity of the negotiated ATT_MTU size. The selected name will be referred to in the remainder of this test procedure as “TEST_NAME”.
  6. To create a filter parameter, the Lower Tester truncates TEST_NAME to a length of less than 19 octets by removing at least one character from the end of the name and writes the value ‘Name Starts With’ (0x01) to Object List Filter1 with the parameter value set to match the truncated name.
  7. The Lower Tester discovers the object name of all objects remaining available on the IUT after applying the filter by running the procedure described in Section 4.2.3 a second time.
  8. The Lower Tester compares the lists of objects obtained before and after filtering.
  9. The Lower Tester writes the value ‘No Filter’ (0x00) to Object List Filter1 with no parameter.

10. To create another filter parameter, the Lower Tester truncates the original TEST_NAME to a length of less than 19 octets by removing at least one character from the beginning of the name and writes the value ‘Name Ends With’ (0x02) to Object List Filter2 with the parameter value set to match the truncated name.
11. The Lower Tester discovers the object name of all objects remaining available on the IUT after applying the filter by running the procedure described in Section 4.2.3 a third time.
12. The Lower Tester compares the lists of objects obtained before and after filtering.
13. The Lower Tester writes the value ‘No Filter’ (0x00) to Object List Filter2 with no parameter.
14. The Lower Tester truncates TEST_NAME to a length of less than 19 octets by removing at least one character from the beginning of the name and at least one character from the end of the name and writes the value ‘Name Contains’ (0x03) to Object List Filter3 with the parameter value set to match the truncated name.
15. The Lower Tester discovers the object name of all objects remaining available on the IUT after applying the filter by running the procedure described in Section 4.2.3 a fourth time.
16. The Lower Tester compares the lists of objects obtained before and after filtering.
17. The Lower Tester writes the value ‘No Filter’ (0x00) to Object List Filter3 with no parameter.
18. The Lower Tester writes the value ‘Name is Exactly’ (0x04) to Object List Filter1 with the parameter value set to match the original value of TEST_NAME.
19. The Lower Tester discovers the object name of all objects remaining available on the IUT after applying the filter by running the procedure described in Section 4.2.3 a fifth time.
20. The Lower Tester compares the lists of objects obtained before and after filtering.
21. The Lower Tester writes the value ‘No Filter’ (0x00) to Object List Filter1 with no parameter.
22. The Lower Tester writes the value ‘Name is Exactly’ (0x04) to Object List Filter2 with the parameter value set to a value that does not match the name of any object discovered in step 4. i.e., the filter conditions now do not match any object on the IUT.
23. The Lower Tester discovers the object name of all objects remaining available on the IUT after applying the filter by running the procedure described in Section 4.2.3 a sixth time.
24. The Lower Tester writes the value ‘No Filter’ (0x00) to Object List Filter2 with no parameter. This step is included to ensure that the Object List Filter conditions do not impact a subsequent test case.

- Expected Outcome

  Pass verdict

  In steps 8, 12, and 16, only the object(s) whose name matches the truncated TEST_NAME in accordance with the specified filter conditions remain in the filtered list of objects.

  In step 20, only the single object whose name exactly matches the name selected in step 5 (the “TEST_NAME”) remains in the filtered list of objects.

  In step 23, a “No Object” error response is received because no object remains in the filtered list.

OTS/SR/OLF/BV-02-C [Object List Filter – Object Type]

- Test Purpose

  Verify that the IUT behaves appropriately when a Client writes the ‘Object Type (UUID)’ filter value to an Object List Filter characteristic.

- Reference

  [3] 3.5.1.6

- Initial Condition
  - At least two objects are stored on the IUT. If possible, the objects are arranged to have different Object Type metadata (i.e., different UUIDs for Object Type). However, having different types is optional given some IUTs may intentionally support only a single type of object.
  - The handle of each characteristic and characteristic descriptor referenced in the test procedure below has been previously discovered by the Lower Tester during the test procedure in Section 4.3 or Section 4.4 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 IUT permissions for the Object Action Control Point characteristic require a specific security mode or security level, establish a connection meeting those requirements. If the IUT requires bonding, then the Lower Tester performs a bonding procedure.
  - Since the IUT supports three instances of the Object List Filter characteristic, these will be referred to in the following test procedure as ‘Object List Filter1’, ‘Object List Filter2’ and ‘Object List Filter3’. It does not matter in which order these labels are assigned to the Object List Filter characteristics.
- Test Procedure
  1. The Lower Tester writes the value ‘No Filter’ (0x00) to Object List Filter1 with no parameter.
  2. The Lower Tester writes the value ‘No Filter’ (0x00) to Object List Filter2 with no parameter.
  3. The Lower Tester writes the value ‘No Filter’ (0x00) to Object List Filter3 with no parameter.
  4. The Lower Tester discovers the Object Name and Object Type of each object presently available on the IUT by running the procedure described in Section 4.2.3.
  5. The Lower Tester selects the Object Type UUID of one of the objects discovered in step 4 to use in the remainder of this test procedure. This value will be referred to as “TEST_TYPE” in the remainder of this test procedure.
  6. The Lower Tester writes the value ‘Object Type (UUID)’ (0x05) to Object List Filter1 with the parameter value set to match the TEST_TYPE.
  7. The Lower Tester discovers the objects remaining available on the IUT after applying the filter by running the procedure described in Section 4.2.3 a second time.
  8. The Lower Tester compares the lists of objects obtained before and after filtering.
  9. The Lower Tester writes the value ‘No Filter’ (0x00) to Object List Filter1 with no parameter.
  10. The Lower Tester writes the value ‘Object Type (UUID)’ (0x05) to Object List Filter2 with the parameter value set to be different from all Object Types that were discovered in step 4. i.e., the filter conditions now do not match any object on the IUT.
  11. The Lower Tester discovers the objects remaining available on the IUT after applying the filter by running the procedure described in Section 4.2.3 a third time.
  12. The Lower Tester writes the value ‘No Filter’ (0x00) to Object List Filter2 with no parameter. This step is included to ensure that the Object List Filter conditions do not impact a subsequent test case.
- Expected Outcome

  Pass verdict

  In step 7, all objects whose Object Type matches the UUID selected in step 5 (the “TEST_TYPE”) remain in the filtered list of objects and no object whose Object Type does not matches the UUID selected in step 5 remains in the filtered list.

  In step 11, a “No Object” error response is received because no object remains in the filtered list.

OTS/SR/OLF/BV-03-C [Object List Filter – Current Size]

- Test Purpose

  Verify that the IUT behaves appropriately when a Client writes the 'Current Size between' filter value to an Object List Filter characteristic. In addition, verify that the IUT responds with the required error response if the Client attempts to write invalid filter criteria.

- Reference

  [3] 3.5.1.9

- Initial Condition

  - At least two objects are stored on the IUT. The objects have different Current Size metadata, except in the case that objects of different sizes are not supported by the IUT.
  - The handle of each characteristic and characteristic descriptor referenced in the test procedure below has been previously discovered by the Lower Tester during the test procedure in Section 4.3 or Section 4.4 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 IUT permissions for the Object Action Control Point characteristic require a specific security mode or security level, establish a connection meeting those requirements. If the IUT requires bonding, then the Lower Tester performs a bonding procedure.
  - Since the IUT supports three instances of the Object List Filter characteristic, these will be referred to in the following test procedure as 'Object List Filter1', 'Object List Filter2' and 'Object List Filter3'. It does not matter in which order these labels are assigned to the Object List Filter characteristics.

- Test Procedure

  1. The Lower Tester writes the value 'No Filter' (0x00) to Object List Filter1 with no parameter.
  2. The Lower Tester writes the value 'No Filter' (0x00) to Object List Filter2 with no parameter.
  3. The Lower Tester writes the value 'No Filter' (0x00) to Object List Filter3 with no parameter.
  4. The Lower Tester discovers the Object Name and Current Size of each object presently available on the IUT by running the procedure described in Section 4.2.3.
  5. The Lower Tester selects the Current Size value of one of the objects discovered in step 4 to use in the remainder of this test procedure. This value will be referred to as "TEST_SIZE" in the remainder of this test procedure.
  6. The Lower Tester writes the value 'Current Size between (inclusive)' (0x08) to Object List Filter1 with the size1 and size2 fields of the parameter value set as follows: size1 = TEST_SIZE and size2 = TEST_SIZE.
  7. The Lower Tester discovers the objects remaining available on the IUT after applying the filter by running the procedure described in Section 4.2.3 a second time.
  8. The Lower Tester compares the lists of objects obtained before and after filtering.
  9. The Lower Tester writes the value 'No Filter' (0x00) to Object List Filter1 with no parameter.
  10. The Lower Tester writes the value 'Current Size between (inclusive)' (0x08) to Object List Filter2 with the size1 and size2 fields of the parameter value set as follows: size1 = (TEST_SIZE + 10) and size2 = (TEST_SIZE + 20).
  11. The Lower Tester discovers the objects remaining available on the IUT after applying the filter by running the procedure described in Section 4.2.3 a third time.
  12. The Lower Tester compares the lists of objects obtained before and after filtering.
  13. The Lower Tester writes the value 'No Filter' (0x00) to Object List Filter2 with no parameter.

14. The Lower Tester writes the value ‘Current Size between (inclusive)’ (0x08) to Object List Filter3 with the size1 and size2 fields of the parameter value set as follows: size1 = (TEST_SIZE – 10) and size2 = (TEST_SIZE + 10).
15. The Lower Tester discovers the objects remaining available on the IUT after applying the filter by running the procedure described in Section 4.2.3 a fourth time.
16. The Lower Tester compares the lists of objects obtained before and after filtering.
17. The Lower Tester writes the value ‘No Filter’ (0x00) to Object List Filter3 with no parameter.
18. The Lower Tester writes the value ‘Current Size between (inclusive)’ (0x08) to Object List Filter1 with the size1 and size2 fields of the parameter value set as follows: size1 = (TEST_SIZE + 10) and size2 = (TEST_SIZE - 10).
19. The IUT rejects the write request in step 18.
20. The Lower Tester writes 0x00 (No Filter) to each Object List Filter characteristic value. This step is performed to ensure that there is no impact on subsequent test cases.

- Expected Outcome

  Pass verdict

  In step 7, the object selected in step 5 and any other object available on the IUT that has an identical Current Size to the TEST_SIZE remain in the filtered list of objects and no object with a different Current Size remains in the filtered list.

  In step 11, the object selected in step 5 is not in the filtered list. Any other object available on the IUT that has an Current Size that is greater than or equal to size1 and also less than or equal to size2, if any, remains in the filtered list of objects (i.e., TEST_SIZE + 10 <= s <= TEST_SIZE + 20). No object with a Current Size that does not meet these criteria remains in the filtered list. In the event that there is no object remaining in the filtered list, a “No Object” error response is received.

  In step 15, the object selected in step 5 and any other object available on the IUT that meets the criteria (i.e., TEST_SIZE-10 <= s <= TEST_SIZE + 10) remains in the filtered list of objects. No object with a Current Size that does not meet these criteria remains in the filtered list.

  In step 19, the IUT recognizes that the request with size1 > size2 is not compliant with the service and sends an Error Response with an Attribute Protocol Application Error Code set to “Write Request Rejected” (0x80). The IUT returns to a stable state.

OTS/SR/OLF/BV-04-C [Object List Filter – Logical Combinations]

- Test Purpose

  Verify that the IUT behaves appropriately when a Client writes a combination of conditions to the Object List Filter characteristics, combining these by using logical “AND” operations.

- Reference

  [3] 3.5.1

- Initial Condition

  - At least two objects are stored on the IUT.
  - The handle of each characteristic and characteristic descriptor referenced in the test procedure below has been previously discovered by the Lower Tester during the test procedure in Section 4.3 or Section 4.4 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 IUT permissions for the Object Action Control Point characteristic require a

specific security mode or security level, establish a connection meeting those requirements. If the IUT requires bonding, then the Lower Tester performs a bonding procedure.

- Since the IUT supports three instances of the Object List Filter characteristic, these will be referred to in the following test procedure as ‘Object List Filter1’, ‘Object List Filter2’ and ‘Object List Filter3’. It does not matter in which order these labels are assigned to the Object List Filter characteristics.

- Test Procedure

1. The Lower Tester writes the value ‘No Filter’ (0x00) to Object List Filter1 with no parameter.
2. The Lower Tester writes the value ‘No Filter’ (0x00) to Object List Filter2 with no parameter.
3. The Lower Tester writes the value ‘No Filter’ (0x00) to Object List Filter3 with no parameter.
4. The Lower Tester discovers the Object Name, Object Type and Allocated Size of each object presently available on the IUT by running the procedure described in Section 4.2.3.
5. The Lower Tester selects the Object Name, Object Type and Allocated Size metadata of one of the objects discovered in step 4 to use in the remainder of this test procedure. The three test values obtained in this step will be referred to as “TEST_NAME”, “TEST_TYPE” and “TEST_SIZE”, respectively, in the remainder of this test procedure. If the Object Name is longer than 19 octets, the Lower Tester truncates the name to a length of greater than 15 octets and less than 20 octets by removing characters from the end of the name and then uses the resulting value as the TEST_NAME.
6. The Lower Tester writes the value ‘Name Contains’ (0x03) to Object List Filter1 with the parameter value set to match the TEST_NAME.
7. The Lower Tester writes the value ‘Object Type (UUID)’ (0x05) to Object List Filter2 with the parameter value set to match the TEST_TYPE.
8. The Lower Tester writes the value ‘Allocated Size between (inclusive)’ (0x09) to Object List Filter3 with the size1 and size2 fields of the parameter value set as follows: size1 = (TEST_SIZE – 10) and size2 = (TEST_SIZE + 10).
9. The Lower Tester discovers the objects remaining available on the IUT after applying the filter by running the procedure described in Section 4.2.3 a second time.
10. The Lower Tester compares the lists of objects obtained before and after filtering.
11. The Lower Tester writes the value ‘Object Type (UUID)’ (0x05) to Object List Filter2 with the parameter value set to be different from all Object Types that were discovered in step 4. i.e., the filter conditions now do not match any object on the IUT.
12. The Lower Tester discovers the objects remaining available on the IUT after applying the filter by running the procedure described in Section 4.2.3 a third time.
13. The Lower Tester receives an error response of ‘No Object’ (0x07) from the IUT.
14. The Lower Tester writes the value ‘Object Type (UUID)’ (0x05) to Object List Filter1 with the parameter value set to match the TEST_TYPE.
15. The Lower Tester writes the value ‘No Filter’ (0x00) to Object List Filter2 with no parameter.
16. The Lower Tester writes the value ‘Allocated Size Between (inclusive)’ (0x09) to Object List Filter3 with the size1 and size2 fields of the parameter value set as follows: size1 = 0 octets and size2 = (2 x TEST_SIZE).
17. The Lower Tester discovers the objects remaining available on the IUT after applying the filter by running the procedure described in Section 4.2.3 a fourth time.
18. The Lower Tester compares the lists of objects obtained before and after filtering.
19. The Lower Tester writes the value ‘No Filter’ (0x00) with no parameter to Object List Filter1 and Object List Filter3.
20. The Lower Tester discovers the objects remaining available on the IUT after applying the filter by running the procedure described in Section 4.2.3 a fifth time.

21. The Lower Tester compares the lists of objects obtained before and after filtering.
22. The Lower Tester writes 0x00 to each Object List Filter characteristic. This step is performed to ensure that there is no impact on subsequent test cases.

- Expected Outcome

  Pass verdict

  In step 9, the object selected in step 5 plus any other object whose metadata match all of the criteria set by Object List Filter1 (TEST_NAME) AND Object List Filter2 (TEST_TYPE) AND Object List Filter3 (TEST_SIZE - 10 <= s <= TEST_SIZE + 10) remain in the filtered list of objects and no other object remains in the filtered list.

  In steps 12 and 13, no object remains in the filtered list.

  In step 17, the object selected in step 5 plus any other object whose metadata match all of the criteria set by Object List Filter1 (TEST_TYPE) AND Object List Filter3 (s <= (2 x TEST_SIZE)) remain in the filtered list of objects and no other object remains in the filtered list.

  In step 20, all available objects are included in the list of objects – the list is not filtered.

OTS/SR/OLF/BI-01-C [Object List Filter – Write Request Rejected]

- Test Purpose

  Verify that the IUT responds appropriately when a Client attempts to write a RFU value to the Object List Filter characteristic.

- Reference

  [3] 1.10, 3.5

- Initial Condition

  - Perform the preamble described in Section 4.2.2.

- Test Procedure

1. The Lower Tester writes the First Op Code (0x01) to the Object List Control Point.
2. The IUT, after sending a Write Response to acknowledge the write to the Object List Control Point, sends an indication of the Object List Control Point characteristic.
3. The Lower Tester receives an ATT_Handle_Value_Indication from the IUT containing the Object List Control Point characteristic handle and value.
4. The Lower Tester sends an ATT_Handle_Value_Confirmation to the IUT.
5. The Lower Tester reads the three Object List Filter characteristics and stores the values it has read.
6. The Lower Tester attempts to write a RFU value to one of the Object List Filter characteristics.
7. Verify that the IUT response meets the requirements of the service.
8. The Lower Tester reads the value of the same Object List Filter characteristic to which it attempted to write in step 6.
9. The Lower Tester writes 0x00 to the Object List Filter characteristic used in step 6. This step is performed to ensure that there is no impact on subsequent test cases.

- Expected Outcome

  Pass verdict

  In step 2, the IUT sends an indication of the Object List Control Point characteristic containing the Response Code Op Code (0x70), the Request Op Code (0x01) followed by the Result Code for ‘Success’ (0x01) with no Response Parameter.

  In step 7, the IUT rejects the Write Request by sending an Error Response with an Attribute Protocol Application Error Code set to “Write Request Rejected” (0x80).

  The value of the Object List Filter characteristic read in step 8 does not equal the value that was attempted to be written in step 6, as the write request was rejected.

  The IUT returns to a stable state.

4.18 Object Changed Operations

Verify compliant operation when the IUT supports the Object Changed characteristic and a change is made to an object or its metadata.

OTS/SR/OC/BV-01-C [Object Changed at Server]

- Test Purpose

  Verify that the IUT sends an indication of the Object Changed characteristic when a change is made to an object’s metadata by means of an action performed at or within the Server (e.g., user action via the Server’s UI).

- Reference

  [3] 3.6

- Initial Condition

  - At least one object is stored on the IUT.
  - The object that is to be used in this test procedure is identified in the IXIT [7] by its Object Name.
  - If the Object List Control Point (OLCP) is supported by the IUT (i.e., the IUT supports more than one object), run the preamble described in Section 4.2.4 to ensure that the object to be tested is selected as the Current Object. Otherwise, establish an ATT Bearer connection between the Lower Tester and IUT as described in Section 4.2.1.
  - If IUT permissions for the Object Action Control Point characteristic require a specific security mode or security level, establish a connection meeting those requirements. If the IUT requires bonding, then the Lower Tester performs a bonding procedure.

- Test Procedure

  1. Configure the Object Changed characteristic for indication by writing value 0x0002 to the Client Characteristic Configuration descriptor of the characteristic.
  2. The Lower Tester reads the value of the Object ID characteristic as described in OTS/SR/CR/BV-07-C [Characteristic Read – Object ID].
  3. The Upper Tester performs an action to change the value of an item of metadata of the object on the IUT via the UI or another out of band mechanism. The metadata available to change depend on the implementation.
  4. The Lower Tester receives an ATT_Handle_Value_Indication from the IUT containing the Object Changed characteristic.

5. The Lower Tester sends an ATT_Handle_Value_Confirmation to the IUT.
6. Verify that the characteristic value meets the requirements of the service.

- Expected Outcome

  Pass verdict

  The IUT sends one indication of the Object Changed characteristic.

  The value of the Object Changed characteristic contains an Object ID field whose value matches the Object ID value read in step 2.

  The value of the Object Changed characteristic received in step 4 contains a Flags field of which all bits are set to 0 except for bit 2 (“Change occurred to the object metadata”) which is set to 1.

OTS/SR/OC/BV-02-C [Object Changed by another Client]

- Test Purpose

  Verify that the IUT sends an indication of the Object Changed characteristic when a change is made to an object’s metadata while the IUT is connected, owing to a new value being written by another Client. The two Clients are connected concurrently.

- Reference

  [3] 3.6

- Initial Condition

  - This test case requires two Lower Testers with different Bluetooth addresses, referred to below as Lower Tester 1 and Lower Tester 2.
  - At least one object is stored on the IUT.
  - The object that is to be used in this test procedure is identified in the IXIT [7] by its Object Name.
  - Lower Tester 1 establishes an ATT Bearer connection between the Lower Tester and IUT as described in Section 4.2.1.
  - Lower Tester 2 establishes an ATT Bearer connection between the Lower Tester and IUT as described in Section 4.2.1.
  - If the Object List Control Point (OLCP) is supported by the IUT (i.e., the IUT supports more than one object), Lower Tester 2 performs the preamble described in Section 4.2.4 to ensure that the object to be used in the test case is selected as the Current Object. Otherwise, Lower Tester 2 establishes an ATT Bearer connection with the IUT as described in Section 4.2.1. If IUT permissions for the Object Action Control Point characteristic require a specific security mode or security level, establish a connection meeting those requirements. If the IUT requires bonding, then the Lower Testers perform a bonding procedure.

- Test Procedure

1. Lower Tester 1 configures the Object Changed characteristic for indication by writing value 0x0002 to the Client Characteristic Configuration descriptor of the characteristic. Lower Tester 1 remains connected.
2. Lower Tester 2 configures the Object Changed characteristic for indication by writing value 0x0002 to the Client Characteristic Configuration descriptor of the characteristic.
3. Lower Tester 2 reads the values of the supported object metadata characteristics, including the Object ID characteristic. If the value of the Object Name is read and is long enough to require the

GATT Read Long sub-procedure to be used, Lower Tester 1 reads the object name in its entirety using the GATT Read Long Characteristic Values sub-procedure.

4. Lower Tester 2 writes a new value to a supported item of metadata of the object, using the procedure described in Section 4.7.
5. Lower Tester 1 receives an ATT_Handle_Value_Indication from the IUT containing the Object Changed characteristic.
6. Lower Tester 1 sends an ATT_Handle_Value_Confirmation to the IUT.
7. Verify that the Object Changed characteristic value received in the indication meets the requirements of the service.
8. Lower Tester 2 writes back the original value, read in step 3, to the characteristic whose value was modified in step 4. This is done to ensure there is no impact on a subsequent test case.

- Expected Outcome

Pass verdict

The IUT sends an indication of the Object Changed characteristic to Lower Tester 1.

The value of the Object Changed characteristic contains an Object ID field whose value matches the Object ID value read in step 3.

The value of the Object Changed characteristic contains a Flags field of which all bits are set to 0 except for bit 0 (Source of Change = Client) and bit 2 (“Change occurred to the object metadata”) which are both set to 1.

Lower Tester 2 does not receive an indication of the Object Changed characteristic.

4.19 Object Transfer Disconnection / Timeout

OTS/SR/OTD/BI-01-C [Malformed Object - Deleted when Connection Terminated]

- Test Purpose

Verify that the IUT behaves appropriately when a new object is created and a disconnection occurs before the new object has been given valid Object Name metadata.

- Reference

[3] 3.7

- Initial Condition

- The handle of each characteristic and characteristic descriptor referenced in the test procedure below has been previously discovered by the Lower Tester during the test procedure in Section 4.3 or Section 4.4 or is known to the Lower Tester by other means.
- Perform the preamble described in Section 4.2.2.
- The object types and object sizes supported by the IUT are known from the IXIT [7].

- Test Procedure

1. The Lower Tester writes the Create Op Code (0x01) to the Object Action Control Point with a Parameter Value representing an object type and size that is supported by the IUT. e.g., the Object Type field could contain the UUID «Firmware» and the Size field could contain 0x00000400 (representing an allocated object size of 1024 octets) - acceptable values will depend on the implementation.
2. The IUT, after sending a Write Response to acknowledge the write to the Object Action Control Point, sends an indication of the Object Action Control Point characteristic.

3. The Lower Tester receives an ATT_Handle_Value_Indication from the IUT containing the Object Action Control Point characteristic handle and value.
4. The Lower Tester sends an ATT_Handle_Value_Confirmation to the IUT.
5. The Lower Tester reads all of the supported Object Metadata characteristic values associated with the newly created object, using the procedure described in Section 4.5. The values reported are noted.
6. The Lower Tester does not write the Object Name characteristic for that particular new object.
7. The Lower Tester terminates the connection.
8. The Lower Tester initiates a new ATT Bearer connection between the Lower Tester and IUT as described in Section 4.2.1. If IUT permissions for the Object Action Control Point characteristic require a specific security mode or security level, establish a connection meeting those requirements. If the IUT requires bonding, then the Lower Tester performs a bonding procedure.
9. If the IUT supports the Object List Filter characteristics, the Lower Tester writes the value “No Filter” (0x00), with no parameter, to all three instances of the Object List Filter characteristic.
10. The Lower Tester performs the procedure described in Section 4.2.3 to compile a list of all the objects available on the IUT, if any, and associated object metadata.
11. The Lower Tester terminates the connection.

- Expected Outcome

  Pass verdict

  Steps 1–7:

  The IUT sends an indication of the Object Action Control Point characteristic containing the Response Code Op Code (0x60), the Request Op Code (0x01) followed by the Result Code for ‘Success’ (0x01) with no Response Parameter.

  The value reported in the Object Type characteristic is a UUID equal to the value of the Type parameter included with the Create Op Code.

  The value reported in the Allocated Size field of the Object Size characteristic is equal to or greater than the value of the Size parameter included with the Create Op Code.

  The value reported in the Current Size field of the Object Size characteristic is 0.

  The value reported in the Object ID characteristic, if supported, is a unique Object ID (i.e., locally unique within the service) that is not a RFU value and meets the requirements of the service.

  The value reported in the Object Name characteristic is a zero length string.

  The value of each field of the Object First-Created characteristic, if supported, is 0.

  The value of each field of the Object Last-Modified characteristic, if supported, is 0.

  The value of the Object Properties characteristic, if supported, includes at least bit 3 set to 1 (i.e., Write is True).

  Step 8–11:

  The Lower Tester does not find any object in the IUT with an Object Name that is a zero length string. The object whose metadata were read in step 5 is no longer present on the IUT.

  The IUT returns to a stable state.

OTS/SR/OTD/BI-02-C [Lock is Released when Connection Terminated]

- Test Purpose

  Verify that the IUT releases the lock of an object when the connection is terminated.

- Reference

  [3] 3.3.2.8.2.8, 3.7

- Initial Condition

  - At least one object is stored on the IUT.
  - This test case requires two Lower Testers, Lower Tester 1 and Lower Tester 2.
  - Lower Tester2 is connected to the IUT concurrently with Lower Tester1.
  - Perform the preamble described in Section 4.2.2 with each Lower Tester.
  - The object that is to be used in this test procedure is identified in the IXIT [7] by its Object Name. If the Object List Control Point (OLCP) is supported by the IUT (i.e., the IUT supports more than one object), run the preamble described in Section 4.2.4 to ensure that the object to be tested is selected as the Current Object.
  - Both Lower Testers select the same object as the Current Object.
  - The object properties of the Current Object permit writing.

- Test Procedure

  1. Both Lower Testers read the value of the Object Size characteristic.
  2. Both Lower Testers open an Object Transfer Channel.
  3. Induce the IUT to lock the Current Object (e.g., by starting an OACP Write operation to the same object using Lower Tester 2).
  4. The Lower Tester 1 writes the Write Op Code (0x06) to the Object Action Control Point with a Length parameter value equal to the Allocated Size field of the Object Size characteristic value read in step 1. The Offset parameter is set to 0. The Mode parameter is set to 0x00.
  5. The IUT, after sending a Write Response to acknowledge the write to the Object Action Control Point, sends an indication of the Object Action Control Point characteristic.
  6. The Lower Tester 1 receives an *ATT_Handle_Value_Indication* from the IUT containing the Object Action Control Point characteristic handle and value.
  7. Lower Tester 1 sends an *ATT_Handle_Value_Confirmation* to the IUT.
  8. Lower Tester 2 terminates its connection with the IUT in order to induce the IUT to remove the lock on the object.
  9. Lower Tester 1 repeats steps 4 to 7.
  10. Lower Tester 1 terminates the connection.

- Expected Outcome

  Pass verdict

  In the first pass through step 6, Lower Tester 1 receives an indication of the Object Action Control Point characteristic from the IUT, containing the Response Code Op Code (0x60), the Request Op Code (0x06) followed by the Result Code for ‘Object Locked’ (0x09) with no Response Parameter.

  In the second pass through step 6, Lower Tester 1 receives an indication of the Object Action Control Point characteristic from the IUT, containing the Response Code Op Code (0x60), the Request Op Code (0x06) followed by the Result Code for ‘Success (0x01) with no Response Parameter.

  The IUT returns to a stable state.

OTS/SR/OTD/BI-03-C [Lock is Released due to Timeout]

- Test Purpose

  Verify that the IUT releases the lock of an object when an object transfer via the Object Transfer Channel has stalled and the transfer has timed out.

- Reference

  [3] 3.8.1

- Initial Condition

  - At least one object is stored on the IUT.
  - This test case requires two Lower Testers, Lower Tester 1 and Lower Tester 2.
  - Perform the preamble described in Section 4.2.2 with both Lower Testers. The two Lower Testers are connected to the IUT concurrently.
  - The object that is to be used in this test procedure is identified in the IXIT [7] by its Object Name. If the Object List Control Point (OLCP) is supported by the IUT (i.e., the IUT supports more than one object), run the preamble described in Section 4.2.4 to ensure that the object to be tested is selected as the Current Object.
  - Both Lower Testers select the same object as the Current Object.
  - The object properties of the Current Object permit writing.

- Test Procedure

  1. Both Lower Testers read the value of the Object Size characteristic.
  2. Both Lower Testers open an Object Transfer Channel.
  3. Induce the IUT to lock the Current Object by starting an OACP Write operation to the same object using Lower Tester 2. i.e., Lower Tester 2 writes the Write Op Code (0x06) to the Object Action Control Point with a Length parameter value equal to the value of the Allocated Size field of the Object Size characteristic read in step 1. The Offset parameter is set to 0. The Mode parameter is set to 0x00.
  4. Lower Tester 2 commences sending object data to the IUT via the open Object Transfer Channel.
  5. The Lower Tester 1 writes the Write Op Code (0x06) to the Object Action Control Point with a Length parameter value equal to the value of the Allocated Size field of the Object Size characteristic read in step 1. The Offset parameter is set to 0. The Mode parameter is set to 0x00.
  6. The IUT, after sending a Write Response to acknowledge the write to the Object Action Control Point, sends an indication of the Object Action Control Point characteristic.
  7. The Lower Tester 1 receives an *ATT_Handle_Value_Indication* from the IUT containing the Object Action Control Point characteristic handle and value.
  8. The Lower Tester 1 sends an *ATT_Handle_Value_Confirmation* to the IUT.
  9. Before completing the transfer of the number of octets specified in the Length parameter, Lower Tester 2 ceases sending any further data to the IUT but does not terminate its connection with the IUT.
  10. Wait for at least 40 seconds.
  11. Lower Tester 1 repeats steps 5 to 8.
  12. Both Lower Testers terminate their connections to the IUT.

- Expected Outcome

  Pass verdict

  In the first pass through step 7, Lower Tester 1 receives an indication of the Object Action Control Point characteristic containing the Response Code Op Code (0x60), the Request Op Code (0x06) followed by the Result Code for 'Object Locked' (0x09) with no Response Parameter.

  In the second pass through step 7, Lower Tester 1 receives an indication of the Object Action Control Point characteristic containing the Response Code Op Code (0x60), the Request Op Code (0x06) followed by the Result Code for 'Success' (0x01) with no Response Parameter.

OTS/SR/OTD/BI-04-C [Object Transfer Operation Timeout – Object Write]

- Test Purpose

  Verify that the IUT can detect and handle a timeout occurring when an object transfer intended to write object data to the Server is only partially completed via the Object Transfer Channel.

  Verify that the IUT closes the Object Transfer Channel.

- Reference

  [3] 3.8.1

- Initial Condition

  - At least one object is stored on the IUT.
  - Perform the preamble described in Section 4.2.2.
  - The object that is to be used in this test procedure is identified in the IXIT [7] by its Object Name. If the Object List Control Point (OLCP) is supported by the IUT (i.e., the IUT supports more than one object), run the preamble described in Section 4.2.4 to ensure that the object to be tested is selected as the Current Object.
  - The object properties permit writing.

- Test Procedure

  1. The Lower Tester opens an Object Transfer Channel.
  2. The Lower Tester reads the value of the Object Size characteristic.
  3. The Lower Tester writes the Write Op Code (0x06) to the Object Action Control Point with a Length parameter value equal to the value of the Allocated Size field of the Object Size characteristic read in step 2. The Offset parameter is set to 0. The Mode parameter is set to 0x00.
  4. The IUT, after sending a Write Response to acknowledge the write to the Object Action Control Point, sends an indication of the Object Action Control Point characteristic.
  5. The Lower Tester receives an *ATT_Handle_Value_Indication* from the IUT containing the Object Action Control Point characteristic handle and value.
  6. The Lower Tester sends an *ATT_Handle_Value_Confirmation* to the IUT.
  7. The Lower Tester sends data consisting of fewer octets than the number of octets specified in the Length parameter (see step 4) to the IUT via the Object Transfer Channel opened in step 1.
  8. Before completing the transfer of the number of octets specified in the Length parameter, the Lower Tester ceases sending any further data to the IUT but does not terminate its connection with the IUT and the Lower Tester does not initiate closure of the Object Transfer Channel.
  9. The Lower Tester waits for at least 30 seconds to ensure that the Object Transfer Timeout has expired.
  10. The Object Transfer Channel is no longer open, having been closed by the IUT.

- Expected Outcome

  Pass verdict

  In step 4, the IUT sends an indication of the Object Action Control Point characteristic containing the Response Code Op Code (0x60), the Request Op Code (0x06) followed by the Result Code for ‘Success’ (0x01) with no Response Parameter.

  Object data is accepted by the IUT via the Object Transfer Channel.

  During steps 8 and 9, the IUT may store the partially transferred object or may delete it; either of these actions is acceptable.

  During step 9, the IUT closes the Object Transfer Channel.

  The IUT returns to a stable state.

OTS/SR/OTD/BI-05-C [Object Transfer Operation Timeout – Object Read – LE Transport]

- Test Purpose

  This test case applies only when using the LE transport.

  Verify that the IUT can detect and handle a timeout occurring when an object transfer intended to read object data from the Server is only partially completed via the Object Transfer Channel when using the LE transport.

  Verify that the IUT closes the Object Transfer Channel.

- Reference

  [3] 3.8.2

- Initial Condition

  - At least one object is stored on the IUT.
  - Perform the preamble described in Section 4.2.2.
  - The object that is to be used in this test procedure is identified in the IXIT [7] by its Object Name. If the Object List Control Point (OLCP) is supported by the IUT (i.e., the IUT supports more than one object), run the preamble described in Section 4.2.4 to ensure that the object to be tested is selected as the Current Object.
  - The object properties permit reading.
  - The object’s current size is greater than 23 octets. This is specified to ensure that it requires more than one LE-frame to be sent for the purposes of the test, given that a Maximum PDU Size (MPS) of 23 octets is used in the test procedure.

- Test Procedure

  1. The Lower Tester reads the value of the Object Size characteristic.
  2. The Lower Tester opens an Object Transfer Channel by sending a valid LE Credit Based Connection Request packet to the IUT which includes the following parameter values: MPS=23, MTU=23, Initial Credits=1.
  3. The IUT sends a positive LE Credit Based Connection Response packet to the Lower Tester.
  4. The Lower Tester writes the Read Op Code (0x05) to the Object Action Control Point with a Length parameter value equal to the value of the Current Size field of the Object Size characteristic read in step 1. The Offset parameter is set to 0.

5. The IUT, after sending a Write Response to acknowledge the write to the Object Action Control Point, sends an indication of the Object Action Control Point characteristic.
6. The Lower Tester receives an *ATT_Handle_Value_Indication* from the IUT containing the Object Action Control Point characteristic handle and value.
7. The Lower Tester sends an *ATT_Handle_Value_Confirmation* to the IUT.
8. The IUT begins sending data to the Lower Tester via the Object Transfer Channel opened in step 2.
9. When the initial credit provided in step 2 is exhausted, the Lower Tester does not provide any further credit, causing the object transfer to stall. i.e., the Lower Tester does not send an LE Flow Control Credit packet to the IUT but the Lower Tester does not otherwise initiate closure of the Object Transfer Channel.
10. The Lower Tester waits for at least 30 seconds to ensure that the Object Transfer Timeout has expired before proceeding to the next step.
11. The Object Transfer Channel is no longer open, having been closed by the IUT.
12. The Lower Tester opens an Object Transfer Channel by sending a valid LE Credit Based Connection Request packet to the IUT which includes the following parameter values: MPS=23, MTU=23, Initial Credits=1.
13. The IUT sends a positive LE Credit Based Connection Response packet to the Lower Tester.
14. The Lower Tester writes the Read Op Code (0x05) to the Object Action Control Point with a Length parameter value equal to the value of the Current Size field of the Object Size characteristic read in step 1. The Offset parameter is set to 0.
15. The IUT, after sending a Write Response to acknowledge the write to the Object Action Control Point, sends an indication of the Object Action Control Point characteristic.
16. The Lower Tester receives an *ATT_Handle_Value_Indication* from the IUT containing the Object Action Control Point characteristic handle and value.
17. The Lower Tester sends an *ATT_Handle_Value_Confirmation* to the IUT.
18. The IUT starts sending data to the Lower Tester via the Object Transfer Channel opened in step 12.
19. When additional credit is required during the object transfer operation, the Lower Tester sends LE Flow Control Credit packets to the IUT in the normal way.
20. The object transfer operation is completed successfully.

- Expected Outcome

Pass verdict

In step 5, the IUT sends an indication of the Object Action Control Point characteristic containing the Response Code Op Code (0x60), the Request Op Code (0x05) followed by the Result Code for ‘Success’ (0x01) with no Response Parameter.

In step 8, some object data is received by the Lower Tester via the Object Transfer Channel but not the complete object contents.

The IUT closes the Object Transfer Channel after the object transfer times out during step 10.

In step 15, the IUT does not respond with the ATT Error “Procedure Already in Progress”; the IUT sends an indication of the Object Action Control Point characteristic containing the Response Code Op Code (0x60), the Request Op Code (0x05) followed by the Result Code for ‘Success’ (0x01) with no Response Parameter.

In step 20, the complete object contents are received by the Lower Tester.

OTS/SR/OTD/BI-06-C [Object Transfer Operation Timeout – Object Read – BR/EDR Transport]

- Test Purpose

  This test case applies only when using the BR/EDR transport.

  Verify that the IUT can detect and handle a timeout occurring when an object transfer intended to read object data from the Server is only partially completed via the Object Transfer Channel when using the BR/EDR transport.

  Verify that the IUT closes the Object Transfer Channel.

- Reference

  [3] 3.8.2

- Initial Condition
  - At least one object is stored on the IUT.
  - Perform the preamble described in Section 4.2.2.
  - The object that is to be used in this test procedure is identified in the IXIT [7] by its Object Name. If the Object List Control Point (OLCP) is supported by the IUT (i.e., the IUT supports more than one object), run the preamble described in Section 4.2.4 to ensure that the object to be tested is selected as the Current Object.
  - The object properties permit reading.
  - The object's current size is greater than 240 octets. This is specified to ensure that it requires several I-frames to be sent for the purposes of the test, given that the minimum default MTU size of 48 octets is used in the test procedure.

- Test Procedure
  1. The Lower Tester reads the value of the Object Size characteristic.
  2. The Lower Tester opens an Object Transfer Channel by sending a valid L2CAP Connection Request packet to the IUT using the BR/EDR transport. The connection is configured to use Enhanced Retransmission Mode and the minimum default MTU size of 48 octets.
  3. The IUT sends a positive Connection Response packet to the Lower Tester.
  4. The Lower Tester writes the Read Op Code (0x05) to the Object Action Control Point with a Length parameter value equal to the value of the Current Size field of the Object Size characteristic read in step 1. The Offset parameter is set to 0.
  5. The IUT, after sending a Write Response to acknowledge the write to the Object Action Control Point, sends an indication of the Object Action Control Point characteristic.
  6. The Lower Tester receives an *ATT_Handle_Value_Indication* from the IUT containing the Object Action Control Point characteristic handle and value.
  7. The Lower Tester sends an *ATT_Handle_Value_Confirmation* to the IUT.
  8. The IUT begins sending data to the Lower Tester via the Object Transfer Channel opened in step 2.
  9. The Lower Tester immediately sends an L2CAP Receiver Not Ready (RNR) frame to the IUT to indicate a busy condition.
  10. The IUT recognizes the *RemoteBusy* condition, ceasing to send object data through the Object Transfer Channel.
  11. The Lower Tester maintains the busy condition by not sending any Receiver Ready (RR), Reject (REJ) or Selective Reject (SREJ) RNR frames, causing the object transfer operation to stall. However, the Lower Tester does not otherwise initiate closure of the Object Transfer Channel.

12. The Lower Tester waits for at least 30 seconds to ensure that the Object Transfer Timeout has expired before proceeding to the next step.
13. The Object Transfer Channel is no longer open, having been closed by the IUT.
14. The Lower Tester opens an Object Transfer Channel by sending a valid L2CAP Connection Request packet to the IUT using the BR/EDR transport. The connection is configured to use Enhanced Retransmission Mode and the minimum default MTU size of 48 octets.
15. The IUT sends a positive Connection Response packet to the Lower Tester.
16. The Lower Tester writes the Read Op Code (0x05) to the Object Action Control Point with a Length parameter value equal to the value of the Current Size field of the Object Size characteristic read in step 1. The Offset parameter is set to 0.
17. The IUT, after sending a Write Response to acknowledge the write to the Object Action Control Point, sends an indication of the Object Action Control Point characteristic.
18. The Lower Tester receives an *ATT_Handle_Value_Indication* from the IUT containing the Object Action Control Point characteristic handle and value.
19. The Lower Tester sends an *ATT_Handle_Value_Confirmation* to the IUT.
20. The IUT starts sending data to the Lower Tester via the Object Transfer Channel opened in step 14.
21. The Lower Tester manages flow control in the normal way using Enhanced Retransmission Mode
22. The object transfer operation is completed successfully.

- Expected Outcome

  Pass verdict

  In step 5, the IUT sends an indication of the Object Action Control Point characteristic containing the Response Code Op Code (0x60), the Request Op Code (0x05) followed by the Result Code for 'Success' (0x01) with no Response Parameter.

  In step 8, some object data is received by the Lower Tester via the Object Transfer Channel but not the complete object contents.

  The IUT closes the Object Transfer Channel after the object transfer times out during step 12.

  In step 17, the IUT does not respond with the ATT Error "Procedure Already in Progress"; the IUT sends an indication of the Object Action Control Point characteristic containing the Response Code Op Code (0x60), the Request Op Code (0x05) followed by the Result Code for 'Success' (0x01) with no Response Parameter.

  In step 22, the complete object contents are received by the Lower Tester.

4.20 Directory Listing Object

OTS/SR/DLO/BV-01-C [Generation of Directory Listing Object]

- Test Purpose

  Verify that the IUT generates a properly formed Directory Listing Object. Verify that the Directory Listing Object is of the correct object type and is read-only. Verify that the object contents when parsed are correct

- Reference

  [3] 4.1

- Initial Condition
  - At least three objects are stored on the IUT.
  - The Lower Tester performs the preamble described in Section 4.2.3 to discover all the objects available on the IUT.
- Test Procedure
  1. The Lower Tester writes the Go To Op Code (0x05) to the Object List Control Point with the parameter value set to the Object ID value of 0x000000000000, the Object ID value reserved for the Directory Listing Object.
  2. The IUT, after sending a Write Response to acknowledge the write to the Object List Control Point, sends an indication of the Object List Control Point characteristic.
  3. The Lower Tester receives an *ATT_Handle_Value_Indication* from the IUT containing the Object List Control Point characteristic handle and value.
  4. The Lower Tester sends an *ATT_Handle_Value_Confirmation* to the IUT.
  5. The Lower Tester reads the values of the Object Size, Object Type and Object Properties characteristics.
  6. The Lower Tester opens an Object Transfer Channel.
  7. The Lower Tester writes the Read Op Code (0x05) to the Object Action Control Point with a Length parameter value equal to the Current Size field of the Object Size characteristic value read in step 5. The Offset parameter is set to 0.
  8. The IUT, after sending a Write Response to acknowledge the write to the Object Action Control Point, sends an indication of the Object Action Control Point characteristic.
  9. The Lower Tester receives an *ATT_Handle_Value_Indication* from the IUT containing the Object Action Control Point characteristic handle and value.
  10. The Lower Tester sends an *ATT_Handle_Value_Confirmation* to the IUT.
  11. The IUT sends data to the Lower Tester via the Object Transfer Channel opened in step 6.
  12. The Lower Tester waits for the object transfer operation to complete.
  13. The Lower Tester parses the contents of the object it has received.
  14. The Lower Tester compares the list of objects and metadata it has obtained from the Directory Listing Object with the list it compiled itself during the pre-amble specified in the Initial Condition section.
- Expected Outcome

  <u>Pass verdict</u>

  In step 5, the value of the Object Type characteristic is «Directory Listing» and the value of the Object Properties characteristic shows that this object is read-only (i.e., the object properties are such that they do not permit a client to modify the object contents nor to delete the object).

  In step 13, the contents of the Directory Listing Object when parsed provide a complete and accurate listing of the objects available on the IUT. The metadata values obtained from the Directory Listing Object agree with the values that the Lower Tester read individually during the pre-amble procedure.

  Note: it is acceptable if optional fields within Object Records are not included in the Directory Listing Object even if the associated metadata was available during the pre-amble procedure.

5 Test case mapping

The Test Case Mapping Table (TCMT) maps test cases to specific requirements in the ICS. The IUT will be 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 Object Transfer Service (OTS) [4].

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

Test Case(s): The applicable test case identifiers, 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 [1].

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

Item	Feature	Test Case(s)
OTS 0/1	Service Definition	OTS/SR/SGGIT/SER/BV-01-C
OTS 2/2	SDP Record	OTS/SR/SGGIT/SDP/BV-01-C
OTS 4/1	OTS Feature Characteristic	OTS/SR/SGGIT/CHA/BV-01-C OTS/SR/CR/BV-01-C
OTS 4/2	Object Name Characteristic	OTS/SR/CR/BV-02-C
OTS 4/2 AND NOT OTS 4/3	Object Name Characteristic – Read Only	OTS/SR/SGGIT/CHA/BV-02-C
OTS 4/3	Object Name Characteristic - Write	OTS/SR/SGGIT/CHA/BV-03-C OTS/SR/CW/BV-01-C
OTS 4/3 AND OTS 3/2	Object Name Characteristic – Write – Multiple Objects Supported	OTS/SR/OME/BI-01-C
OTS 4/4	Object Name Characteristic – Read Long	OTS/SR/CRL/BV-01-C
OTS 4/5	Object Name Characteristic – Write Long	OTS/SR/CWL/BV-01-C
OTS 4/6	Object Type Characteristic	OTS/SR/SGGIT/CHA/BV-04-C OTS/SR/CR/BV-03-C
OTS 4/7	Object Size Characteristic	OTS/SR/SGGIT/CHA/BV-05-C OTS/SR/CR/BV-04-C
OTS 4/8	Object First-Created Characteristic	OTS/SR/CR/BV-05-C
OTS 4/8 AND NOT OTS 4/9	Object First-Created Characteristic – Read Only	OTS/SR/SGGIT/CHA/BV-06-C
OTS 4/9	Object First-Created Characteristic - Write	OTS/SR/SGGIT/CHA/BV-07-C OTS/SR/CW/BV-02-C
OTS 4/10	Object Last-Modified Characteristic	OTS/SR/CR/BV-06-C
OTS 4/10 AND NOT OTS 4/11	Object Last-Modified Characteristic - RTC	OTS/SR/SGGIT/CHA/BV-08-C OTS/SR/RTC/BV-01-C
OTS 4/11	Object Last-Modified Characteristic - Write	OTS/SR/SGGIT/CHA/BV-09-C OTS/SR/CW/BV-05-C OTS/SR/RTC/BV-02-C

Item	Feature	Test Case(s)
OTS 4/12	Object ID Characteristic	OTS/SR/SGGIT/CHA/BV-10-C OTS/SR/CR/BV-07-C
OTS 4/13	Object Properties Characteristic	OTS/SR/CR/BV-08-C
OTS 4/13 AND NOT OTS 4/14	Object Properties Characteristic – Read Only	OTS/SR/SGGIT/CHA/BV-11-C
OTS 4/14	Object Properties Characteristic - Write	OTS/SR/SGGIT/CHA/BV-12-C OTS/SR/CW/BV-03-C OTS/SR/OME/BI-03-C
OTS 4/15	Object Action Control Point (OACP)	OTS/SR/SGGIT/CHA/BV-13-C OTS/SR/CON/BV-01-C OTS/SR/OAE/BI-01-C OTS/SR/OAE/BI-12-C
OTS 4/17 AND (OTS 5/2 OR OTS 5/3 OR OTS 5/4 OR OTS 5/5 OR OTS 5/6)	Object Action Control Point (OACP) - Invalid Object	OTS/SR/OAE/BI-02-C
OTS 4/16	Object List Control Point (OLCP)	OTS/SR/SGGIT/CHA/BV-14-C OTS/SR/CON/BV-02-C OTS/SR/OLE/BI-01-C OTS/SR/OLE/BI-05-C
OTS 4/16 AND OTS 3/2	Object List Control Point (OLCP) - First, Last, Previous and Next	OTS/SR/OLSP/BV-01-C
OTS 4/16 AND OTS 3/3	Object List Control Point (OLCP) – Zero Objects	OTS/SR/OLE/BI-03-C
OTS 4/17	Object List Filter Characteristic	OTS/SR/DEC/BV-11-C OTS/SR/CR/BV-09-C OTS/SR/CW/BV-04-C OTS/SR/OLF/BV-01-C OTS/SR/OLF/BV-02-C OTS/SR/OLF/BV-03-C OTS/SR/OLF/BV-04-C OTS/SR/OLF/BI-01-C OTS/SR/OME/BI-02-C OTS/SR/INV/BV-01-C
OTS 4/18	Object List Filter Characteristic – Read Long	OTS/SR/CRL/BV-02-C
OTS 4/19	Object List Filter Characteristic – Write Long	OTS/SR/CWL/BV-02-C
OTS 4/20	Object Changed Characteristic	OTS/SR/SGGIT/CHA/BV-15-C OTS/SR/CON/BV-03-C OTS/SR/OC/BV-01-C
OTS 3/1 AND OTS 4/20	Object Changed Characteristic	OTS/SR/OC/BV-02-C

Item	Feature	Test Case(s)
OTS 5/1	OACP Create Procedure	OTS/SR/OASP/BV-01-C OTS/SR/OAE/BI-05-C
OTS 5/1 AND OTS 3/2	OACP Create Procedure – Multiple Objects Supported	OTS/SR/OTD/BI-01-C
OTS 5/2	OACP Delete Procedure	OTS/SR/OASP/BV-02-C OTS/SR/OAE/BI-06-C
OTS 5/3	OACP Calculate Checksum Procedure	OTS/SR/OASP/BV-03-C
OTS 5/4	OACP Execute Procedure	OTS/SR/OAE/BI-07-C
OTS 5/5	OACP Read Procedure	OTS/SR/OASP/BV-04-C OTS/SR/OASP/BV-05-C OTS/SR/OAE/BI-03-C OTS/SR/OAE/BI-08-C
OTS 5/5 AND OTS 2/1	Object Transfer - Using an LE L2CAP Connection-Oriented Channel	OTS/SR/OTD/BI-05-C
OTS 5/5 AND OTS 2/2	Object Transfer - Using a BR/EDR L2CAP Connection-Oriented Channel	OTS/SR/OTD/BI-06-C
OTS 5/6	OACP Write Procedure	OTS/SR/OASP/BV-06-C OTS/SR/OAE/BI-04-C OTS/SR/OAE/BI-09-C OTS/SR/OTD/BI-04-C
OTS 5/6 AND OTS 3/1	OACP Write – Simultaneous Clients	OTS/SR/OTD/BI-02-C OTS/SR/OAE/BI-10-C OTS/SR/OTD/BI-03-C
OTS 5/7	OACP Write – Increased Allocated Size	OTS/SR/OASP/BV-08-C
OTS 5/8	OACP Write – Truncate Mode	OTS/SR/OASP/BV-09-C
OTS 5/9	OACP Write – Patching of Objects	OTS/SR/OASP/BV-07-C
OTS 5/10	OACP Abort Procedure	OTS/SR/OASP/BV-10-C OTS/SR/OAE/BI-11-C
OTS 5/3 OR OTS 5/5 OR OTS 5/6 OR OTS 5/7 OR OTS 5/8 OR OTS 5/9	OACP Op Codes with Offset Parameter	OTS/SR/OAE/BI-13-C
OTS 6/5	OLCP Go To Procedure	OTS/SR/OLSP/BV-02-C OTS/SR/OLE/BI-04-C
OTS 6/6	OLCP Order Procedure	OTS/SR/OLSP/BV-03-C OTS/SR/OLE/BI-02-C
OTS 6/7	OLCP Request Number of Objects Procedure	OTS/SR/OLSP/BV-04-C
OTS 6/8	OLCP Clear Marking Procedure	OTS/SR/OLSP/BV-05-C
OTS 7/1	Generation of Directory Listing Object	OTS/SR/DLO/BV-01-C

Table 5.1: Test case mapping

6 Revision history and acknowledgments

Revision History

Publication Number	Revision Number	Date	Comments
	D0.5.0	2014-05-16	Initial draft for review based on UDS.TS
	D0.5.1	2014-06-08	Incorporated feedback from Guillaume and Abhirup. Addressed previous comments and created the content of some more of the test cases. Implemented the new TP identifiers from Table 4.1 and updated the ordering of the test groups. Completed Object Action Control Point Service Procedures and OACP Error Handling test cases.
	D0.9.0	2014-08-05	Rolled revision to draft 0.9. Incorporated feedback from informal IOP test event and BTI. Moved general metadata error handling test cases into common group.
	D0.9.1	2014-08-06	Added test cases for Object Properties and OTS Feature characteristics. Added test cases for OLCP procedures and OLCP error handling. Added several new test cases to OLCP Service Procedure and OLCP Error Handing groups.
	D0.9.2	2014-08-12	Added preamble for object discovery (Section 4.2.3) and created test cases for the test groups for timeout and disconnection during an object transfer (OTS/SR/OC/BV-02-C [Object Changed by another Client]), more OLCP Error Handling (Section 4.14), Object List Filter Operations (Section 4.17) plus various edits to existing test cases.
	D0.9.3	2014-08-13	Updated following discussion in SFWG conference call. Reduced reliance on IXIT values where possible. Added a preamble (Section 4.2.4) to initialize the Current Object by enabling the tester to find and select an object on the IUT by object name. Modified affected test cases to use this preamble. Added TP/OAE/BI-15-C [OACP – Invalid Parameter] test case.
	D0.9.4	2014-08-13	Corrected the Pass Verdict section of TP/OLSP/BV-07-C. Minor editing. Added TCMT.
	D0.9.5	2014-08-22	Updated to address BTI review comments from Megan, Magnus and Miles. Also simplified the OLCP Previous and Next test case to align it with the updated test case for OLCP First and Last which was overlapping. Corrected the description of the test purpose in OTS/SR/OTD/BI-03-C [Lock is Released due to Timeout].
	D0.9.6	2014-08-26	Updated TCMT as a result if ICS changes to Table 3. Fixed some IXIT-related issues.
	D0.9.7	2015-03-10	Incorporated feedback from December IOP. Test cases updated throughout to align with multiple changes in the OTS D09r04 spec. including new

Publication Number	Revision Number	Date	Comments
			OACP Op Code parameters and deletion of redundant characteristics. Test case TP/OASP/BV-09-C added to test the Truncate Mode that has been introduced in the OTS. TCMT updated, including updated OTS ICS item numbers. Added two BR/EDR-specific test cases. Removed references to the IXIT that were no longer needed. Resolved WG comments from Bob and Leif. Updated the RTC test group.
	D0.9.8	2015-03-19	Minor update to TP/OME/BI-01-C to utilize the new “Object Name Already Exists” ATT Application Error response which was added during resolution of BARB comments on OTS D09r04. Integrated resolution of BTI review feedback. Version used at Formal IOP.
	D1.0.0r00	2015-05-27	Deleted TP/OAE/BI-14-C [OACP – Procedure Already In Progress] and addressed minor feedback from Formal IOP. Minor update to TP/OASP/BV-01-C [OACP - Create]. Added two new test cases to cover reading and writing long attributes in the Object List Filter characteristics: TP/CRL/BV-02-C and TP/CWL/BV-02-C. Updated the Overview in Section 3.1 in line with the similar section in the OTP.TS.
	D1.0.0r01	2015-06-22	Added test case TP/DLO/BV-01-C to verify that the IUT generates a properly formed Directory Listing Object meeting the requirements of the service, following discussion with the BARB.
	D1.0.0r02	2015-09-17	Clarification of TP/DLO/BV-01-C [Generation of Directory Listing Object] following F2F discussion of the feedback from the IOP performed on 7-9 September 2015.
	D1.0.0r03	2015-10-19	Accepted all changes from Daniel Cowling, Alicia Courtney and Miles Louis, resolving BTI feedback.
0	1.0.0	2015-11-17	Spec adopted by BoD. Prepared for publication.
	1.0.1r00	2016-05-25	Converted to new Test Case ID conventions as defined in TSTO v4.1.
1	1.0.1	2016-07-14	Prepared for TCRL 2016-1 publication.
	1.0.2r00	2017-08-21	TSE 9635: Revised the TCMT for OTS/SR/OLSP/BV-01-C.
	1.0.2r01	2017-09-22	Updated TS Template.
2	1.0.2	2017-11-28	Approved by BTI. Prepared for TCRL 2017-2 publication.
	1.0.3r00	2018-10-02	TSE 11013 (rating 2): Revised the TCMT for OTS/SR/OC/BV-02-C
3	1.0.3	2018-11-21	Approved by BTI. Prepared for TCRL 2018-2 publication.
	p4r00–r03	2022-03-22 – 2022-05-18	TSE 17266 (rating 2): Converted tests to GGIT: the new GGIT TCIDs are OTS/SR/SGGIT/CHA/BV-01-C – OTS/SR/SGGIT/CHA/BV-15-C, OTS/SR/SGGIT/SDP/BV-01-C, and OTS/SR/SGGIT/SER/BV-01-C, and the deleted TCIDs

Publication Number	Revision Number	Date	Comments
			are OTS/SR/DEC/BV-01-C – -10-C and -12-C, OTS/SR/DES/BV-01-C – -03-C, and OTS/SR/SD/BV-01-C and -02-C. Updated the “Test groups” section and added the GGIT material to the TCID conventions section. Updated the Test Purpose, Initial Condition, and the Pass verdict for the section containing OTS/SR/DEC/BV-11-C to cite a GGIT test case. Updated cross-references to point to the new GATT Integrated Tests section globally. Updated the TCMT accordingly. TSE 18420 (rating 1): Removed direct references to GATT test cases in the ATT Bearer preambles and in the Initial Configuration, Object Discovery, and Initialization of Current Object sections, as well as in OTS/SR/DEC/BV-11-C, OTS/SR/CR/BV-01-C – -09-C, OTS/SR/CRL/BV-01-C and -02-C, OTS/SR/CW/BV-01-C – -05-C, OTS/SR/CWL/BV-01-C and -02-C, OTS/SR/CON/BV-01-C – -03-C, OTS/SR/OLSP/BV-01-C – -05-C, OTS/SR/RTC/BV-01-C and -02-C, OTS/SR/OLF/BV-01-C – -04-C, OTS/SR/OC/BV-01-C and -02-C, and OTS/SR/OTD/BI-01-C. TSE 18723 (rating 1): Editorials to align the document with the latest TS template in anticipation of a future .Z release. Performed template-related formatting fixes. Assigned publication number 3 to previous v1.0.3 and aligned copyright page with v2 of the DNMD.
4	p4	2022-06-28	Approved by BTI on 2022-05-31. Prepared for TCRL 2022-1 publication.

Acknowledgments

Name	Company
Dejan Berec	Bluetooth SIG, Inc.
Jawid Mirani	Bluetooth SIG, Inc.
Abhirup Ghosh	Intel Corporation
Robert D. Hughes	Intel Corporation
Leif-Alexandre Aschehoug	Nordic Semiconductor ASA
Guillaume Schatz	Polar Electro Oy
Laurence Richardson	Qualcomm Technologies, Inc.